C8051F96x

Ultra Low Power 128K, LCD MCU Family

Ultra Low Power Consumption at 3.6V
- 130 µA/MHz Low-Power Active mode with dc-dc enabled
- 110 nA sleep current w/ data retention; POR monitor enabled
- 400 nA sleep mode with SmaRTClock (internal LFO)
- 700 nA sleep mode with SmaRTClock (ext. crystal)
- 2 µs wakeup time; 1.5 µA analog settling time

12-Bit; 16 Ch. Analog-to-Digital Converter
- Up to 75 ksps (12-bit mode) or 300 ksps (10-bit mode)
- External pin or internal VREF (no ext cap required)
- On-chip voltage reference; 0.5x gain allows measuring voltages up to twice the reference voltage
- Autonomous burst mode with 16-bit auto-averaging accumulator
- Integrated temperature sensor

Two Low Current Comparators
- Programmable hysteresis and response time
- Configurable as wake-up or reset source

Internal 6-Bit Current Reference
- Up to ±500 µA; source and sink capability
- Enhanced resolution via PWM interpolation

Integrated LCD Controller
- Supports up to 128 segments (32x4)
- LCD controller consumes only 400 nA for 32-segment static display
- Integrated charge pump for contrast control

Metering-Specific Peripherals
- DC-DC buck converter allows dynamic voltage scaling for maximum efficiency (250 mW output)
- Sleep-mode pulse accumulator with programmable switch, de-bounce and pull-up control; interfaces directly to metering sensor
- Data Packet Processing Engine (DPPE) includes hardware AES, DMA, CRC and encoding blocks for acceleration of wireless protocols

High-Speed 8051 µC Core
- Pipelined instruction architecture; executes 70% of instructions in 1 or 2 system clocks

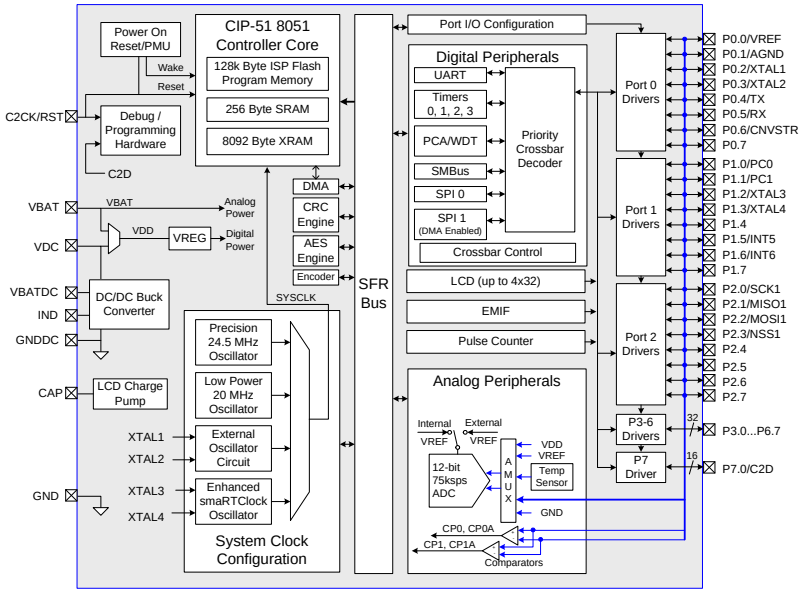
Memory
- Up to 128 kB Flash; In-system programmable; Full read/write/erase functionality over supply range
- Up to 8 kB internal data RAM

Digital Peripherals
- 57 or 34 port I/O; All 5 V tolerant with high sink current and programmable drive strength
- Hardware SMBus™ (I^2C™ Compatible), 2 x SPI™, and UART serial ports available concurrently
- Four general purpose 16-bit counter/timers
- Programmable 16-bit counter/timer array with six capture/compare modules and watchdog timer

Clock Sources
- Precision Internal oscillator: 24.5 MHz, 2% accuracy supports UART operation; spread-spectrum mode for reduced EMI
- Low power internal oscillator: 20 MHz
- External oscillator: Crystal, RC, C, or CMOS Clock
- SmaRTClock oscillator: 32 kHz Crystal or 16.4 kHz internal LFO

On-Chip Debug
- On-chip debug circuitry facilitates full-speed, non-intrusive in-system debug (no emulator required)
- Provides 4 breakpoints, single stepping

Packages
- 76-pin DQFN (6 x 6 mm)
- 40-pin QFN (6 x 6 mm)
- 80-pin TQFP (12 x 12 mm)
- **Temperature Range: –40 to +85 °C**



Copyright © 2012 by Silicon Laboratories

This information applies to a product under development. Its characteristics and specifications are subject to change without notice.

Table of Contents

1. System Overview 23
1.1. CIP-51™ Microcontroller Core 29
1.1.1. Fully 8051 Compatible 29
1.1.2. Improved Throughput 29
1.1.3. Additional Features 29
1.2. Port Input/Output 30
1.3. Serial Ports 31
1.4. Programmable Counter Array 31
1.5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode 32
1.6. Programmable Current Reference (IREF0) 33
1.7. Comparators 33
2. Ordering Information 35
3. Pinout and Package Definitions 36
3.1. DQFN-76 Package Specifications 46
3.1.1. Package Drawing 46
3.1.2. Land Pattern 47
3.1.3. Soldering Guidelines 48
3.2. QFN-40 Package Specifications 50
3.3. TQFP-80 Package Specifications 52
3.3.1. Soldering Guidelines 55
4. Electrical Characteristics 56
4.1. Absolute Maximum Specifications 56
4.2. Electrical Characteristics 57
5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode 78
5.1. Output Code Formatting 78
5.2. Modes of Operation 80
5.2.1. Starting a Conversion 80
5.2.2. Tracking Modes 80
5.2.3. Burst Mode 82
5.2.4. Settling Time Requirements 83
5.2.5. Gain Setting 83
5.3. 8-Bit Mode 84
5.4. 12-Bit Mode 84
5.5. Low Power Mode 85
5.6. Programmable Window Detector 91
5.6.1. Window Detector In Single-Ended Mode 93
5.6.2. ADC0 Specifications 94
5.7. ADC0 Analog Multiplexer 95
5.8. Temperature Sensor 97
5.8.1. Calibration 97
5.9. Voltage and Ground Reference Options 100

5.10. External Voltage Reference 101
5.11. Internal Voltage Reference 101
5.12. Analog Ground Reference 101
5.13. Temperature Sensor Enable 101
5.14. Voltage Reference Electrical Specifications 102
6. Programmable Current Reference (IREF0) 103
6.1. PWM Enhanced Mode 103
6.2. IREF0 Specifications 104
7. Comparators 105
7.1. Comparator Inputs 105
7.2. Comparator Outputs 106
7.3. Comparator Response Time 107
7.4. Comparator Hysterisis 107
7.5. Comparator Register Descriptions 108
7.6. Comparator0 and Comparator1 Analog Multiplexers 112
8. CIP-51 Microcontroller 115
8.1. Instruction Set 116
8.1.1. Instruction and CPU Timing 116
8.2. CIP-51 Register Descriptions 121
9. Memory Organization 124
9.1. Program Memory 124
9.1.1. MOVX Instruction and Program Memory 127
9.2. Data Memory 127
9.2.1. Internal RAM 128
9.2.2. External RAM 128
10. External Data Memory Interface and On-Chip XRAM 129
10.1. Accessing XRAM 129
10.1.1. 16-Bit MOVX Example 129
10.1.2. 8-Bit MOVX Example 129
10.2. Configuring the External Memory Interface 130
10.3. Port Configuration 130
10.4. Multiplexed and Non-multiplexed Selection 134
10.4.1. Multiplexed Configuration 134
10.4.2. Non-multiplexed Configuration 134
10.5. Memory Mode Selection 135
10.5.1. Internal XRAM Only 136
10.5.2. Split Mode without Bank Select 136
10.5.3. Split Mode with Bank Select 136
10.5.4. External Only 136
10.6. Timing 137
10.6.1. Non-Multiplexed Mode 139
10.6.2. Multiplexed Mode 142
11. Direct Memory Access (DMA0) 146
11.1. DMA0 Architecture 147
11.2. DMA0 Arbitration 148

11.2.1. DMA0 Memory Access Arbitration 148
11.2.2. DMA0 Channel Arbitration 148
11.3. DMA0 Operation in Low Power Modes 148
11.4. Transfer Configuration 149
12. Cyclic Redundancy Check Unit (CRC0) 160
12.1. 16-bit CRC Algorithm 160
12.3. Preparing for a CRC Calculation 163
12.4. Performing a CRC Calculation 163
12.5. Accessing the CRC0 Result 163
12.6. CRC0 Bit Reverse Feature 167
13. DMA-Enabled Cyclic Redundancy Check Module (CRC1) 168
13.1. Polynomial Specification 168
13.2. Endianness 169
13.3. CRC Seed Value 170
13.4. Inverting the Final Value 170
13.5. Flipping the Final Value 170
13.6. Using CRC1 with SFR Access 171
13.7. Using the CRC1 module with the DMA 171
14. Advanced Encryption Standard (AES) Peripheral 175
14.1. Hardware Description 176
14.1.1. AES Encryption/Decryption Core 177
14.1.2. Data SFRs 177
14.1.3. Configuration sfrs 178
14.1.4. Input Multiplexer 178
14.1.5. Output Multiplexer 178
14.1.6. Internal State Machine 178
14.2. Key Inversion 179
14.2.1. Key Inversion using DMA 180
14.2.2. Key Inversion using SFRs 181
14.2.3. Extended Key Output Byte Order 182
14.2.4. Using the DMA to unwrap the extended Key 183
14.3. AES Block Cipher 184
14.4. AES Block Cipher Data Flow 185
14.4.1. AES Block Cipher Encryption using DMA 186
14.4.2. AES Block Cipher Encryption using SFRs 187
14.5. AES Block Cipher Decryption 188
14.5.1. AES Block Cipher Decryption using DMA 188
14.5.2. AES Block Cipher Decryption using SFRs 189
14.6. Block Cipher Modes 190
14.6.1. Cipher Block Chaining Mode 190
14.6.2. CBC Encryption Initialization Vector Location 192
14.6.3. CBC Encryption using DMA 192
14.6.4. CBC Decryption 195
14.6.5. Counter Mode 198
14.6.6. CTR Encryption using DMA 200

15. Encoder/Decoder 207
15.1. Manchester Encoding 208
15.2. Manchester Decoding 209
15.3. Three-out-of-Six Encoding 210
15.4. Three-out-of-Six Decoding 211
15.5. Encoding/Decoding with SFR Access 212
15.6. Decoder Error Interrupt 212
15.7. Using the ENC0 module with the DMA 213
16. Special Function Registers 216
16.1. SFR Paging 216
16.2. Interrupts and SFR Paging 216
17. Interrupt Handler 232
17.1. Enabling Interrupt Sources 232
17.2. MCU Interrupt Sources and Vectors 232
17.3. Interrupt Priorities 233
17.4. Interrupt Latency 233
17.5. Interrupt Register Descriptions 235
17.6. External Interrupts INT0 and INT1 242
18. Flash Memory 244
18.1. Programming the Flash Memory 244
18.1.1. Flash Lock and Key Functions 244
18.1.2. Flash Erase Procedure 244
18.1.3. Flash Write Procedure 245
18.1.4. Flash Write Optimization 246
18.2. Non-volatile Data Storage 247
18.3. Security Options 247
18.4. Determining the Device Part Number at Run Time 249
18.5. Flash Write and Erase Guidelines 250
18.5.1. VDD Maintenance and the VDD Monitor 250
18.5.2. PSWE Maintenance 251
18.5.3. System Clock 251
18.6. Minimizing Flash Read Current 252
19. Power Management 257
19.1. Normal Mode 258
19.2. Idle Mode 258
19.3. Stop Mode 259
19.4. Low Power Idle Mode 259
19.5. Suspend Mode 263
19.6. Sleep Mode 263
19.7. Configuring Wakeup Sources 264
19.8. Determining the Event that Caused the Last Wakeup 264
19.9. Power Management Specifications 268
20. On-Chip DC-DC Buck Converter (DC0) 269
20.1. Startup Behavior 270
20.4. Optimizing Board Layout 271

20.5. Selecting the Optimum Switch Size 271
20.6. **DC-DC Converter Clocking Options 271**
20.7. Bypass Mode 272
20.8. DC-DC Converter Register Descriptions 272
20.9. DC-DC Converter Specifications 276
21. Voltage Regulator (VREG0) 277
21.1. Voltage Regulator Electrical Specifications 277
22. Reset Sources 278
22.1. Power-On Reset 279
22.2. Power-Fail Reset 280
22.3. External Reset 283
22.4. Missing Clock Detector Reset 283
22.5. Comparator0 Reset 283
22.6. PCA Watchdog Timer Reset 283
22.7. Flash Error Reset 284
22.8. SmaRTClock (Real Time Clock) Reset 284
22.9. Software Reset 284
23. Clocking Sources 286
23.1. Programmable Precision Internal Oscillator 287
23.2. Low Power Internal Oscillator 287
23.3. External Oscillator Drive Circuit 287
23.3.1. External Crystal Mode 287
23.3.2. External RC Mode 289
23.3.3. External Capacitor Mode 290
23.3.4. External CMOS Clock Mode 290
23.4. Special Function Registers for Selecting and Configuring the System Clock 291
24. SmaRTClock (Real Time Clock) 295
24.1. SmaRTClock Interface 296
24.1.1. SmaRTClock Lock and Key Functions 297
24.1.2. Using RTC0ADR and RTC0DAT to Access SmaRTClock Internal Registers 297
24.1.3. SmaRTClock Interface Autoread Feature 297
24.1.4. RTC0ADR Autoincrement Feature 297
24.2. SmaRTClock Clocking Sources 300
24.2.1. Using the SmaRTClock Oscillator with a Crystal or External CMOS Clock 300
24.2.2. Using the SmaRTClock Oscillator in Self-Oscillate Mode 301
24.2.3. Using the Low Frequency Oscillator (LFO) 301
24.2.4. Programmable Load Capacitance 301
24.2.5. Automatic Gain Control (Crystal Mode Only) and SmaRTClock Bias Doubling 302
24.2.6. Missing SmaRTClock Detector 304
24.2.7. SmaRTClock Oscillator Crystal Valid Detector 304
24.3. SmaRTClock Timer and Alarm Function 304
24.3.1. Setting and Reading the SmaRTClock Timer Value 304

24.3.2. Setting a SmaRTClock Alarm 305
24.3.3. Software Considerations for using the SmaRTClock Timer and Alarm 305
25. Low-Power Pulse Counter 312
25.1. Counting Modes 313
25.2. Reed Switch Types..... 314
25.3. Programmable Pull-Up Resistors 315
25.4. Automatic Pull-Up Resistor Calibration 317
25.5. Sample Rate..... 317
25.6. Debounce 317
25.7. Reset Behavior..... 318
25.8. Wake up and Interrupt Sources..... 318
25.9. Real-Time Register Access 319
25.10. Advanced Features 319
25.10.1. Quadrature Error..... 319
25.10.2. Flutter Detection..... 320
26. LCD Segment Driver 334
26.1. Configuring the LCD Segment Driver 334
26.2. Mapping Data Registers to LCD Pins..... 335
26.3. LCD Contrast Adjustment..... 338
26.3.1. Contrast Control Mode 1 (Bypass Mode)..... 338
26.3.2. Contrast Control Mode 2 (Minimum Contrast Mode) 339
26.3.3. Contrast Control Mode 3 (Constant Contrast Mode)..... 339
26.3.4. Contrast Control Mode 4 (Auto-Bypass Mode) 340
26.4. Adjusting the VBAT Monitor Threshold 344
26.5. Setting the LCD Refresh Rate..... 345
26.6. Blinking LCD Segments..... 346
26.7. Advanced LCD Optimizations..... 348
27. Port Input/Output 351
27.1. Port I/O Modes of Operation..... 352
27.1.1. Port Pins Configured for Analog I/O..... 352
27.1.2. Port Pins Configured For Digital I/O..... 352
27.1.3. Interfacing Port I/O to High Voltage Logic..... 353
27.1.4. Increasing Port I/O Drive Strength 353
27.2. Assigning Port I/O Pins to Analog and Digital Functions..... 353
27.2.1. Assigning Port I/O Pins to Analog Functions 353
27.2.2. Assigning Port I/O Pins to Digital Functions..... 354
27.2.3. Assigning Port I/O Pins to External Digital Event Capture Functions ... 354
27.3. Priority Crossbar Decoder 355
27.4. Port Match 361
27.5. Special Function Registers for Accessing and Configuring Port I/O 363
28. SMBus..... 381
28.1. Supporting Documents..... 382
28.2. SMBus Configuration..... 382
28.3. SMBus Operation 382
28.3.1. Transmitter Vs. Receiver..... 383

28.3.2. Arbitration 383
28.3.3. Clock Low Extension 383
28.3.4. SCL Low Timeout 383
28.3.5. SCL High (SMBus Free) Timeout 384
28.4. Using the SMBus 384
28.4.1. SMBus Configuration Register 384
28.4.2. SMB0CN Control Register 388
28.4.3. Hardware Slave Address Recognition 390
28.4.4. Data Register 393
28.5. SMBus Transfer Modes 393
28.5.1. Write Sequence (Master) 393
28.5.2. Read Sequence (Master) 394
28.5.3. Write Sequence (Slave) 395
28.5.4. Read Sequence (Slave) 396
28.6. SMBus Status Decoding 397
29. UART0 402
29.1. Enhanced Baud Rate Generation 403
29.2. Operational Modes 404
29.2.1. 8-Bit UART 404
29.2.2. 9-Bit UART 404
29.3. Multiprocessor Communications 405
30. Enhanced Serial Peripheral Interface (SPI0) 411
30.1. Signal Descriptions 412
30.1.1. Master Out, Slave In (MOSI) 412
30.1.2. Master In, Slave Out (MISO) 412
30.1.3. Serial Clock (SCK) 412
30.1.4. Slave Select (NSS) 412
30.2. SPI0 Master Mode Operation 412
30.3. SPI0 Slave Mode Operation 414
30.4. SPI0 Interrupt Sources 415
30.5. Serial Clock Phase and Polarity 415
30.6. SPI Special Function Registers 417
32. Timers 444
32.1. Timer 0 and Timer 1 446
32.1.1. Mode 0: 13-bit Counter/Timer 446
32.1.2. Mode 1: 16-bit Counter/Timer 447
32.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload 447
32.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only) 448
32.2. Timer 2 454
32.2.1. 16-bit Timer with Auto-Reload 454
32.2.2. 8-bit Timers with Auto-Reload 455
32.2.3. Comparator 0/SmaRTClock Capture Mode 455
32.3. Timer 3 460
32.3.1. 16-bit Timer with Auto-Reload 460
32.3.2. 8-Bit Timers with Auto-Reload 461

32.3.3. SmaRTClock/External Oscillator Capture Mode 461
33. Programmable Counter Array 466
33.1. PCA Counter/Timer 467
33.2. PCA0 Interrupt Sources 468
33.3. Capture/Compare Modules 469
33.3.1. Edge-triggered Capture Mode 470
33.3.2. Software Timer (Compare) Mode 471
33.3.3. High-Speed Output Mode 472
33.3.4. Frequency Output Mode 473
33.3.5. 8-Bit, 9-Bit, 10-Bit and 11-Bit Pulse Width Modulator Modes 474
33.3.6. 16-Bit Pulse Width Modulator Mode 476
33.4. Watchdog Timer Mode 477
33.4.1. Watchdog Timer Operation 477
33.4.2. Watchdog Timer Usage 478
33.5. Register Descriptions for PCA0 480
34. C2 Interface 486
34.1. C2 Interface Registers 486
34.2. C2 Pin Sharing 489
Document Change List 490
Contact Information 492

List of Figures

Figure 1.1. C8051F960 Block Diagram 24
Figure 1.2. C8051F961 Block Diagram 24
Figure 1.3. C8051F962 Block Diagram 25
Figure 1.4. C8051F963 Block Diagram 25
Figure 1.5. C8051F964 Block Diagram 26
Figure 1.6. C8051F965 Block Diagram 26
Figure 1.7. C8051F966 Block Diagram 27
Figure 1.8. C8051F967 Block Diagram 27
Figure 1.9. C8051F968 Block Diagram 28
Figure 1.10. C8051F969 Block Diagram 28
Figure 1.11. Port I/O Functional Block Diagram 30
Figure 1.12. PCA Block Diagram 31
Figure 1.13. ADC0 Functional Block Diagram 32
Figure 1.14. ADC0 Multiplexer Block Diagram 33
Figure 1.15. Comparator 0 Functional Block Diagram 34
Figure 1.16. Comparator 1 Functional Block Diagram 34
Figure 3.1. DQFN-76 Pinout Diagram (Top View) 43
Figure 3.2. QFN-40 Pinout Diagram (Top View) 44
Figure 3.3. TQFP-80 Pinout Diagram (Top View) 45
Figure 3.4. DQFN-76 Package Drawing 46
Figure 3.5. DQFN-76 Land Pattern 47
Figure 3.6. Recomended Inner Via Placement 49
Figure 3.7. Typical QFN-40 Package Drawing 50
Figure 3.8. QFN-40 Landing Diagram 51
Figure 3.9. TQFP-80 Package Drawing 52
Figure 3.10. TQFP80 Landing Diagram 54
Figure 4.1. Frequency Sensitivity (External CMOS Clock, 25°C) 64
Figure 4.2. Typical VOH Curves, 1.8–3.6 V 66
Figure 4.3. Typical VOL Curves, 1.8–3.6 V 67
Figure 5.1. ADC0 Functional Block Diagram 78
Figure 5.2. 10-Bit ADC Track and Conversion Example Timing (BURSTEN = 0) ... 81
Figure 5.3. Burst Mode Tracking Example with Repeat Count Set to 4 82
Figure 5.4. ADC0 Equivalent Input Circuits 83
Figure 5.5. ADC Window Compare Example: Right-Justified Single-Ended Data .. 94
Figure 5.6. ADC Window Compare Example: Left-Justified Single-Ended Data 94
Figure 5.7. ADC0 Multiplexer Block Diagram 95
Figure 5.8. Temperature Sensor Transfer Function 97
Figure 5.9. Temperature Sensor Error with 1-Point Calibration (V_{REF} = 1.68 V) 98
Figure 5.10. Voltage Reference Functional Block Diagram 100
Figure 7.1. Comparator 0 Functional Block Diagram 105
Figure 7.2. Comparator 1 Functional Block Diagram 106
Figure 7.3. Comparator Hysteresis Plot 107
Figure 7.4. CPn Multiplexer Block Diagram 112

Figure 8.1. CIP-51 Block Diagram 115
Figure 9.1. C8051F96x Memory Map 124
Figure 9.2. Flash Program Memory Map 125
Figure 9.3. Address Memory Map for Instruction Fetches 126
Figure 10.1. Multiplexed Configuration Example 134
Figure 10.2. Non-multiplexed Configuration Example 135
Figure 10.3. EMIF Operating Modes 135
Figure 10.4. Non-multiplexed 16-bit MOVX Timing 139
Figure 10.5. Non-multiplexed 8-bit MOVX without Bank Select Timing 140
Figure 10.6. Non-multiplexed 8-bit MOVX with Bank Select Timing 141
Figure 10.7. Multiplexed 16-bit MOVX Timing 142
Figure 10.8. Multiplexed 8-bit MOVX without Bank Select Timing 143
Figure 10.9. Multiplexed 8-bit MOVX with Bank Select Timing 144
Figure 11.1. DMA0 Block Diagram 147
Figure 12.1. CRC0 Block Diagram 160
Figure 12.2. Bit Reverse Register 167
Figure 13.1. Polynomial Representation 168
Figure 14.1. AES Peripheral Block Diagram 176
Figure 14.2. Key Inversion Data Flow 179
Figure 14.3. AES Block Cipher Data Flow 185
Figure 14.4. Cipher Block Chaining Mode 190
Figure 14.5. CBC Encryption Data Flow 191
Figure 14.6. CBC Decryption Data Flow 195
Figure 14.7. Counter Mode 198
Figure 14.8. Counter Mode Data Flow 199
Figure 16.1. SFR Page Stack 217
Figure 18.1. Flash Security Example 247
Figure 19.1. C8051F96x Power Distribution 258
Figure 19.2. Clock Tree Distribution 259
Figure 20.1. Step Down DC-DC Buck Converter Block Diagram 269
Figure 22.1. Reset Sources 278
Figure 22.2. Power-On Reset Timing Diagram 279
Figure 23.1. Clocking Sources Block Diagram 286
Figure 23.2. 25 MHz External Crystal Example 288
Figure 24.1. SmaRTClock Block Diagram 295
Figure 24.2. Interpreting Oscillation Robustness (Duty Cycle) Test Results 303
Figure 25.1. Pulse Counter Block Diagram 312
Figure 25.2. Mode Examples 313
Figure 25.3. Reed Switch Configurations 314
Figure 25.4. Debounce Timing 318
Figure 25.5. Flutter Example 320
Figure 26.1. LCD Segment Driver Block Diagram 334
Figure 26.2. LCD Data Register to LCD Pin Mapping 336
Figure 26.3. Contrast Control Mode 1 338
Figure 26.4. Contrast Control Mode 2 339

Figure 26.5. Contrast Control Mode 3 339
Figure 26.6. Contrast Control Mode 4 340
Figure 27.1. Port I/O Functional Block Diagram 351
Figure 27.2. Port I/O Cell Block Diagram 352
Figure 27.3. Crossbar Priority Decoder with No Pins Skipped 356
Figure 27.4. Crossbar Priority Decoder with Crystal Pins Skipped 357
Figure 28.1. SMBus Block Diagram 381
Figure 28.2. Typical SMBus Configuration 382
Figure 28.3. SMBus Transaction 383
Figure 28.4. Typical SMBus SCL Generation 385
Figure 28.5. Typical Master Write Sequence 394
Figure 28.6. Typical Master Read Sequence 395
Figure 28.7. Typical Slave Write Sequence 396
Figure 28.8. Typical Slave Read Sequence 397
Figure 29.1. UART0 Block Diagram 402
Figure 29.2. UART0 Baud Rate Logic 403
Figure 29.3. UART Interconnect Diagram 404
Figure 29.4. 8-Bit UART Timing Diagram 404
Figure 29.5. 9-Bit UART Timing Diagram 405
Figure 29.6. UART Multi-Processor Mode Interconnect Diagram 406
Figure 30.1. SPI Block Diagram 411
Figure 30.2. Multiple-Master Mode Connection Diagram 414
Figure 30.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram 414
Figure 30.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram 414
Figure 30.5. Master Mode Data/Clock Timing 416
Figure 30.6. Slave Mode Data/Clock Timing (CKPHA = 0) 416
Figure 30.7. Slave Mode Data/Clock Timing (CKPHA = 1) 417
Figure 30.8. SPI Master Timing (CKPHA = 0) 421
Figure 30.9. SPI Master Timing (CKPHA = 1) 421
Figure 30.10. SPI Slave Timing (CKPHA = 0) 422
Figure 30.11. SPI Slave Timing (CKPHA = 1) 422
Figure 32.1. T0 Mode 0 Block Diagram 447
Figure 32.2. T0 Mode 2 Block Diagram 448
Figure 32.3. T0 Mode 3 Block Diagram 449
Figure 32.4. Timer 2 16-Bit Mode Block Diagram 454
Figure 32.5. Timer 2 8-Bit Mode Block Diagram 455
Figure 32.6. Timer 2 Capture Mode Block Diagram 456
Figure 32.7. Timer 3 16-Bit Mode Block Diagram 460
Figure 32.8. Timer 3 8-Bit Mode Block Diagram 461
Figure 32.9. Timer 3 Capture Mode Block Diagram 462
Figure 33.1. PCA Block Diagram 466
Figure 33.2. PCA Counter/Timer Block Diagram 468
Figure 33.3. PCA Interrupt Block Diagram 469

Figure 33.4. PCA Capture Mode Diagram 471
Figure 33.5. PCA Software Timer Mode Diagram 472
Figure 33.6. PCA High-Speed Output Mode Diagram 473
Figure 33.7. PCA Frequency Output Mode 474
Figure 33.8. PCA 8-Bit PWM Mode Diagram 475
Figure 33.9. PCA 9, 10 and 11-Bit PWM Mode Diagram 476
Figure 33.10. PCA 16-Bit PWM Mode 477
Figure 33.11. PCA Module 5 with Watchdog Timer Enabled 478
Figure 34.1. Typical C2 Pin Sharing 489

List of Tables

Table 2.1. Product Selection Guide 35
Table 3.1. Pin Definitions for the C8051F96x 36
Table 3.2. DQFN-76 Package Dimensions 46
Table 3.3. DQFN-76 Land Pattern Dimensions 47
Table 3.4. Recomended Inner Via Placement Dimensions 49
Table 3.5. QFN-40 Package Dimensions 50
Table 3.6. QFN-40 Landing Diagram Dimensions 51
Table 3.7. TQFP-80 Package Dimensions 52
Table 3.8. TQFP80 Landing Diagram Dimensions 54
Table 4.1. Absolute Maximum Ratings 56
Table 4.2. Global Electrical Characteristics 57
Table 4.3. Digital Supply Current at VBAT pin with DC-DC Converter Enabled 57
Table 4.4. Digital Supply Current with DC-DC Converter Disabled 58
Table 4.5. Port I/O DC Electrical Characteristics 65
Table 4.6. Reset Electrical Characteristics 68
Table 4.7. Power Management Electrical Specifications 69
Table 4.8. Flash Electrical Characteristics 69
Table 4.9. Internal Precision Oscillator Electrical Characteristics 69
Table 4.10. Internal Low-Power Oscillator Electrical Characteristics 69
Table 4.11. SmaRTClock Characteristics 70
Table 4.12. ADC0 Electrical Characteristics 70
Table 4.13. Temperature Sensor Electrical Characteristics 71
Table 4.14. Voltage Reference Electrical Characteristics 72
Table 4.15. IREF0 Electrical Characteristics 73
Table 4.16. Comparator Electrical Characteristics 74
Table 4.17. VREG0 Electrical Characteristics 75
Table 4.18. LCD0 Electrical Characteristics 76
Table 4.19. PC0 Electrical Characteristics 76
Table 4.20. DC0 (Buck Converter) Electrical Characteristics 77
Table 5.1. Representative Conversion Times and Energy Consumption for the SAR ADC with 1.65 V High-Speed VREF 85
Table 8.1. CIP-51 Instruction Set Summary 117
Table 10.1. EMIF Pinout (C8051F960/2/4/6/8) 131
Table 10.2. AC Parameters for External Memory Interface 145
Table 12.1. Example 16-bit CRC Outputs 161
Table 12.2. Example 32-bit CRC Outputs 163
Table 14.1. Extended Key Output Byte Order 182
Table 14.2. 192-Bit Key DMA Usage 183
Table 14.3. 256-bit Key DMA Usage 183
Table 15.1. Encoder Input and Output Data Sizes 207
Table 15.2. Manchester Encoding 208
Table 15.3. Manchester Decoding 209
Table 15.4. Three-out-of-Six Encoding Nibble 210

Table 15.5. Three-out-of-Six Decoding 211
Table 16.1. SFR Map (0xC0–0xFF) 222
Table 16.2. SFR Map (0x80–0xBF) 223
Table 16.3. Special Function Registers 224
Table 17.1. Interrupt Summary 234
Table 18.1. Flash Security Summary 248
Table 19.1. Power Modes 257
Table 20.1. IPeak Inductor Current Limit Settings 270
Table 23.1. Recommended XFCN Settings for Crystal Mode 288
Table 23.2. Recommended XFCN Settings for RC and C modes 289
Table 24.1. SmaRTClock Internal Registers 296
Table 24.2. SmaRTClock Load Capacitance Settings 302
Table 24.3. SmaRTClock Bias Settings 303
Table 25.1. Pull-Up Resistor Current 315
Table 25.2. Sample Rate Duty-Cycle Multiplier 315
Table 25.3. Pull-Up Duty-Cycle Multiplier 315
Table 25.4. Average Pull-Up Current (Sample Rate = 250 µs) 316
Table 25.5. Average Pull-Up Current (Sample Rate = 500 µs) 316
Table 25.6. Average Pull-Up Current (Sample Rate = 1 ms) 316
Table 25.7. Average Pull-Up Current (Sample Rate = 2 ms) 316
Table 26.1. Bit Configurations to select Contrast Control Modes 338
Table 27.1. Port I/O Assignment for Analog Functions 353
Table 27.2. Port I/O Assignment for Digital Functions 354
Table 27.3. Port I/O Assignment for External Digital Event Capture Functions 354
Table 28.1. SMBus Clock Source Selection 385
Table 28.2. Minimum SDA Setup and Hold Times 386
Table 28.3. Sources for Hardware Changes to SMB0CN 390
Table 28.4. Hardware Address Recognition Examples (EHACK = 1) 391
Table 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0) 398
Table 28.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1) 400
Table 29.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator 409
Table 29.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator 409
Table 30.1. SPI Slave Timing Parameters 423
Table 31.1. SPI Slave Timing Parameters 443
Table 32.1. Timer 0 Running Modes 446
Table 33.1. PCA Timebase Input Options 467
Table 33.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules 469
Table 33.3. Watchdog Timer Timeout Intervals1 479

List of Registers

SFR Definition 5.1. ADC0CN: ADC0 Control 86
SFR Definition 5.2. ADC0CF: ADC0 Configuration 87
SFR Definition 5.3. ADC0AC: ADC0 Accumulator Configuration 88
SFR Definition 5.4. ADC0PWR: ADC0 Burst Mode Power-Up Time 89
SFR Definition 5.5. ADC0TK: ADC0 Burst Mode Track Time 90
SFR Definition 5.6. ADC0H: ADC0 Data Word High Byte 91
SFR Definition 5.7. ADC0L: ADC0 Data Word Low Byte 91
SFR Definition 5.8. ADC0GTH: ADC0 Greater-Than High Byte 92
SFR Definition 5.9. ADC0GTL: ADC0 Greater-Than Low Byte 92
SFR Definition 5.10. ADC0LTH: ADC0 Less-Than High Byte 93
SFR Definition 5.11. ADC0LTL: ADC0 Less-Than Low Byte 93
SFR Definition 5.12. ADC0MX: ADC0 Input Channel Select 96
SFR Definition 5.13. TOFFH: Temperature Sensor Offset High Byte 99
SFR Definition 5.14. TOFFL: Temperature Sensor Offset Low Byte 99
SFR Definition 5.15. REF0CN: Voltage Reference Control 102
SFR Definition 6.1. IREF0CN: Current Reference Control 103
SFR Definition 6.2. IREF0CF: Current Reference Configuration 104
SFR Definition 7.1. CPT0CN: Comparator 0 Control 108
SFR Definition 7.2. CPT0MD: Comparator 0 Mode Selection 109
SFR Definition 7.3. CPT1CN: Comparator 1 Control 110
SFR Definition 7.4. CPT1MD: Comparator 1 Mode Selection 111
SFR Definition 7.5. CPT0MX: Comparator0 Input Channel Select 113
SFR Definition 7.6. CPT1MX: Comparator1 Input Channel Select 114
SFR Definition 8.1. DPL: Data Pointer Low Byte 121
SFR Definition 8.2. DPH: Data Pointer High Byte 121
SFR Definition 8.3. SP: Stack Pointer 122
SFR Definition 8.4. ACC: Accumulator 122
SFR Definition 8.5. B: B Register 122
SFR Definition 8.6. PSW: Program Status Word 123
SFR Definition 9.1. PSBANK: Program Space Bank Select 127
SFR Definition 10.1. EMI0CN: External Memory Interface Control 132
SFR Definition 10.2. EMI0CF: External Memory Configuration 133
SFR Definition 10.3. EMI0TC: External Memory Timing Control 138
SFR Definition 11.1. DMA0EN: DMA0 Channel Enable 150
SFR Definition 11.2. DMA0INT: DMA0 Full-Length Interrupt 151
SFR Definition 11.3. DMA0MINT: DMA0 Mid-Point Interrupt 152
SFR Definition 11.4. DMA0BUSY: DMA0 Busy 153
SFR Definition 11.5. DMA0SEL: DMA0 Channel Select for Configuration 154
SFR Definition 11.6. DMA0NMD: DMA Channel Mode 155
SFR Definition 11.7. DMA0NCF: DMA Channel Configuration 156
SFR Definition 11.8. DMA0NBAH: Memory Base Address High Byte 157
SFR Definition 11.9. DMA0NBAL: Memory Base Address Low Byte 157
SFR Definition 11.10. DMA0NAOH: Memory Address Offset High Byte 158

SFR Definition 11.11. DMA0NAOL: Memory Address Offset Low Byte 158
SFR Definition 11.12. DMA0NSZH: Transfer Size High Byte 159
SFR Definition 11.13. DMA0NSZL: Memory Transfer Size Low Byte 159
SFR Definition 12.1. CRC0CN: CRC0 Control 164
SFR Definition 12.2. CRC0IN: CRC0 Data Input 165
SFR Definition 12.3. CRC0DAT: CRC0 Data Output 165
SFR Definition 12.4. CRC0AUTO: CRC0 Automatic Control 166
SFR Definition 12.5. CRC0CNT: CRC0 Automatic Flash Sector Count 166
SFR Definition 12.6. CRC0FLIP: CRC0 Bit Flip 167
SFR Definition 13.1. CRC1CN: CRC1 Control 172
SFR Definition 13.2. CRC1IN: CRC1 Data IN 173
SFR Definition 13.3. CRC1POLL: CRC1 Polynomial LSB 173
SFR Definition 13.4. CRC1POLH: CRC1 Polynomial MSB 173
SFR Definition 13.5. CRC1OUTL: CRC1 Output LSB 174
SFR Definition 13.6. CRC1OUTH: CRC1 Output MSB 174
SFR Definition 14.1. AES0BCFG: AES Block Configuration 202
SFR Definition 14.2. AES0DCFG: AES Data Configuration 203
SFR Definition 14.3. AES0BIN: AES Block Input 204
SFR Definition 14.4. AES0XIN: AES XOR Input 205
SFR Definition 14.5. AES0KIN: AES Key Input 205
SFR Definition 14.6. AES0YOUT: AES Y Output 206
SFR Definition 15.1. ENC0CN: Encoder Decoder 0 Control 214
SFR Definition 15.2. ENC0L: ENC0 Data Low Byte 215
SFR Definition 15.3. ENC0M: ENC0 Data Middle Byte 215
SFR Definition 15.4. ENC0H: ENC0 Data High Byte 215
SFR Definition 16.1. SFRPGCN: SFR Page Control 218
SFR Definition 16.2. SFRPAGE: SFR Page 219
SFR Definition 16.3. SFRNEXT: SFR Next 220
SFR Definition 16.4. SFRLAST: SFR Last 221
SFR Definition 17.1. IE: Interrupt Enable 236
SFR Definition 17.2. IP: Interrupt Priority 237
SFR Definition 17.3. EIE1: Extended Interrupt Enable 1 238
SFR Definition 17.4. EIP1: Extended Interrupt Priority 1 239
SFR Definition 17.5. EIE2: Extended Interrupt Enable 2 240
SFR Definition 17.6. EIP2: Extended Interrupt Priority 2 241
SFR Definition 17.7. IT01CF: INT0/INT1 Configuration 243
SFR Definition 18.1. DEVICEID: Device Identification 249
SFR Definition 18.2. REVID: Revision Identification 249
SFR Definition 18.3. PSCTL: Program Store R/W Control 253
SFR Definition 18.4. FLKEY: Flash Lock and Key 254
SFR Definition 18.5. FLSCL: Flash Scale 255
SFR Definition 18.6. FLWR: Flash Write Only 255
SFR Definition 18.7. FRBCN: Flash Read Buffer Control 256
SFR Definition 19.1. PCLKACT: Peripheral Active Clock Enable 260
SFR Definition 19.2. PCLKEN: Peripheral Clock Enable 261

SFR Definition 19.3. CLKMODE: Clock Mode 262
SFR Definition 19.4. PMU0CF: Power Management Unit Configuration[1,2,3] 265
SFR Definition 19.5. PMU0FL: Power Management Unit Flag[1,2] 266
SFR Definition 19.6. PMU0MD: Power Management Unit Mode 267
SFR Definition 19.7. PCON: Power Management Control Register 268
SFR Definition 20.1. DC0CN: DC-DC Converter Control 273
SFR Definition 20.2. DC0CF: DC-DC Converter Configuration 274
SFR Definition 20.3. DC0MD: DC-DC Converter Mode 275
SFR Definition 20.4. DC0RDY: DC-DC Converter Ready Indicator 276
SFR Definition 21.1. REG0CN: Voltage Regulator Control 277
SFR Definition 22.1. VDM0CN: VDD Supply Monitor Control 282
SFR Definition 22.2. RSTSRC: Reset Source 285
SFR Definition 23.1. CLKSEL: Clock Select 291
SFR Definition 23.2. OSCICN: Internal Oscillator Control 292
SFR Definition 23.3. OSCICL: Internal Oscillator Calibration 293
SFR Definition 23.4. OSCXCN: External Oscillator Control 294
SFR Definition 24.1. RTC0KEY: SmaRTClock Lock and Key 298
SFR Definition 24.2. RTC0ADR: SmaRTClock Address 298
SFR Definition 24.3. RTC0DAT: SmaRTClock Data 299
Internal Register Definition 24.4. RTC0CN: SmaRTClock Control 306
Internal Register Definition 24.5. RTC0XCN: SmaRTClock Oscillator Control 307
Internal Register Definition 24.6. RTC0XCF: SmaRTClock Oscillator Configuration . 308
Internal Register Definition 24.7. RTC0CF: SmaRTClock Configuration 309
Internal Register Definition 24.8. CAPTUREn: SmaRTClock Timer Capture 310
Internal Register Definition 24.9. ALARM0Bn: SmaRTClock Alarm 0 Match Value . . 310
Internal Register Definition 24.10. ALARM1Bn: SmaRTClock Alarm 1 Match Value . 311
Internal Register Definition 24.11. ALARM2Bn: SmaRTClock Alarm 2 Match Value . 311
SFR Definition 25.1. PC0MD: PC0 Mode Configuration 321
SFR Definition 25.2. PC0PCF: PC0 Mode Pull-Up Configuration 322
SFR Definition 25.3. PC0TH: PC0 Threshold Configuration 323
SFR Definition 25.4. PC0STAT: PC0 Status 324
SFR Definition 25.5. PC0DCH: PC0 Debounce Configuration High 325
SFR Definition 25.6. PC0DCL: PC0 Debounce Configuration Low 326
SFR Definition 25.7. PC0CTR0H: PC0 Counter 0 High (MSB) 327
SFR Definition 25.8. PC0CTR0M: PC0 Counter 0 Middle 327
SFR Definition 25.9. PC0CTR0L: PC0 Counter 0 Low (LSB) 327
SFR Definition 25.10. PC0CTR1H: PC0 Counter 1 High (MSB) 328
SFR Definition 25.11. PC0CTR1M: PC0 Counter 1 Middle 328
SFR Definition 25.12. PC0CTR1L: PC0 Counter 1 Low (LSB) 328
SFR Definition 25.13. PC0CMP0H: PC0 Comparator 0 High (MSB) 329
SFR Definition 25.14. PC0CMP0M: PC0 Comparator 0 Middle 329
SFR Definition 25.15. PC0CMP0L: PC0 Comparator 0 Low (LSB) 329
SFR Definition 25.16. PC0CMP1H: PC0 Comparator 1 High (MSB) 330
SFR Definition 25.17. PC0CMP1M: PC0 Comparator 1 Middle 330
SFR Definition 25.18. PC0CMP1L: PC0 Comparator 1 Low (LSB) 330

SFR Definition 25.19. PC0HIST: PC0 History 331
SFR Definition 25.20. PC0INT0: PC0 Interrupt 0 332
SFR Definition 25.21. PC0INT1: PC0 Interrupt 1 333
SFR Definition 26.1. LCD0Dn: LCD0 Data 335
SFR Definition 26.2. LCD0CN: LCD0 Control Register 337
SFR Definition 26.3. LCD0CNTRST: LCD0 Contrast Adjustment 341
SFR Definition 26.4. LCD0MSCN: LCD0 Master Control 342
SFR Definition 26.5. LCD0MSCF: LCD0 Master Configuration 343
SFR Definition 26.6. LCD0PWR: LCD0 Power 343
SFR Definition 26.7. LCD0VBMCN: LCD0 VBAT Monitor Control 344
SFR Definition 26.8. LCD0CLKDIVH: LCD0 Refresh Rate Prescaler High Byte 345
SFR Definition 26.9. LCD0CLKDIVL: LCD Refresh Rate Prescaler Low Byte 345
SFR Definition 26.10. LCD0BLINK: LCD0 Blink Mask 346
SFR Definition 26.11. LCD0TOGR: LCD0 Toggle Rate 347
SFR Definition 26.12. LCD0CF: LCD0 Configuration 348
SFR Definition 26.13. LCD0CHPCN: LCD0 Charge Pump Control 348
SFR Definition 26.14. LCD0CHPCF: LCD0 Charge Pump Configuration 349
SFR Definition 26.15. LCD0CHPMD: LCD0 Charge Pump Mode 349
SFR Definition 26.16. LCD0BUFCN: LCD0 Buffer Control 349
SFR Definition 26.17. LCD0BUFCF: LCD0 Buffer Configuration 350
SFR Definition 26.18. LCD0BUFMD: LCD0 Buffer Mode 350
SFR Definition 26.19. LCD0VBMCF: LCD0 VBAT Monitor Configuration 350
SFR Definition 27.1. XBR0: Port I/O Crossbar Register 0 358
SFR Definition 27.2. XBR1: Port I/O Crossbar Register 1 359
SFR Definition 27.3. XBR2: Port I/O Crossbar Register 2 360
SFR Definition 27.4. P0MASK: Port0 Mask Register 361
SFR Definition 27.5. P0MAT: Port0 Match Register 361
SFR Definition 27.6. P1MASK: Port1 Mask Register 362
SFR Definition 27.7. P1MAT: Port1 Match Register 362
SFR Definition 27.8. P0: Port0 364
SFR Definition 27.9. P0SKIP: Port0 Skip 364
SFR Definition 27.10. P0MDIN: Port0 Input Mode 365
SFR Definition 27.11. P0MDOUT: Port0 Output Mode 365
SFR Definition 27.12. P0DRV: Port0 Drive Strength 366
SFR Definition 27.13. P1: Port1 366
SFR Definition 27.14. P1SKIP: Port1 Skip 367
SFR Definition 27.15. P1MDIN: Port1 Input Mode 367
SFR Definition 27.16. P1MDOUT: Port1 Output Mode 368
SFR Definition 27.17. P1DRV: Port1 Drive Strength 368
SFR Definition 27.18. P2: Port2 369
SFR Definition 27.19. P2SKIP: Port2 Skip 369
SFR Definition 27.20. P2MDIN: Port2 Input Mode 370
SFR Definition 27.21. P2MDOUT: Port2 Output Mode 370
SFR Definition 27.22. P2DRV: Port2 Drive Strength 371
SFR Definition 27.23. P3: Port3 371

SFR Definition 27.24. P3MDIN: Port3 Input Mode 372
SFR Definition 27.25. P3MDOUT: Port3 Output Mode 372
SFR Definition 27.26. P3DRV: Port3 Drive Strength 373
SFR Definition 27.27. P4: Port4 373
SFR Definition 27.28. P4MDIN: Port4 Input Mode 374
SFR Definition 27.29. P4MDOUT: Port4 Output Mode 374
SFR Definition 27.30. P4DRV: Port4 Drive Strength 375
SFR Definition 27.31. P5: Port5 375
SFR Definition 27.32. P5MDIN: Port5 Input Mode 376
SFR Definition 27.33. P5MDOUT: Port5 Output Mode 376
SFR Definition 27.34. P5DRV: Port5 Drive Strength 377
SFR Definition 27.35. P6: Port6 377
SFR Definition 27.36. P6MDIN: Port6 Input Mode 378
SFR Definition 27.37. P6MDOUT: Port6 Output Mode 378
SFR Definition 27.38. P6DRV: Port6 Drive Strength 379
SFR Definition 27.39. P7: Port7 379
SFR Definition 27.40. P7MDOUT: Port7 Output Mode 380
SFR Definition 27.41. P7DRV: Port7 Drive Strength 380
SFR Definition 28.1. SMB0CF: SMBus Clock/Configuration 387
SFR Definition 28.2. SMB0CN: SMBus Control 389
SFR Definition 28.3. SMB0ADR: SMBus Slave Address 391
SFR Definition 28.4. SMB0ADM: SMBus Slave Address Mask 392
SFR Definition 28.5. SMB0DAT: SMBus Data 393
SFR Definition 29.1. SCON0: Serial Port 0 Control 407
SFR Definition 29.2. SBUF0: Serial (UART0) Port Data Buffer 408
SFR Definition 30.1. SPI0CFG: SPI0 Configuration 418
SFR Definition 30.2. SPI0CN: SPI0 Control 419
SFR Definition 30.3. SPI0CKR: SPI0 Clock Rate 420
SFR Definition 30.4. SPI0DAT: SPI0 Data 420
SFR Definition 31.1. SPI1CFG: SPI1 Configuration 438
SFR Definition 31.2. SPI1CN: SPI1 Control 439
SFR Definition 31.3. SPI1CKR: SPI1 Clock Rate 440
SFR Definition 31.4. SPI1DAT: SPI1 Data 440
SFR Definition 32.1. CKCON: Clock Control 445
SFR Definition 32.2. TCON: Timer Control 450
SFR Definition 32.3. TMOD: Timer Mode 451
SFR Definition 32.4. TL0: Timer 0 Low Byte 452
SFR Definition 32.5. TL1: Timer 1 Low Byte 452
SFR Definition 32.6. TH0: Timer 0 High Byte 453
SFR Definition 32.7. TH1: Timer 1 High Byte 453
SFR Definition 32.8. TMR2CN: Timer 2 Control 457
SFR Definition 32.9. TMR2RLL: Timer 2 Reload Register Low Byte 458
SFR Definition 32.10. TMR2RLH: Timer 2 Reload Register High Byte 458
SFR Definition 32.11. TMR2L: Timer 2 Low Byte 459
SFR Definition 32.12. TMR2H Timer 2 High Byte 459

SFR Definition 32.13. TMR3CN: Timer 3 Control 463
SFR Definition 32.14. TMR3RLL: Timer 3 Reload Register Low Byte 464
SFR Definition 32.15. TMR3RLH: Timer 3 Reload Register High Byte 464
SFR Definition 32.16. TMR3L: Timer 3 Low Byte 465
SFR Definition 32.17. TMR3H Timer 3 High Byte 465
SFR Definition 33.1. PCA0CN: PCA Control 480
SFR Definition 33.2. PCA0MD: PCA Mode 481
SFR Definition 33.3. PCA0PWM: PCA PWM Configuration 482
SFR Definition 33.4. PCA0CPMn: PCA Capture/Compare Mode 483
SFR Definition 33.5. PCA0L: PCA Counter/Timer Low Byte 484
SFR Definition 33.6. PCA0H: PCA Counter/Timer High Byte 484
SFR Definition 33.7. PCA0CPLn: PCA Capture Module Low Byte 485
SFR Definition 33.8. PCA0CPHn: PCA Capture Module High Byte 485
C2 Register Definition 34.1. C2ADD: C2 Address 486
C2 Register Definition 34.2. DEVICEID: C2 Device ID 487
C2 Register Definition 34.3. REVID: C2 Revision ID 487
C2 Register Definition 34.4. FPCTL: C2 Flash Programming Control 488
C2 Register Definition 34.5. FPDAT: C2 Flash Programming Data 488

1. System Overview

C8051F96x devices are fully integrated mixed-signal system-on-a-chip MCUs. Highlighted features are listed below. Refer to Table 2.1 for specific product feature selection and part ordering numbers.

- Power efficient on-chip dc-dc buck converter
- High-speed pipelined 8051-compatible microcontroller core (up to 25 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 10-bit 300 ksps, or 12-bit 75 ksps single-ended ADC with 16 external analog inputs and 4 internal inputs such as various power supply voltages and the temperature sensor
- 6-bit programmable current reference
- Precision programmable 24.5 MHz internal oscillator with spread spectrum technology
- 128, 64, 32, or 16 kB of on-chip flash memory
- 8448 or 4352 bytes of on-chip RAM
- 128 segment LCD driver
- SMBus/I^2C, enhanced UART, and two enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- Programmable counter/timer array (PCA) with six capture/compare modules and watchdog timer function
- Hardware AES, DMA, and pulse counter
- On-chip power-on reset, V_{DD} monitor, and temperature sensor
- Two on-chip voltage comparators
- 57 or 34 Port I/O

With on-chip power-on reset, V_{DD} monitor, watchdog timer, and clock oscillator, the C8051F96x devices are truly standalone system-on-a-chip solutions. The flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The on-chip Silicon Labs 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins.

Each device is specified for 1.8 to 3.8 V operation over the industrial temperature range (–40 to +85 °C). The Port I/O and $\overline{\text{RST}}$ pins are tolerant of input signals up to VIO + 2.0 V. The C8051F960/2/4/6/8 are available in a 76-pin DQFN package and an 80-pin TQFP package. The C8051F961/3/5/7/9 are available in a 40-pin QFN package. All package options are lead-free and RoHS compliant. See Table 2.1 for ordering information. Block diagrams are included in Figure 1.1 through Figure 1.16.

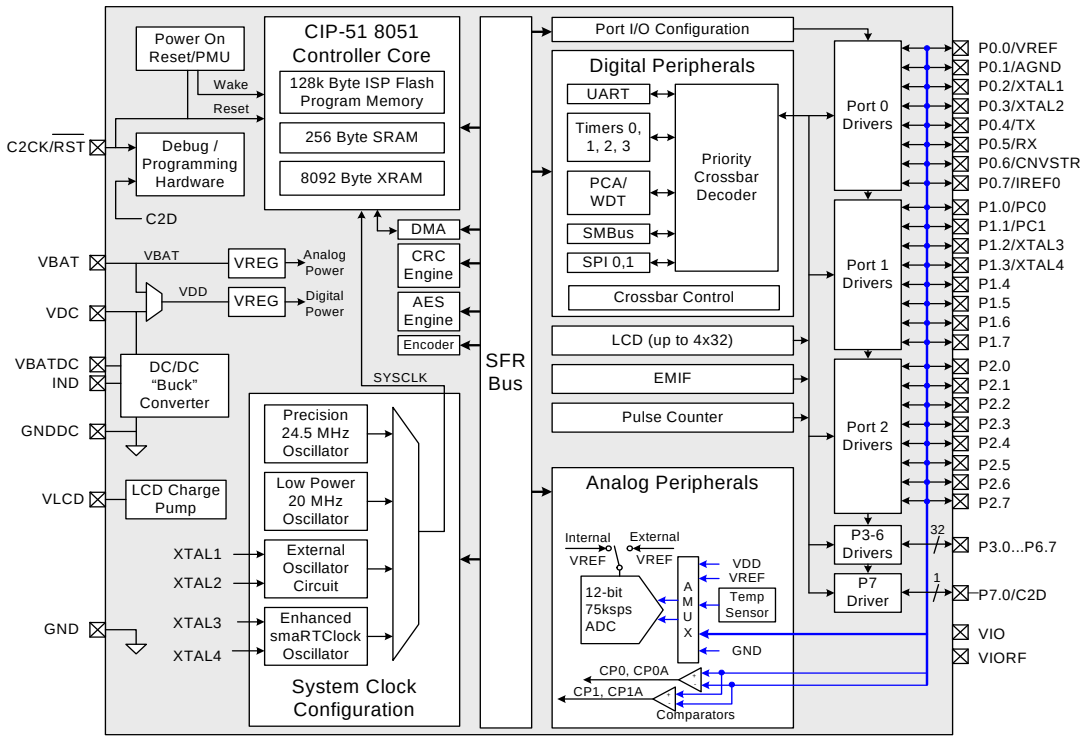


Figure 1.1. C8051F960 Block Diagram

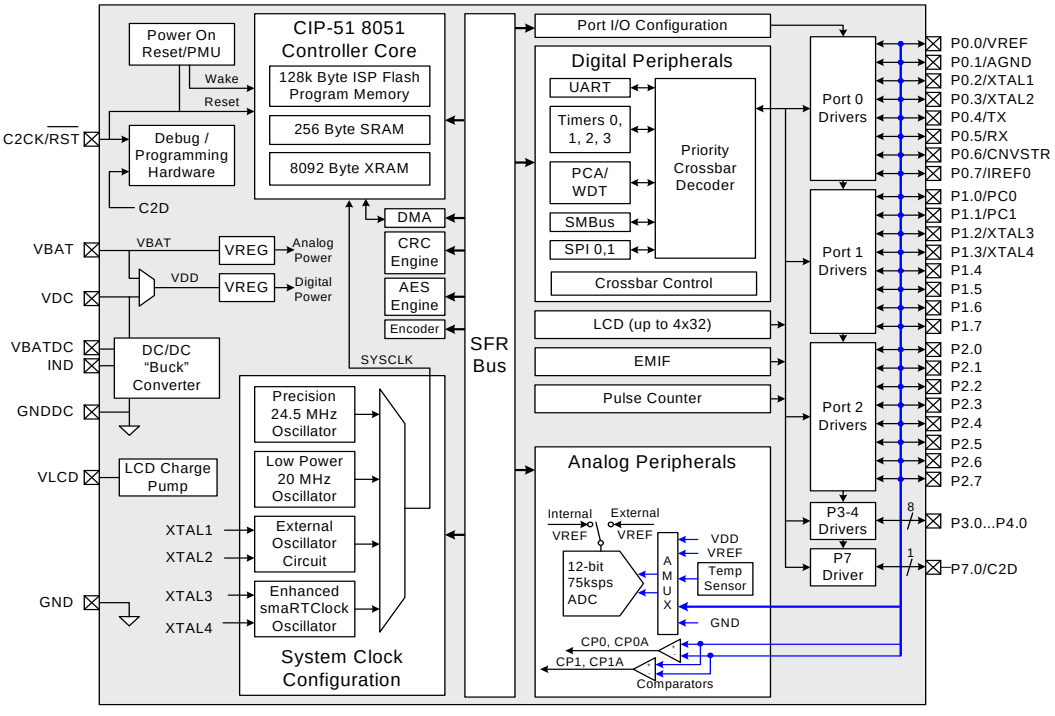


Figure 1.2. C8051F961 Block Diagram

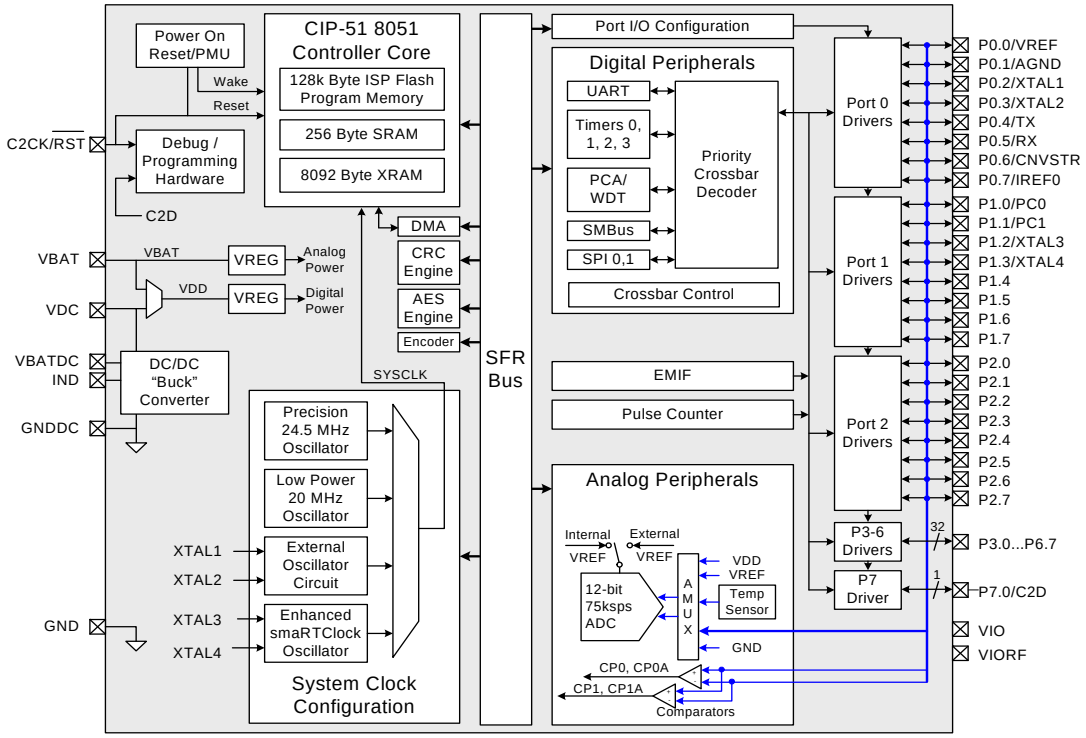


Figure 1.3. C8051F962 Block Diagram

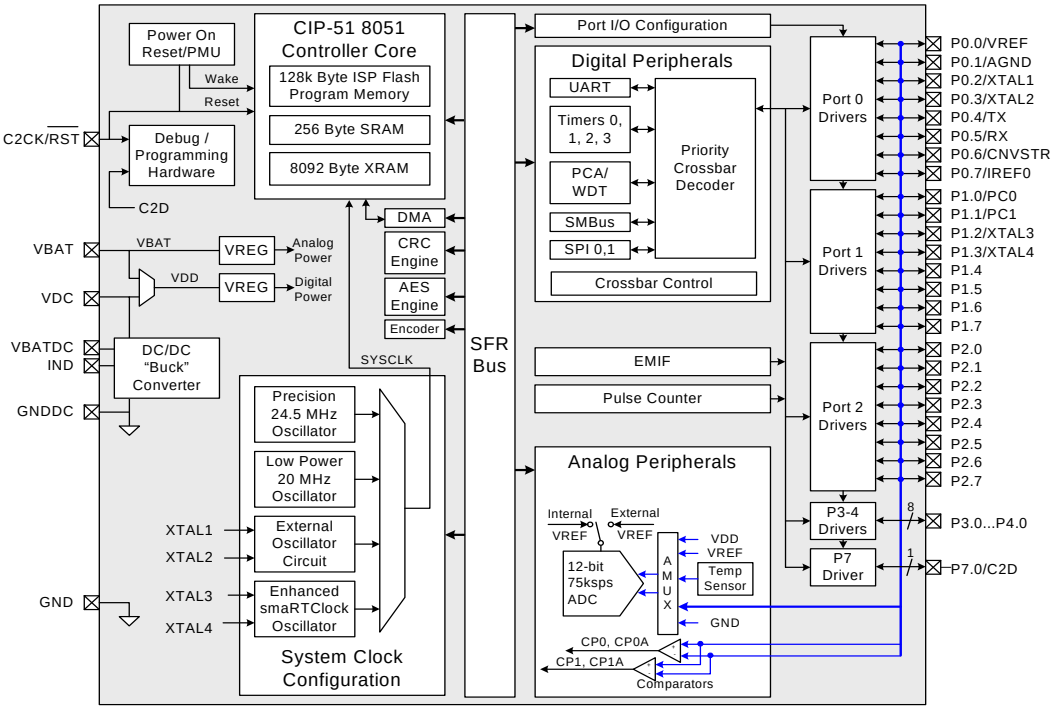


Figure 1.4. C8051F963 Block Diagram

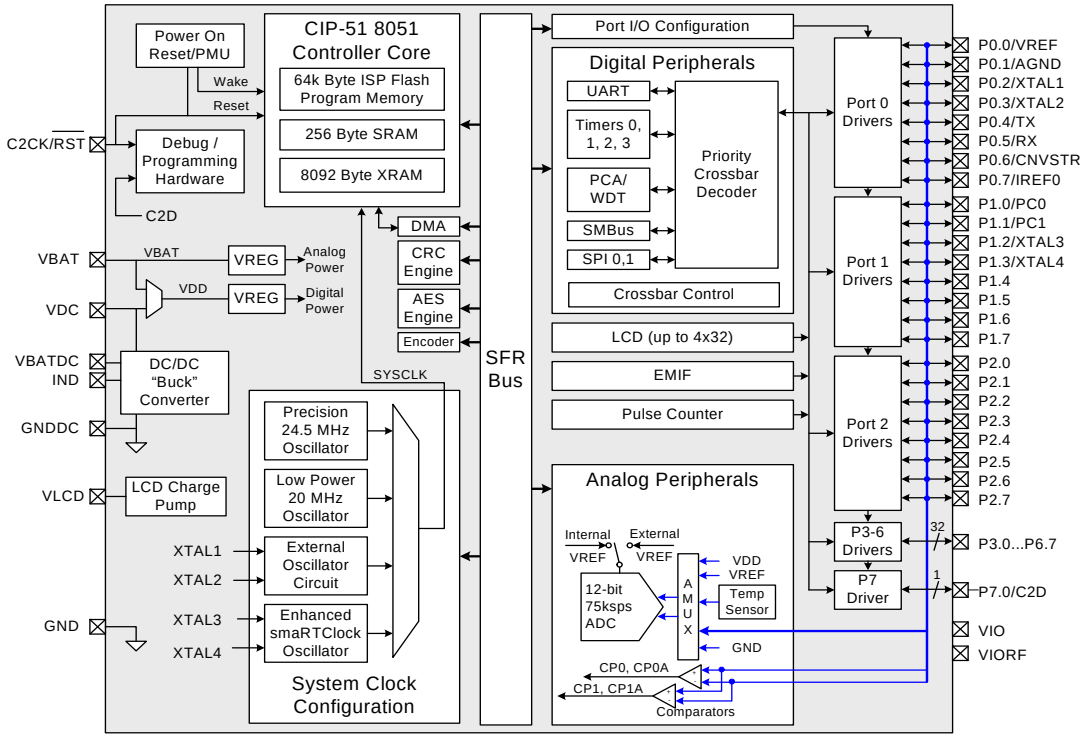


Figure 1.5. C8051F964 Block Diagram

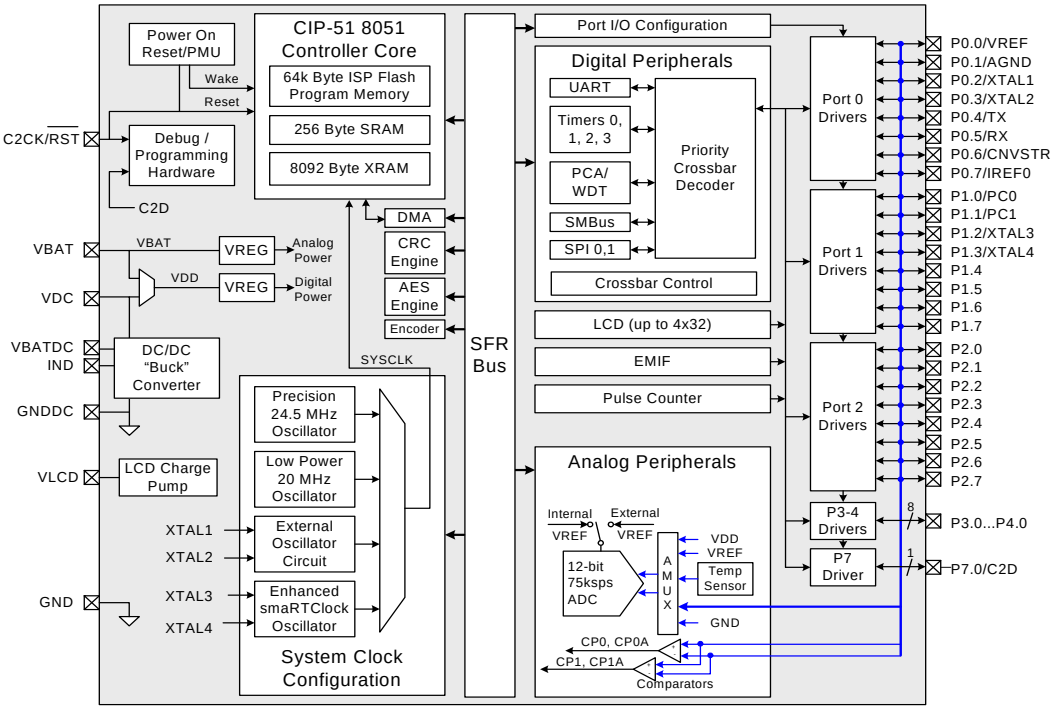


Figure 1.6. C8051F965 Block Diagram

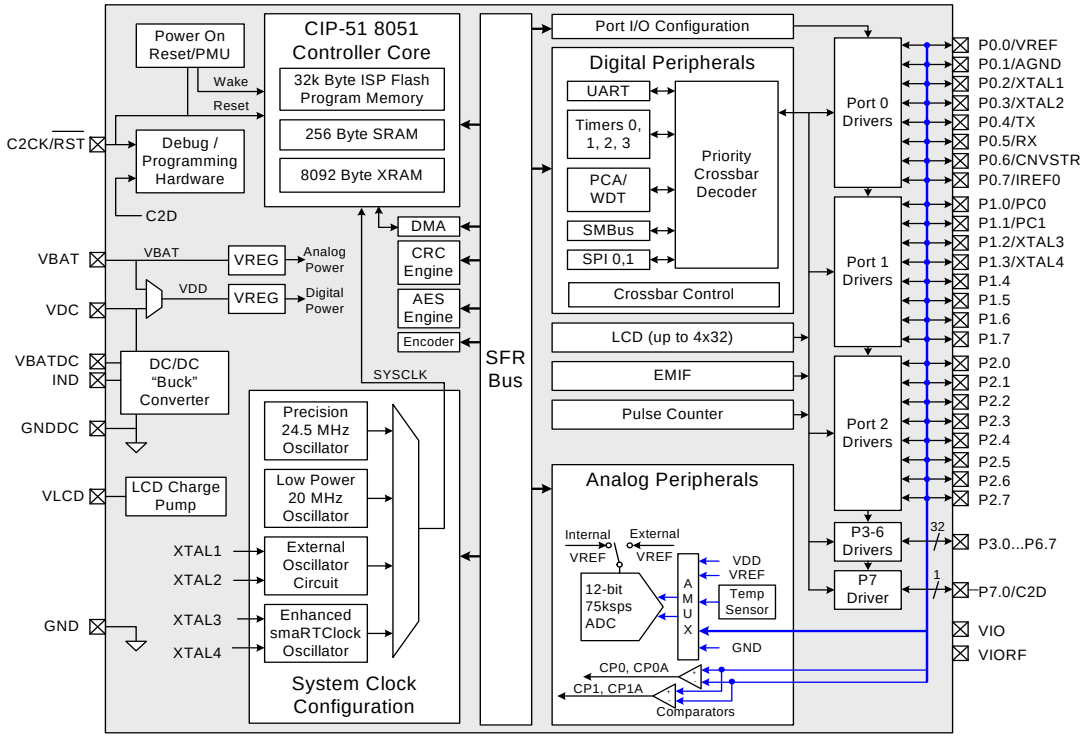


Figure 1.7. C8051F966 Block Diagram

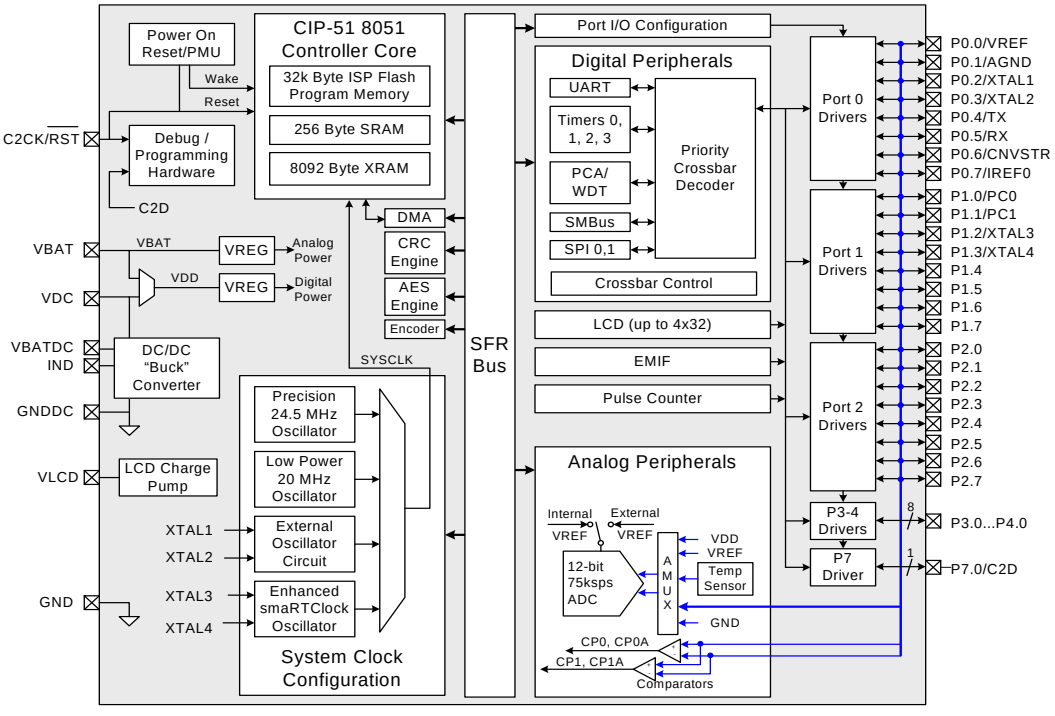


Figure 1.8. C8051F967 Block Diagram

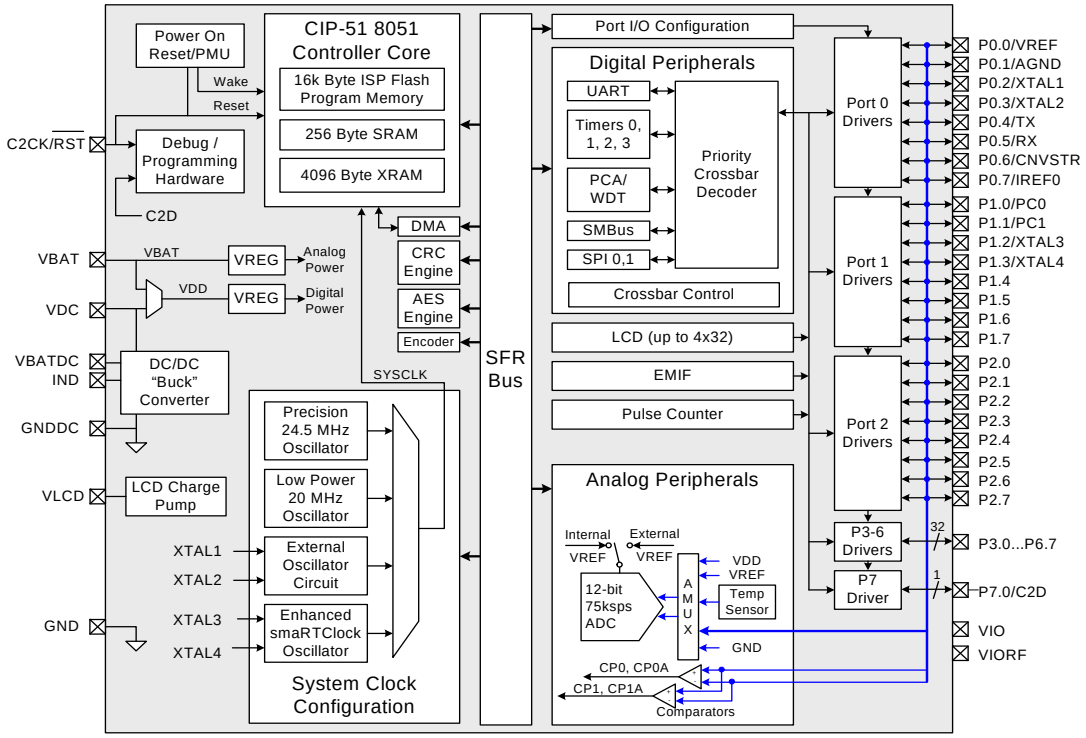


Figure 1.9. C8051F968 Block Diagram

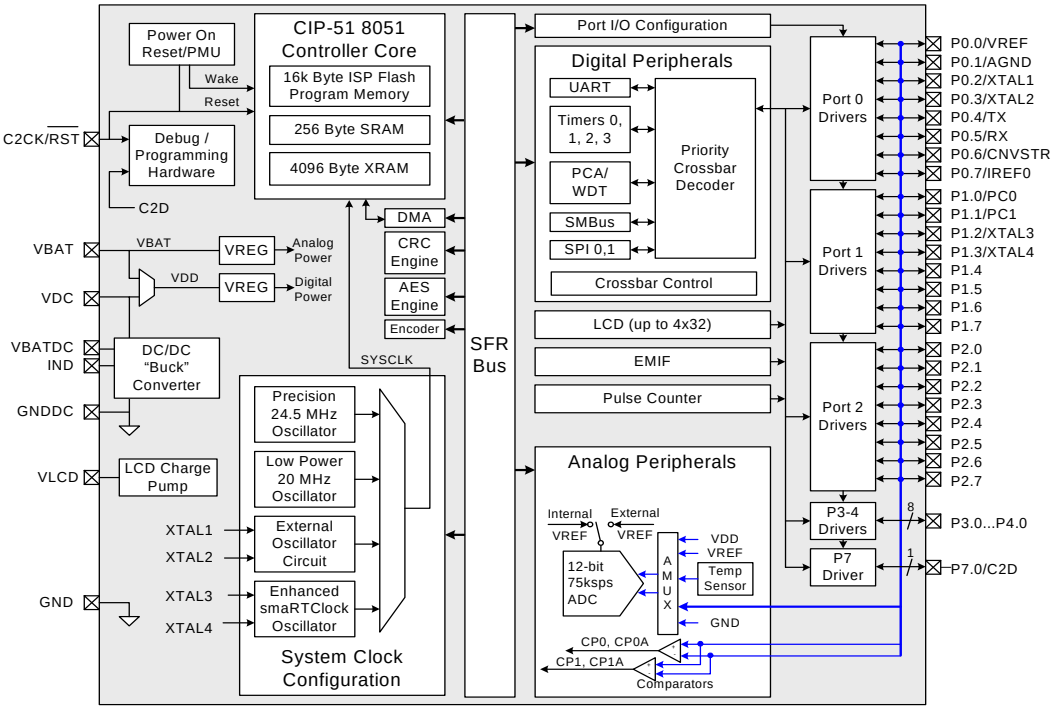


Figure 1.10. C8051F969 Block Diagram

1.1. CIP-51™ Microcontroller Core

1.1.1. Fully 8051 Compatible

The C8051F96x family utilizes Silicon Labs' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The CIP-51 core offers all the peripherals included with a standard 8052.

1.1.2. Improved Throughput

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute with a maximum system clock of 12–24 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with only four instructions taking more than four system clock cycles.

The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS.

1.1.3. Additional Features

The C8051F96x SoC family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

The extended interrupt handler provides multiple interrupt sources into the CIP-51 allowing numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

Eight reset sources are available: power-on reset circuitry (POR), an on-chip V_{DD} monitor (forces reset when power supply voltage drops below safe levels), a watchdog timer (WDT), a missing clock detector, SmaRTClock oscillator fail or alarm, a voltage level detection from Comparator0, a forced software reset, an external reset pin, and an illegal flash access protection circuit. Each reset source except for the POR, reset input pin, or flash error may be disabled by the user in software. The WDT may be permanently disabled in software after a power-on reset during MCU initialization.

The internal oscillator factory calibrated to 24.5 MHz and is accurate to ±2% over the full temperature and supply range. The internal oscillator period can also be adjusted by user firmware. An additional 20 MHz low power oscillator is also available which facilitates low-power operation. An external oscillator drive circuit is included, allowing an external crystal, ceramic resonator, capacitor, RC, or CMOS clock source to generate the system clock. If desired, the system clock source may be switched on-the-fly between both internal and external oscillator circuits. An external oscillator can also be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) source, while periodically switching to the fast (up to 25 MHz) internal oscillator as needed.

1.2. Port Input/Output

Digital and analog resources are available through 57 I/O pins (C8051F960/2/4/6/8) or 34 I/O pins (C8051F961/3/5/7/9). Port pins are organized as eight byte-wide ports. Port pins can be defined as digital or analog I/O. Digital I/O pins can be assigned to one of the internal digital resources or used as general purpose I/O (GPIO). Analog I/O pins are used by the internal analog resources. P7.0 can be used as GPIO and is shared with the C2 Interface Data signal (C2D). See Section “34. C2 Interface” on page 486 for more details.

The designer has complete control over which digital and analog functions are assigned to individual port pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. See Section “27. Port Input/Output” on page 351 for more information on the Crossbar.

For Port I/Os configured as push-pull outputs, current is sourced from the VIO, VIORF, or VBAT supply pin. Port I/Os used for analog functions can operate up to the supply voltage. See Section “27. Port Input/Output” on page 351 for more information on Port I/O operating modes and the electrical specifications chapter for detailed electrical specifications.

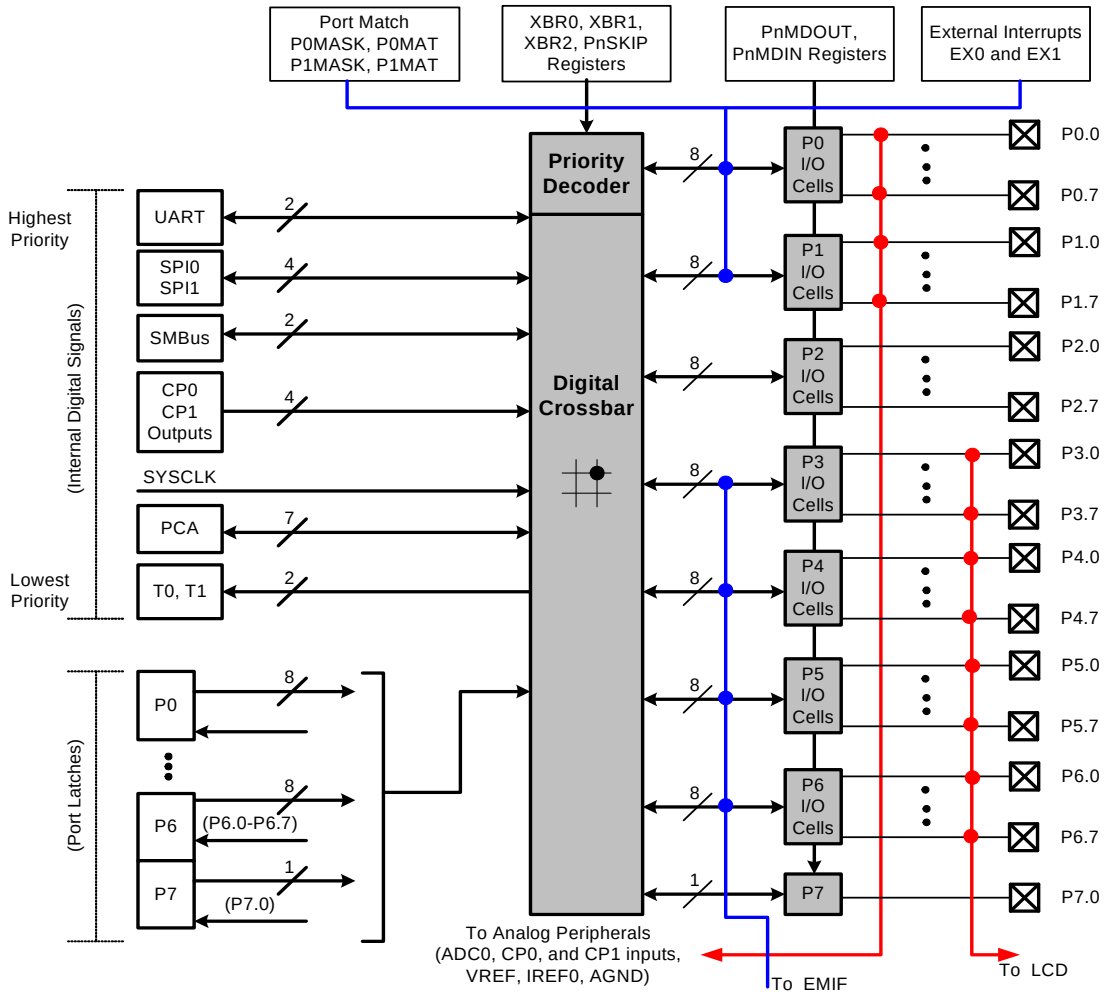


Figure 1.11. Port I/O Functional Block Diagram

1.3. Serial Ports

The C8051F96x Family includes an SMBus/I^2C interface, a full-duplex UART with enhanced baud rate configuration, and two Enhanced SPI interfaces. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little CPU intervention.

1.4. Programmable Counter Array

An on-chip Programmable Counter/Timer Array (PCA) is included in addition to the four 16-bit general purpose counter/timers. The PCA consists of a dedicated 16-bit counter/timer time base with six programmable capture/compare modules. The PCA clock is derived from one of six sources: the system clock divided by 12, the system clock divided by 4, Timer 0 overflows, an External Clock Input (ECI), the system clock, or the external oscillator clock source divided by 8.

Each capture/compare module can be configured to operate in a variety of modes: edge-triggered capture, software timer, high-speed output, pulse width modulator (8, 9, 10, 11, or 16-bit), or frequency output. Additionally, Capture/Compare Module 5 offers watchdog timer (WDT) capabilities. Following a system reset, Module 5 is configured and enabled in WDT mode. The PCA Capture/Compare Module I/O and External Clock Input may be routed to Port I/O via the Digital Crossbar.

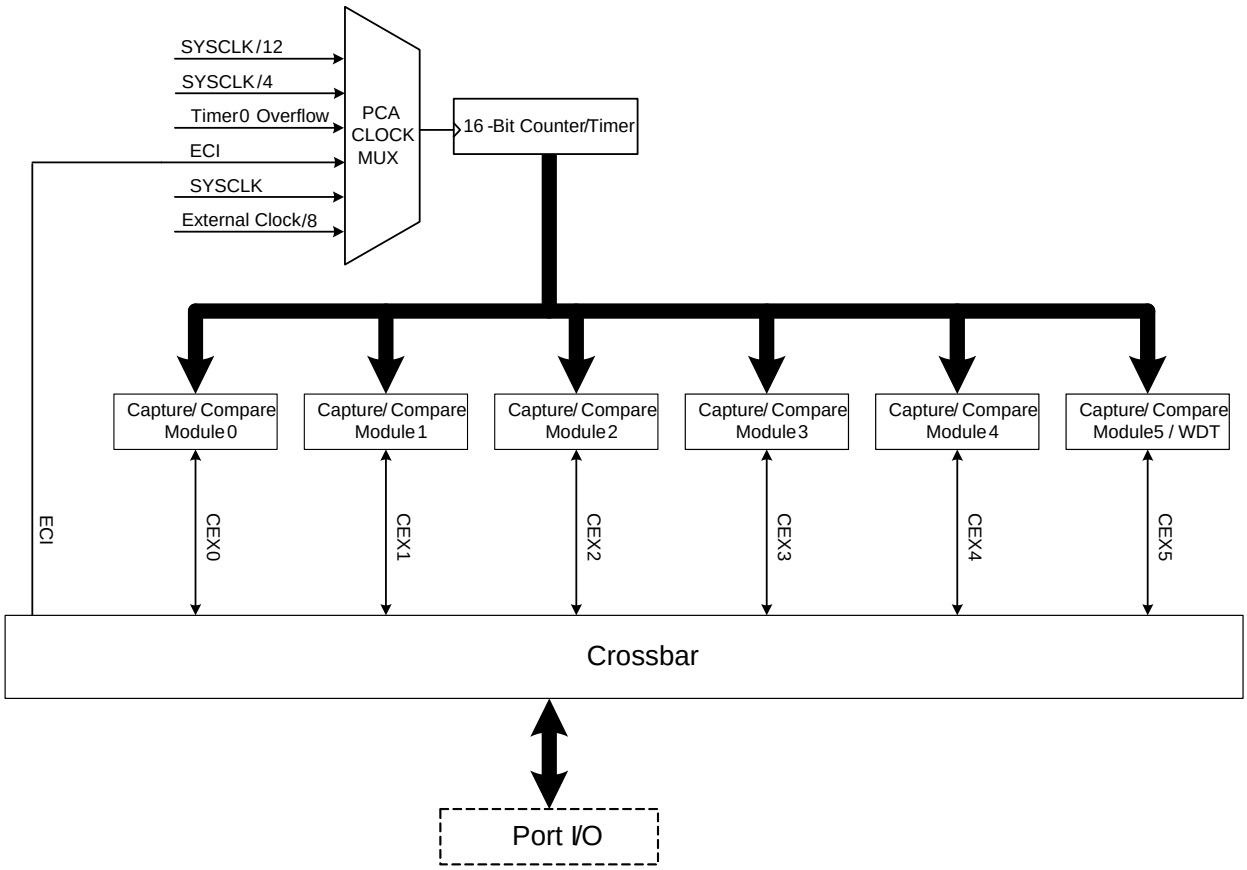


Figure 1.12. PCA Block Diagram

1.5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode

The ADC0 on C8051F96x devices is a 300 ksps, 10-bit or 75 ksps, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold and programmable window detector. ADC0 also has an autonomous low power Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. It also has a 16-bit accumulator that can automatically oversample and average the ADC results. See Section “5.4. 12-Bit Mode” on page 84 for more details on using the ADC in 12-bit mode.

The ADC is fully configurable under software control via Special Function Registers. The ADC0 operates in single-ended mode and may be configured to measure various different signals using the analog multiplexer described in Section “5.7. ADC0 Analog Multiplexer” on page 95. The voltage reference for the ADC is selected as described in Section “5.9. Voltage and Ground Reference Options” on page 100.

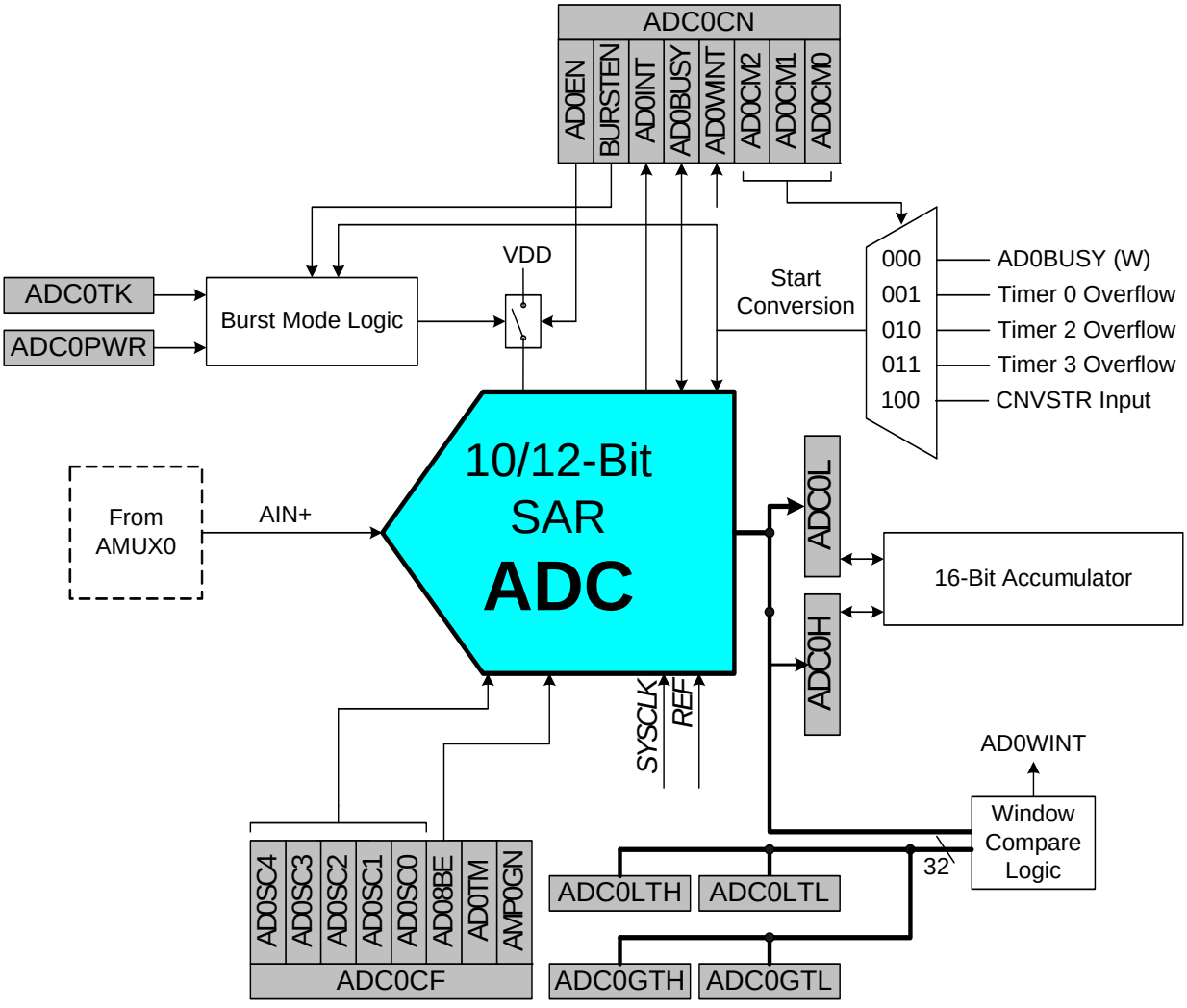


Figure 1.13. ADC0 Functional Block Diagram

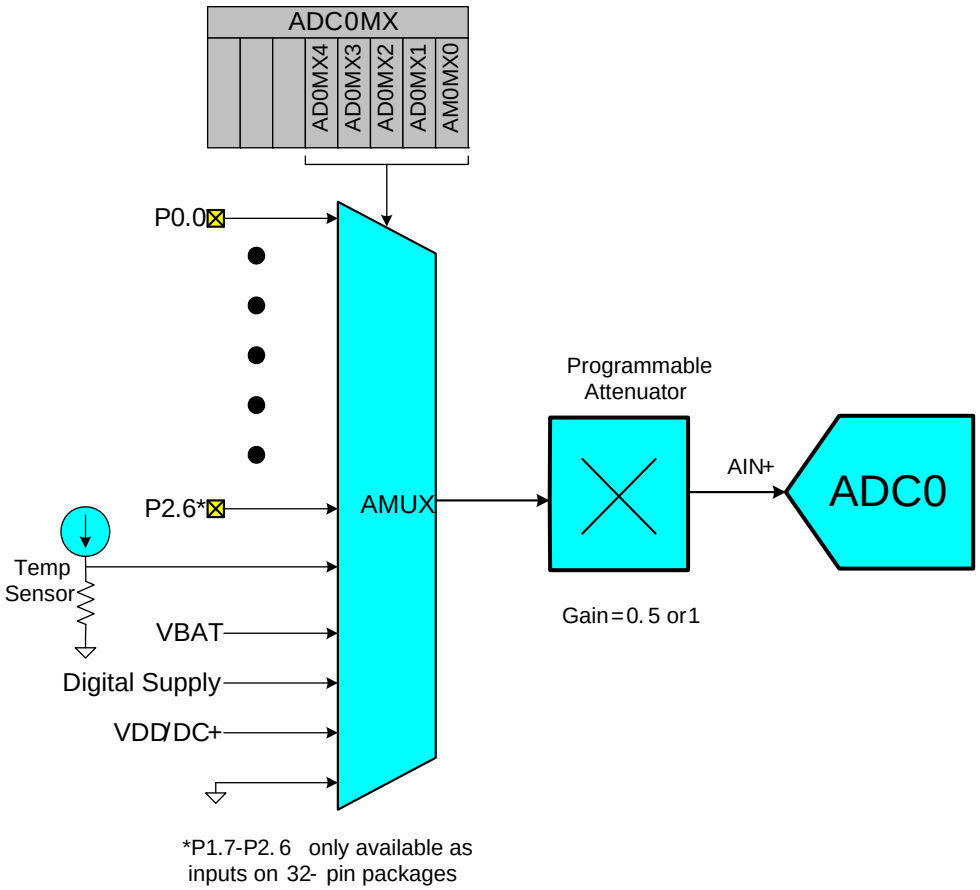


Figure 1.14. ADC0 Multiplexer Block Diagram

1.6. Programmable Current Reference (IREF0)

C8051F96x devices include an on-chip programmable current reference (source or sink) with two output current settings: low power mode and high current mode. The maximum current output in low power mode is 63 µA (1 µA steps) and the maximum current output in high current mode is 504 µA (8 µA steps).

1.7. Comparators

C8051F96x devices include two on-chip programmable voltage comparators: Comparator 0 (CPT0) which is shown in Figure 1.15; Comparator 1 (CPT1) which is shown in Figure 1.16. The two comparators operate identically but may differ in their ability to be used as reset or wake-up sources. See Section “22. Reset Sources” on page 278 and the Section “19. Power Management” on page 257 for details on reset sources and low power mode wake-up sources, respectively.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output when the device is in some low power modes.

The comparator inputs may be connected to Port I/O pins or to other internal signals. Port pins may also be used to directly sense capacitive touch switches. See Application Note AN338 for details on Capacitive Touch Switch sensing.

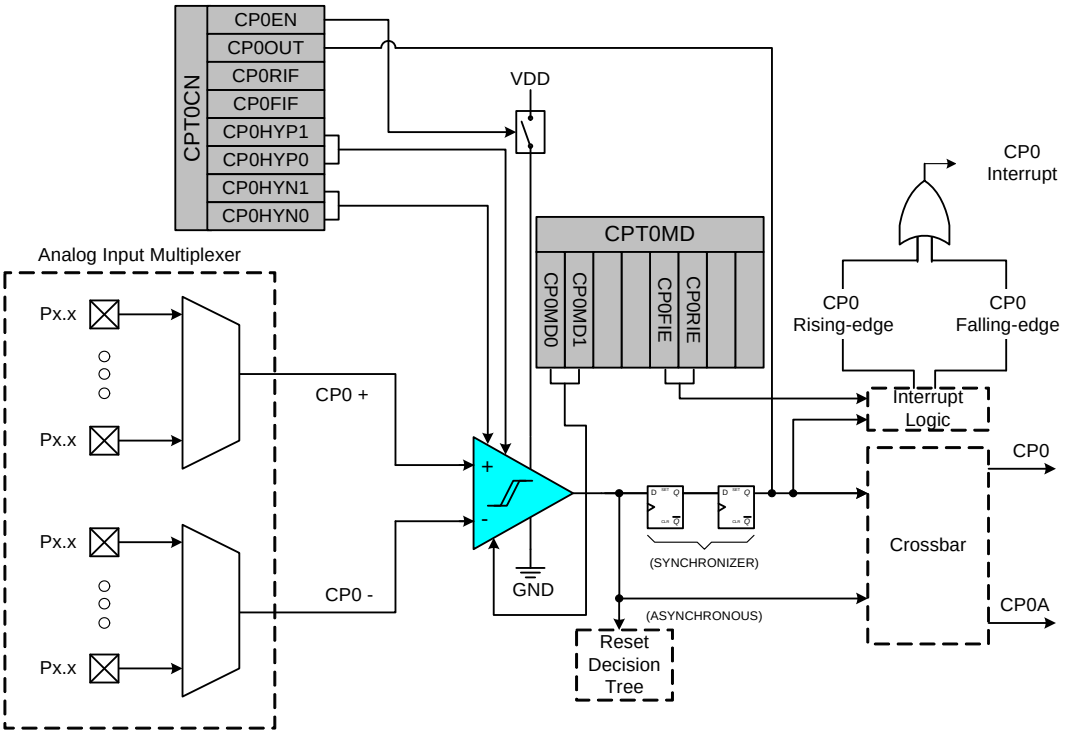


Figure 1.15. Comparator 0 Functional Block Diagram

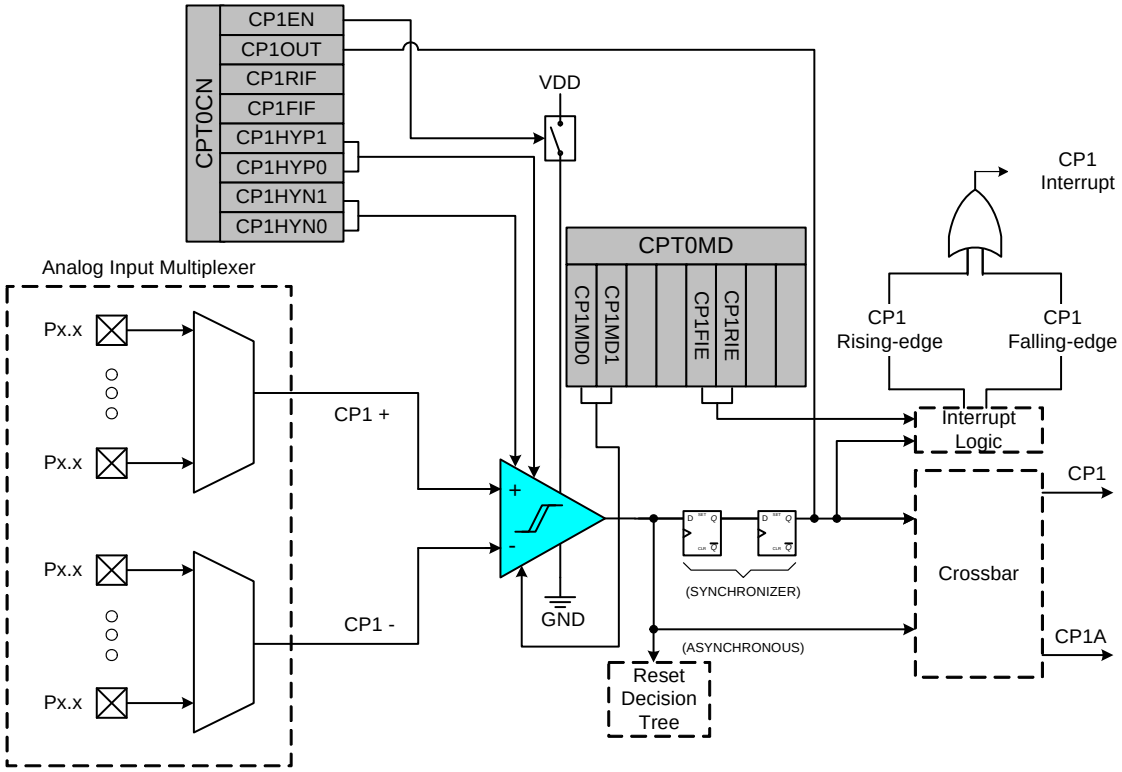


Figure 1.16. Comparator 1 Functional Block Diagram

2. Ordering Information

Table 2.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	Flash Memory (kB)	RAM (bytes)	Digital Port I/Os	AES 128, 192, 256 Encryption	LCD Segments (4-MUX)	SmaRTClock Real Time Clock	SMBus/I^2C	UART	Enhanced SPI	Timers (16-bit)	PCA Channels	10/12-bit 300/75 ksps ADC channels with internal VREF and temp sensor	Analog Comparators	Package
C8051F960-A-GM	25	128	8448	57	✓	128	✓	1	1	2	4	6	16	2	DQFN-76 (6x6)
C8051F960-A-GQ	25	128	8448	57	✓	128	✓	1	1	2	4	6	16	2	TQFP80 (12x12)
C8051F961-A-GM	25	128	8448	34	✓	36	✓	1	1	2	4	6	16	2	QFN-40 (6x6)
C8051F962-A-GM	25	128	8448	57	✓	—	✓	1	1	2	4	6	16	2	DQFN-76 (6x6)
C8051F962-A-GQ	25	128	8448	57	✓	—	✓	1	1	2	4	6	16	2	TQFP80 (12x12)
C8051F963-A-GM	25	128	8448	34	✓	—	✓	1	1	2	4	6	16	2	QFN-40 (6x6)
C8051F964-A-GM	25	64	8448	57	✓	128	✓	1	1	2	4	6	16	2	DQFN-76 (6x6)
C8051F964-A-GQ	25	64	8448	57	✓	128	✓	1	1	2	4	6	16	2	TQFP80 (12x12)
C8051F965-A-GM	25	64	8448	34	✓	36	✓	1	1	2	4	6	16	2	QFN-40 (6x6)
C8051F966-A-GM	25	32	8448	57	✓	128	✓	1	1	2	4	6	16	2	DQFN-76 (6x6)
C8051F966-A-GQ	25	32	8448	57	✓	128	✓	1	1	2	4	6	16	2	TQFP80 (12x12)
C8051F967-A-GM	25	32	8448	34	✓	36	✓	1	1	2	4	6	16	2	QFN-40 (6x6)
C8051F968-A-GM	25	16	4352	57	✓	128	✓	1	1	2	4	6	16	2	DQFN-76 (6x6)
C8051F968-A-GQ	25	16	4352	57	✓	128	✓	1	1	2	4	6	16	2	TQFP80 (12x12)
C8051F969-A-GM	25	16	4352	34	✓	36	✓	1	1	2	4	6	16	2	QFN-40 (6x6)

All packages are Lead-free (RoHS Compliant).

3. Pinout and Package Definitions

Table 3.1. Pin Definitions for the C8051F96x

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
VBAT	A5	8	5	P In	Battery Supply Voltage. Must be 1.8 to 3.6 V.
VBATDC	A6	10	5	P In	DC0 Input Voltage. Must be 1.8 to 3.6 V.
VDC	A8	14	8	P In P Out	Alternate Power Supply Voltage. Must be 1.8 to 3.6 V. This supply voltage must always be ≤ VBAT. Software may select this supply voltage to power the digital logic. Positive output of the dc-dc converter. A 1uF to 10uF ceramic capacitor is required on this pin when using the dc-dc converter. This pin can supply power to external devices when the dc-dc converter is enabled.
GNDDC	A	12	7	P In	DC-DC converter return current path. This pin is typically tied to the ground plane.
GND	B6	13,64,66,68	7	G	Required Ground.
IND	B5	11	6	P In	DC-DC Inductor Pin. This pin requires a 560 nH inductor to VDC if the dc-dc converter is used.
VIO	B4	9	5	P In	I/O Power Supply for P0.0–P1.4 and P2.4–P7.0 pins. This supply voltage must always be ≤ VBAT.
VIORF	B7	15	8	P In	I/O Power Supply for P1.5–P2.3 pins. This supply voltage must always be ≤ VBAT.
$\overline{\text{RST}}$/ C2CK	A9	16	9	D I/O D I/O	Device Reset. Open-drain output of internal POR or V_{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 µs. A 1 kΩ to 5 kΩ pullup to V_{DD} is recommended. See Reset Sources Section for a complete description. Clock signal for the C2 Debug Interface.
P7.0/ C2D	A10	17	10	D I/O D I/O	Port 7.0. This pin can only be used as GPIO. The Crossbar cannot route signals to this pin and it cannot be configured as an analog input. See Port I/O Section for a complete description. Bi-directional data signal for the C2 Debug Interface.
VLCD	A32	61	32	P I/O	LCD Power Supply. This pin requires a 10 µF capacitor to stabilize the charge pump.

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P0.0	A4	6	4	D I/O or A In	Port 0.0. See Port I/O Section for a complete description.
V_{REF}				A In A Out	External V_{REF} Input. Internal V_{REF} Output. External V_{REF} decoupling capacitors are recommended. See ADC0 Section for details.
P0.1	A3	4	3	D I/O or A In	Port 0.1. See Port I/O Section for a complete description.
AGND				G	Optional Analog Ground. See ADC0 Section for details.
P0.2	A2	2	2	D I/O or A In	Port 0.2. See Port I/O Section for a complete description.
XTAL1				A In	External Clock Input. This pin is the external oscillator return for a crystal or resonator. See Oscillator Section.
P0.3	A1	1	1	D I/O or A In	Port 0.3. See Port I/O Section for a complete description.
XTAL2				A Out D In A In	External Clock Output. This pin is the excitation driver for an external crystal or resonator. External Clock Input. This pin is the external clock input in external CMOS clock mode. External Clock Input. This pin is the external clock input in capacitor or RC oscillator configurations. See Oscillator Section for complete details.
P0.4	A40	79	40	D I/O or A In	Port 0.4. See Port I/O Section for a complete description.
TX				D Out	UART TX Pin. See Port I/O Section.
P0.5	A39	78	39	D I/O or A In	Port 0.5. See Port I/O Section for a complete description.
RX				D In	UART RX Pin. See Port I/O Section.
P0.6	A38	76	38	D I/O or A In	Port 0.6. See Port I/O Section for a complete description.
CNVSTR				D In	External Convert Start Input for ADC0. See ADC0 section for a complete description.
P0.7	A37	74	37	D I/O or A In	Port 0.7. See Port I/O Section for a complete description.
IREF0				A Out	IREF0 Output. See IREF Section for complete description.

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P1.0	A36	72	36	D I/O or A In	Port 1.0. See Port I/O Section for a complete description. May also be used as SCK for SPI0.
PC0				D I/O	Pulse Counter 0.
P1.1	A35	70	35	D I/O or A In	Port 1.1. See Port I/O Section for a complete description. May also be used as MISO for SPI0.
PC1				D I/O	Pulse Counter 1.
P1.2	A34	67	34	D I/O or A In	Port 1.2. See Port I/O Section for a complete description. May also be used as MOSI for SPI0.
XTAL3				A In	SmaRTClock Oscillator Crystal Input.
P1.3	A33	65	33	D I/O or A In	Port 1.3. See Port I/O Section for a complete description. May also be used as NSS for SPI0.
XTAL4				A Out	SmaRTClock Oscillator Crystal Output.
P1.4	A31	60	31	D I/O or A In	Port 1.4. See Port I/O Section for a complete description.
P1.5	A30	57	30	D I/O or A In	Port 1.5. See Port I/O Section for a complete description.
P1.6	A29	56	29	D I/O or A In	Port 1.6. See Port I/O Section for a complete description.
P1.7	A28	54	28	D I/O or A In	Port 1.7. See Port I/O Section for a complete description.
P2.0	A27	53	27	D I/O or A In	Port 2.0. See Port I/O Section for a complete description.
P2.1	A26	49	26	D I/O or A In	Port 2.1. See Port I/O Section for a complete description.
P2.2	A25	48	25	D I/O or A In	Port 2.2. See Port I/O Section for a complete description.
P2.3	A24	47	24	D I/O or A In	Port 2.3. See Port I/O Section for a complete description.
P2.4	A23	46	23	D I/O or A In	Port 2.4. See Port I/O Section for a complete description.
COM0				A O	LCD Common Pin 0 (Backplane Driver)
P2.5	A22	45	22	D I/O or A In	Port 2.5. See Port I/O Section for a complete description.
COM1				A O	LCD Common Pin 1 (Backplane Driver)

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P2.6	A21	43	21	D I/O or A In	Port 2.6. See Port I/O Section for a complete description.
COM2				A O	LCD Common Pin 2 (Backplane Driver)
P2.7	A20	41	20	D I/O or A In	Port 2.7. See Port I/O Section for a complete description.
COM2				A O	LCD Common Pin 3 (Backplane Driver)
P3.0	A19	39	19	D I/O or A In	Port 3.0. See Port I/O Section for a complete description.
LCD0				A O	LCD Segment Pin 0
P3.1	A18	38	18	D I/O or A In	Port 3.1. See Port I/O Section for a complete description.
LCD1				A O	LCD Segment Pin 1
P3.2	A17	36	17	D I/O or A In	Port 3.2. See Port I/O Section for a complete description.
LCD2				A O	LCD Segment Pin 2
P3.3	A16	34	16	D I/O or A In	Port 3.3. See Port I/O Section for a complete description.
LCD3				A O	LCD Segment Pin 3
P3.4	A15	32	15	D I/O or A In	Port 3.4. See Port I/O Section for a complete description.
LCD4				A O	LCD Segment Pin 4
P3.5	A14	28	14	D I/O or A In	Port 3.5. See Port I/O Section for a complete description.
LCD5				A O	LCD Segment Pin 5
P3.6	A13	26	13	D I/O or A In	Port 3.6. See Port I/O Section for a complete description.
LCD6				A O	LCD Segment Pin 6
P3.7	A12	24	12	D I/O or A In	Port 3.7. See Port I/O Section for a complete description.
LCD7				A O	LCD Segment Pin 7

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P4.0	A11	23	11	D I/O or A In	Port 4.0. See Port I/O Section for a complete description.
LCD8				A O	LCD Segment Pin 8
P4.1	B3	7		D I/O or A In	Port 4.1. See Port I/O Section for a complete description.
LCD9				A O	LCD Segment Pin 9
P4.2	B2	5		D I/O or A In	Port 4.2. See Port I/O Section for a complete description.
LCD10				A O	LCD Segment Pin 10
P4.3	B1	3		D I/O or A In	Port 4.3. See Port I/O Section for a complete description.
LCD11				A O	LCD Segment Pin 11
P4.4	D1	80		D I/O or A In	Port 4.4. See Port I/O Section for a complete description.
LCD12				A O	LCD Segment Pin 12
P4.5	B28	77		D I/O or A In	Port 4.5. See Port I/O Section for a complete description.
LCD13				A O	LCD Segment Pin 13
P4.6	B27	75		D I/O or A In	Port 4.6. See Port I/O Section for a complete description.
LCD14				A O	LCD Segment Pin 14
P4.7	B26	73		D I/O or A In	Port 4.7. See Port I/O Section for a complete description.
LCD15				A O	LCD Segment Pin 15
P5.0	B25	71		D I/O or A In	Port 5.0. See Port I/O Section for a complete description.
LCD16				A O	LCD Segment Pin 16
P5.1	B24	69		D I/O or A In	Port 5.1. See Port I/O Section for a complete description.
LCD17				A O	LCD Segment Pin 17

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P5.2	B23	63		D I/O or A In	Port 5.2. See Port I/O Section for a complete description.
LCD18				A O	LCD Segment Pin 18
P5.3	B22	62		D I/O or A In	Port 5.3. See Port I/O Section for a complete description.
LCD19				A O	LCD Segment Pin 19
P5.4	D4	59		D I/O or A In	Port 5.4. See Port I/O Section for a complete description.
LCD20				A O	LCD Segment Pin 20
P5.5	B21	55		D I/O or A In	Port 5.5. See Port I/O Section for a complete description.
LCD21				A O	LCD Segment Pin 21
P5.6	B15	44		D I/O or A In	Port 5.6. See Port I/O Section for a complete description.
LCD22				A O	LCD Segment Pin 22
P5.7	D3	42		D I/O or A In	Port 5.7. See Port I/O Section for a complete description.
LCD23				A O	LCD Segment Pin 23
P6.0	B14	40		D I/O or A In	Port 6.0. See Port I/O Section for a complete description.
LCD24				A O	LCD Segment Pin 24
P6.1	B13	37		D I/O or A In	Port 6.1. See Port I/O Section for a complete description.
LCD25				A O	LCD Segment Pin 25
P6.2	B12	35		D I/O or A In	Port 6.2. See Port I/O Section for a complete description.
LCD26				A O	LCD Segment Pin 26
P6.3	B11	33		D I/O or A In	Port 6.3. See Port I/O Section for a complete description.
LCD27				A O	LCD Segment Pin 27

Table 3.1. Pin Definitions for the C8051F96x (Continued)

Name	Pin Numbers			Type	Description
	DQFN76	TQFP80	QFN40		
P6.4	B10	29		D I/O or A In	Port 6.4. See Port I/O Section for a complete description.
LCD28				A O	LCD Segment Pin 28
P6.5	B9	27		D I/O or A In	Port 6.5. See Port I/O Section for a complete description.
LCD29				A O	LCD Segment Pin 29
P6.6	B8	25		D I/O or A In	Port 6.6. See Port I/O Section for a complete description.
LCD30				A O	LCD Segment Pin 30
P6.7	D2	18		D I/O or A In	Port 6.7. See Port I/O Section for a complete description.
LCD31				A O	LCD Segment Pin 31

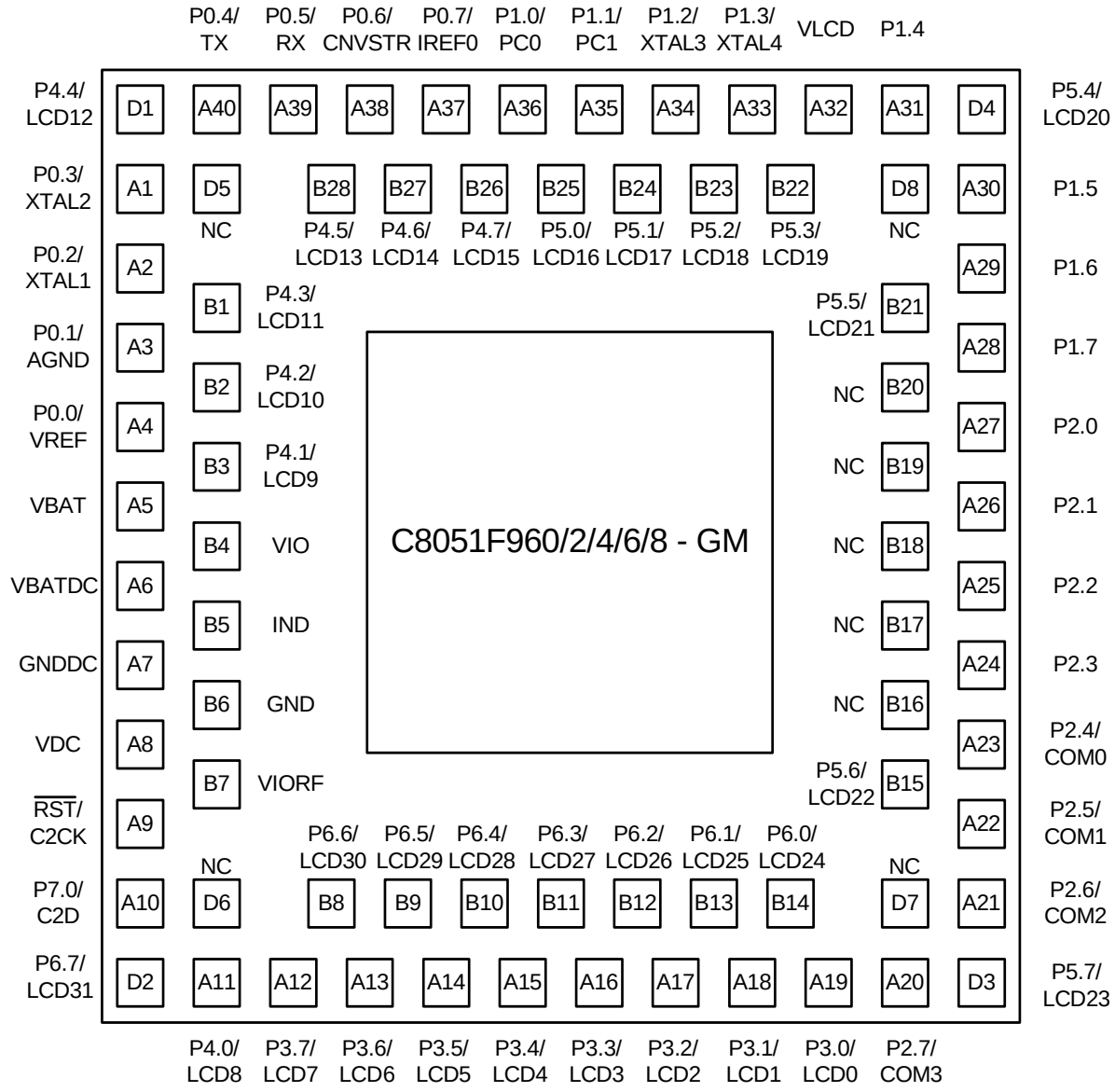


Figure 3.1. DQFN-76 Pinout Diagram (Top View)

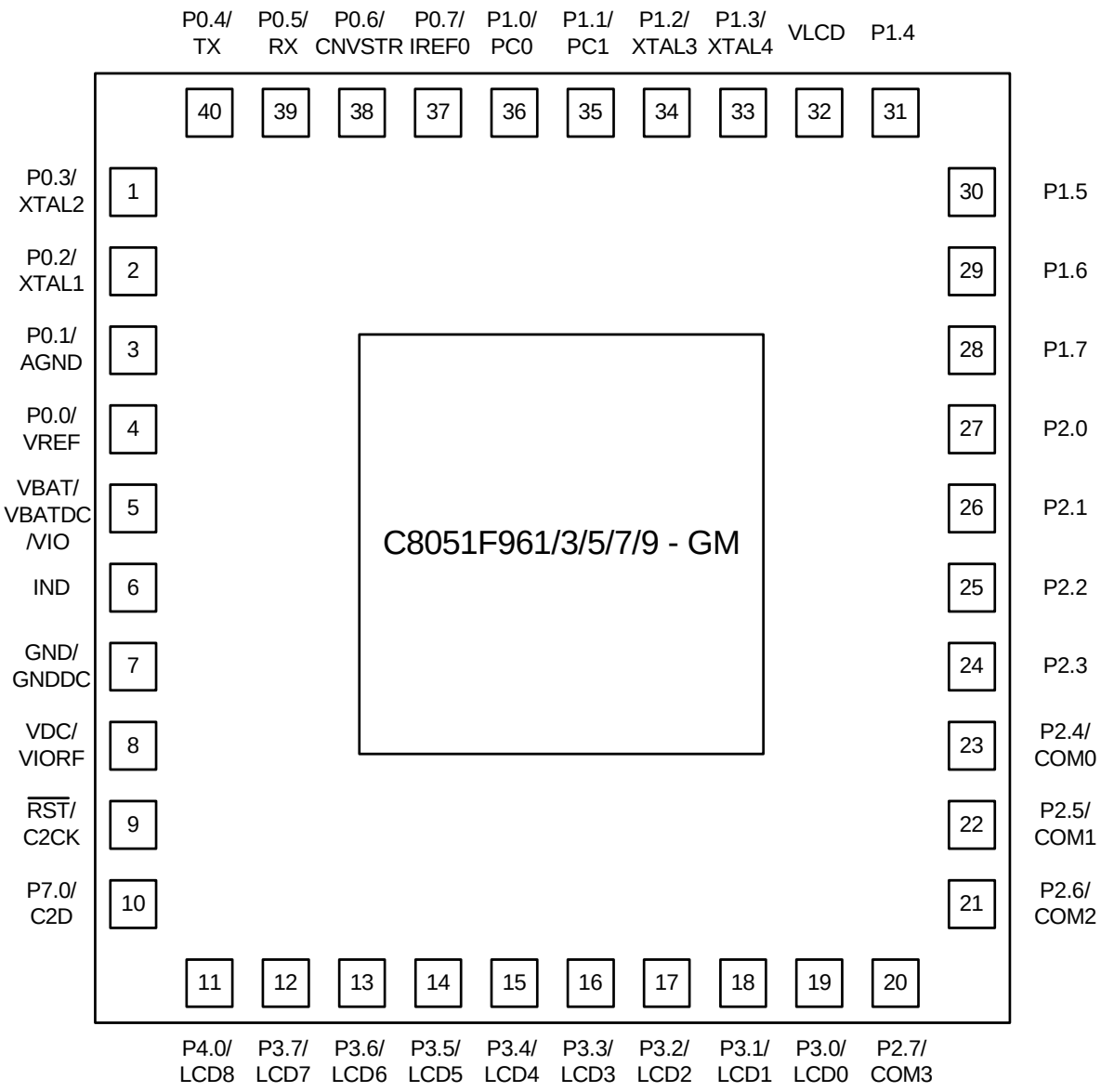


Figure 3.2. QFN-40 Pinout Diagram (Top View)

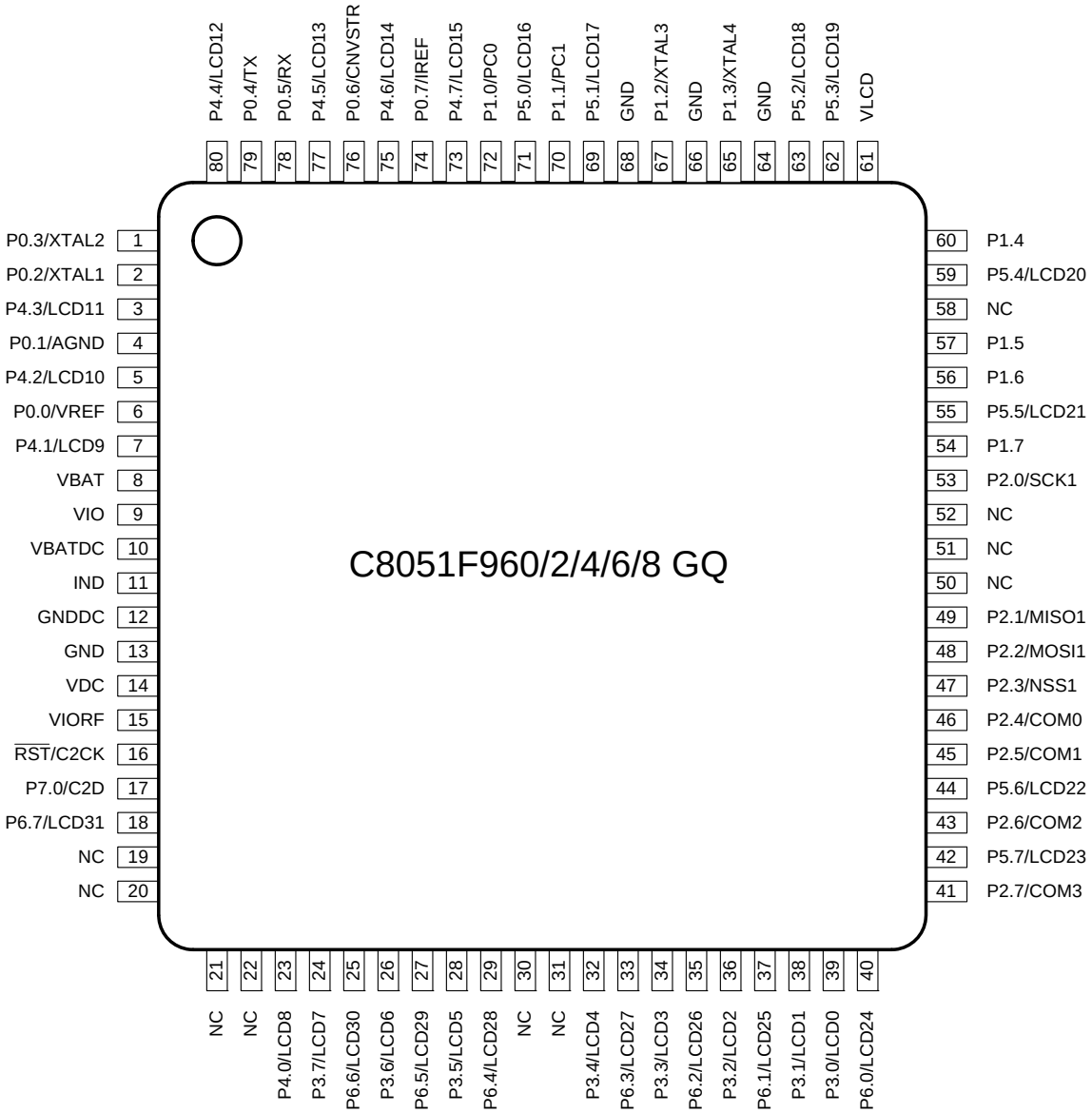


Figure 3.3. TQFP-80 Pinout Diagram (Top View)

3.1. DQFN-76 Package Specifications

3.1.1. Package Drawing

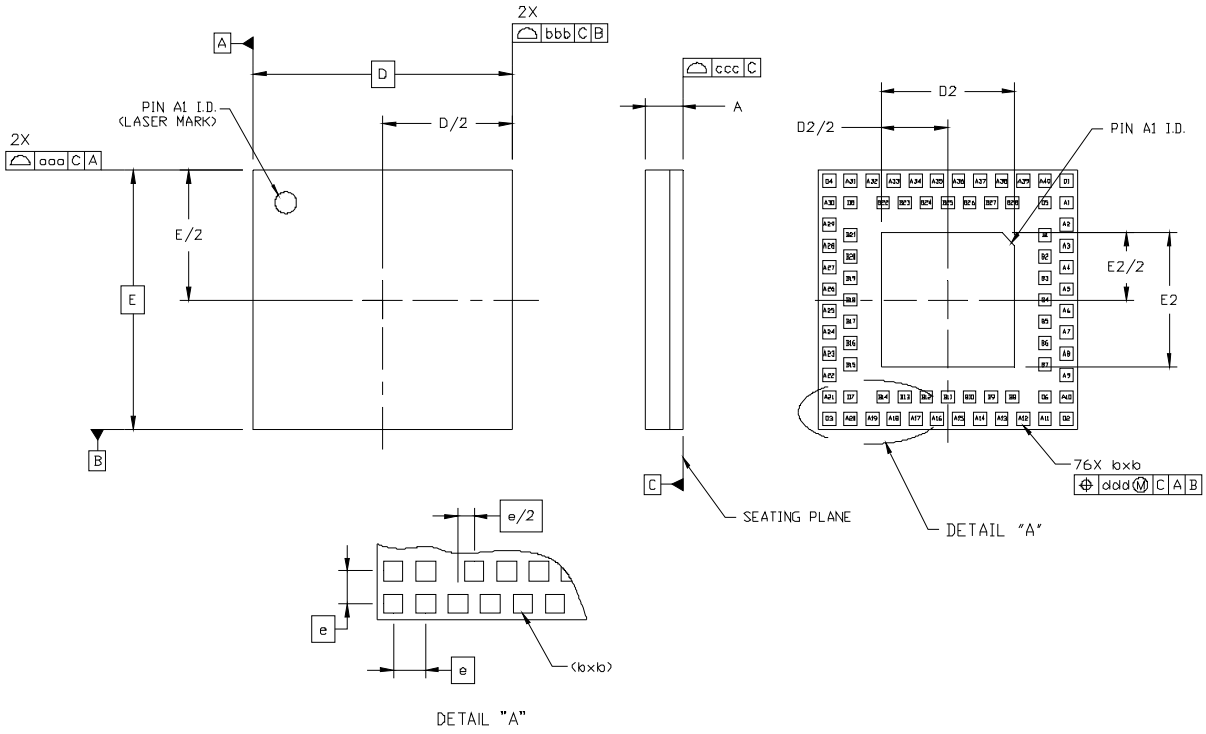


Figure 3.4. DQFN-76 Package Drawing

Table 3.2. DQFN-76 Package Dimensions

Dimension	Min	Typ	Max
A	0.74	0.84	0.94
b	0.25	0.30	0.35
D		6.00 BSC	
D2	3.00	3.10	3.20
e		0.50 BSC	
E		6.00 BSC	

Dimension	Min	Typ	Max
E2	3.00	3.10	3.20
aaa	—	—	0.10
bbb	—	—	0.10
ddd	—	—	0.08
eee	—	—	0.10

Notes:
1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

3.1.2. Land Pattern

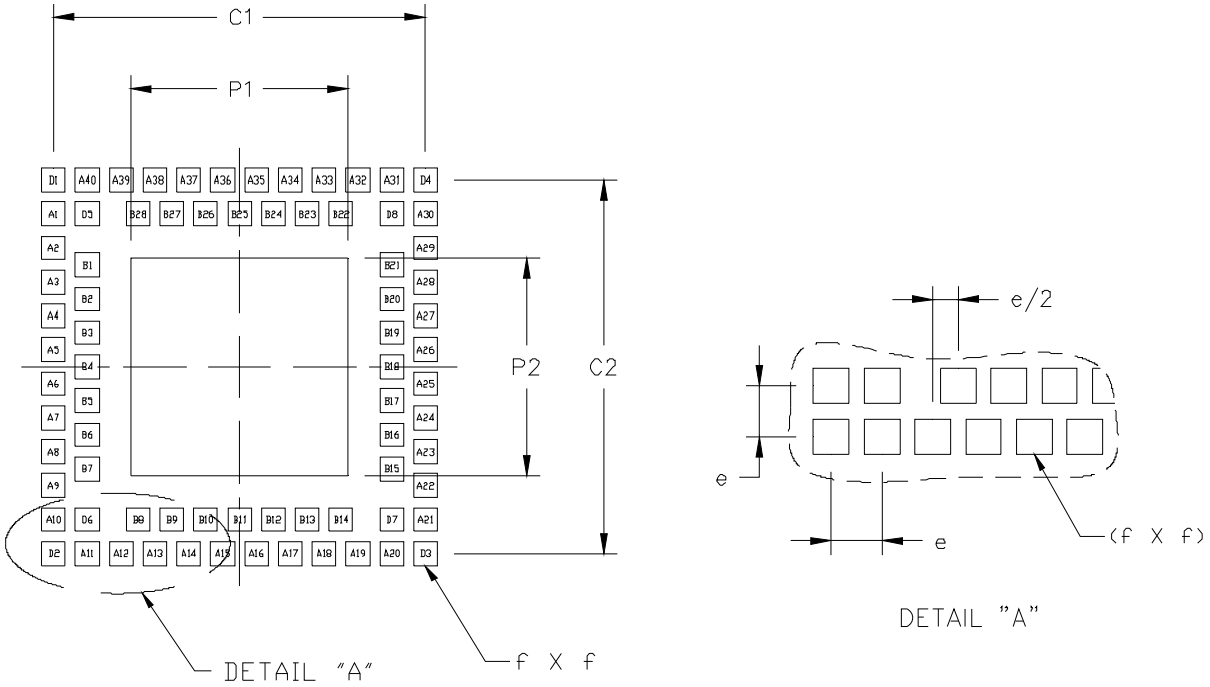


Figure 3.5. DQFN-76 Land Pattern

Table 3.3. DQFN-76 Land Pattern Dimensions

	Dimension (mm)	
Symbol	**Typ**	**Max**
C1	5.50	—
C2	5.50	—
e	0.50	—
f	—	0.35
P1	—	3.20
P2	—	3.20

Notes:

1. All feature sizes shown are at Maximum Material Condition (MMC) and a card fabrication tolerance of 0.05 mm is assumed.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-7351 guidelines.

3.1.3. Soldering Guidelines

3.1.3.1. Solder Mask Design

All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μm minimum, all the way around the pad.

3.1.3.2. Stencil Design

1. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
2. The stencil thickness should be 0.125 mm (5 mils).
3. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
4. A 2x2 array of 1.25 mm square openings on 1.60 mm pitch should be used for the center ground pad.

3.1.3.3. Card Assembly

1. A No-Clean, Type-3 solder paste is recommended.
2. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

3.1.3.4. Inner via placement

1. Inner via placement per Figure 3.6.
2. Reccomended via hole size is 0.150 mm (6 mil) laser drilled holes.

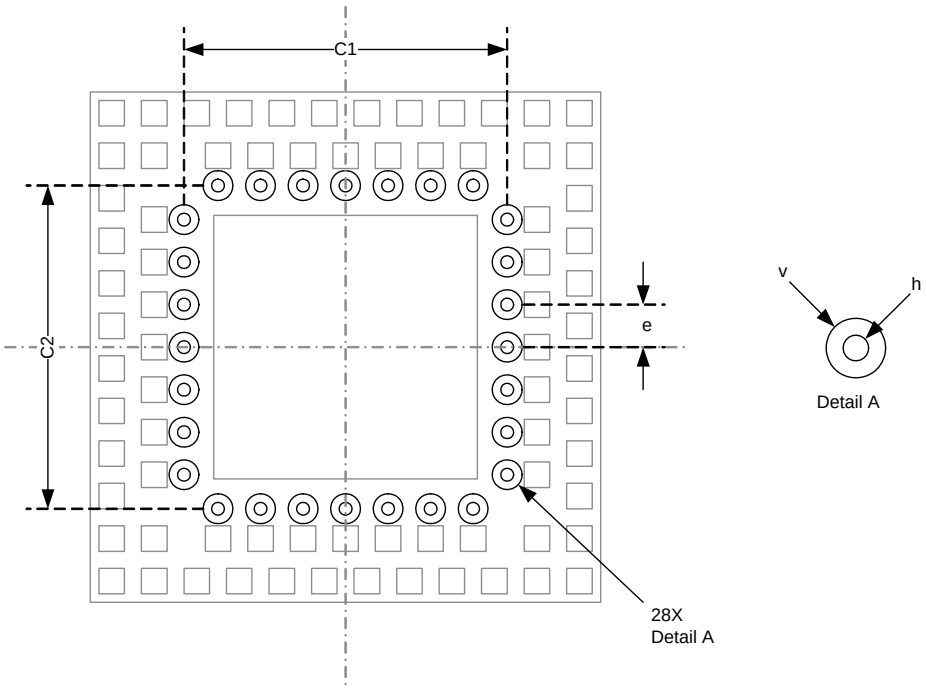


Figure 3.6. Recomended Inner Via Placement

Table 3.4. Recomended Inner Via Placement Dimensions

Dimension	Min	Nominal	Max
C1	—	3.8	—
C2	—	3.8	—
v	—	0.35	—
h	—	0.150	—

Notes:
1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Via hole should be 0.150 mm (6 mil) laser drilled.

3.2. QFN-40 Package Specifications

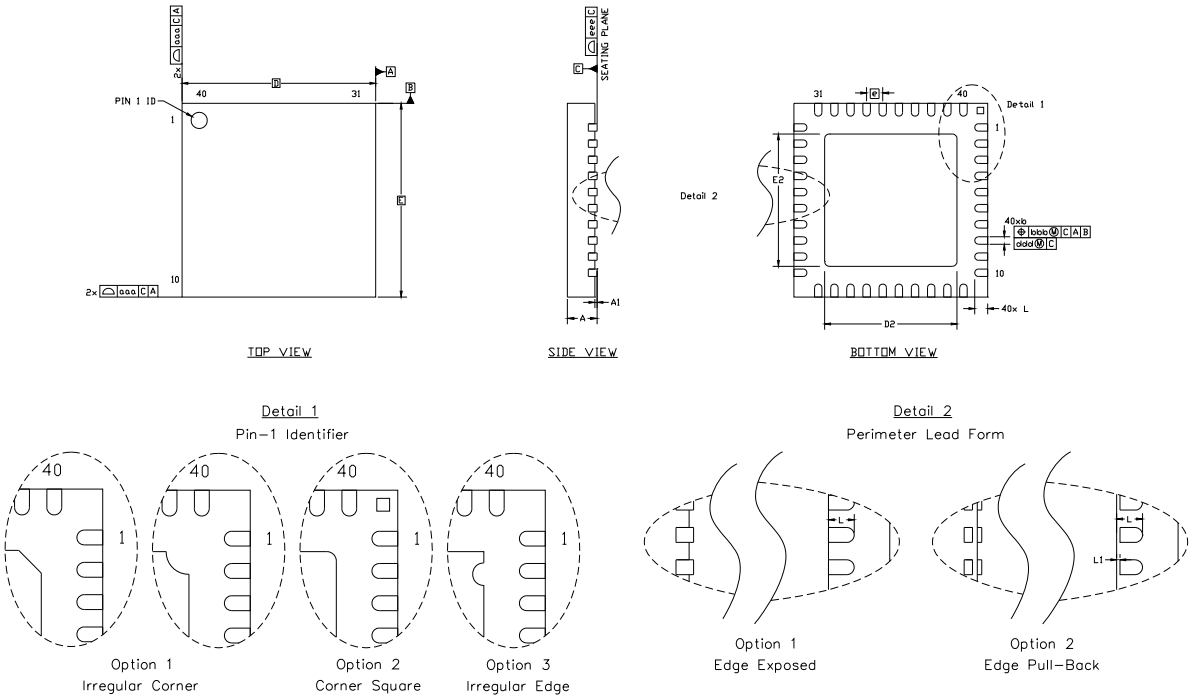


Figure 3.7. Typical QFN-40 Package Drawing

Table 3.5. QFN-40 Package Dimensions

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	0.80	0.85	0.90	E2	4.00	4.10	4.20
A1	0.00	—	0.05	L	0.35	0.40	0.45
b	0.18	0.23	0.28	L1	—	—	0.10
D	6.00 BSC			aaa	—	—	0.10
D2	4.00	4.10	4.20	bbb	—	—	0.10
e	0.50 BSC			ddd	—	—	0.05
E	6.00 BSC			eee	—	—	0.08

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation VJJD-5, except for features A, D2, and E2 which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

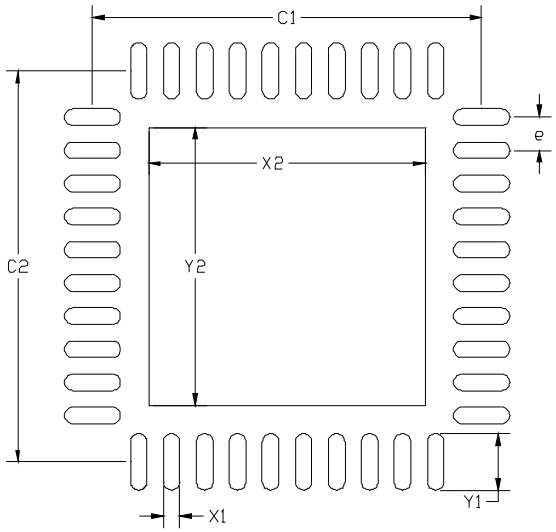


Figure 3.8. QFN-40 Landing Diagram

Table 3.6. QFN-40 Landing Diagram Dimensions

Dimension	Min	Max	Dimension	Min	Max
C1	5.80	5.90	X2	4.10	4.20
C2	5.80	5.90	Y1	0.75	0.85
e	0.50 BSC		Y2	4.10	4.20
X1	0.15	0.25			

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimension and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-SM-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

5. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μm minimum, all the way around the pad.

Stencil Design

6. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
7. The stencil thickness should be 0.125 mm (5 mils).
8. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
9. A 4x4 array of 0.80 mm square openings on a 1.05 mm pitch should be used for the center ground pad.

Card Assembly

10. A No-Clean, Type-3 solder paste is recommended.
11. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

3.3. TQFP-80 Package Specifications

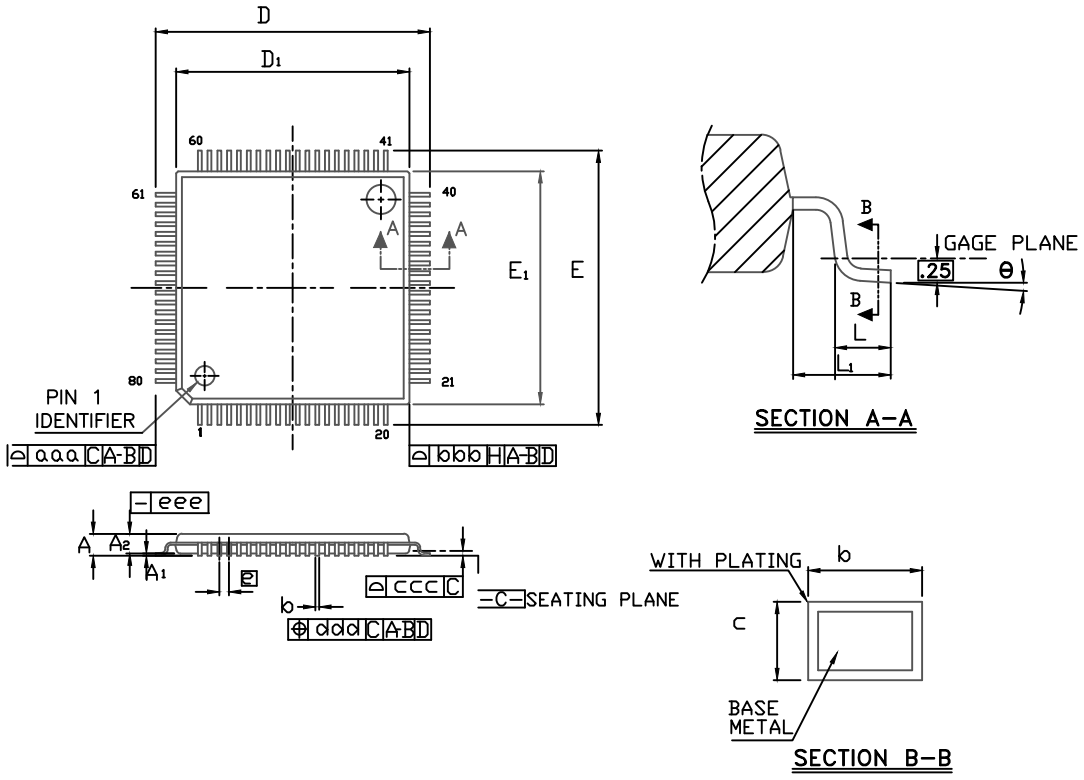


Figure 3.9. TQFP-80 Package Drawing

Table 3.7. TQFP-80 Package Dimensions

Dimension	Min	Nominal	Max
A	—	—	1.20
A1	0.05	—	0.15
A2	0.95	1.00	1.05
b	0.17	0.20	0.27
c	0.09	—	0.20
D	14.00 BSC		
D1	12.00 BSC		
e	0.50 BSC		
E	14.00 BSC		
E1	12.00 BSC		
L	0.45	0.60	0.75
L1	1.00 Ref		

Table 3.7. TQFP-80 Package Dimensions

Dimension	Min	Nominal	Max
Θ	0°	3.5°	7°
aaa	0.20		
bbb	0.20		
ccc	0.08		
ddd	0.08		
eee	0.05		

Notes:
1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This package outline conforms to JEDEC MS-026, variant ADD.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

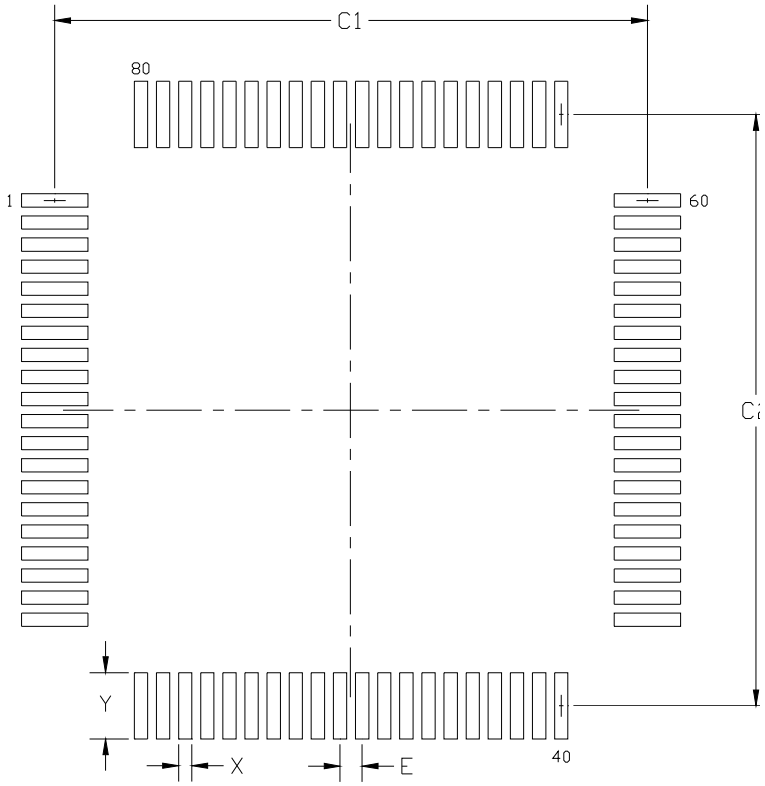


Figure 3.10. TQFP80 Landing Diagram

Table 3.8. TQFP80 Landing Diagram Dimensions

Dimension	Min	Max
C1	13.30	13.40
C2	13.30	13.40
E	0.50 BSC	
X	0.20	0.30
Y	1.40	1.50

Notes:
1. All feature sizes shown are in mm unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

3.3.1. Soldering Guidelines

3.3.1.1. Solder Mask Design

All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μm minimum, all the way around the pad.

3.3.1.2. Stencil Design

1. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
2. The stencil thickness should be 0.125 mm (5 mils).
3. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
4. A 2x2 array of 1.25 mm square openings on 1.60 mm pitch should be used for the center ground pad.

3.3.1.3. Card Assembly

1. A No-Clean, Type-3 solder paste is recommended.
2. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

4. Electrical Characteristics

Throughout the Electrical Characteristics chapter:

- "VIO" refers to the VIO or VIORF Supply Voltage.

4.1. Absolute Maximum Specifications

Table 4.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient Temperature under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on any VIO Port I/O Pin (all Port I/O pins except P1.5/6/7 and P2.0/1/2/3) or $\overline{\text{RST}}$ with respect to GND		–0.3	—	VIO + 2	V
Voltage on P1.5/6/7 or P2.0/1/2/3 with respect to GND.		–0.3	—	VIORF + 2	V
Voltage on VBAT, VBATDC, VIO, or VIORF with respect to GND		–0.3	—	4.0	V
Maximum Total Current through VBAT or GND		—	—	500	mA
Maximum Current through $\overline{\text{RST}}$ or any Port Pin		—	—	100	mA
Maximum Total Current through all Port Pins		—	—	200	mA

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

4.2. Electrical Characteristics

Table 4.2. Global Electrical Characteristics

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Supply Voltage (V_{BAT})		1.8		3.8	V
Minimum RAM Data Retention Voltage[1]	Not in sleep mode in sleep mode	— —	1.4 0.3	— 0.5	V
SYSCLK (System Clock)[2]		0	—	25	MHz
T_{SYSH} (SYSCLK High Time)		18	—	—	ns
T_{SYSL} (SYSCLK Low Time)		18	—	—	ns
Specified Operating Temperature Range		–40	—	+85	°C

Notes:

1. Based on device characterization data; Not production tested.
2. SYSCLK must be at least 32 kHz to enable debugging.

Table 4.3. Digital Supply Current at VBAT pin with DC-DC Converter Enabled

–40 to +85 °C, VBAT = 3.6V, VDC = 1.9 V, 24.5 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from flash, no external load)					
I_{BAT} [1,2,3]	V_{BAT}= 3.0 V	—	4.1	—	mA
	V_{BAT}= 3.3 V	—	4.0	—	mA
	V_{BAT}= 3.6 V	—	3.8	—	mA
Digital Supply Current—CPU Inactive (Sleep Mode, sourcing current to external device)					
I_{BAT}[1]	sourcing 9 mA to external device	—	6.5	—	mA
	sourcing 19 mA to external device	—	13	—	mA

Notes:

1. Based on device characterization data; Not production tested.
2. Digital Supply Current depends upon the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
3. Includes oscillator and regulator supply current.

Table 4.4. Digital Supply Current with DC-DC Converter Disabled

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current - Active Mode, No Clock Gating (PCLKACT=0x0F) (CPU Active, fetching instructions from flash)					
I_{BAT} [1, 2]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	4.9	5.5	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	3.9	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	175 190	— —	µA µA
	V_{BAT} = 1.8–3.8 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	85	—	µA
I_{BAT} Frequency Sensitivity[1,3,4]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	183	—	µA/MHz
Digital Supply Current - Active Mode, All Peripheral Clocks Disabled (PCLKACT=0x00) (CPU Active, fetching instructions from flash)					
I_{BAT} [1, 2]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	3.9	—	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	3.1	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	165 180	— —	µA µA
I_{BAT} Frequency Sensitivity[1, 3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	TBD	—	µA/MHz

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—Idle Mode (CPU Inactive, not Fetching Instructions from Flash)					
I_{BAT}[2]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	3.5	—	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	2.6	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	340 360	— —	µA µA
	V_{BAT} = 1.8–3.8 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	230[5]	—	µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	135	—	µA/MHz

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current— Low Power Idle Mode, All peripheral clocks enabled (PCLKEN = 0x0F) (CPU Inactive, not fetching instructions from flash)					
I_{BAT}[2, 6]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	1.5	1.9	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	1.07	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	270 280	— —	µA µA
	V_{BAT} = 1.8–3.8 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	232[5]	—	µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	47[5]	—	µA/MHz
Digital Supply Current— Low Power Idle Mode, All Peripheral Clocks Disabled (PCLKEN = 0x00) (CPU Inactive, not fetching instructions from flash)					
I_{BAT}[2, 7]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	487	—	µA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	340	—	µA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	90 94	— —	µA µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	11[5]	—	µA/MHz
Digital Supply Current—Suspend Mode					
Digital Supply Current (Suspend Mode)	V_{BAT} = 1.8 V V_{BAT} = 3.8 V	— —	77 84	— —	µA

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—Sleep Mode (LCD Enabled, RTC enabled)					
Digital Supply Current (Sleep Mode, SmaRTClock running, internal LFO, LCD Contrast Mode 1, charge pump disabled, 60 Hz refresh rate, driving 32 segment pins w/ no load)	1.8 V, T = 25 °C, static LCD	—	0.4	—	µA
	3.0 V, T = 25 °C, static LCD	—	0.6	—	
	3.6 V, T = 25 °C, static LCD	—	0.8	—	
	1.8 V, T = 25 °C, 2-Mux LCD	—	0.7	—	µA
	3.0 V, T = 25 °C, 2-Mux LCD	—	1.0	—	
	3.6 V, T = 25 °C, 2-Mux LCD	—	1.2	—	
	1.8 V, T = 25 °C, 4-Mux LCD	—	0.7	—	µA
	3.0 V, T = 25 °C, 4-Mux LCD	—	1.1	—	
	3.6 V, T = 25 °C, 4-Mux LCD	—	1.2	—	
Digital Supply Current (Sleep Mode, SmaRTClock running, 32.768 kHz Crystal, LCD Contrast Mode 1, charge pump disabled, 60 Hz refresh rate, driving 32 segment pins w/ no load)	1.8 V, T = 25 °C, static LCD	—	0.8	—	µA
	3.0 V, T = 25 °C, static LCD	—	1.1	—	
	3.6 V, T = 25 °C, static LCD	—	1.4	—	
	1.8 V, T = 25 °C, 2-Mux LCD	—	1.1	—	µA
	3.0 V, T = 25 °C, 2-Mux LCD	—	1.5	—	
	3.6 V, T = 25 °C, 2-Mux LCD	—	1.8	—	
	1.8 V, T = 25 °C, 4-Mux LCD	—	1.2	—	µA
	3.0 V, T = 25 °C, 4-Mux LCD	—	1.6	—	
	3.6 V, T = 25 °C, 4-Mux LCD	—	1.9	—	
Digital Supply Current (Sleep Mode, SmaRTClock running, internal LFO, LCD Contrast Mode 3 (2.7 V), charge pump enabled, 60 Hz refresh rate, driving 32 segment pins w/ no load)	1.8 V, T = 25 °C, static LCD	—	1.2	—	µA
	1.8 V, T = 25 °C, 2-Mux LCD	—	1.6	—	
	1.8 V, T = 25 °C, 3-Mux LCD	—	1.8	—	
	1.8 V, T = 25 °C, 4-Mux LCD	—	2.0	—	

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current (Sleep Mode, SmaRTClock running, 32.768 kHz Crystal, LCD Contrast Mode 3 (2.7 V), charge pump enabled, 60 Hz refresh rate, driving 32 segment pins w/ no load)	1.8 V, T = 25 °C, static LCD 1.8 V, T = 25 °C, 2-Mux LCD 1.8 V, T = 25 °C, 3-Mux LCD 1.8 V, T = 25 °C, 4-Mux LCD	— — — —	1.3 1.8 1.8 2.0	— — — —	µA

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—Sleep Mode (LCD disabled, RTC enabled)					
Digital Supply Current (Sleep Mode, SmaRTClock running, 32.768 kHz crystal)	1.8 V, T = 25 °C 3.0 V, T = 25 °C 3.6 V, T = 25 °C 1.8 V, T = 85 °C 3.0 V, T = 85 °C 3.6 V, T = 85 °C (includes SmaRTClock oscillator and V_{BAT} Supply Monitor)	 — — — — — —	 0.35 0.55 0.60 1.56 2.38 2.79	 — — — — — —	µA
Digital Supply Current (Sleep Mode, SmaRTClock running, internal LFO)	1.8 V, T = 25 °C 3.0 V, T = 25 °C 3.6 V, T = 25 °C 1.8 V, T = 85 °C 3.0 V, T = 85 °C 3.6 V, T = 85 °C (includes SmaRTClock oscillator and V_{BAT} Supply Monitor)	— — — — — —	0.20 0.35 0.45 1.30 2.06 2.41	— — — — — —	µA
Digital Supply Current—Sleep Mode (LCD disabled, RTC disabled)					
Digital Supply Current (Sleep Mode)	1.8 V, T = 25 °C 3.0 V, T = 25 °C 3.6 V, T = 25 °C 1.8 V, T = 85 °C 3.0 V, T = 85 °C 3.6 V, T = 85 °C (includes POR supply monitor)	— — — — — —	0.05 0.08 0.12 1.13 1.83 2.25	— — — — — —	µA
Digital Supply Current (Sleep Mode, POR Supply Monitor Disabled)	1.8 V, T = 25 °C 3.0 V, T = 25 °C 3.6 V, T = 25 °C 1.8 V, T = 85 °C 3.0 V, T = 85 °C 3.6 V, T = 85 °C	— — — — — —	0.01 0.02 0.06 TBD TBD TBD	— — — — — —	µA

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

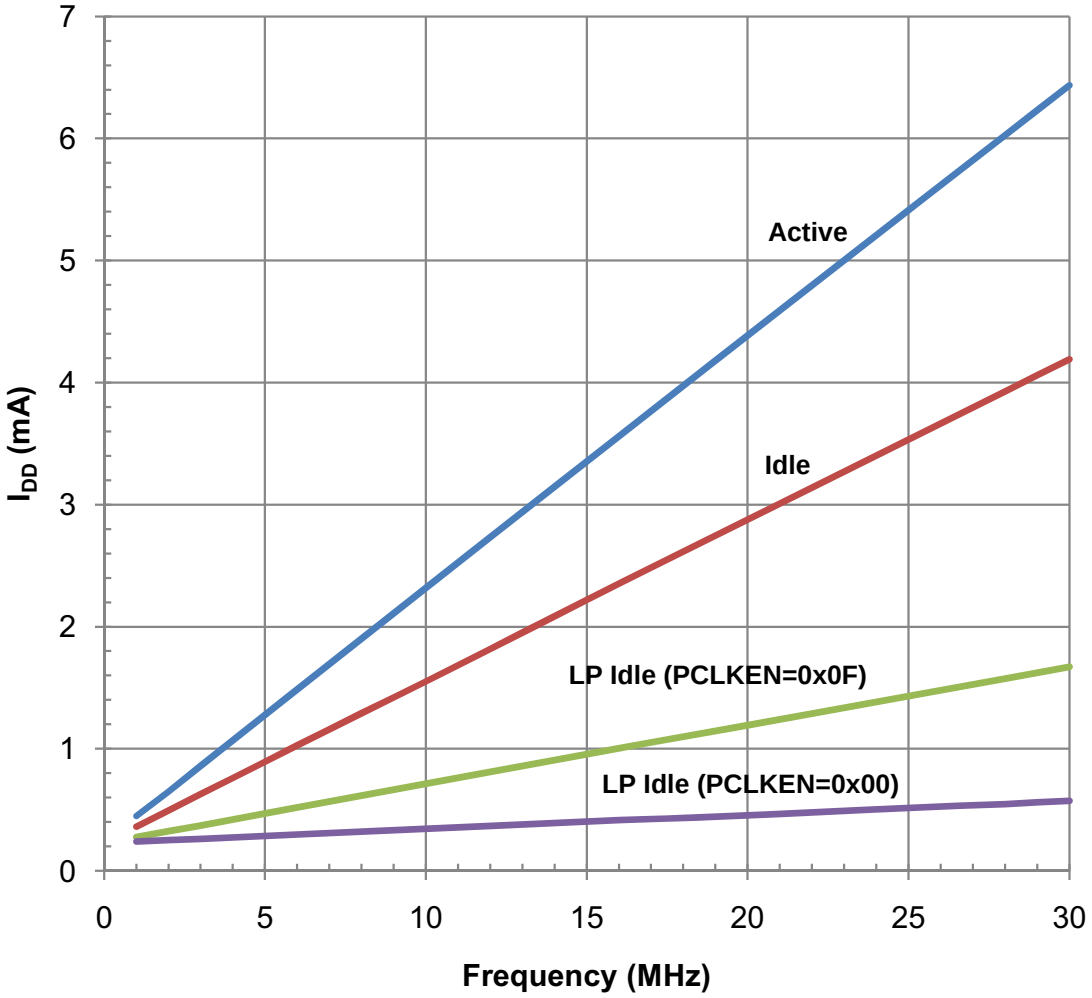
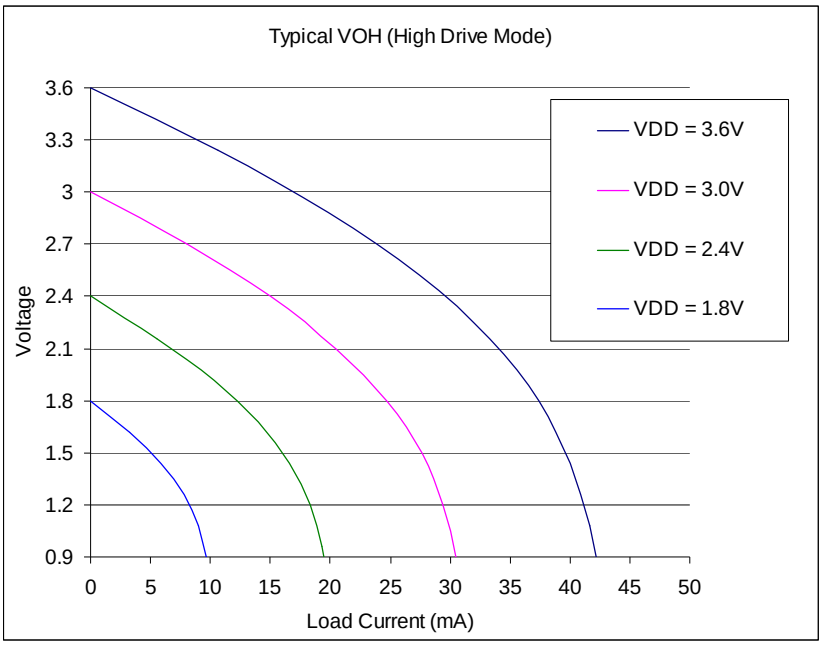


Figure 4.1. Frequency Sensitivity (External CMOS Clock, 25°C)

Table 4.5. Port I/O DC Electrical Characteristics

V_{IO} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	High Drive Strength, PnDRV.n = 1				
	IOH = –3 mA, Port I/O push-pull	V_{IO}– 0.7	—	—	
	IOH = –10 µA, Port I/O push-pull	V_{IO} – 0.1	—	—	
	IOH = –10 mA, Port I/O push-pull		See Chart		
					V
	Low Drive Strength, PnDRV.n = 0				
	IOH = –1 mA, Port I/O push-pull	V_{IO} – 0.7	—	—	
	IOH = –10 µA, Port I/O push-pull	V_{IO} – 0.1	—	—	
	IOH = –3 mA, Port I/O push-pull	—	See Chart	—	
Output Low Voltage	High Drive Strength, PnDRV.n = 1				
	I_{OL} = 8.5 mA	—	—	0.6	
	I_{OL} = 10 µA	—	—	0.1	
	I_{OL} = 25 mA	—	See Chart	—	
					V
	Low Drive Strength, PnDRV.n = 0				
	I_{OL} = 1.4 mA	—	—	0.6	
	I_{OL} = 10 µA	—	—	0.1	
	I_{OL} = 4 mA	—	See Chart	—	
Input High Voltage	V_{BAT} = 2.0 to 3.8 V	V_{IO} – 0.6	—	—	V
	V_{BAT} = 1.8 to 2.0 V	0.7 x V_{IO}	—	—	V
Input Low Voltage	V_{BAT} = 2.0 to 3.8 V	—	—	0.6	V
	V_{BAT} = 1.8 to 2.0 V	—	—	0.3 x V_{IO}	V
Input Leakage Current	Weak Pullup Off				
	Weak Pullup On, V_{IN} = 0 V, V_{BAT} = 1.8 V	—	—	±1	
		—	4	—	µA
	Weak Pullup On, Vin = 0 V, V_{BAT} = 3.8 V	—	20	35	



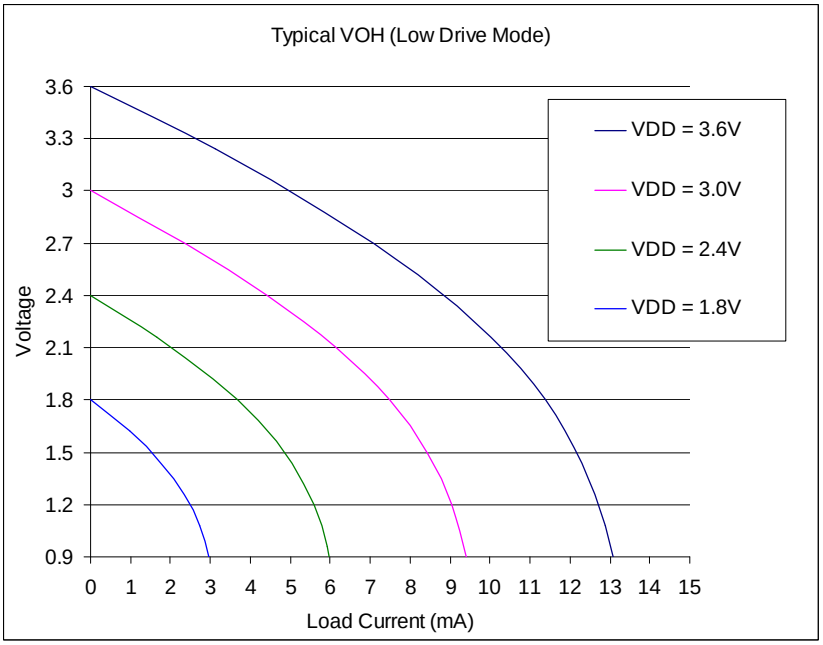


Figure 4.2. Typical VOH Curves, 1.8–3.6 V

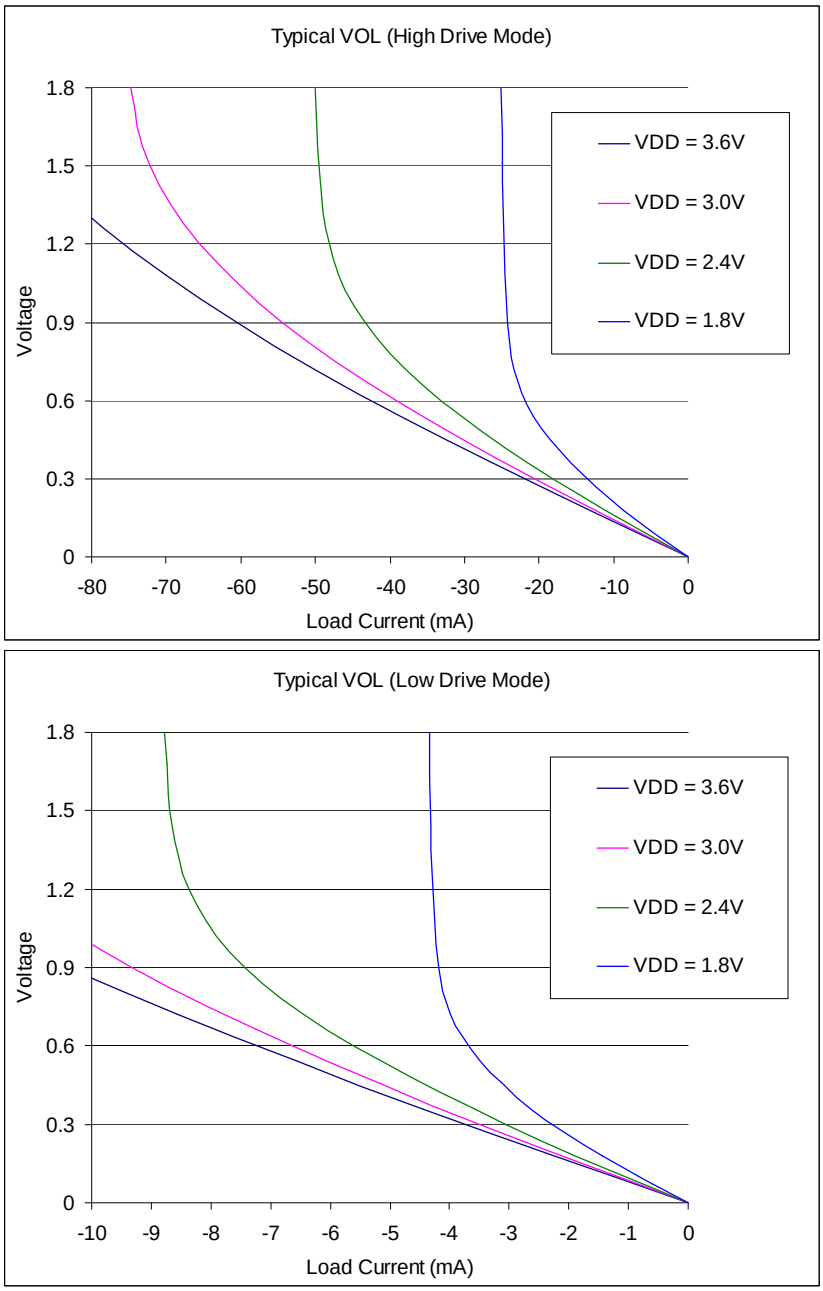


Figure 4.3. Typical VOL Curves, 1.8–3.6 V

Table 4.6. Reset Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{RST}$ Output Low Voltage	I_{OL} = 1.4 mA,	—	—	0.6	V
$\overline{RST}$ Input High Voltage	V_{BAT} = 2.0 to 3.8 V	V_{BAT} – 0.6	—	—	V
	V_{BAT} = 1.8 to 2.0 V	0.7 x V_{BAT}	—	—	V
$\overline{RST}$ Input Low Voltage	V_{BAT} = 2.0 to 3.8 V	—	—	0.6	V
	V_{BAT} = 1.8 to 2.0 V	—	—	0.3 x V_{BAT}	V
$\overline{RST}$ Input Pullup Current	$\overline{RST}$ = 0.0 V, V_{BAT} = 1.8 V	—	4	—	µA
	$\overline{RST}$ = 0.0 V, V_{BAT} = 3.8 V	—	20	35	
VBAT Monitor Threshold (V_{RST})*	Early Warning	1.8	1.85	1.9	V
	Reset Trigger (all power modes except Sleep)	1.7	1.75	1.8	
VBAT Ramp Time for Power On*	VBAT Ramp from 0–1.8 V	—	—	3	ms
POR Monitor Threshold (V_{POR})	Brownout Condition (V_{BAT} Falling)	0.45	0.7	1.0	V
	Recovery from Brownout (V_{BAT} Rising)	—	1.75	—	
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	650	1000	µs
Minimum System Clock w/ Missing Clock Detector Enabled	System clock frequency which triggers a missing clock detector timeout	—	7	10	kHz
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	—	10	—	µs
Minimum $\overline{RST}$ Low Time to Generate a System Reset		15	—	—	µs
Digital/Analog Monitor Turn-on Time		—	300	—	ns
Digital Monitor Supply Current		—	14	—	µA
Analog Monitor Supply Current		—	14	—	µA

***Note:** The VBAT monitor electical specifications apply to both the analog and digital VBAT monitors (“SFR Definition 22.1. VDM0CN: VDD Supply Monitor Control” on page 282).

Table 4.7. Power Management Electrical Specifications

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Idle Mode Wake-up Time		2	—	3	SYSCLKs
Suspend Mode Wake-up Time	CLKDIV = 0x00 Low Power or Precision Osc.	—	400	—	ns
Sleep Mode Wake-up Time		—	2	—	µs

Table 4.8. Flash Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.,

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F960/1/2/3	131072	—	—	bytes
	C8051F964/5	65536	—	—	bytes
	C8051F966/7	32768	—	—	bytes
	C8051F968/9	16384	—	—	bytes
Endurance		20 k	100k	—	Erase/Write Cycles
Erase Cycle Time		28	32	36	ms
Write Cycle Time		57	64	71	µs

Table 4.9. Internal Precision Oscillator Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	–40 to +85 °C, V_{BAT} = 1.8–3.8 V	24	24.5	25	MHz
Oscillator Supply Current (from V_{BAT})	25 °C; includes bias current of 50 µA typical	—	300*	—	µA
***Note:** Does not include clock divider or clock tree supply current.					

Table 4.10. Internal Low-Power Oscillator Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	–40 to +85 °C, V_{BAT} = 1.8–3.8 V	18	20	22	MHz
Oscillator Supply Current (from V_{BAT})	25 °C No separate bias current required	—	100*	—	µA
***Note:** Does not include clock divider or clock tree supply current.					

Table 4.11. SmaRTClock Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency (LFO)		13.1	16.4	19.7	kHz

Table 4.12. ADC0 Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, VREF = 1.65 V (REFSL[1:0] = 11), –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution	12-bit mode 10-bit mode		12 10		bits
Integral Nonlinearity	12-bit mode[1] 10-bit mode	— —	±1 ±0.5	±3 ±1	LSB
Differential Nonlinearity (Guaranteed Monotonic)	12-bit mode[1] 10-bit mode	— —	±0.8 ±0.5	±2 ±1	LSB
Offset Error	12-bit mode 10-bit mode	— —	±<1 ±<1	±3 ±3	LSB
Full Scale Error	12-bit mode[2] 10-bit mode	— —	±1 ±1	±4 ±2.5	LSB
Dynamic performance (10 kHz sine-wave single-ended input, 1 dB below Full Scale, maximum sampling rate)					
Signal-to-Noise Plus Distortion[3]	12-bit mode 10-bit mode	62 54	65 58	— —	dB
Signal-to-Distortion[3]	12-bit mode 10-bit mode	— —	76 73	— —	dB
Spurious-Free Dynamic Range[3]	12-bit mode 10-bit mode	— —	82 75	— —	dB
Conversion Rate					
SAR Conversion Clock	Normal Power Mode Low Power Mode	— —	— —	8.33 4.4	MHz
Conversion Time in SAR Clocks	10-bit Mode 8-bit Mode	13 11	— —	— —	clocks
Track/Hold Acquisition Time	Initial Acquisition Subsequent Acquisitions (dc input, burst mode)	1.5 1.1	— —	— —	us
Throughput Rate	12-bit mode 10-bit mode	— —	— —	75 300	ksps

1. INL and DNL specifications for 12-bit mode do not include the first or last four ADC codes.
2. The maximum code in 12-bit mode is 0xFFFC. The Full Scale Error is referenced from the maximum code.
3. Performance in 8-bit mode is similar to 10-bit mode.

Table 4.12. ADC0 Electrical Characteristics (Continued)

V_{BAT} = 1.8 to 3.8 V, VREF = 1.65 V (REFSL[1:0] = 11), –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ – GND)	0	—	V_{REF}	V
Absolute Pin Voltage with respect to GND	Single Ended	0	—	V_{BAT}	V
Sampling Capacitance	1x Gain 0.5x Gain	—	16 13	—	pF
Input Multiplexer Impedance		—	5	—	kΩ
Power Specifications					
Power Supply Current (V_{BAT} supplied to ADC0)	Normal Power Mode: Conversion Mode (300 ksps) Tracking Mode (0 ksps) Low Power Mode: Conversion Mode (150 ksps) Tracking Mode (0 ksps)	 — — — —	 650 740 370 400	 — — — —	µA
Power Supply Rejection	Internal High Speed VREF External VREF	— —	67 74	— —	dB

1. INL and DNL specifications for 12-bit mode do not include the first or last four ADC codes.
2. The maximum code in 12-bit mode is 0xFFFC. The Full Scale Error is referenced from the maximum code.
3. Performance in 8-bit mode is similar to 10-bit mode.

Table 4.13. Temperature Sensor Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Linearity		—	±1	—	°C
Slope		—	3.40	—	mV/°C
Slope Error*		—	40	—	µV/°C
Offset	Temp = 25 °C	—	1025	—	mV
Offset Error*	Temp = 25 °C	—	18	—	mV
Temperature Sensor Turn-On Time		—	1.7	—	µs
Supply Current		—	35	—	µA

***Note:** Represents one standard deviation from the mean.

Table 4.14. Voltage Reference Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal High-Speed Reference (REFSL[1:0] = 11)					
Output Voltage	–40 to +85 °C, V_{BAT} = 1.8–3.8 V	1.62	1.65	1.68	V
VREF Turn-on Time		—	—	1.5	µs
Supply Current	Normal Power Mode Low Power Mode	— —	260 140	— —	µA
External Reference (REFSL[1:0] = 00, REFOE = 0)					
Input Voltage Range		0	—	V_{BAT}	V
Input Current	Sample Rate = 300 ksps; VREF = 3.0 V	—	5.25	—	µA

Table 4.15. IREF0 Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C, unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Static Performance					
Resolution			6		bits
Output Compliance Range	Low Power Mode, Source	0	—	V_{BAT} – 0.4	V
	High Current Mode, Source	0	—	V_{BAT} – 0.8	
	Low Power Mode, Sink	0.3	—	V_{BAT}	
	High Current Mode, Sink	0.8	—	V_{BAT}	
Integral Nonlinearity		—	<±0.2	±1.0	LSB
Differential Nonlinearity		—	<±0.2	±1.0	LSB
Offset Error		—	<±0.1	±0.5	LSB
Full Scale Error	Low Power Mode, Source	—	—	±5	%
	High Current Mode, Source	—	—	±6	%
	Low Power Mode, Sink	—	—	±8	%
	High Current Mode, Sink	—	—	±8	%
Absolute Current Error	Low Power Mode Sourcing 20 µA	—	<±1	±3	%
Dynamic Performance					
Output Settling Time to 1/2 LSB		—	300	—	ns
Startup Time		—	1	—	µs
Power Consumption					
Net Power Supply Current (V_{BAT} supplied to IREF0 minus any output source current)	Low Power Mode, Source				
	IREF0DAT = 000001	—	10	—	µA
	IREF0DAT = 111111	—	10	—	µA
	High Current Mode, Source				
	IREF0DAT = 000001	—	10	—	µA
	IREF0DAT = 111111	—	10	—	µA
	Low Power Mode, Sink				
	IREF0DAT = 000001	—	1	—	µA
	IREF0DAT = 111111	—	11	—	µA
	High Current Mode, Sink				
	IREF0DAT = 000001	—	12	—	µA
	IREF0DAT = 111111	—	81	—	µA

Note: Refer to “6.1. PWM Enhanced Mode” on page 103 for information on how to improve IREF0 resolution.

Table 4.16. Comparator Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	120	—	ns
	CP0+ – CP0– = –100 mV	—	110	—	ns
Response Time: Mode 1, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	180	—	ns
	CP0+ – CP0– = –100 mV	—	220	—	ns
Response Time: Mode 2, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	350	—	ns
	CP0+ – CP0– = –100 mV	—	600	—	ns
Response Time: Mode 3, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	1240	—	ns
	CP0+ – CP0– = –100 mV	—	3200	—	ns
Common-Mode Rejection Ratio		—	1.5	—	mV/V
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{BAT} + 0.25	V
Input Capacitance		—	12	—	pF
Input Bias Current		—	1	—	nA
Input Offset Voltage		–10	—	+10	mV
Power Supply					
Power Supply Rejection		—	0.1	—	mV/V
Power-up Time	V_{BAT} = 3.8 V	—	0.6	—	µs
	V_{BAT} = 3.0 V	—	1.0	—	µs
	V_{BAT} = 2.4 V	—	1.8	—	µs
	V_{BAT} = 1.8 V	—	10	—	µs
Supply Current at DC	Mode 0	—	23	—	µA
	Mode 1	—	8.8	—	µA
	Mode 2	—	2.6	—	µA
	Mode 3	—	0.4	—	µA

***Note:** Vcm is the common-mode voltage on CP0+ and CP0–.

Table 4.16. Comparator Electrical Characteristics (Continued)
V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Hysteresis					
Mode 0					
Hysteresis 1	(CPnHYP/N1–0 = 00)	—	0	—	mV
Hysteresis 2	(CPnHYP/N1–0 = 01)	—	8.5	—	mV
Hysteresis 3	(CPnHYP/N1–0 = 10)	—	17	—	mV
Hysteresis 4	(CPnHYP/N1–0 = 11)	—	34	—	mV
Mode 1					
Hysteresis 1	(CPnHYP/N1–0 = 00)	—	0	—	mV
Hysteresis 2	(CPnHYP/N1–0 = 01)	—	6.5	—	mV
Hysteresis 3	(CPnHYP/N1–0 = 10)	—	13	—	mV
Hysteresis 4	(CPnHYP/N1–0 = 11)	—	26	—	mV
Mode 2					
Hysteresis 1	(CPnHYP/N1–0 = 00)	—	0	1	mV
Hysteresis 2	(CPnHYP/N1–0 = 01)	2	5	10	mV
Hysteresis 3	(CPnHYP/N1–0 = 10)	5	10	20	mV
Hysteresis 4	(CPnHYP/N1–0 = 11)	12	20	30	mV
Mode 3					
Hysteresis 1	(CPnHYP/N1–0 = 00)	—	0	—	mV
Hysteresis 2	(CPnHYP/N1–0 = 01)	—	4.5	—	mV
Hysteresis 3	(CPnHYP/N1–0 = 10)	—	9	—	mV
Hysteresis 4	(CPnHYP/N1–0 = 11)	—	17	—	mV
***Note:** Vcm is the common-mode voltage on CP0+ and CP0–.					

Table 4.17. VREG0 Electrical Characteristics
V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range		1.8	—	3.8	V
Bias Current	Normal, idle, suspend, or stop mode	—	20	—	µA

Table 4.18. LCD0 Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Charge Pump Output Voltage Error		—	±30	—	mV
LCD Clock Frequency		16	—	33	kHz

Table 4.19. PC0 Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Supply Current (25 °C, 2 ms sample rate)	1.8 V	—	145	—	nA
	2.2 V	—	175	—	
	3.0 V	—	235	—	
	3.8 V	—	285	—	

Table 4.20. DC0 (Buck Converter) Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Condition	Min	Typ	Max	Units
Input Voltage Range		1.8	—	3.8	V
Input Supply to Output Voltage Differential (for regulation)		0.45	—	—	V
Output Voltage Range	Programmable from 1.8 to 3.5 V	1.8	1.9	3.5	V
Output Power	V_{DC} = 1.8 to 3.0 V. $V_{BAT} \geq V_{DC}$ + 0.5.	—	—	250	mW
Inductor Value[1]		0.47	0.56	0.68	µH
Inductor Current Rating	For load currents less than 50 mA For load currents greater than 50 mA	450 550	— —	— —	mA
Output Capacitor Value[2]		1	2.2	10	µF
Input Capacitor[2]		—	4.7	—	µF
Output Load Current (based on output power specification)	Target output = 1.8 to 3.0 V[3] Target output = 3.1 V[3] Target output = 3.3 V[3] Target output = 3.5 V[4]	— — — — —	— — — — —	85[3] 70[3] 50[3] 10[4]	mA
Load Regulation	Output = 1.9 V; Load current up to 85 mA; Supply range = 2.4–3.8 V	—	0.03	—	mv/mA
Maximum DC Load Current During Startup		—	—	5	mA
Switching Clock Frequency		1.9	2.9	3.8	MHz

Notes:

1. Recommended: Inductor similar to NLV32T-R56J-PF (0.56 µH)
2. Recommended: X7R or X5R ceramic capacitors with low ESR. Example: Murata GRM21BR71C225K with ESR < 10 mΩ (@ frequency > 1 MHz)
3. $V_{BAT} \geq V_{DC}$ + 0.5. Auto-Bypass enabled (DC0MD.2 = 1).
4. V_{BAT} = 3.8 V. Auto-Bypass disabled (DC0MD.2 = 0).

5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode

The ADC0 on C8051F96x devices is a 300 ksps, 10-bit or 75 ksps, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold and programmable window detector. ADC0 also has an autonomous low power Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. It also has a 16-bit accumulator that can automatically oversample and average the ADC results. See Section 5.4 for more details on using the ADC in 12-bit mode.

The ADC is fully configurable under software control via Special Function Registers. The ADC0 operates in Single-ended mode and may be configured to measure various different signals using the analog multiplexer described in “5.7. ADC0 Analog Multiplexer” on page 95. The voltage reference for the ADC is selected as described in “5.9. Voltage and Ground Reference Options” on page 100.

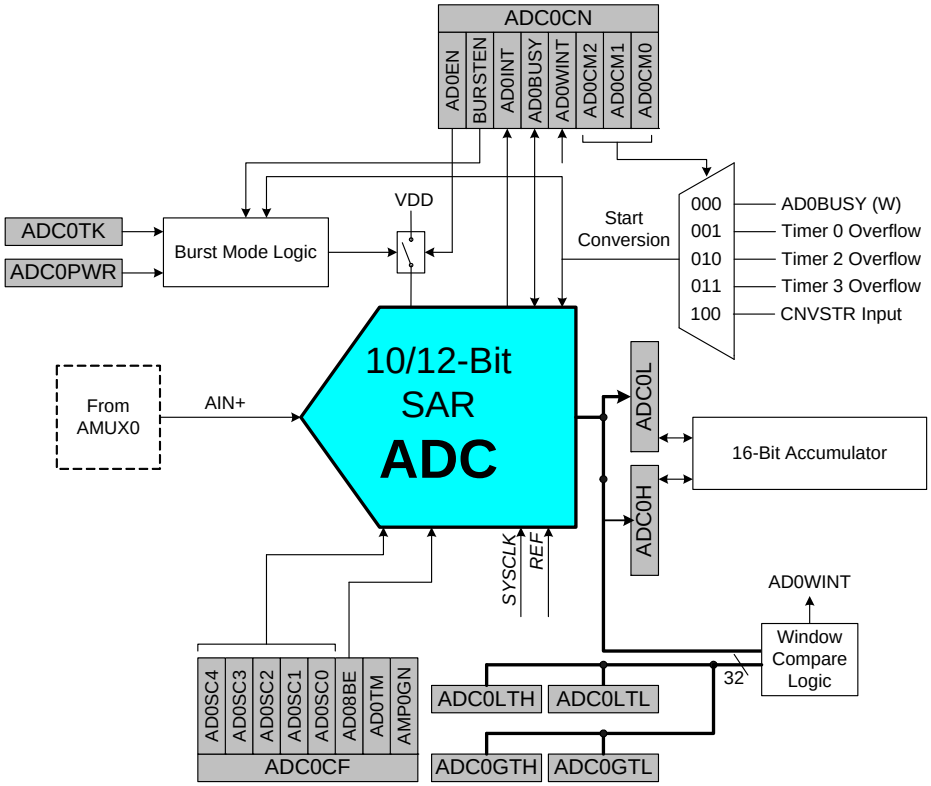


Figure 5.1. ADC0 Functional Block Diagram

5.1. Output Code Formatting

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0SJST[2:0]. When the repeat count is set to 1, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0SJST = 000)	Left-Justified ADC0H:ADC0L (AD0SJST = 100)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When the repeat count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, 16, 32, or 64 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the AD0RPT bits in the ADC0AC register. When a repeat count higher than 1, the ADC output must be right-justified (AD0SJST = 0xx); unused bits in the ADC0H and ADC0L registers are set to 0. The example below shows the right-justified result for various input voltages and repeat counts. Notice that accumulating 2^n samples is equivalent to left-shifting by n bit positions when all samples returned from the ADC have the same value.

Input Voltage	Repeat Count = 4	Repeat Count = 16	Repeat Count = 64
V_{REF} x 1023/1024	0x0FFC	0x3FF0	0xFFC0
V_{REF} x 512/1024	0x0800	0x2000	0x8000
V_{REF} x 511/1024	0x07FC	0x1FF0	0x7FC0
0	0x0000	0x0000	0x0000

The AD0SJST bits can be used to format the contents of the 16-bit accumulator. The accumulated result can be shifted right by 1, 2, or 3 bit positions. Based on the principles of oversampling and averaging, the effective ADC resolution increases by 1 bit each time the oversampling rate is increased by a factor of 4. The example below shows how to increase the effective ADC resolution by 1, 2, and 3 bits to obtain an effective ADC resolution of 11-bit, 12-bit, or 13-bit respectively without CPU intervention.

Input Voltage	Repeat Count = 4 Shift Right = 1 11-Bit Result	Repeat Count = 16 Shift Right = 2 12-Bit Result	Repeat Count = 64 Shift Right = 3 13-Bit Result
V_{REF} x 1023/1024	0x07F7	0x0FFC	0x1FF8
V_{REF} x 512/1024	0x0400	0x0800	0x1000
V_{REF} x 511/1024	0x03FE	0x04FC	0x0FF8
0	0x0000	0x0000	0x0000

5.2. Modes of Operation

ADC0 has a maximum conversion speed of 300 ksps in 10-bit mode. The ADC0 conversion clock (SAR-CLK) is a divided version of the system clock when burst mode is disabled (BURSTEN = 0), or a divided version of the low power oscillator when burst mode is enabled (BURSEN = 1). The clock divide value is determined by the AD0SC bits in the ADC0CF register.

5.2.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a 1 to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 3 overflow
5. A rising edge on the CNVSTR input signal (pin P0.6)

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. When Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See “32. Timers” on page 444 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as Port pin P0.6. When the CNVSTR input is used as the ADC0 conversion source, Port pin P0.6 should be skipped by the Digital Crossbar. To configure the Crossbar to skip P0.6, set to 1 Bit 6 in register P0SKIP. See “27. Port Input/Output” on page 351 for details on Port I/O configuration.

5.2.2. Tracking Modes

Each ADC0 conversion must be preceded by a minimum tracking time in order for the converted result to be accurate. The minimum tracking time is given in Table 4.12. The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state when Burst Mode is disabled, the ADC0 input is continuously tracked, except when a conversion is in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 5.2). Tracking can also be disabled (shutdown) when the device is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX settings are frequently changed, due to the settling time requirements described in “5.2.4. Settling Time Requirements” on page 83.

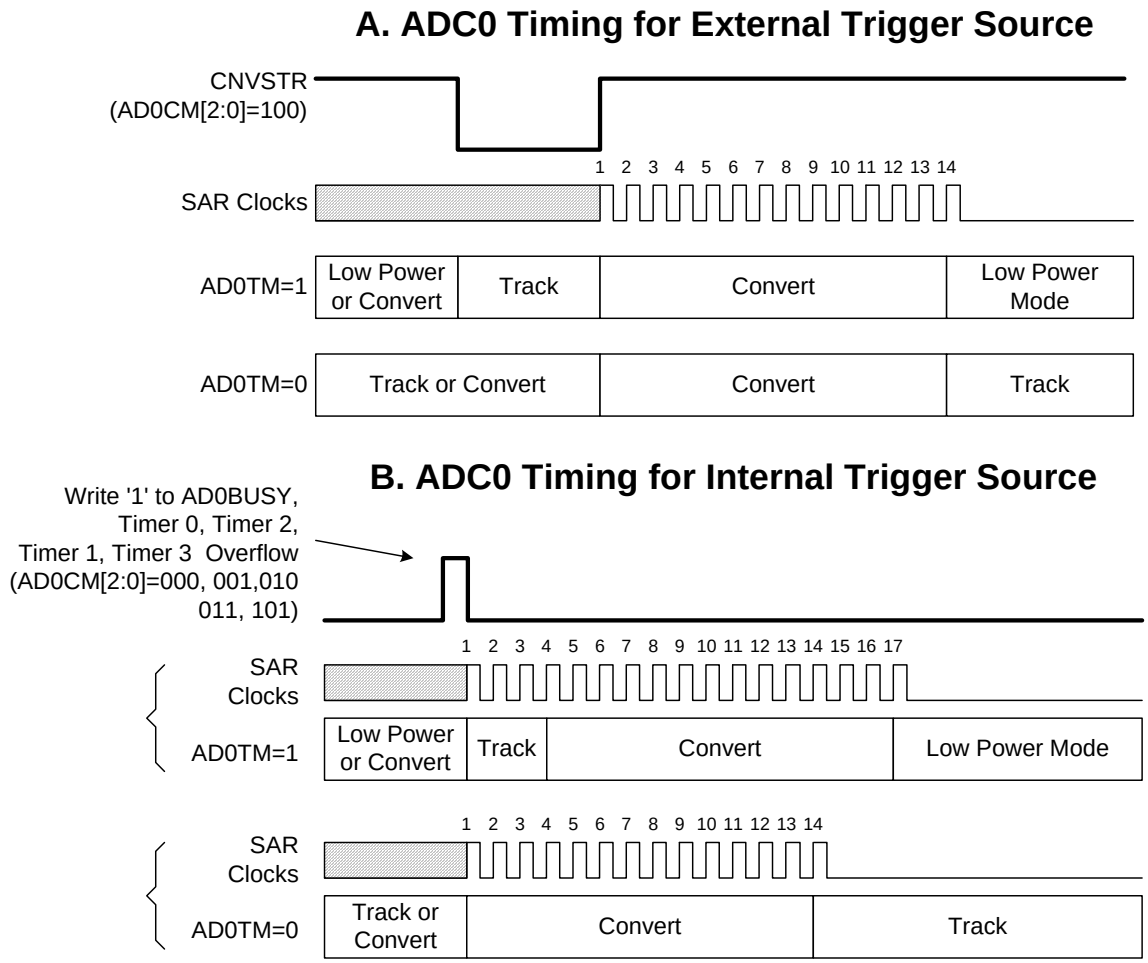


Figure 5.2. 10-Bit ADC Track and Conversion Example Timing (BURSTEN = 0)

5.2.3. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a low power state, accumulates 1, 4, 8, 16, 32, or 64 using an internal Burst Mode clock (approximately 20 MHz), then re-enters a low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a low power state within a single system clock cycle, even if the system clock is slow (e.g. 32.768 kHz), or suspended.

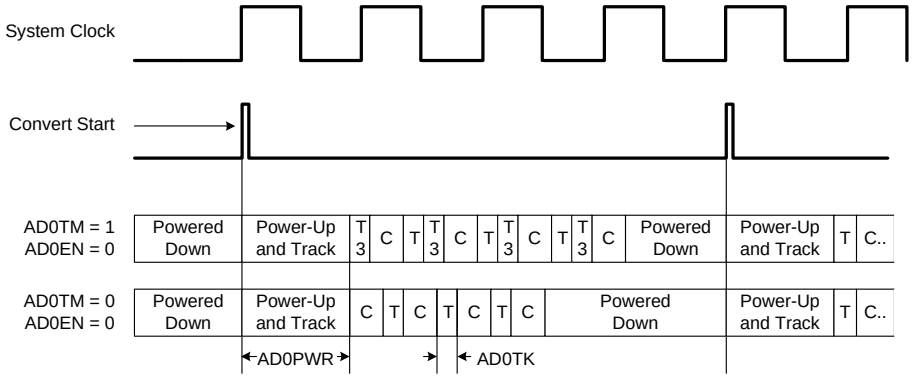
Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e. the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 5.3 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

In Burst Mode, tracking is determined by the settings in AD0PWR and AD0TK. The default settings for these registers will work in most applications without modification; however, settling time requirements may need adjustment in some applications. Refer to “5.2.4. Settling Time Requirements” on page 83 for more details.

Notes:

- Setting AD0TM to 1 will insert an additional 3 SAR clocks of tracking before each conversion, regardless of the settings of AD0PWR and AD0TK.
- When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.



T = Tracking set by AD0TK
T3 = Tracking set by AD0TM (3 SAR clocks)
C = Converting

Figure 5.3. Burst Mode Tracking Example with Repeat Count Set to 4

5.2.4. Settling Time Requirements

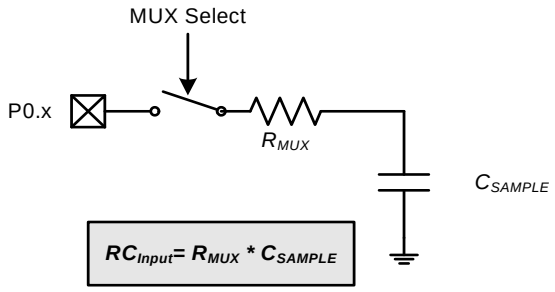
A minimum amount of tracking time is required before each conversion can be performed, to allow the sampling capacitor voltage to settle. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For many applications, these three SAR clocks will meet the minimum tracking time requirements, and higher values for the external source impedance will increase the required tracking time.

Figure 5.4 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation . When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 4.12 for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (10).



Note: The value of CSAMPLE depends on the PGA Gain. See Table 4.12 for details.

Figure 5.4. ADC0 Equivalent Input Circuits

5.2.5. Gain Setting

The ADC has gain settings of 1x and 0.5x. In 1x mode, the full scale reading of the ADC is determined directly by V_{REF}. In 0.5x mode, the full-scale reading of the ADC occurs when the input voltage is V_{REF} x 2. The 0.5x gain setting can be useful to obtain a higher input Voltage range when using a small V_{REF} voltage, or to measure input voltages that are between V_{REF} and V_{DD}. Gain settings for the ADC are controlled by the AMP0GN bit in register ADC0CF.

5.3. 8-Bit Mode

Setting the ADC08BE bit in register ADC0CF to 1 will put the ADC in 8-bit mode.In 8-bit mode, only the 8 MSBs of data are converted, allowing the conversion to be completed in two fewer SAR clock cycles than a 10-bit conversion. This can result in an overall lower power consumption since the system can spend more time in a low power mode. The two LSBs of a conversion are always 00 in this mode, and the ADC0L register will always read back 0x00.

5.4. 12-Bit Mode

C8051F96x devices have an enhanced SAR converter that provides 12-bit resolution while retaining the 10- and 8-bit operating modes of the other devices in the family. When configured for 12-bit conversions, the ADC performs four 10-bit conversions using four different reference voltages and combines the results into a single 12-bit value. Unlike simple averaging techniques, this method provides true 12-bit resolution of ac or dc input signals without depending on noise to provide dithering. The converter also employs a hardware Dynamic Element Matching algorithm that reconfigures the largest elements of the internal DAC for each of the four 10-bit conversions to cancel the any matching errors, enabling the converter to achieve 12-bit linearity performance to go along with its 12-bit resolution. For best performance, the Low Power Oscillator should be selected as the system clock source while taking 12-bit ADC measurements.

The 12-bit mode is enabled by setting the AD012BE bit (ADC0AC.7) to logic 1 and configuring Burst Mode for four conversions as described in Section 5.2.3. The conversion can be initiated using any of the methods described in Section 5.2.1, and the 12-bit result will appear in the ADC0H and ADC0L registers. Since the 12-bit result is formed from a combination of four 10-bit results, the maximum output value is 4 x (1023) = 4092, rather than the max value of (2^12 – 1) = 4095 that is produced by a traditional 12-bit converter. To further increase resolution, the burst mode repeat value may be configured to any multiple of four conversions. For example, if a repeat value of 16 is selected, the ADC0 output will be a 14-bit number (sum of four 12-bit numbers) with 13 effective bits of resolution.

5.5. Low Power Mode

The SAR converter provides a low power mode that allows a significant reduction in operating current when operating at low SAR clock frequencies. Low power mode is enabled by setting the AD0LPM bit (ADC0PWR.7) to 1. In general, low power mode is recommended when operating with SAR conversion clock frequency at 4 MHz or less. See the Electrical Characteristics chapter for details on power consumption and the maximum clock frequencies allowed in each mode. Setting the Low Power Mode bit reduces the bias currents in both the SAR converter and in the High-Speed Voltage Reference.

Table 5.1. Representative Conversion Times and Energy Consumption for the SAR ADC with 1.65 V High-Speed VREF

	Normal Power Mode			Low Power Mode		
	8 bit	**10 bit**	**12 bit**	**8 bit**	**10 bit**	**12 bit**
Highest nominal SAR clock frequency	8.17 MHz (24.5/3)	8.17 MHz (24.5/3)	6.67 MHz (20.0/3)	4.08 MHz (24.5/6)	4.08 MHz (24.5/6)	4.00 MHz (20.0/5)
Total number of conversion clocks required	11	13	52 (13 x 4)	11	13	52 (13*4)
Total tracking time (min)	1.5 µs	1.5 µs	4.8 µs (1.5+3 x 1.1)	1.5 µs	1.5 µs	4.8 µs (1.5+3 x 1.1)
Total time for one conversion	2.85 µs	3.09 µs	12.6 µs	4.19 µs	4.68 µs	17.8 µs
ADC Throughput	351 ksps	323 ksps	79 ksps	238 ksps	214 ksps	56 ksps
Energy per conversion	8.2 nJ	8.9 nJ	36.5 nJ	6.5 nJ	7.3 nJ	27.7 nJ

Note: This table assumes that the 24.5 MHz precision oscillator is used for 8- and 10-bit modes, and the 20 MHz low power oscillator is used for 12-bit mode. The values in the table assume that the oscillators run at their nominal frequencies. The maximum SAR clock values given in Table 4.12 allow for maximum oscillation frequencies of 25.0 MHz and 22 MHz for the precision and low-power oscillators, respectively, when using the given SAR clock divider values. Energy calculations are for the ADC subsystem only and do not include CPU current.

SFR Definition 5.1. ADC0CN: ADC0 Control

Bit	7	6	5	4	3	2	1	0
Name	AD0EN	BURSTEN	AD0INT	AD0BUSY	AD0WINT	ADC0CM[2:0]		
Type	R/W	R/W	R/W	W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xE8; bit-addressable;

Bit	Name	Function
7	AD0EN	**ADC0 Enable.** 0: ADC0 Disabled (low-power shutdown). 1: ADC0 Enabled (active and ready for data conversions).
6	BURSTEN	**ADC0 Burst Mode Enable.** 0: ADC0 Burst Mode Disabled. 1: ADC0 Burst Mode Enabled.
5	AD0INT	**ADC0 Conversion Complete Interrupt Flag.** Set by hardware upon completion of a data conversion (BURSTEN=0), or a burst of conversions (BURSTEN=1). Can trigger an interrupt. Must be cleared by software.
4	AD0BUSY	**ADC0 Busy.** Writing 1 to this bit initiates an ADC conversion when ADC0CM[2:0] = 000.
3	AD0WINT	**ADC0 Window Compare Interrupt Flag.** Set by hardware when the contents of ADC0H:ADC0L fall within the window specified by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL. Can trigger an interrupt. Must be cleared by software.
2:0	ADC0CM[2:0]	**ADC0 Start of Conversion Mode Select.** Specifies the ADC0 start of conversion source. 000: ADC0 conversion initiated on write of 1 to AD0BUSY. 001: ADC0 conversion initiated on overflow of Timer 0. 010: ADC0 conversion initiated on overflow of Timer 2. 011: ADC0 conversion initiated on overflow of Timer 3. 1xx: ADC0 conversion initiated on rising edge of CNVSTR.

SFR Definition 5.2. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD08BE	AD0TM	AMP0GN
Type	R/W					R/W	R/W	R/W
Reset	1	1	1	1	1	0	0	0

SFR Page = 0x0; SFR Address = 0xBC

Bit	Name	Function
7:3	AD0SC[4:0]	**ADC0 SAR Conversion Clock Divider.** SAR Conversion clock is derived from FCLK by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC[4:0]. SAR Conversion clock requirements are given in Table 4.12. BURSTEN = 0: FCLK is the current system clock. BURSTEN = 1: FCLK is the 20 MHz low power oscillator, independent of the system clock. $AD0SC = \frac{FCLK}{CLK_{SAR}} - 1$ * *Round the result up. or $CLK_{SAR} = \frac{FCLK}{AD0SC + 1}$
2	AD08BE	**ADC0 8-Bit Mode Enable.** 0: ADC0 operates in 10-bit mode (normal operation). 1: ADC0 operates in 8-bit mode.
1	AD0TM	**ADC0 Track Mode.** Selects between Normal or Delayed Tracking Modes. 0: Normal Track Mode: When ADC0 is enabled, conversion begins immediately following the start-of-conversion signal. 1: Delayed Track Mode: When ADC0 is enabled, conversion begins 3 SAR clock cycles following the start-of-conversion signal. The ADC is allowed to track during this time.
0	AMP0GN	**ADC0 Gain Control.** 0: The on-chip PGA gain is 0.5. 1: The on-chip PGA gain is 1.

SFR Definition 5.3. ADC0AC: ADC0 Accumulator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD012BE	AD0AE	AD0SJST[2:0]			AD0RPT[2:0]		
Type	R/W	W	R/W			R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xBA

Bit	Name	Function
7	AD012BE	**ADC0 12-Bit Mode Enable.** Enables 12-bit Mode. 0: 12-bit Mode Disabled. 1: 12-bit Mode Enabled.
6	AD0AE	**ADC0 Accumulate Enable.** Enables multiple conversions to be accumulated when burst mode is disabled. 0: ADC0H:ADC0L contain the result of the latest conversion when Burst Mode is disabled. 1: ADC0H:ADC0L contain the accumulated conversion results when Burst Mode is disabled. Software must write 0x0000 to ADC0H:ADC0L to clear the accumulated result. This bit is write-only. Always reads 0b.
5:3	AD0SJST[2:0]	**ADC0 Accumulator Shift and Justify.** Specifies the format of data read from ADC0H:ADC0L. 000: Right justified. No shifting applied. 001: Right justified. Shifted right by 1 bit. 010: Right justified. Shifted right by 2 bits. 011: Right justified. Shifted right by 3 bits. 100: Left justified. No shifting applied. All remaining bit combinations are reserved.
2:0	AD0RPT[2:0]	**ADC0 Repeat Count.** Selects the number of conversions to perform and accumulate in Burst Mode. This bit field must be set to 000 if Burst Mode is disabled. 000: Perform and Accumulate 1 conversion. 001: Perform and Accumulate 4 conversions. 010: Perform and Accumulate 8 conversions. 011: Perform and Accumulate 16 conversions. 100: Perform and Accumulate 32 conversions. 101: Perform and Accumulate 64 conversions. All remaining bit combinations are reserved.

SFR Definition 5.4. ADC0PWR: ADC0 Burst Mode Power-Up Time

Bit	7	6	5	4	3	2	1	0
Name	AD0LPM				AD0PWR[3:0]			
Type	R/W	R	R	R	R/W			
Reset	0	0	0	0	1	1	1	1

SFR Page = 0xF; SFR Address = 0xBA

Bit	Name	Function
7	AD0LPM	**ADC0 Low Power Mode Enable.** Enables Low Power Mode Operation. 0: Low Power Mode disabled. 1: Low Power Mode enabled.
6:4	Unused	Read = 0000b; Write = Don't Care.
3:0	AD0PWR[3:0]	**ADC0 Burst Mode Power-Up Time.** Sets the time delay required for ADC0 to power up from a low power state. For BURSTEN = 0: ADC0 power state controlled by AD0EN. For BURSTEN = 1 and AD0EN = 1: ADC0 remains enabled and does not enter a low power state after all conversions are complete. Conversions can begin immediately following the start-of-conversion signal. For BURSTEN = 1 and AD0EN = 0: ADC0 enters a low power state after all conversions are complete. Conversions can begin a programmed delay after the start-of-conversion signal. The ADC0 Burst Mode Power-Up time is programmed according to the following equation: $$AD0PWR = \frac{Tstartup}{400ns} - 1$$ or $$Tstartup = (AD0PWR + 1)400ns$$ **Note:** Setting AD0PWR to 0x04 provides a typical tracking time of 2 us for the first sample taken after the start of conversion.

SFR Definition 5.5. ADC0TK: ADC0 Burst Mode Track Time

Bit	7	6	5	4	3	2	1	0
Name			AD0TK[5:0]					
Type	R	R	R/W					
Reset	0	0	0	1	1	1	1	0

SFR Page = 0xF; SFR Address = 0xBB

Bit	Name	Function
7	Reserved	Read = 0b; Write = Must Write 0b.
6	Unused	Read = 0b; Write = Don't Care.
5:0	AD0TK[5:0]	**ADC0 Burst Mode Track Time.** Sets the time delay between consecutive conversions performed in Burst Mode. The ADC0 Burst Mode Track time is programmed according to the following equation: $$AD0TK = 63 - \left(\frac{Ttrack}{50ns} - 1\right)$$ or $$Ttrack = (64 - AD0TK)50ns$$

Notes:
1. If AD0TM is set to 1, an additional 3 SAR clock cycles of Track time will be inserted prior to starting the conversion.
2. The Burst Mode Track delay is not inserted prior to the first conversion. The required tracking time for the first conversion should be met by the Burst Mode Power-Up Time.

SFR Definition 5.6. ADC0H: ADC0 Data Word High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0[15:8]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xBE

Bit	Name	Description	Read	Write
7:0	ADC0[15:8]	**ADC0 Data Word High Byte.**	Most Significant Byte of the 16-bit ADC0 Accumulator formatted according to the settings in AD0SJST[2:0].	Set the most significant byte of the 16-bit ADC0 Accumulator to the value written.
Note: If Accumulator shifting is enabled, the most significant bits of the value read will be zeros. This register should not be written when the SYNC bit is set to 1.				

SFR Definition 5.7. ADC0L: ADC0 Data Word Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xBD;

Bit	Name	Description	Read	Write
7:0	ADC0[7:0]	**ADC0 Data Word Low Byte.**	Least Significant Byte of the 16-bit ADC0 Accumulator formatted according to the settings in AD0SJST[2:0].	Set the least significant byte of the 16-bit ADC0 Accumulator to the value written.
Note: If Accumulator shifting is enabled, the most significant bits of the value read will be the least significant bits of the accumulator high byte. This register should not be written when the SYNC bit is set to 1.				

5.6. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

SFR Definition 5.8. ADC0GTH: ADC0 Greater-Than High Byte

Bit	7	6	5	4	3	2	1	0
Name	AD0GT[15:8]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xC4

Bit	Name	Function
7:0	AD0GT[15:8]	**ADC0 Greater-Than High Byte.** Most Significant Byte of the 16-bit Greater-Than window compare register.

SFR Definition 5.9. ADC0GTL: ADC0 Greater-Than Low Byte

Bit	7	6	5	4	3	2	1	0
Name	AD0GT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xC3

Bit	Name	Function
7:0	AD0GT[7:0]	**ADC0 Greater-Than Low Byte.** Least Significant Byte of the 16-bit Greater-Than window compare register.
Note: In 8-bit mode, this register should be set to 0x00.		

SFR Definition 5.10. ADC0LTH: ADC0 Less-Than High Byte

Bit	7	6	5	4	3	2	1	0
Name	AD0LT[15:8]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC6

Bit	Name	Function
7:0	AD0LT[15:8]	**ADC0 Less-Than High Byte.** Most Significant Byte of the 16-bit Less-Than window compare register.

SFR Definition 5.11. ADC0LTL: ADC0 Less-Than Low Byte

Bit	7	6	5	4	3	2	1	0
Name	AD0LT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC5

Bit	Name	Function
7:0	AD0LT[7:0]	**ADC0 Less-Than Low Byte.** Least Significant Byte of the 16-bit Less-Than window compare register.
Note: In 8-bit mode, this register should be set to 0x00.		

5.6.1. Window Detector In Single-Ended Mode

Figure 5.5 shows two example window comparisons for right-justified data, with ADC0LTH:ADC0LTL = 0x0080 (128d) and ADC0GTH:ADC0GTL = 0x0040 (64d). The input voltage can range from 0 to VREF x (1023/1024) with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0040 < ADC0H:ADC0L < 0x0080). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0040 or ADC0H:ADC0L > 0x0080). Figure 5.6 shows an example using left-justified data with the same comparison values.

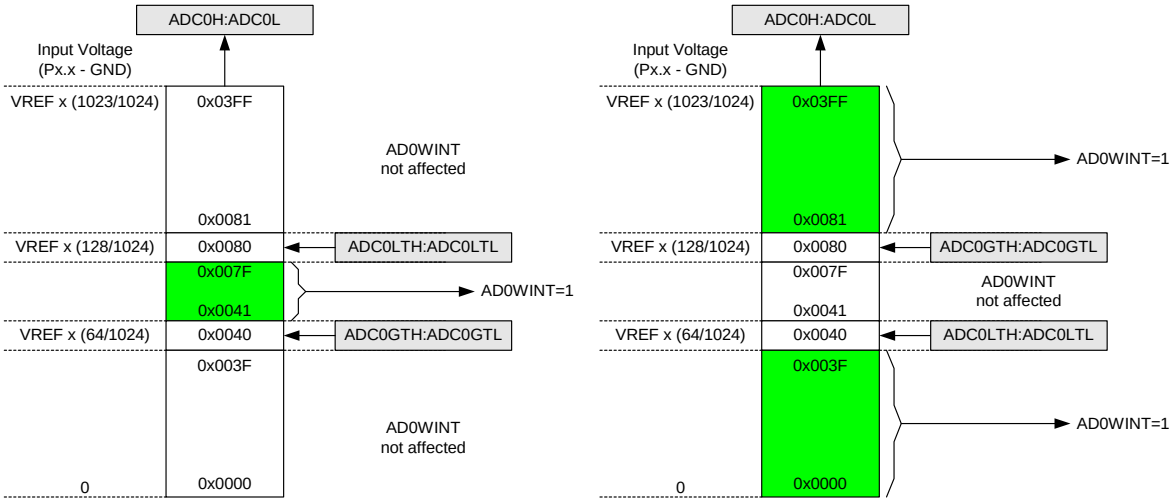


Figure 5.5. ADC Window Compare Example: Right-Justified Single-Ended Data

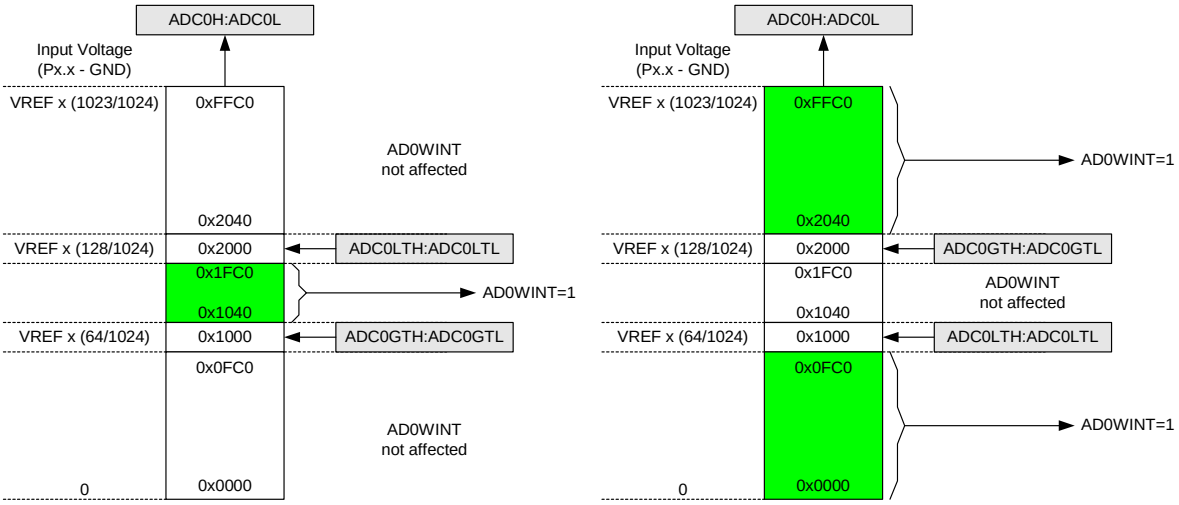


Figure 5.6. ADC Window Compare Example: Left-Justified Single-Ended Data

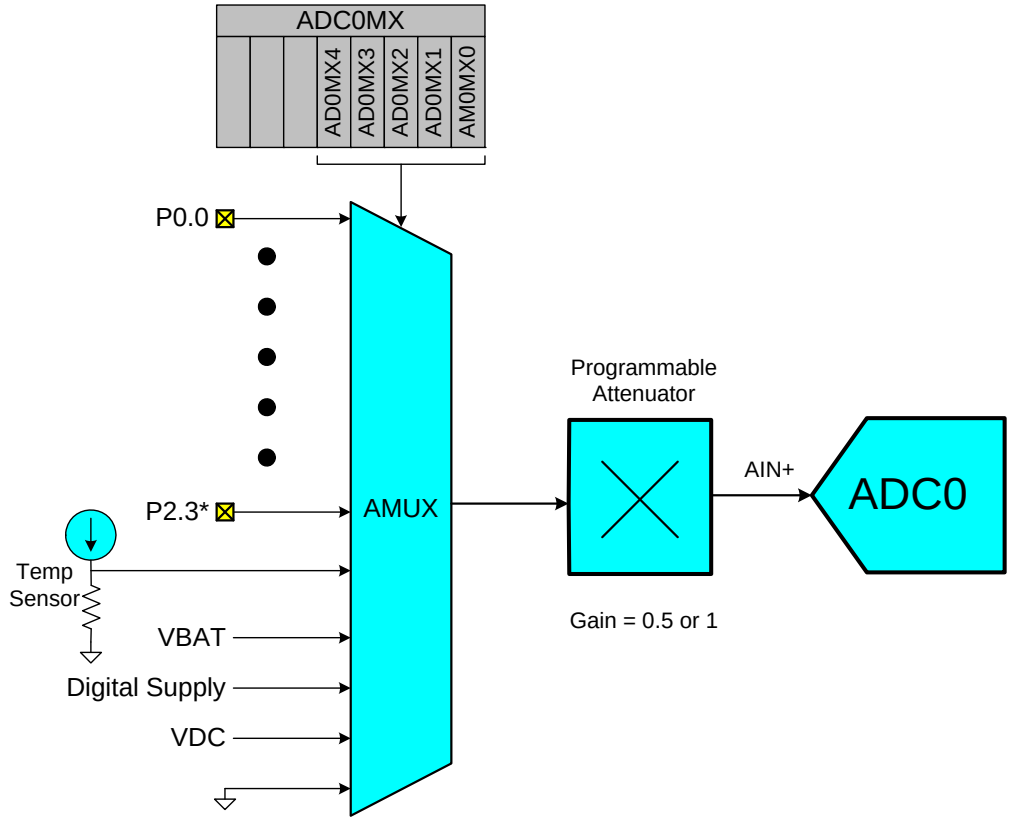
5.6.2. ADC0 Specifications

See “4. Electrical Characteristics” on page 56 for a detailed listing of ADC0 specifications.

5.7. ADC0 Analog Multiplexer

ADC0 on C8051F96x has an analog multiplexer, referred to as AMUX0.

AMUX0 selects the positive inputs to the single-ended ADC0. Any of the following may be selected as the positive input: Port I/O pins, the on-chip temperature sensor, the VBAT Power Supply, Regulated Digital Supply Voltage (Output of VREG0), VDC Supply, or the positive input may be connected to GND. The ADC0 input channels are selected in the ADC0MX register described in SFR Definition 5.12.



*P1.0 – P1.3 are not available as analog inputs

Figure 5.7. ADC0 Multiplexer Block Diagram

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN and disable the digital driver (PnMDOUT = 0 and Port Latch = 1). To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section “27. Port Input/Output” on page 351 for more Port I/O configuration details.

SFR Definition 5.12. ADC0MX: ADC0 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name				AD0MX				
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xBB

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care.
4:0	AD0MX	**AMUX0 Positive Input Selection.** Selects the positive input channel for ADC0.

00000:	P0.0	10000:	P2.0
00001:	P0.1	10001:	P2.1
00010:	P0.2	10010:	P2.2
00011:	P0.3	10011:	P2.3
00100:	P0.4	10100:	Reserved.
00101:	P0.5	10101:	Reserved.
00110:	P0.6	10110:	Reserved.
00111:	P0.7	10111:	Reserved.
01000:	Reserved	11000:	Reserved.
01001:	Reserved	11001:	Reserved.
01010:	Reserved	11010:	Reserved.
01011:	Reserved	11011:	Temperature Sensor
01100:	P1.4	11100:	VBAT Supply Voltage (1.8–3.6 V)
01101:	P1.5		
01110:	P1.6	11101:	Digital Supply Voltage (VREG0 Output, 1.7 V Typical)
01111:	P1.7		
		11110:	VDC Supply Voltage (1.8–3.6 V)
		11111:	Ground

5.8. Temperature Sensor

An on-chip temperature sensor is included on the C8051F96x which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC mux channel should select the temperature sensor. The temperature sensor transfer function is shown in Figure 5.8. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 5.15. REF0CN: Voltage Reference Control. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 4.12 for the slope and offset parameters of the temperature sensor.

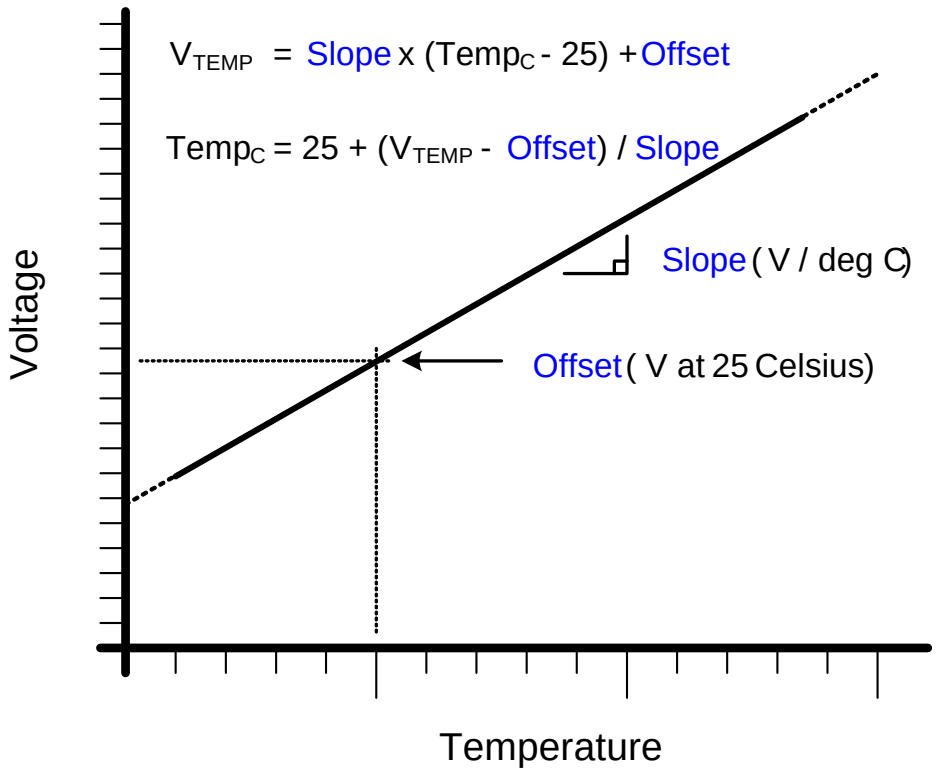


Figure 5.8. Temperature Sensor Transfer Function

5.8.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 4.13 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.9 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

A single-point offset measurement of the temperature sensor is performed on each device during production test. The measurement is performed at 25 °C ±5 °C, using the ADC with the internal high speed reference buffer selected as the Voltage Reference. The direct ADC result of the measurement is stored in the SFR registers TOFFH and TOFFL, shown in SFR Definition 5.13 and SFR Definition 5.14.

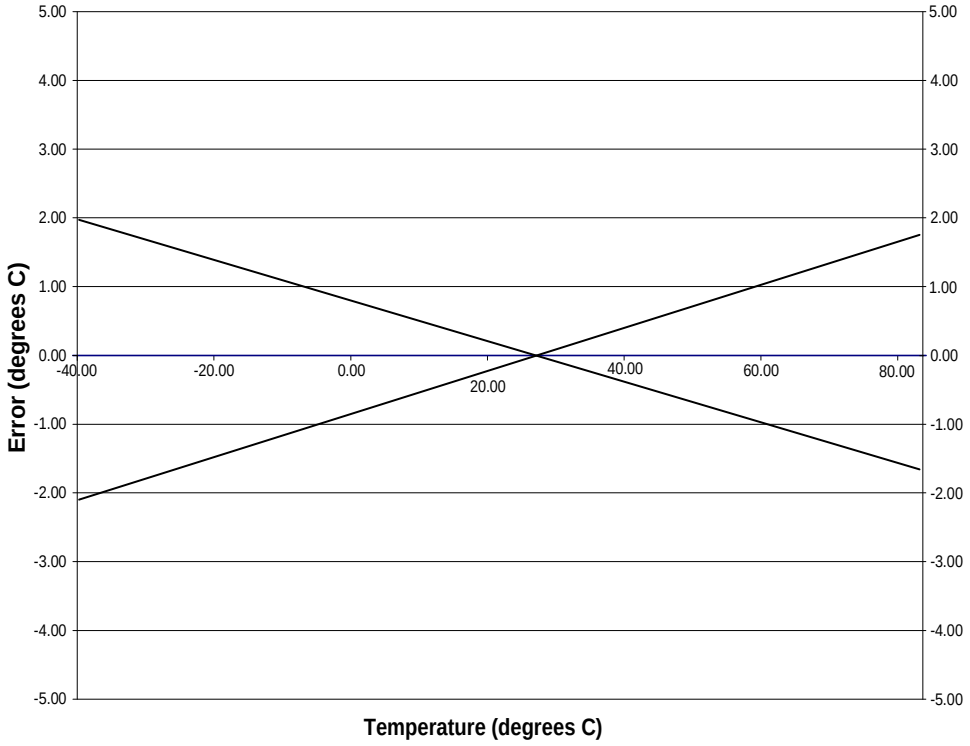


Figure 5.9. Temperature Sensor Error with 1-Point Calibration (V_{REF} = 1.68 V)

SFR Definition 5.13. TOFFH: Temperature Sensor Offset High Byte

Bit	7	6	5	4	3	2	1	0
Name	TOFF[9:2]							
Type	R	R	R	R	R	R	R	R
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Page = 0xF; SFR Address = 0xBE

Bit	Name	Function
7:0	TOFF[9:2]	**Temperature Sensor Offset High Bits.** Most Significant Bits of the 10-bit temperature sensor offset measurement.

SFR Definition 5.14. TOFFL: Temperature Sensor Offset Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TOFF[1:0]							
Type	R	R						
Reset	Varies	Varies	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xBD

Bit	Name	Function
7:6	TOFF[1:0]	**Temperature Sensor Offset Low Bits.** Least Significant Bits of the 10-bit temperature sensor offset measurement.
5:0	Unused	Read = 0; Write = Don't Care.

5.9. Voltage and Ground Reference Options

The voltage reference MUX is configurable to use an externally connected voltage reference, the internal voltage reference, or one of two power supply voltages (see Figure 5.10). The ground reference MUX allows the ground reference for ADC0 to be selected between the ground pin (GND) or a port pin dedicated to analog ground (P0.1/AGND).

The voltage and ground reference options are configured using the REF0CN SFR described on SFR Definition 5.15. REF0CN: Voltage Reference Control. Electrical specifications are can be found in the Electrical Specifications Chapter.

Important Note About the V_{REF} and AGND Inputs: Port pins are used as the external V_{REF} and AGND inputs. When using an external voltage reference or the internal precision reference, P0.0/VREF should be configured as an analog input and skipped by the Digital Crossbar. When using AGND as the ground reference to ADC0, P0.1/AGND should be configured as an analog input and skipped by the Digital Crossbar. Refer to Section “27. Port Input/Output” on page 351 for complete Port I/O configuration details. The external reference voltage must be within the range $0 \le V_{REF} \le VDD$ and the external ground reference must be at the same dc voltage potential as GND.

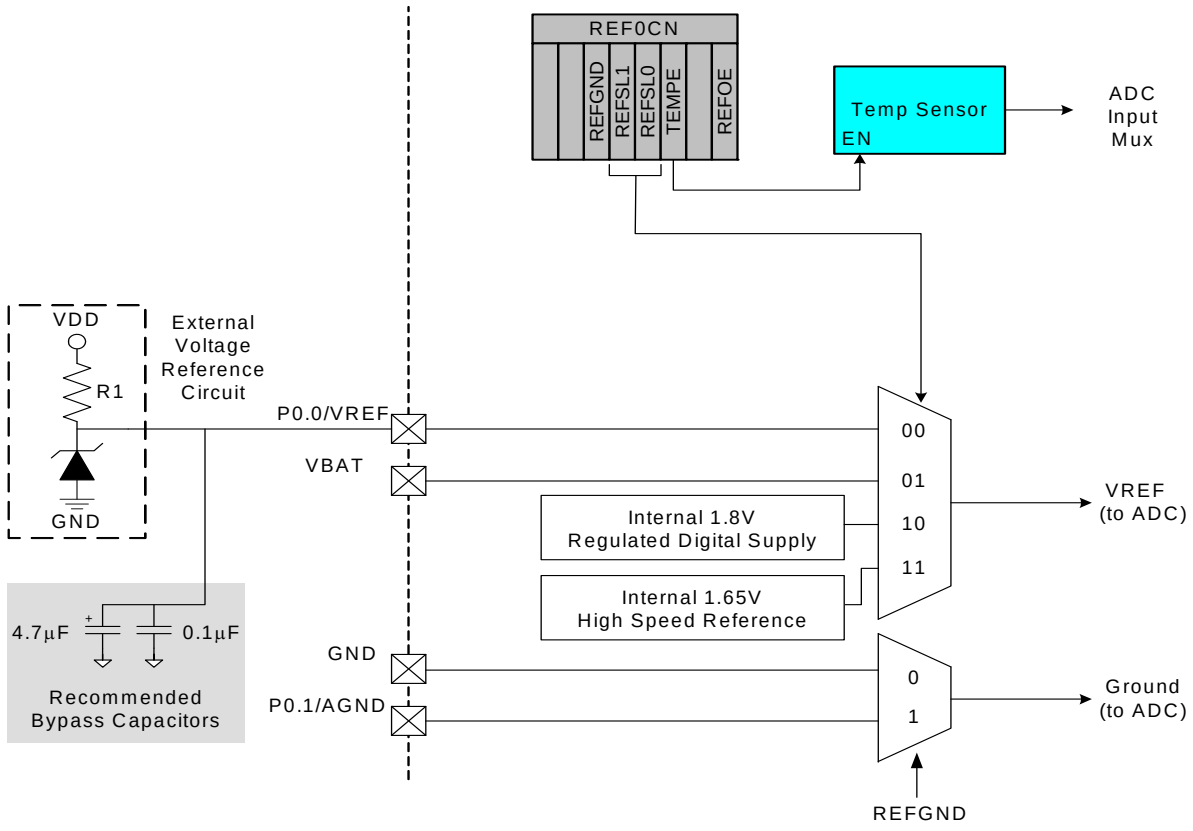


Figure 5.10. Voltage Reference Functional Block Diagram

5.10. External Voltage Reference

To use an external voltage reference, REFSL[1:0] should be set to 00. Bypass capacitors should be added as recommended by the manufacturer of the external voltage reference. If the manufacturer does not provide recommendations, a 4.7uF in parallel with a 0.1uF capacitor is recommended.

5.11. Internal Voltage Reference

For applications requiring the maximum number of port I/O pins, or very short VREF turn-on time, the 1.65 V high-speed reference will be the best internal reference option to choose. The high speed internal reference is selected by setting REFSL[1:0] to 11. When selected, the high speed internal reference will be automatically enabled/disabled on an as-needed basis by ADC0.

For applications with a non-varying power supply voltage, using the power supply as the voltage reference can provide ADC0 with added dynamic range at the cost of reduced power supply noise rejection. To use the 1.8 to 3.6 V power supply voltage (V_{DD}) or the 1.8 V regulated digital supply voltage as the reference source, REFSL[1:0] should be set to 01 or 10, respectively.

5.12. Analog Ground Reference

To prevent ground noise generated by switching digital logic from affecting sensitive analog measurements, a separate analog ground reference option is available. When enabled, the ground reference for ADC0 during both the tracking/sampling and the conversion periods is taken from the P0.1/AGND pin. Any external sensors sampled by ADC0 should be referenced to the P0.1/AGND pin. This pin should be connected to the ground terminal of any external sensors sampled by ADC0. If an external voltage reference is used, the P0.1/AGND pin should be connected to the ground of the external reference and its associated decoupling capacitor. The separate analog ground reference option is enabled by setting REFGND to 1. Note that when sampling the internal temperature sensor, the internal chip ground is always used for the sampling operation, regardless of the setting of the REFGND bit. Similarly, whenever the internal 1.65 V high-speed reference is selected, the internal chip ground is always used during the conversion period, regardless of the setting of the REFGND bit.

5.13. Temperature Sensor Enable

The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data. See Section “5.8. Temperature Sensor” on page 97 for details on temperature sensor characteristics when it is enabled.

SFR Definition 5.15. REF0CN: Voltage Reference Control

Bit	7	6	5	4	3	2	1	0
Name			REFGND	REFSL		TEMPE		
Type	R	R	R/W	R/W	R/W	R/W	R	R
Reset	0	0	0	1	1	0	0	0

SFR Page = 0x0; SFR Address = 0xD1

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care.
5	REFGND	**Analog Ground Reference.** Selects the ADC0 ground reference. 0: The ADC0 ground reference is the GND pin. 1: The ADC0 ground reference is the P0.1/AGND pin.
4:3	REFSL	**Voltage Reference Select.** Selects the ADC0 voltage reference. 00: The ADC0 voltage reference is the P0.0/VREF pin. 01: The ADC0 voltage reference is the VDD pin. 10: The ADC0 voltage reference is the internal 1.8 V digital supply voltage. 11: The ADC0 voltage reference is the internal 1.65 V high speed voltage reference.
2	TEMPE	**Temperature Sensor Enable.** Enables/Disables the internal temperature sensor. 0: Temperature Sensor Disabled. 1: Temperature Sensor Enabled.
1:0	Unused	Read = 00b; Write = Don't Care.

5.14. Voltage Reference Electrical Specifications

See Table 4.14 on page 72 for detailed Voltage Reference Electrical Specifications.

6. Programmable Current Reference (IREF0)

C8051F96x devices include an on-chip programmable current reference (source or sink) with two output current settings: Low Power Mode and High Current Mode. The maximum current output in Low Power Mode is 63 µA (1 µA steps) and the maximum current output in High Current Mode is 504 µA (8 µA steps).

The current source/sink is controlled though the IREF0CN special function register. It is enabled by setting the desired output current to a non-zero value. It is disabled by writing 0x00 to IREF0CN. The port I/O pin associated with ISRC0 should be configured as an analog input and skipped in the Crossbar. See “Port Input/Output” on page 351 for more details.

SFR Definition 6.1. IREF0CN: Current Reference Control

Bit	7	6	5	4	3	2	1	0
Name	SINK	MODE	IREF0DAT					
Type	R/W	R/W	R/W					
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xB9

Bit	Name	Function
7	SINK	**IREF0 Current Sink Enable.** Selects if IREF0 is a current source or a current sink. 0: IREF0 is a current source. 1: IREF0 is a current sink.
6	MDSEL	**IREF0 Output Mode Select.** Selects Low Power or High Current Mode. 0: Low Power Mode is selected (step size = 1 µA). 1: High Current Mode is selected (step size = 8 µA).
5:0	IREF0DAT[5:0]	**IREF0 Data Word.** Specifies the number of steps required to achieve the desired output current. Output current = direction x step size x IREF0DAT. IREF0 is in a low power state when IREF0DAT is set to 0x00.

6.1. PWM Enhanced Mode

The precision of the current reference can be increased by fine tuning the IREF0 output using a PWM signal generated by the PCA. This mode allows the IREF0DAT bits to perform a course adjustment on the IREF0 output. Any available PCA channel can perform a fine adjustment on the IREF0 output. When enabled (PWMEN = 1), the CEX signal selected using the PWMSS bit field is internally routed to IREF0 to control the on time of a current source having the weight of 2 LSBs. With the two least significant bits of IREF0DAT set to 00b, applying a 100% duty cycle on the CEX signal will be equivalent to setting the two LSBs of IREF0DAT to 10b. PWM enhanced mode is enabled and setup using the IREF0CF register.

SFR Definition 6.2. IREF0CF: Current Reference Configuration

Bit	7	6	5	4	3	2	1	0
Name	PWMEN					PWMSS[2:0]		
Type	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xB9

Bit	Name	Function
7	PWMEN	**PWM Enhanced Mode Enable.** Enables the PWM Enhanced Mode. 0: PWM Enhanced Mode disabled. 1: PWM Enhanced Mode enabled.
6:3	Unused	Read = 0000b, Write = don't care.
2:0	PWMSS[2:0]	**PWM Source Select.** Selects the PCA channel to use for the fine-tuning control signal. 000: CEX0 selected as fine-tuning control signal. 001: CEX1 selected as fine-tuning control signal. 010: CEX2 selected as fine-tuning control signal. 011: CEX3 selected as fine-tuning control signal. 100: CEX4 selected as fine-tuning control signal. 101: CEX5 selected as fine tuning control signal. All Other Values: Reserved.

6.2. IREF0 Specifications

See Table 4.15 on page 73 for a detailed listing of IREF0 specifications.

7. Comparators

C8051F96x devices include two on-chip programmable voltage comparators: Comparator 0 (CPT0) is shown in Figure 7.1; Comparator 1 (CPT1) is shown in Figure 7.2. The two comparators operate identically, but may differ in their ability to be used as reset or wake-up sources. See the Reset Sources chapter and the Power Management chapter for details on reset sources and low power mode wake-up sources, respectively.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output when the device is in some low power modes.

7.1. Comparator Inputs

Each Comparator performs an analog comparison of the voltage levels at its positive (CP0+ or CP1+) and negative (CP0- or CP1-) input. Both comparators support multiple port pin inputs multiplexed to their positive and negative comparator inputs using analog input multiplexers. The analog input multiplexers are completely under software control and configured using SFR registers. See Section “7.6. Comparator0 and Comparator1 Analog Multiplexers” on page 112 for details on how to select and configure Comparator inputs.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs and skipped by the Crossbar. See the Port I/O chapter for more details on how to configure Port I/O pins as Analog Inputs. The Comparator may also be used to compare the logic level of digital signals, however, Port I/O pins configured as digital inputs must be driven to a valid logic state (HIGH or LOW) to avoid increased power consumption.

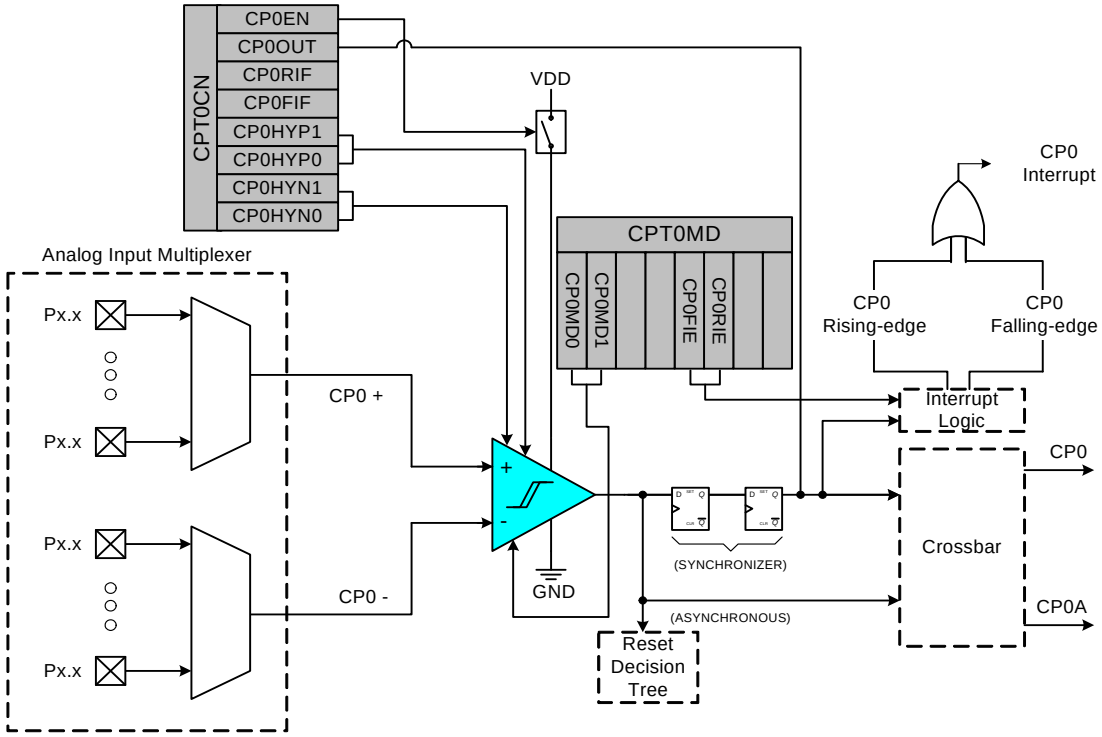


Figure 7.1. Comparator 0 Functional Block Diagram

7.2. Comparator Outputs

When a comparator is enabled, its output is a logic 1 if the voltage at the positive input is higher than the voltage at the negative input. When disabled, the comparator output is a logic 0. The comparator output is synchronized with the system clock as shown in Figure 7.2. The synchronous “latched” output (CP0, CP1) can be polled in software (CPnOUT bit), used as an interrupt source, or routed to a Port pin through the Crossbar.

The asynchronous “raw” comparator output (CP0A, CP1A) is used by the low power mode wakeup logic and reset decision logic. See the Power Options chapter and the Reset Sources chapter for more details on how the asynchronous comparator outputs are used to make wake-up and reset decisions. The asynchronous comparator output can also be routed directly to a Port pin through the Crossbar, and is available for use outside the device even if the system clock is stopped.

When using a Comparator as an interrupt source, Comparator interrupts can be generated on rising-edge and/or falling-edge comparator output transitions. Two independent interrupt flags (CPnRIF and CPnFIF) allow software to determine which edge caused the Comparator interrupt. The comparator rising-edge and falling-edge interrupt flags are set by hardware when a corresponding edge is detected regardless of the interrupt enable state. Once set, these bits remain set until cleared by software.

The rising-edge and falling-edge interrupts can be individually enabled using the CPnRIE and CPnFIE interrupt enable bits in the CPTnMD register. In order for the CPnRIF and/or CPnFIF interrupt flags to generate an interrupt request to the CPU, the Comparator must be enabled as an interrupt source and global interrupts must be enabled. See the Interrupt Handler chapter for additional information.

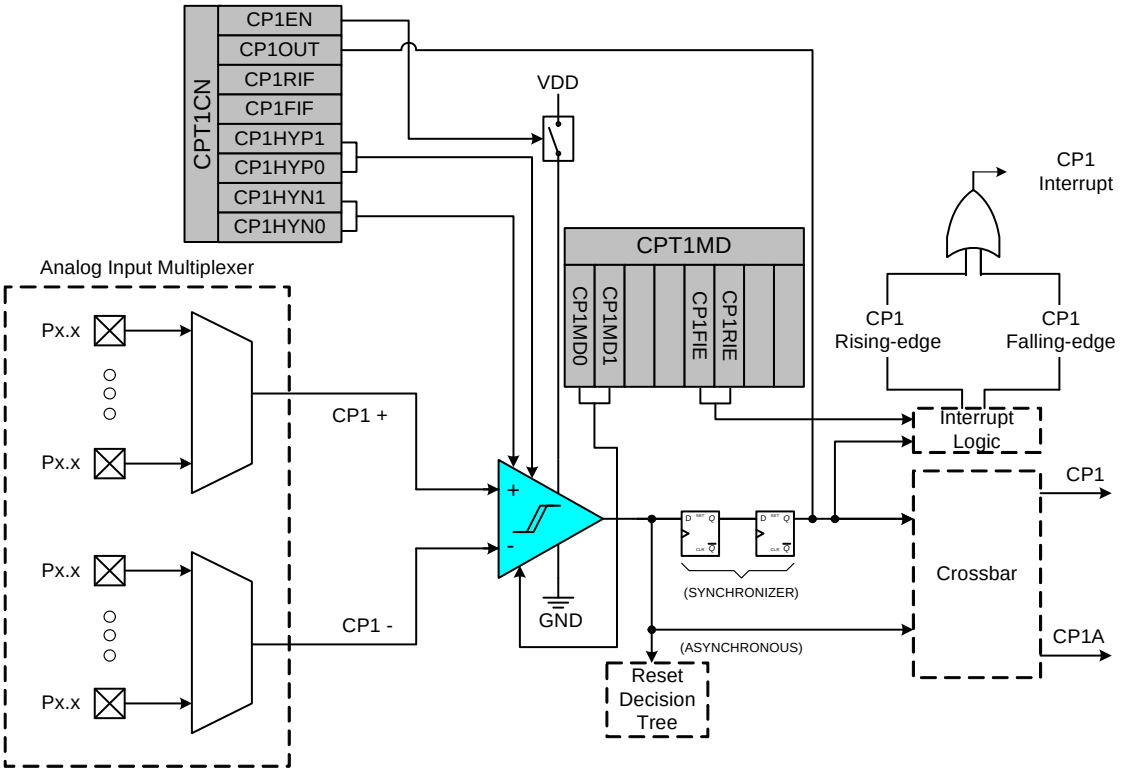


Figure 7.2. Comparator 1 Functional Block Diagram

7.3. Comparator Response Time

Comparator response time may be configured in software via the CPTnMD registers described on "CPT0MD: Comparator 0 Mode Selection" on page 109 and "CPT1MD: Comparator 1 Mode Selection" on page 111. Four response time settings are available: Mode 0 (Fastest Response Time), Mode 1, Mode 2, and Mode 3 (Lowest Power). Selecting a longer response time reduces the Comparator active supply current. The Comparators also have low power shutdown state, which is entered any time the comparator is disabled. Comparator rising edge and falling edge response times are typically not equal. See Table 4.16 on page 74 for complete comparator timing and supply current specifications.

7.4. Comparator Hysterisis

The Comparators feature software-programmable hysterisis that can be used to stabilize the comparator output while a transition is occurring on the input. Using the CPTnCN registers, the user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage (i.e., the comparator negative input).

Figure 7.3 shows that when positive hysterisis is enabled, the comparator output does not transition from logic 0 to logic 1 until the comparator positive input voltage has exceeded the threshold voltage by an amount equal to the programmed hysterisis. It also shows that when negative hysterisis is enabled, the comparator output does not transition from logic 1 to logic 0 until the comparator positive input voltage has fallen below the threshold voltage by an amount equal to the programmed hysterisis.

The amount of positive hysterisis is determined by the settings of the CPnHYP bits in the CPTnCN register and the amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits in the same register. Settings of 20 mV, 10 mV, 5 mV, or 0 mV can be programmed for both positive and negative hysterisis. See Section "Table 4.16. Comparator Electrical Characteristics" on page 74 for complete comparator hysterisis specifications.

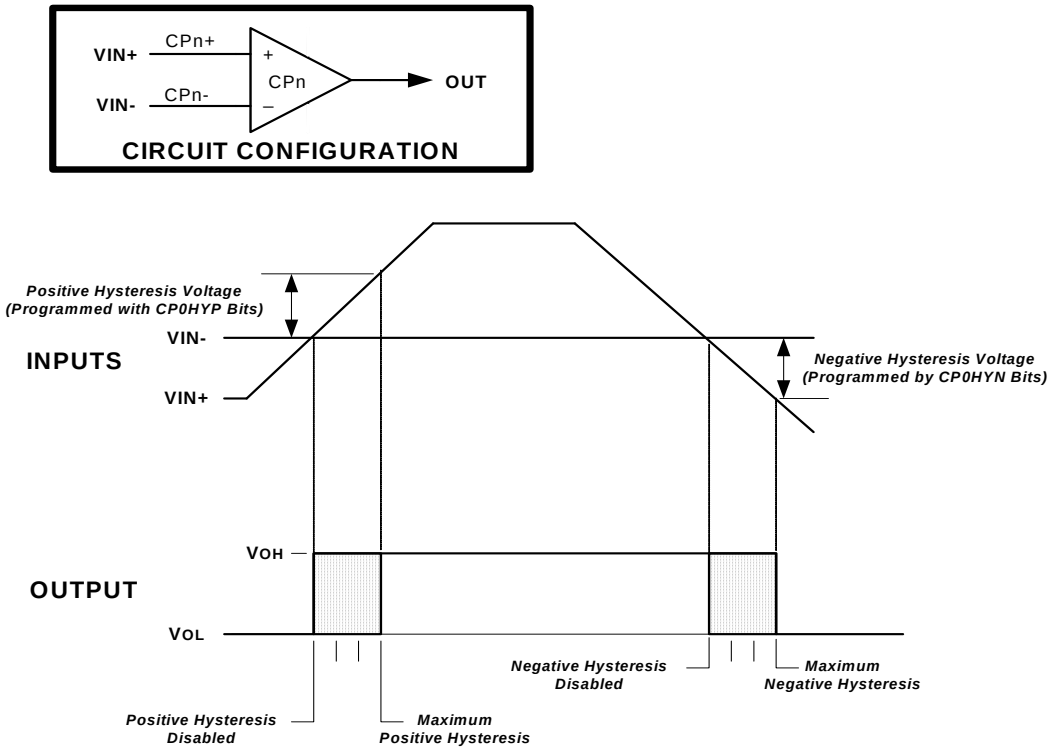


Figure 7.3. Comparator Hysteresis Plot

7.5. Comparator Register Descriptions

The SFRs used to enable and configure the comparators are described in the following register descriptions. A Comparator must be enabled by setting the CPnEN bit to logic 1 before it can be used. From an enabled state, a comparator can be disabled and placed in a low power state by clearing the CPnEN bit to logic 0.

Important Note About Comparator Settings: False rising and falling edges can be detected by the Comparator while powering on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed. The Comparator Power Up Time is specified in Section “Table 4.16. Comparator Electrical Characteristics” on page 74.

SFR Definition 7.1. CPT0CN: Comparator 0 Control

Bit	7	6	5	4	3	2	1	0
Name	CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP[1:0]		CP0HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0x9B

Bit	Name	Function
7	CP0EN	**Comparator0 Enable Bit.** 0: Comparator0 Disabled. 1: Comparator0 Enabled.
6	CP0OUT	**Comparator0 Output State Flag.** 0: Voltage on CP0+ < CP0–. 1: Voltage on CP0+ > CP0–.
5	CP0RIF	**Comparator0 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator0 Rising Edge has occurred since this flag was last cleared. 1: Comparator0 Rising Edge has occurred.
4	CP0FIF	**Comparator0 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator0 Falling-Edge has occurred since this flag was last cleared. 1: Comparator0 Falling-Edge has occurred.
3-2	CP0HYP[1:0]	**Comparator0 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1-0	CP0HYN[1:0]	**Comparator0 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

SFR Definition 7.2. CPT0MD: Comparator 0 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP0RIE	CP0FIE			CP0MD[1:0]	
Type	R/W	R	R/W	R/W	R	R	R/W	
Reset	1	0	0	0	0	0	1	0

SFR Page = 0x0; SFR Address = 0x9D

Bit	Name	Function
7	Reserved	Read = 1b, Must Write 1b.
6	Unused	Read = 0b, Write = don't care.
5	CP0RIE	**Comparator0 Rising-Edge Interrupt Enable.** 0: Comparator0 Rising-edge interrupt disabled. 1: Comparator0 Rising-edge interrupt enabled.
4	CP0FIE	**Comparator0 Falling-Edge Interrupt Enable.** 0: Comparator0 Falling-edge interrupt disabled. 1: Comparator0 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP0MD[1:0]	**Comparator0 Mode Select** These bits affect the response time and power consumption for Comparator0. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

SFR Definition 7.3. CPT1CN: Comparator 1 Control

Bit	7	6	5	4	3	2	1	0
Name	CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP[1:0]		CP1HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0x9A

Bit	Name	Function
7	CP1EN	**Comparator1 Enable Bit.** 0: Comparator1 Disabled. 1: Comparator1 Enabled.
6	CP1OUT	**Comparator1 Output State Flag.** 0: Voltage on CP1+ < CP1–. 1: Voltage on CP1+ > CP1–.
5	CP1RIF	**Comparator1 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator1 Rising Edge has occurred since this flag was last cleared. 1: Comparator1 Rising Edge has occurred.
4	CP1FIF	**Comparator1 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator1 Falling-Edge has occurred since this flag was last cleared. 1: Comparator1 Falling-Edge has occurred.
3:2	CP1HYP[1:0]	**Comparator1 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CP1HYN[1:0]	**Comparator1 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

SFR Definition 7.4. CPT1MD: Comparator 1 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP1RIE	CP1FIE			CP1MD[1:0]	
Type	R/W	R	R/W	R/W	R	R	R/W	
Reset	1	0	0	0	0	0	1	0

SFR Page = 0x0; SFR Address = 0x9C

Bit	Name	Function
7	Reserved	Read = 1b, Must Write 1b.
6	Unused	**Unused.** Read = 00b, Write = don't care.
5	CP1RIE	**Comparator1 Rising-Edge Interrupt Enable.** 0: Comparator1 Rising-edge interrupt disabled. 1: Comparator1 Rising-edge interrupt enabled.
4	CP1FIE	**Comparator1 Falling-Edge Interrupt Enable.** 0: Comparator1 Falling-edge interrupt disabled. 1: Comparator1 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP1MD[1:0]	**Comparator1 Mode Select** These bits affect the response time and power consumption for Comparator1. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

7.6. Comparator0 and Comparator1 Analog Multiplexers

Comparator0 and Comparator1 on C8051F96x devices have analog input multiplexers to connect Port I/O pins and internal signals the comparator inputs; CP0+/CP0- are the positive and negative input multiplexers for Comparator0 and CP1+/CP1- are the positive and negative input multiplexers for Comparator1.

The comparator input multiplexers directly support capacitive sensors. When the Compare input is selected on the positive or negative multiplexer, any Port I/O pin connected to the other multiplexer can be directly connected to a capacitive sensor with no additional external components. The Compare signal provides the appropriate reference level for detecting when the capacitive sensor has charged or discharged through the on-chip Rsense resistor. The Comparator0 output can be routed to Timer2 for capturing the capacitor's charge and discharge time. See Section "32. Timers" on page 444 for details.

Any of the following may be selected as comparator inputs: Port I/O pins, Capacitive Touch Sense Compare, VDD/DC+ Supply Voltage, Regulated Digital Supply Voltage (Output of VREG0), the VBAT Supply voltage or ground. The Comparator's supply voltage divided by 2 is also available as an input; the resistors used to divide the voltage only draw current when this setting is selected. The Comparator input multiplexers are configured using the CPT0MX and CPT1MX registers described in SFR Definition 7.5 and SFR Definition 7.6.

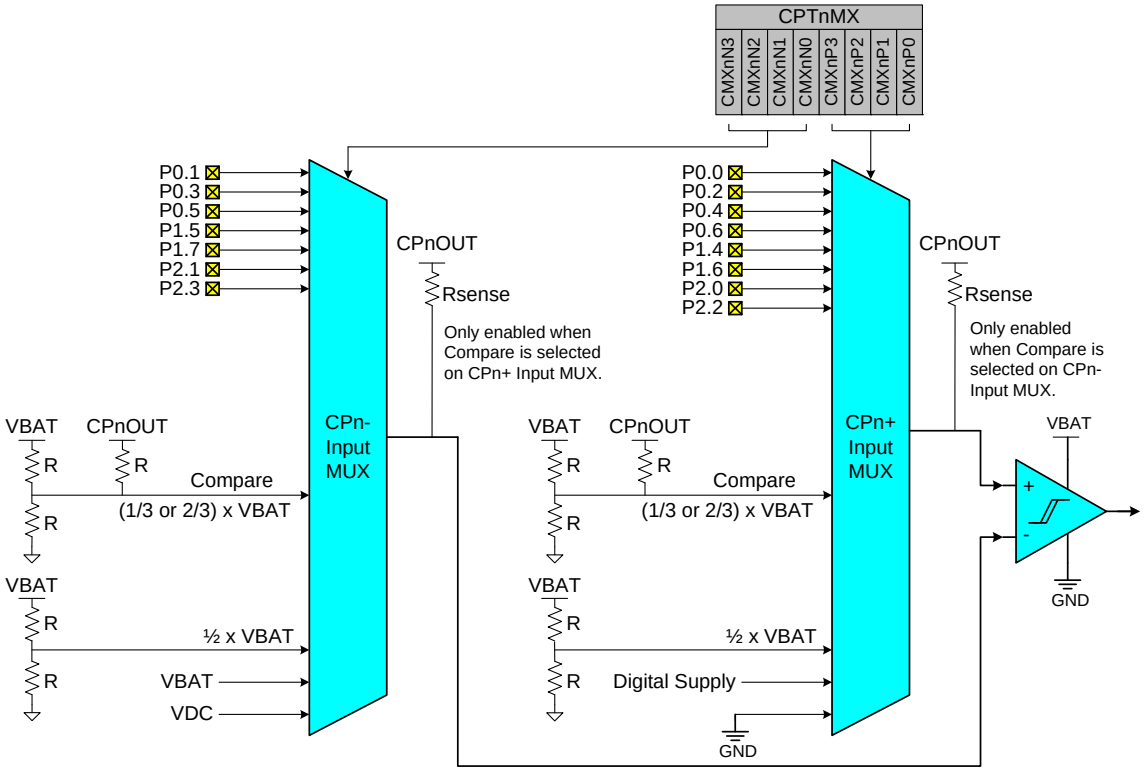


Figure 7.4. CPn Multiplexer Block Diagram

Important Note About Comparator Input Configuration: Port pins selected as comparator inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to 0 the corresponding bit in register PnMDIN and disable the digital driver (PnMDOUT = 0 and Port Latch = 1). To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP. See Section "27. Port Input/Output" on page 351 for more Port I/O configuration details.

SFR Definition 7.5. CPT0MX: Comparator0 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX0N[3:0]				CMX0P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0x9F

Bit	Name	Function			
7:4	CMX0N	**Comparator0 Negative Input Selection.** Selects the negative input channel for Comparator0.			
		0000:	P0.1	1000:	P2.1
		0001:	P0.3	1001:	P2.3
		0010:	P0.5	1010:	Reserved
		0011:	Reserved	1011:	Reserved
		0100:	Reserved	1100:	Compare
		0101:	Reserved	1101:	VBAT divided by 2
		0110:	P1.5	1110:	Digital Supply Voltage
		0111:	P1.7	1111:	Ground
3:0	CMX0P	**Comparator0 Positive Input Selection.** Selects the positive input channel for Comparator0.			
		0000:	P0.0	1000:	P2.0
		0001:	P0.2	1001:	P2.2
		0010:	P0.4	1010:	Reserved
		0011:	P0.6	1011:	Reserved
		0100:	Reserved	1100:	Compare
		0101:	Reserved	1101:	VBAT divided by 2
		0110:	P1.4	1110:	VBAT Supply Voltage
		0111:	P1.6	1111:	VBAT Supply Voltage

SFR Definition 7.6. CPT1MX: Comparator1 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX1N[3:0]				CMX1P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0x9E

Bit	Name	Function			
7:4	CMX1N	**Comparator1 Negative Input Selection.** Selects the negative input channel for Comparator1.			
		0000:	P0.1	1000:	P2.1
		0001:	P0.3	1001:	P2.3
		0010:	P0.5	1010:	Reserved
		0011:	Reserved	1011:	Reserved
		0100:	Reserved	1100:	Compare
		0101:	Reserved	1101:	VBAT divided by 2
		0110:	P1.5	1110:	Digital Supply Voltage
		0111:	P1.7	1111:	Ground
3:0	CMX1P	**Comparator1 Positive Input Selection.** Selects the positive input channel for Comparator1.			
		0000:	P0.0	1000:	P2.0
		0001:	P0.2	1001:	P2.2
		0010:	P0.4	1010:	Reserved
		0011:	P0.6	1011:	Reserved
		0100:	Reserved	1100:	Compare
		0101:	Reserved	1101:	VBAT divided by 2
		0110:	P1.4	1110:	VBAT Supply Voltage
		0111:	P1.6	1111:	VDC Supply Voltage

8. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 34), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 8.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

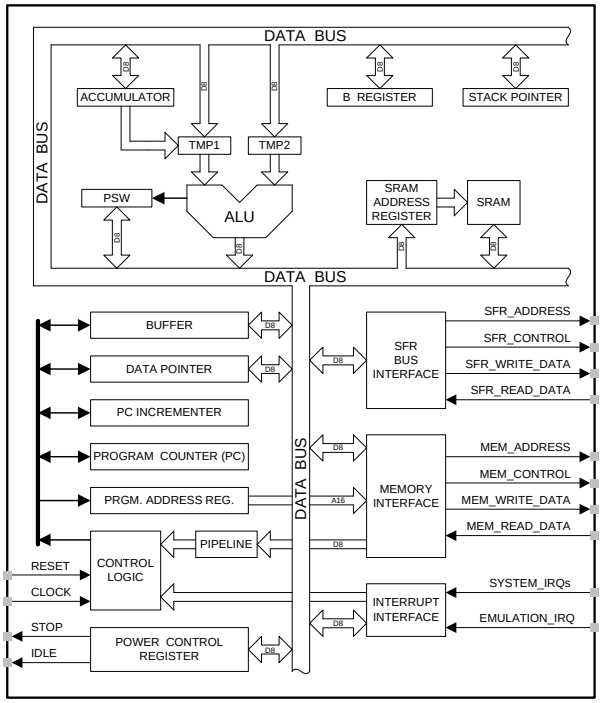


Figure 8.1. CIP-51 Block Diagram

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

Programming and Debugging Support
In-system programming of the flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2).

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debugging is completely non-intrusive, requiring no RAM, Stack, timers, or other on-chip resources. C2 details can be found in Section “34. C2 Interface” on page 486.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

8.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

8.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 8.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

Table 8.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
Program Branching			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3
JNZ rel	Jump if A does not equal zero	2	2/3
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/4
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4
NOP	No operation	1	1

Notes on Registers, Operands and Addressing Modes:

Rn—Register R0–R7 of the currently selected register bank.

@Ri—Data RAM location addressed indirectly through R0 or R1.

rel—8-bit, signed (twos complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).

#data—8-bit constant

#data16—16-bit constant

bit—Direct-accessed bit in Data RAM or SFR

addr11—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

addr16—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 8 kB program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.
All mnemonics copyrighted © Intel Corporation 1980.

8.2. CIP-51 Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic l. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the data sheet associated with their corresponding system function.

SFR Definition 8.1. DPL: Data Pointer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	DPL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0x82

Bit	Name	Function
7:0	DPL[7:0]	**Data Pointer Low.** The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed flash memory or XRAM.

SFR Definition 8.2. DPH: Data Pointer High Byte

Bit	7	6	5	4	3	2	1	0
Name	DPH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0x83

Bit	Name	Function
7:0	DPH[7:0]	**Data Pointer High.** The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed flash memory or XRAM.

SFR Definition 8.3. SP: Stack Pointer

Bit	7	6	5	4	3	2	1	0
Name	SP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	1	1	1

SFR Page = All Pages; SFR Address = 0x81

Bit	Name	Function
7:0	SP[7:0]	**Stack Pointer.** The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 8.4. ACC: Accumulator

Bit	7	6	5	4	3	2	1	0
Name	ACC[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xE0; Bit-Addressable

Bit	Name	Function
7:0	ACC[7:0]	**Accumulator.** This register is the accumulator for arithmetic operations.

SFR Definition 8.5. B: B Register

Bit	7	6	5	4	3	2	1	0
Name	B[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xF0; Bit-Addressable

Bit	Name	Function
7:0	B[7:0]	**B Register.** This register serves as a second accumulator for certain arithmetic operations.

SFR Definition 8.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	PARITY
Type	R/W	R/W	R/W	R/W		R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xD0; Bit-Addressable

Bit	Name	Function
7	CY	**Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	**Auxiliary Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under software control.
4:3	RS[1:0]	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: ■ An ADD, ADDC, or SUBB instruction causes a sign-change overflow. ■ A MUL instruction results in an overflow (result is greater than 255). ■ A DIV instruction causes a divide-by-zero condition. The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under software control.
0	PARITY	**Parity Flag.** This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

9. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051F96x device family is shown in Figure 9.1

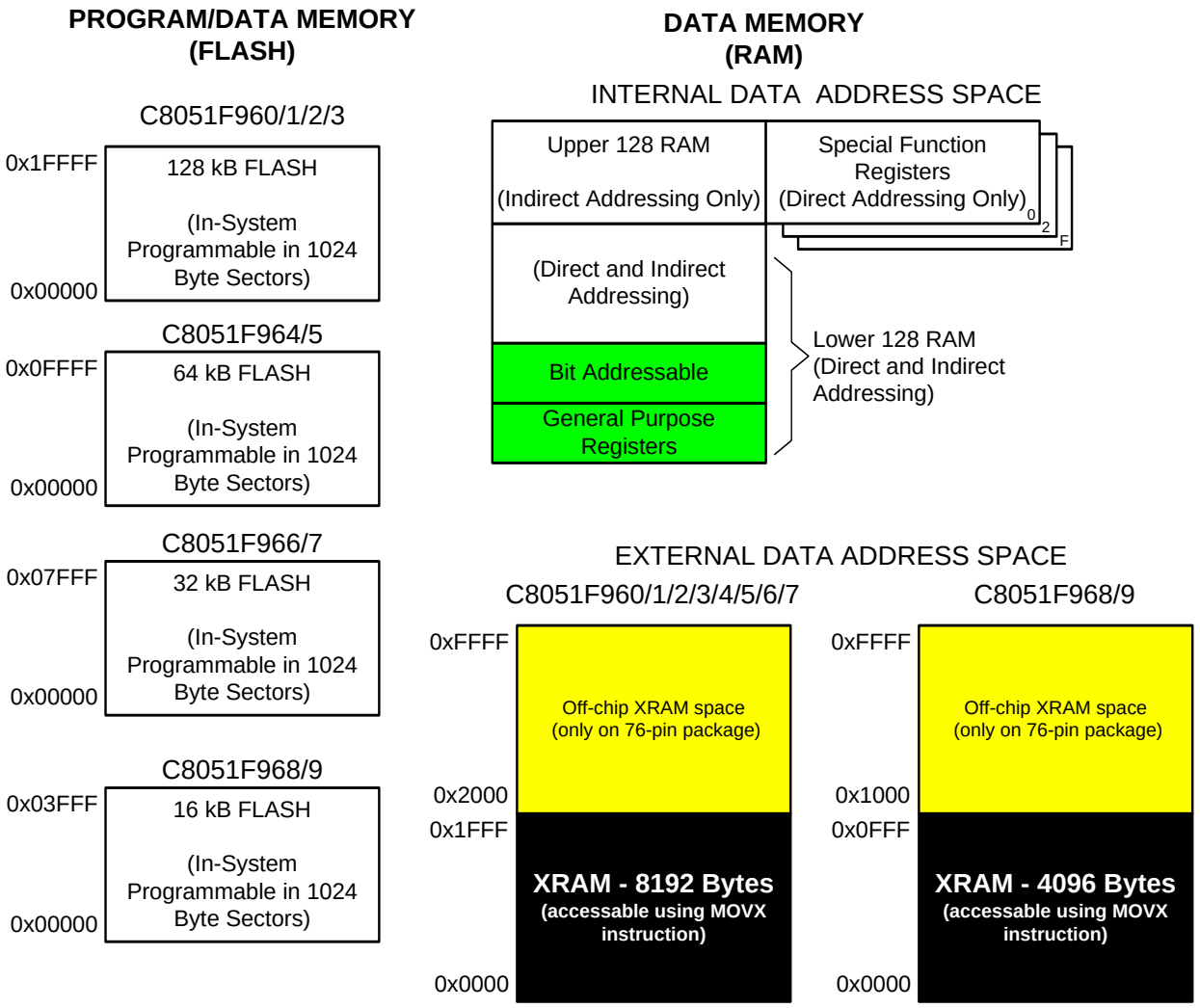


Figure 9.1. C8051F96x Memory Map

9.1. Program Memory

The C8051F960/1/2/3 device flashs have a 128 kB program memory space, C8051F964/5 devices have 64 kB program memory space, C8051F966/7 devices have 32 kB program memory space, and C8051F968/9 devices have a 16 kB program memory space. The devices with 128 kB flash implement this program memory space as in-system re-programmable flash memory in four 32 kB code banks. A common code bank (Bank 0) of 32 kB is always accessible from addresses 0x0000 to 0x7FFF. The upper code banks (Bank 1, Bank 2, and Bank 3) are each mapped to addresses 0x8000 to 0xFFFF, depending on the

selection of bits in the PSBANK register, as described in SFR Definition 9.1. All other devices with 64 kB or less program memory can be used as non-banked devices.

The IFBANK bits select which of the upper banks are used for code execution, while the COBANK bits select the bank to be used for direct writes and reads of the flash memory.

The address 0x1FFFF (C8051F960/1/2/3), 0xFFFF (C8051F964/5), 0x07FFF (C8051F966/7), or 0x3FFF (C8051F9608/9) serves as the security lock byte for the device. Any addresses above the lock byte are reserved.

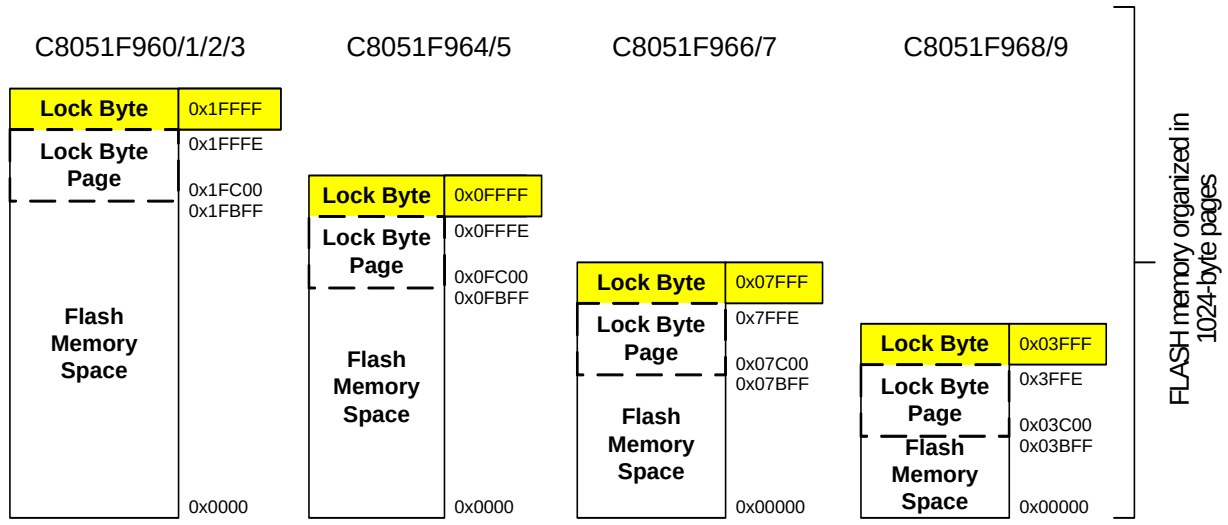


Figure 9.2. Flash Program Memory Map

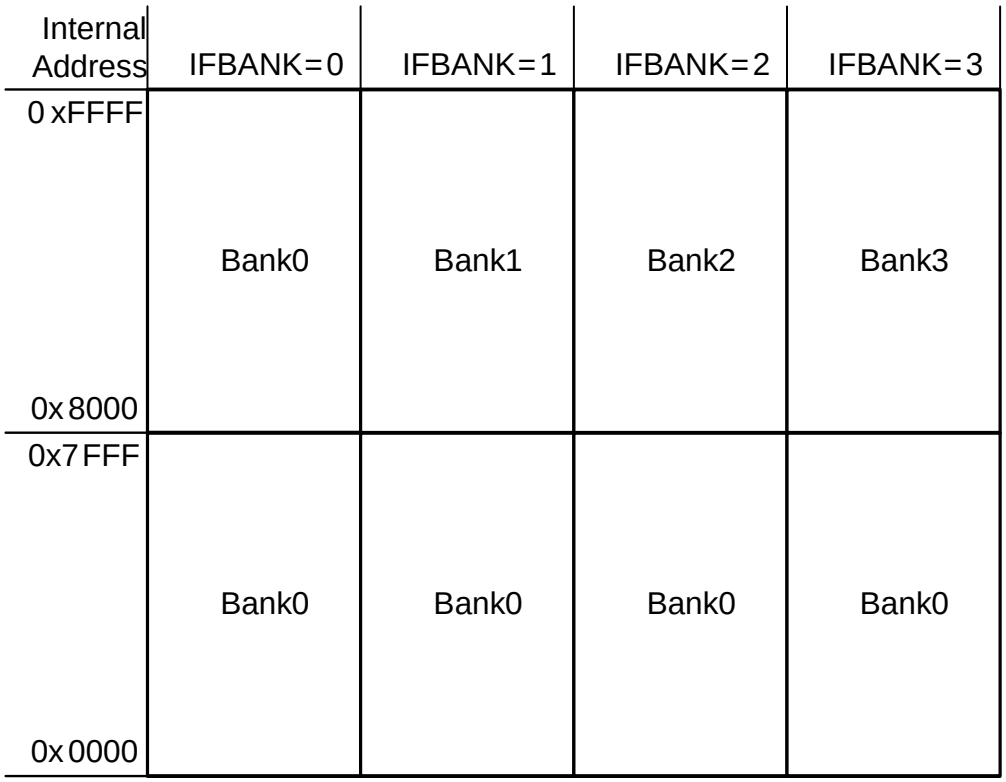


Figure 9.3. Address Memory Map for Instruction Fetches

SFR Definition 9.1. PSBANK: Program Space Bank Select

Bit	7	6	5	4	3	2	1	0
Name			COBANK[1:0]				IFBANK[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0x84

Bit	Name	Function
7:6	Reserved	Read = 00b, Must Write = 00b.
5:4	COBANK[1:0]	**Constant Operations Bank Select.** These bits select which flash bank is targeted during constant operations (MOVC and flash MOVX) involving address 0x8000 to 0xFFFF. 00: Constant Operations Target Bank 0 (note that Bank 0 is also mapped between 0x0000 to 0x7FFF). 01: Constant operations target Bank 1. 10: Constant operations target Bank 2. 11: Constant operations target Bank 3.
3:2	Reserved	Read = 00b, Must Write = 00b.
1:0	IFBANK[1:0]	**Instruction Fetch Operations Bank Select.** These bits select which flash bank is used for instruction fetches involving address 0x8000 to 0xFFFF. These bits can only be changed from code in Bank 0. 00: Instructions fetch from Bank 0 (note that Bank 0 is also mapped between 0x0000 to 0x7FFF). 01: Instructions fetch from Bank 1. 10: Instructions fetch from Bank 2. 11: Instructions fetch from Bank 3.

Note:
1. COBANK[1:0] and IFBANK[1:0] should not be set to select Bank 3 (11b) on the C8051F963/4/5 devices.
2. COBANK[1:0] and IFBANK[1:0] should not be set to select Bank 2 (10b) or Bank 3 (11b) on the C8051F966/7/8 devices.

9.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F96x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip flash memory space. MOVC instructions are always used to read flash memory, while MOVX write instructions are used to erase and write flash. This flash access feature provides a mechanism for the C8051F96x to update program code and use the program memory space for non-volatile data storage. Refer to Section “18. Flash Memory” on page 244 for further details.

9.2. Data Memory

The C8051F96x device family includes 8448 bytes (C8051F960/1/2/3/4/5/6/7) or 4352 bytes (C8051F968/9) of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. 8192 or 4096 bytes of this memory is on-chip “external” memory. The data memory map is shown in Figure 9.1 for reference.

9.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 9.1 illustrates the data memory organization of the C8051F96x.

9.2.1.1. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 8.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

9.2.1.2. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV   C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

9.2.1.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

9.2.2. External RAM

There are 8192 bytes or 4096 bytes of on-chip RAM mapped into the external data memory space. All of these address locations may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode (such as @R1) in combination with the EMI0CN register. Additional off-chip memory or memory-mapped devices may be mapped to the external memory address space and accessed using the external memory interface. See Section “10. External Data Memory Interface and On-Chip XRAM” on page 129 for further details.

10. External Data Memory Interface and On-Chip XRAM

For C8051F96x devices, 8 kB of RAM are included on-chip and mapped into the external data memory space (XRAM). Additionally, an External Memory Interface (EMIF) is available on the C8051F960/2/4/6/8 devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 10.1).

Note: The MOVX instruction can also be used for writing to the flash memory. See Section "18. Flash Memory" on page 244 for details. The MOVX instruction accesses XRAM by default.

10.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

10.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

10.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

10.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Configure the Output Modes of the associated port pins as either push-pull or open-drain (push-pull is most common). The Input Mode of the associated port pins should be set to digital (reset value).
2. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic 1).
3. Select Multiplexed mode or Non-multiplexed mode.
4. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
5. Set up timing to interface with off-chip memory or peripherals.

Each of these five steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in SFR Definition .

10.3. Port Configuration

The External Memory Interface appears on Ports 3, 4, 5, and 6 when it is used for off-chip memory access. The external memory interface and the LCD cannot be used simultaneously. When using EMIF, all pins on Port 3-6 may only be used for EMIF purposes or as general purpose I/O. The EMIF pinout is shown in Table 10.1 on page 131.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar settings for those pins. See Section “27. Port Input/Output” on page 351 for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to “park” the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode.

The C8051F960/2/4/6/8 devices support both the multiplexed and non-multiplexed modes. Accessing off-chip memory is not supported by the C8051F961/3/5/7/9 devices.

Table 10.1. EMIF Pinout (C8051F960/2/4/6/8)

Multiplexed Mode		
Signal Name	**Port Pin**	
	8-Bit Mode[1]	**16-Bit Mode[2]**
$\overline{RD}$	P3.6	P3.6
$\overline{WR}$	P3.7	P3.7
ALE	P3.5	P3.5
AD0	P6.0	P6.0
AD1	P6.1	P6.1
AD2	P6.2	P6.2
AD3	P6.3	P6.3
AD4	P6.4	P6.4
AD5	P6.5	P6.5
AD6	P6.6	P6.6
AD7	P6.7	P6.7
A8	—	P5.0
A9	—	P5.1
A10	—	P5.2
A11	—	P5.3
A12	—	P5.4
A13	—	P5.5
A14	—	P5.6
A15	—	P5.7
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—
Required I/O:	**11**	**19**

Non Multiplexed Mode		
Signal Name	**Port Pin**	
	8-Bit Mode [1]	**16-Bit Mode[2]**
$\overline{RD}$	P3.6	P3.6
$\overline{WR}$	P3.7	P3.7
D0	P6.0	P6.0
D1	P6.1	P6.1
D2	P6.2	P6.2
D3	P6.3	P6.3
D4	P6.4	P6.4
D5	P6.5	P6.5
D6	P6.6	P6.6
D7	P6.7	P6.7
A0	P5.0	P5.0
A1	P5.1	P5.1
A2	P5.2	P5.2
A3	P5.3	P5.3
A4	P5.4	P5.4
A5	P5.5	P5.5
A6	P5.6	P5.6
A7	P5.7	P5.7
A8	—	P4.0
A9	—	P4.1
A10	—	P4.2
A11	—	P4.3
A12	—	P4.4
A13	—	P4.5
A14	—	P4.6
A15	—	P4.7
Required I/O:	**18**	**26**

Notes:
1. Using 8-bit movx instruction without bank select.
2. Using 16-bit movx instruction.

SFR Definition 10.1. EMI0CN: External Memory Interface Control

Bit	7	6	5	4	3	2	1	0
Name	PGSEL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAA

Bit	Name	Function
7:0	PGSEL[7:0]	**XRAM Page Select Bits.** The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. 0x00: 0x0000 to 0x00FF 0x01: 0x0100 to 0x01FF ... 0xFE: 0xFE00 to 0xFEFF 0xFF: 0xFF00 to 0xFFFF

SFR Definition 10.2. EMI0CF: External Memory Configuration

Bit	7	6	5	4	3	2	1	0
Name				EMD2	EMD[1:0]		EALE[1:0]	
Type	R/W							
Reset	0	0	0	0	0	0	1	1

SFR Page = 0x0; SFR Address = 0xAB

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care.
4	EMD2	**EMIF Multiplex Mode Select Bit.** 0: EMIF operates in multiplexed address/data mode 1: EMIF operates in non-multiplexed mode (separate address and data pins)
3:2	EMD[1:0]	**EMIF Operating Mode Select Bits.** 00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space 01: Split Mode without Bank Select: Accesses below the 8 kB boundary are directed on-chip. Accesses above the 8 kB boundary are directed off-chip. 8-bit off-chip MOVX operations use current contents of the Address high port latches to resolve the upper address byte. To access off chip space, EMI0CN must be set to a page that is not contained in the on-chip address space. 10: Split Mode with Bank Select: Accesses below the 8 kB boundary are directed on-chip. Accesses above the 8 kB boundary are directed off-chip. 8-bit off-chip MOVX operations uses the contents of EMI0CN to determine the high-byte of the address. 11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.
1:0	EALE[1:0]	**ALE Pulse-Width Select Bits.** These bits only have an effect when EMD2 = 0. 00: ALE high and ALE low pulse width = 1 SYSCLK cycle. 01: ALE high and ALE low pulse width = 2 SYSCLK cycles. 10: ALE high and ALE low pulse width = 3 SYSCLK cycles. 11: ALE high and ALE low pulse width = 4 SYSCLK cycles.

10.4. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMI0CF.4) bit.

10.4.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 10.1.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the Q outputs reflect the states of the D inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time $\overline{RD}$ or $\overline{WR}$ is asserted.

See Section “10.6.2. Multiplexed Mode” on page 142 for more information.

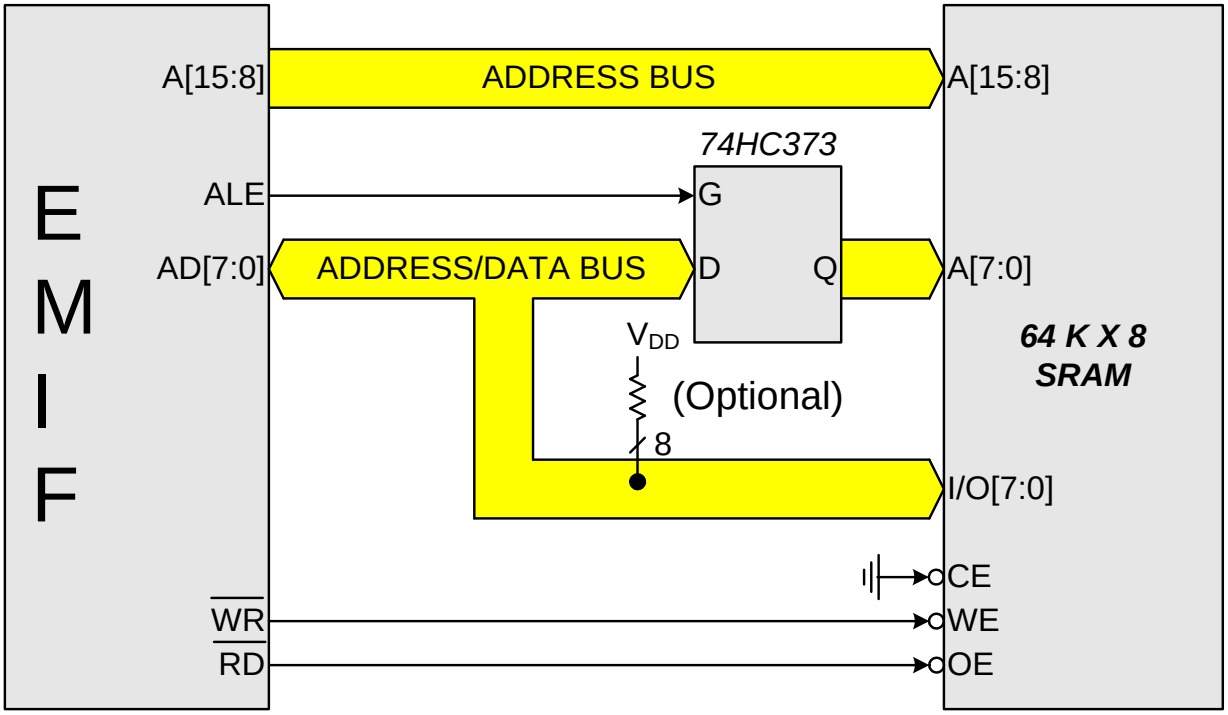


Figure 10.1. Multiplexed Configuration Example

10.4.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 10.2. See Section “10.6.1. Non-Multiplexed Mode” on page 139 for more information about Non-multiplexed operation.

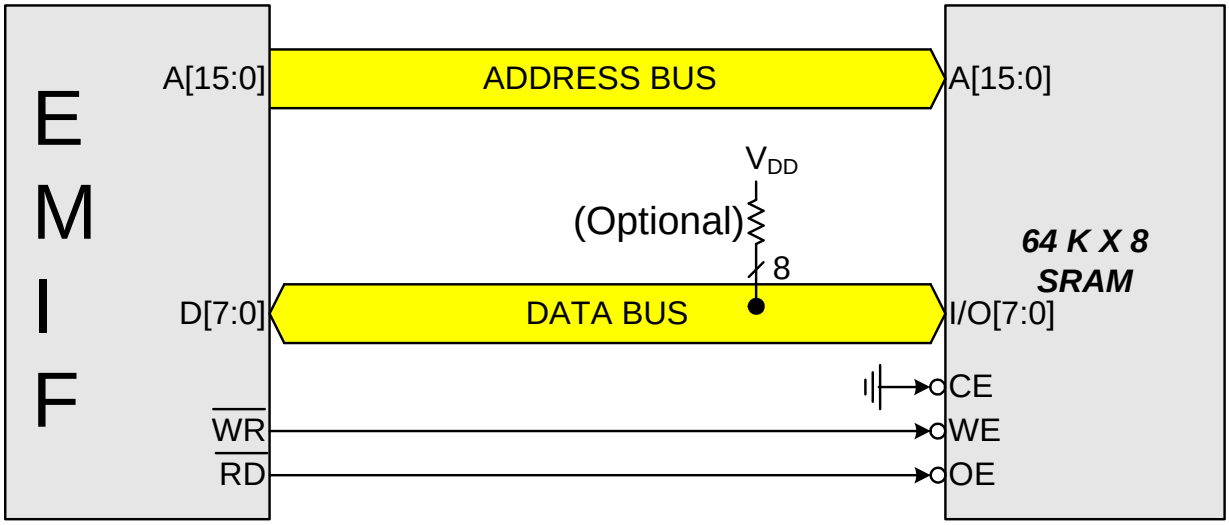


Figure 10.2. Non-multiplexed Configuration Example

10.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 10.3, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 10.2). These modes are summarized below. More information about the different modes can be found in Section “10.6. Timing” on page 137.

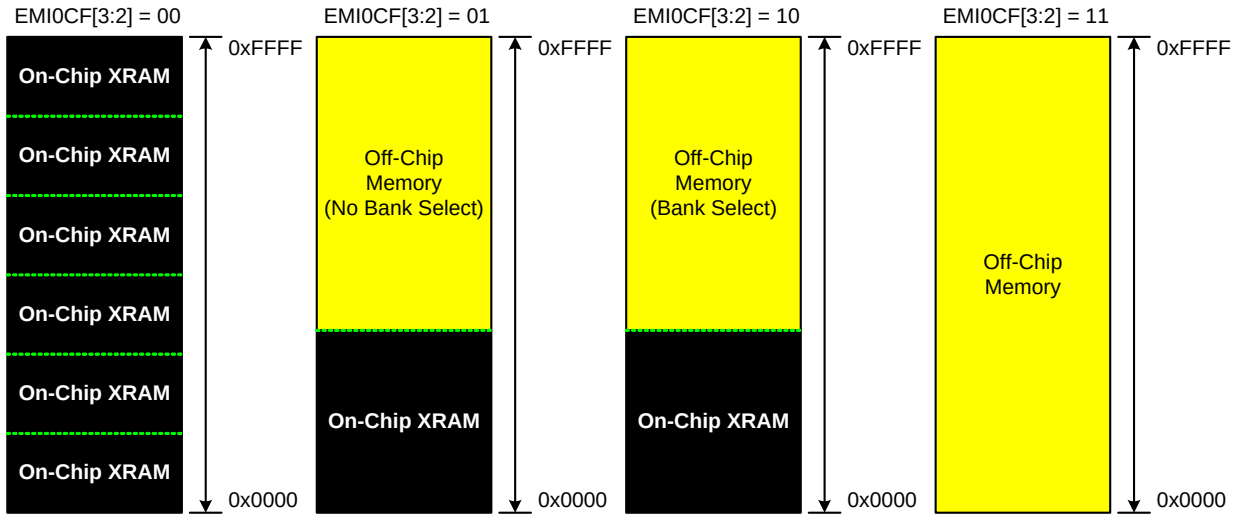


Figure 10.3. EMIF Operating Modes

10.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 8 kB boundaries. As an example, the addresses 0x2000 and 0x4000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

10.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.5.3. Split Mode with Bank Select

When EMI0CF[3:2] are set to 10, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. The upper 8-bits of the Address Bus A[15:8] are determined by EMI0CN, and the lower 8-bits of the Address Bus A[7:0] are determined by R0 or R1. All 16-bits of the Address Bus A[15:0] are driven in “Bank Select” mode.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.5.4. External Only

When EMI0CF[3:2] are set to 11, all MOVX operations are directed to off-chip space. On-chip XRAM is not visible to the CPU. This mode is useful for accessing off-chip memory located between 0x0000 and the internal XRAM size boundary.

- 8-bit MOVX operations ignore the contents of EMI0CN. The upper Address bits A[15:8] are not driven (identical behavior to an off-chip access in “Split Mode without Bank Select” described above). This allows the user to manipulate the upper address bits at will by setting the Port state directly. The lower 8-bits of the effective address A[7:0] are determined by the contents of R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine the effective address A[15:0]. The full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.6. Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, $\overline{RD}$ and $\overline{WR}$ strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in SFR Definition 10.3, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for $\overline{RD}$ or $\overline{WR}$ pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for /ALE + 1 for $\overline{RD}$ or $\overline{WR}$ + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 10.2 lists the ac parameters for the External Memory Interface, and Figure 10.4 through Figure 10.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

SFR Definition 10.3. EMI0TC: External Memory Timing Control

Bit	7	6	5	4	3	2	1	0
Name	EAS[1:0]		EWR[3:0]				EAH[1:0]	
Type	R/W		R/W				R/W	
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xAF

Bit	Name	Function
7:6	EAS[1:0]	**EMIF Address Setup Time Bits.** 00: Address setup time = 0 SYSCLK cycles. 01: Address setup time = 1 SYSCLK cycle. 10: Address setup time = 2 SYSCLK cycles. 11: Address setup time = 3 SYSCLK cycles.
5:2	EWR[3:0]	**EMIF $\overline{WR}$ and $\overline{RD}$ Pulse-Width Control Bits.** 0000: $\overline{WR}$ and $\overline{RD}$ pulse width = 1 SYSCLK cycle. 0001: $\overline{WR}$ and $\overline{RD}$ pulse width = 2 SYSCLK cycles. 0010: $\overline{WR}$ and $\overline{RD}$ pulse width = 3 SYSCLK cycles. 0011: $\overline{WR}$ and $\overline{RD}$ pulse width = 4 SYSCLK cycles. 0100: $\overline{WR}$ and $\overline{RD}$ pulse width = 5 SYSCLK cycles. 0101: $\overline{WR}$ and $\overline{RD}$ pulse width = 6 SYSCLK cycles. 0110: $\overline{WR}$ and $\overline{RD}$ pulse width = 7 SYSCLK cycles. 0111: $\overline{WR}$ and $\overline{RD}$ pulse width = 8 SYSCLK cycles. 1000: $\overline{WR}$ and $\overline{RD}$ pulse width = 9 SYSCLK cycles. 1001: $\overline{WR}$ and $\overline{RD}$ pulse width = 10 SYSCLK cycles. 1010: $\overline{WR}$ and $\overline{RD}$ pulse width = 11 SYSCLK cycles. 1011: $\overline{WR}$ and $\overline{RD}$ pulse width = 12 SYSCLK cycles. 1100: $\overline{WR}$ and $\overline{RD}$ pulse width = 13 SYSCLK cycles. 1101: $\overline{WR}$ and $\overline{RD}$ pulse width = 14 SYSCLK cycles. 1110: $\overline{WR}$ and $\overline{RD}$ pulse width = 15 SYSCLK cycles. 1111: $\overline{WR}$ and $\overline{RD}$ pulse width = 16 SYSCLK cycles.
1:0	EAH[1:0]	**EMIF Address Hold Time Bits.** 00: Address hold time = 0 SYSCLK cycles. 01: Address hold time = 1 SYSCLK cycle. 10: Address hold time = 2 SYSCLK cycles. 11: Address hold time = 3 SYSCLK cycles.

10.6.1. Non-Multiplexed Mode

10.6.1.1. 16-bit MOVX: EMI0CF[4:2] = 101, 110, or 111

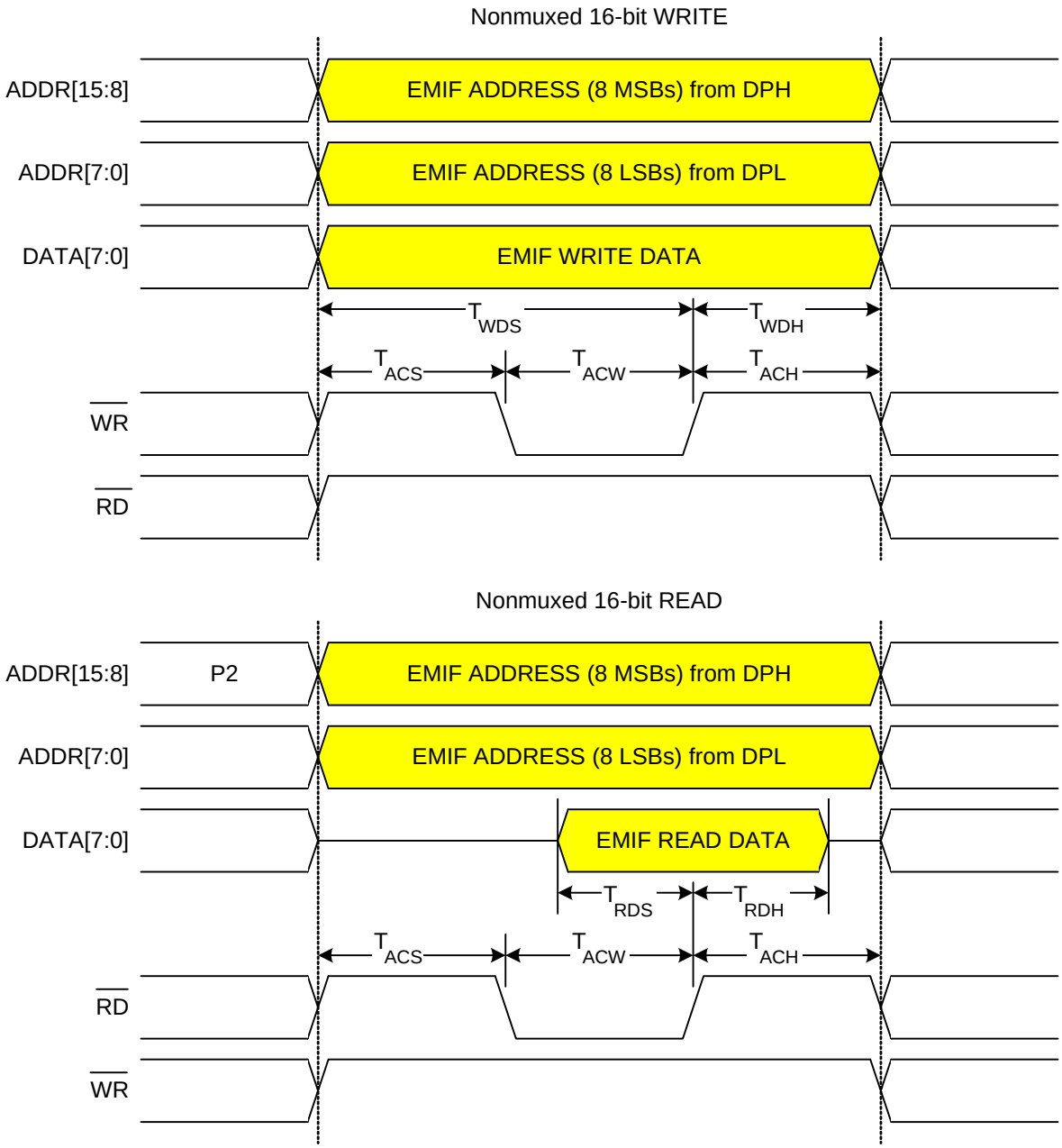


Figure 10.4. Non-multiplexed 16-bit MOVX Timing

10.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 101 or 111

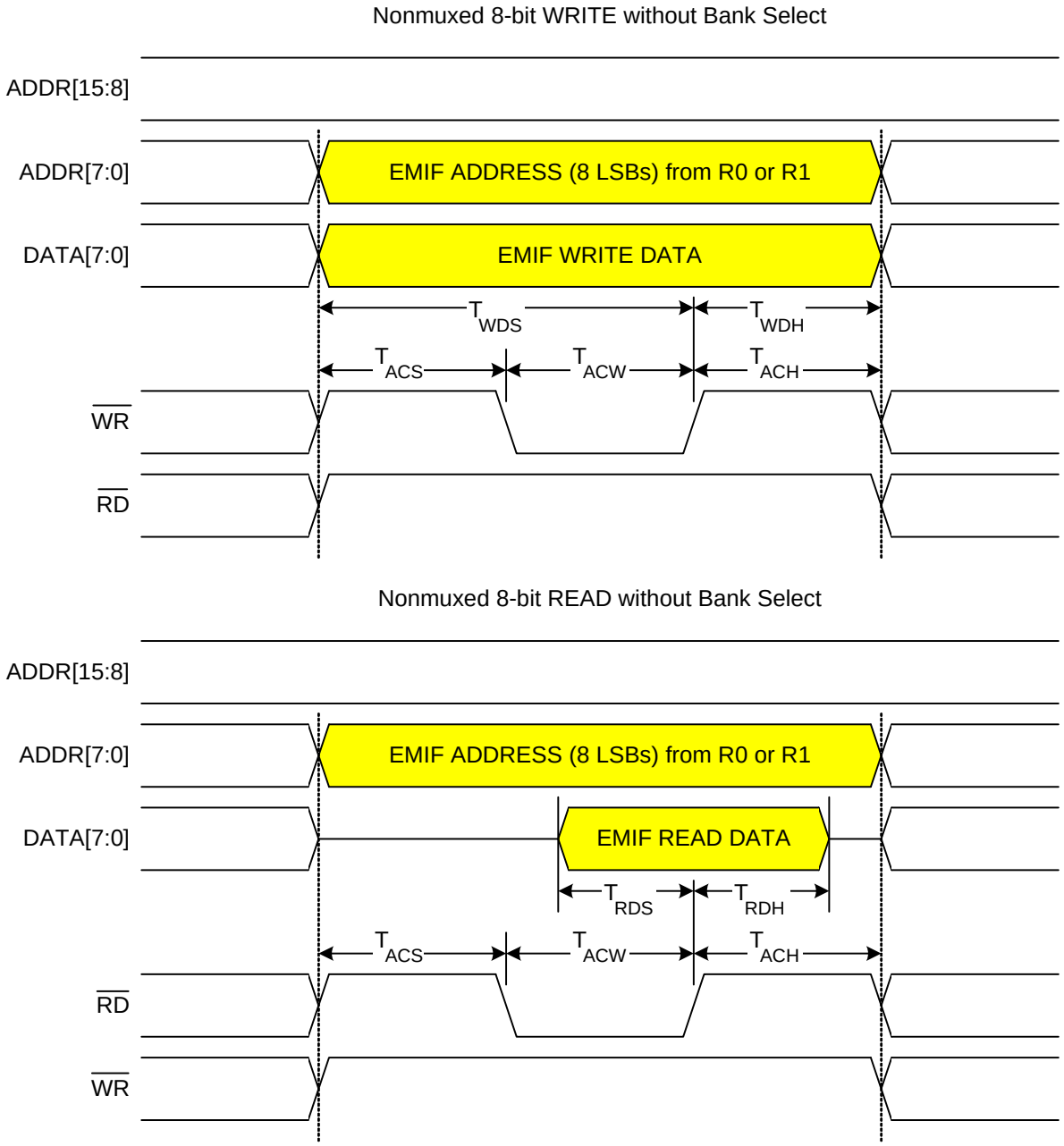


Figure 10.5. Non-multiplexed 8-bit MOVX without Bank Select Timing

10.6.1.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 110

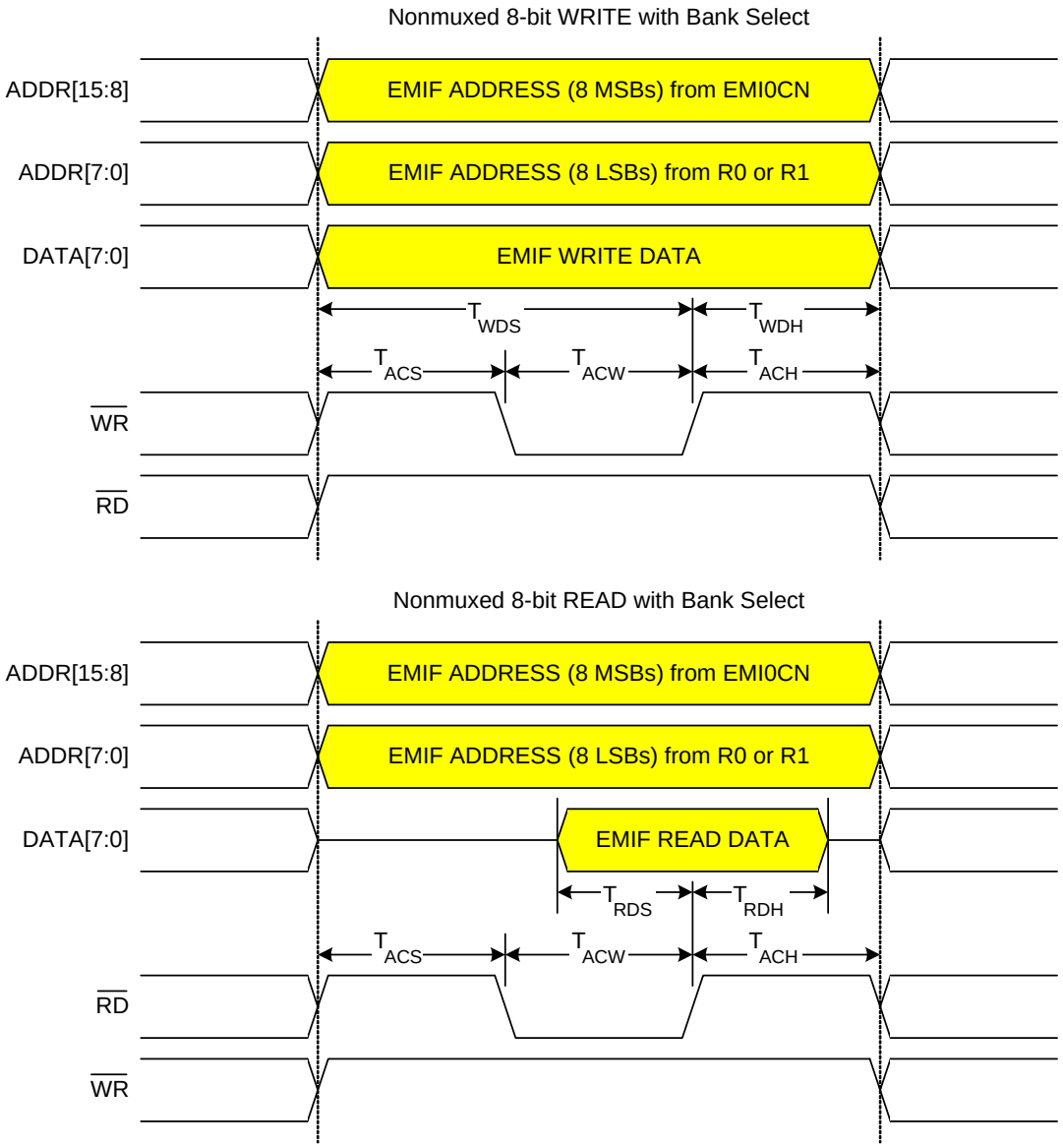


Figure 10.6. Non-multiplexed 8-bit MOVX with Bank Select Timing

10.6.2. Multiplexed Mode

10.6.2.1. 16-bit MOVX: EMI0CF[4:2] = 001, 010, or 011

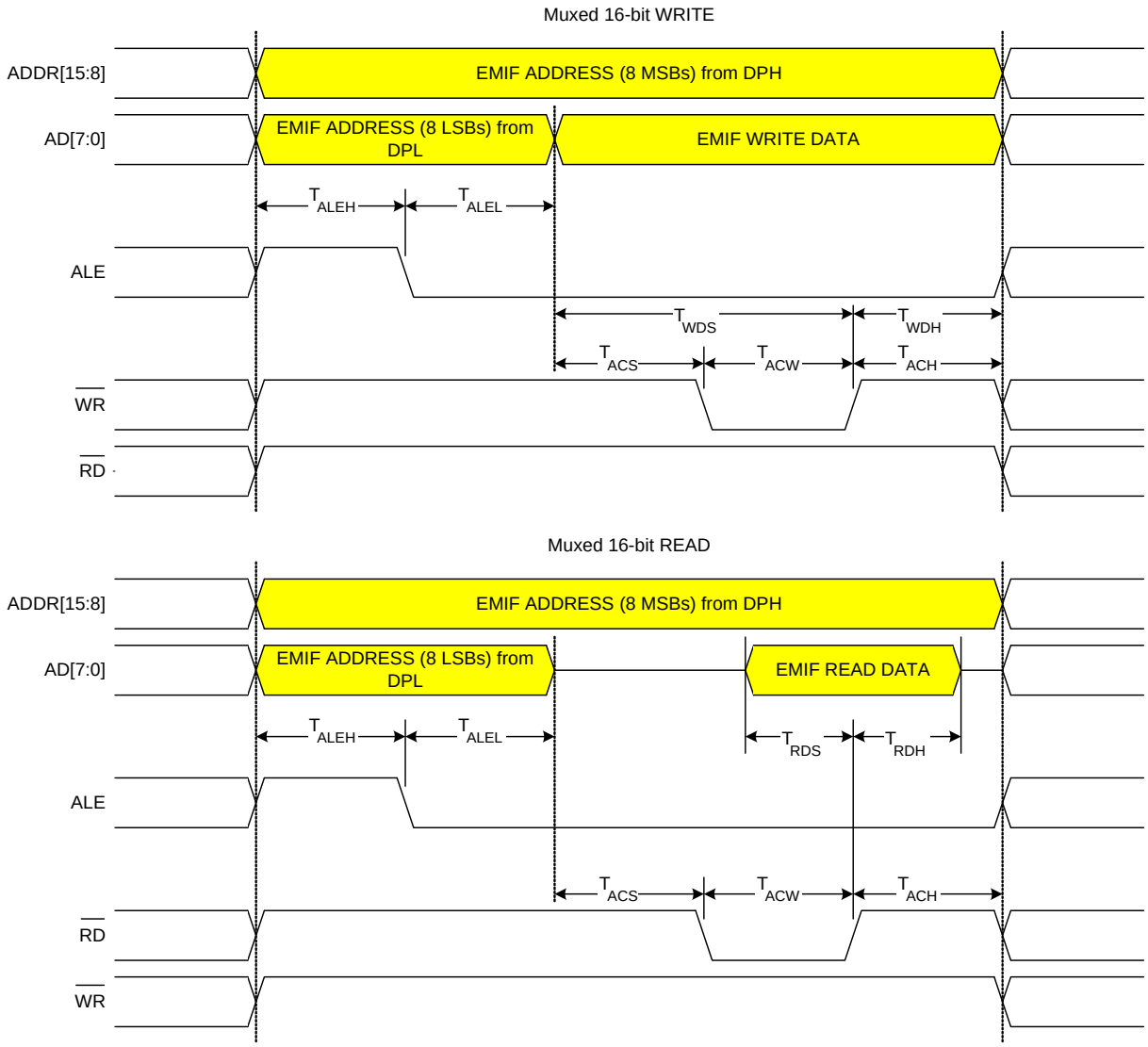


Figure 10.7. Multiplexed 16-bit MOVX Timing

10.6.2.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 001 or 011

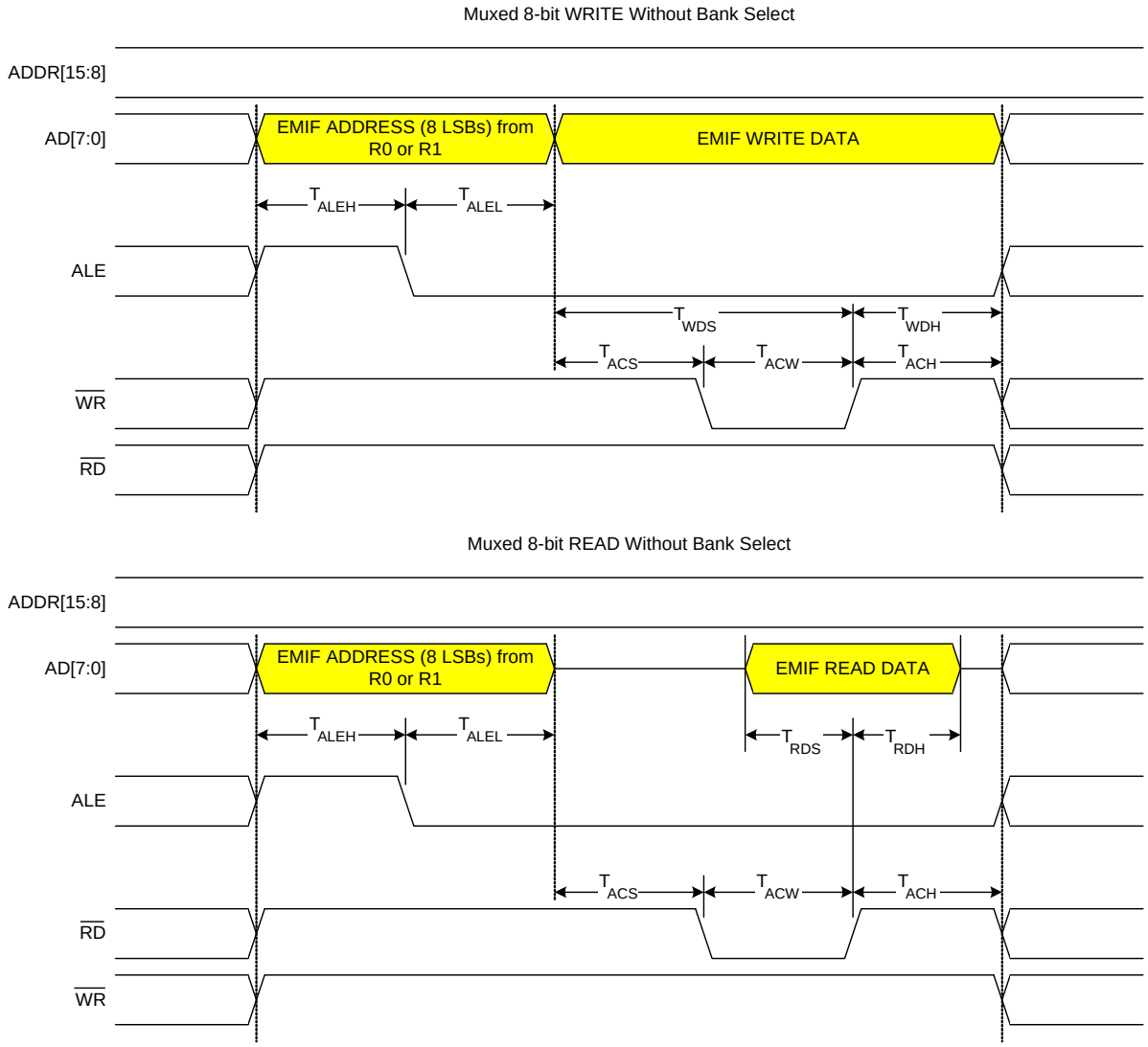


Figure 10.8. Multiplexed 8-bit MOVX without Bank Select Timing

10.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 010

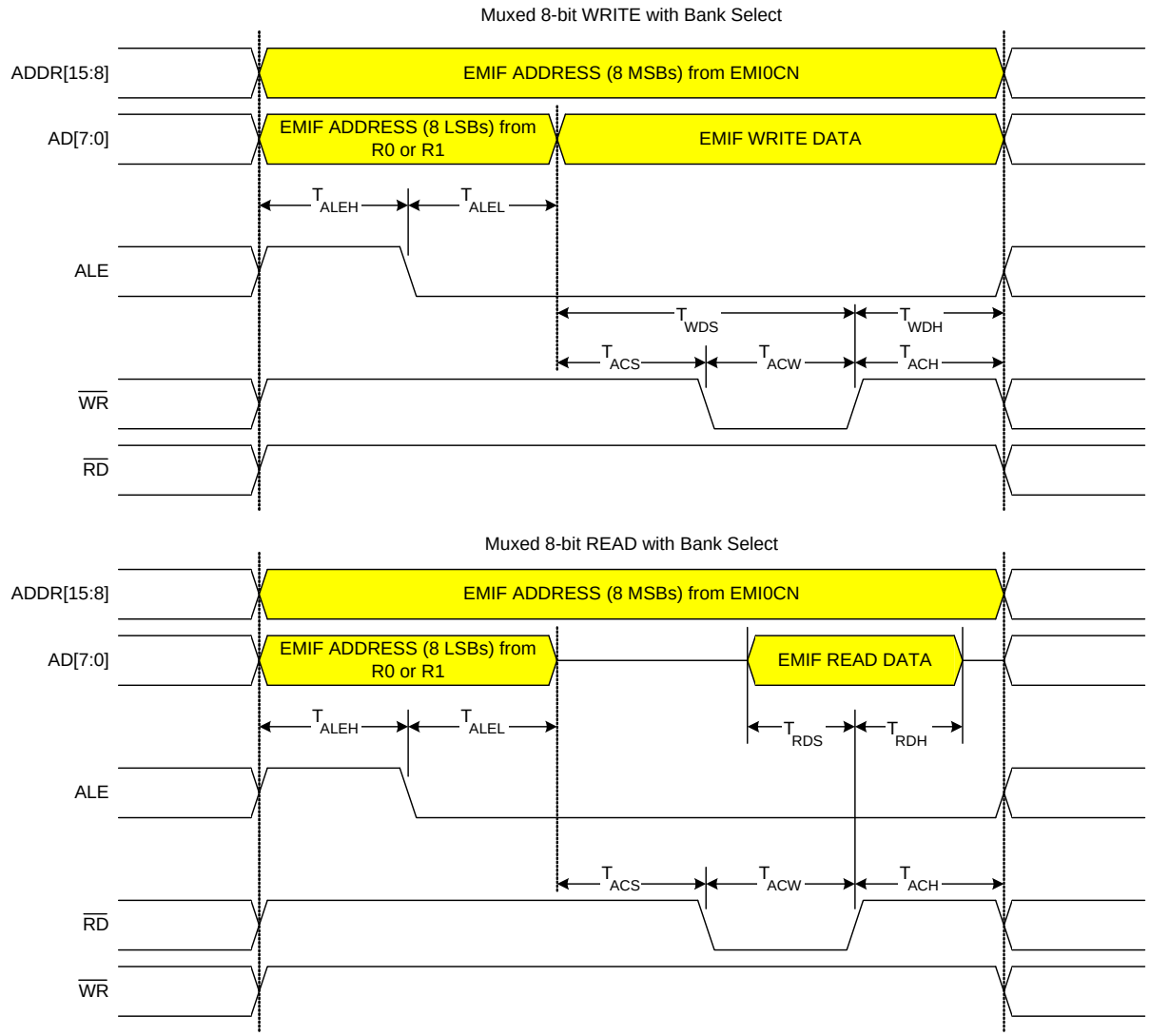


Figure 10.9. Multiplexed 8-bit MOVX with Bank Select Timing

Table 10.2. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20		ns
T_{RDH}	Read Data Hold Time	0		ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

11. Direct Memory Access (DMA0)

An on-chip direct memory access (DMA0) is included on the C8051F96x devices. The DMA0 subsystem allows autonomous variable-length data transfers between XRAM and peripheral SFR registers without CPU intervention. During DMA0 operation, the CPU is free to perform some other tasks. In order to save total system power consumption, the CPU and flash can be powered down. DMA0 improves the system performance and efficiency with high data throughput peripherals.

DMA0 contains seven independent channels, common control registers, and a DMA0 Engine (see Figure 11.1). Each channel includes a register that assigns a peripheral to the channel, a channel control register, and a set of SFRs that include XRAM address information and SFR address information used by the channel during a data transfer. The DMA0 architecture is described in detail in Section 11.1.

The DMA0 in C8051F96x devices supports four peripherals: AES0, ENC0, CRC1, and SPI1. Peripherals with DMA0 capability should be configured to work with the DMA0 through their own registers. The DMA0 provides up to seven channels, and each channel can be configured for one of nine possible data transfer functions:

- XRAM to ENC0L/M/H
- ENC0L/M/H sfrs to XRAM
- XRAM to CRC1IN sfr
- XRAM to SPI1DAT sfr
- SPI1DAT sfr to XRAM
- XRAM to AES0KIN sfr
- XRAM to AES0BIN sfr
- XRAM to AES0XIN sfr
- AES0YOUT sfr to XRAM

The DMA0 subsystem signals the MCU through a set of interrupt service routine flags. Interrupts can be generated when the DMA0 transfers half of the data length or full data length on any channel.

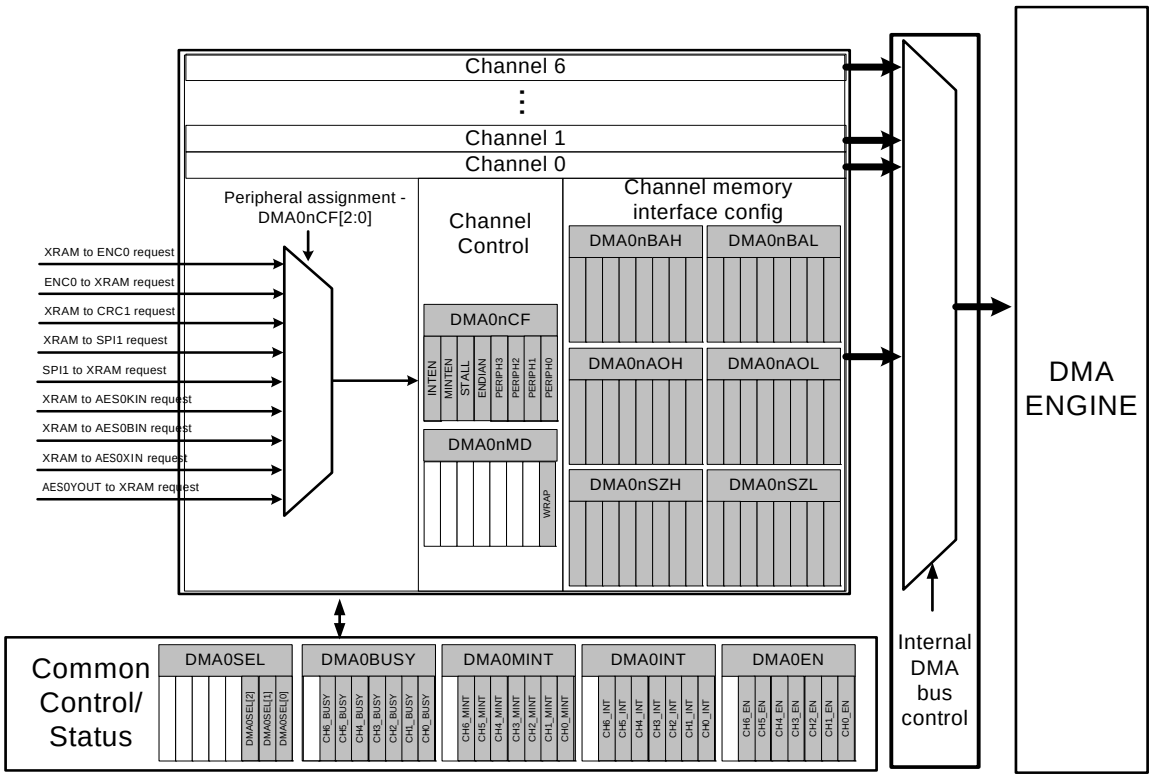


Figure 11.1. DMA0 Block Diagram

11.1. DMA0 Architecture

The first step in configuring a DMA0 channel is to select the desired channel for data transfer using DMA0SEL[2:0] bits (DMA0SEL). After setting the DMA0 channel, firmware can address channel-specific registers such as DMA0NCF, DMA0NBAH/L, DMA0NAOH/L, and DMA0NSZH/L. Once firmware selects a channel, the subsequent SFR configuration applies to the DMA0 transfer of that selected channel.

Each DMA0 channel consists of an SFR assigning the channel to a peripheral, a channel control register and a set of SFRs that describe XRAM and SFR addresses to be used during data transfer (See Figure 11.1). The peripheral assignment bits of DMA0nCF select one of the eight data transfer functions. The selected channel can choose the desired function by writing to the PERIPH[2:0] bits (DMA0NCF[2:0]).

The control register DMA0NCF of each channel configures the endian-ness of the data in XRAM, stall enable, full-length interrupt enable and mid-point interrupt enable. When a channel is stalled by setting the STALL bit (DMA0NCF.5), DMA0 transfers in progress will not be aborted, but new DMA0 transfers will be blocked until the stall status of the channel is reset. After the stall bit is set, software should poll the corresponding DMA0BUSY to verify that there are no more DMA transfers for that channel.

The memory interface configuration SFRs of a channel define the linear region of XRAM involved in the transfer through a 12-bit base address register DMA0NBAH:L, a 10-bit address offset register DMA0NAOH:L and a 10-bit data transfer size DMA0NSZH:L. The effective memory address is the address involved in the current DMA0 transaction.

Effective Memory Address = Base Address + Address Offset

The address offset serves as byte counter. The address offset should be always less than data transfer length. The address offset increments by one after each byte transferred. For DMA0 configuration of any channel, address offsets of active channels should be reset to 0 before DMA0 transfers occur.

Data transfer size DMA0NSZH:L defines the maximum number of bytes for the DMA0 transfer of the selected channel. If the address offset reaches data transfer size, the full-length interrupt flag bit CHn_INT (DMA0INT) of the selected channel will be asserted. Similarly, the mid-point interrupt flag bit CHn_MINT is set when the address offset is equal to half of data transfer size if the transfer size is an even number or when the address offset is equal to half of the transfer size plus one if the transfer size is an odd number. Interrupt flags must be cleared by software so that the next DMA0 data transfer can proceed.

The DMA0 subsystem permits data transfer between SFR registers and XRAM. The DMA0 subsystem executes its task based on settings of a channel's control and memory interface configuration SFRs. When data is copied from XRAM to SFR registers, it takes two cycles for DMA0 to read from XRAM and the SFR write occurs in the second cycle. If more than one byte is involved, a pipeline is used. When data is copied from SFR registers to XRAM, the DMA0 only requires one cycle for one byte transaction.

The selected DMA0 channel for a peripheral should be enabled through the enable bits CHn_EN (DMA0EN.n) to allow the DMA0 to transfer the data. When the DMA0 is transferring data on a channel, the busy status bit of the channel CHn_BUSY (DMA0BUSY.n) is set. During the transaction, writes to DMA0NSZH:L, DMA0NBAH:L, and DMA0NAOH:L are disabled.

Each peripheral is responsible for asserting the peripheral transfer requests necessary to service the particular peripheral. Some peripherals may have a complex state machine to manage the peripheral requests. Please refer to the DMA enabled peripheral chapters for additional information (AES0, CRC1, ENC0 and SPI1).

Besides reporting transaction status of a channel, DMA0BUSY can be used to force a DMA0 transfer on an already configured channel by setting the CHn_BUSY bit (DMA0BUSY.n).

The DMA0NMD sfr has a wrap bit that supports address offset wrapping. The size register DMA0NSZ sets the transfer size. Normally the address offset starts at zero and increases until it reaches size minus one. At this point the transfer is complete and the interrupt bit will be set. When the wrap bit is set, the address offset will automatically be reset to zero and transfers will continue as long as the peripheral keeps requesting data.

The wrap feature can be used to support key wrapping for the AES0 module. Normally the same key is used over and over with additional data blocks. So the wrap bit should be set when using the XRAM to AES0KIN request. This feature supports multiple-block encryption operations.

11.2. DMA0 Arbitration

11.2.1. DMA0 Memory Access Arbitration

If both DMA0 and CPU attempt to access SFR register or XRAM at the same time, the CPU pre-empts the DMA0 module. DMA0 will be stalled until CPU completes its bus activity.

11.2.2. DMA0 Channel Arbitration

Multiple DMA0 channels can request transfer simultaneously, but only one DMA0 channel will be granted the bus to transfer the data. Channel 0 has the highest priority. DMA0 channels are serviced based on their priority. A higher priority channel is serviced first. Channel arbitration occurs at the end of the data transfer granularity (transaction boundary) of the DMA. When there is a DMA0 request at the transaction boundary from higher priority channel, lower priority ones will be stalled until the highest priority one completes its transaction. So, for 16-bit transfers, the transaction boundary is at every 2 bytes.

11.3. DMA0 Operation in Low Power Modes

DMA0 remains functional in normal active, low power active, idle, low power idle modes but not in sleep or suspend mode. CPU will wait for DMA0 to complete all pending requests before it enters sleep mode. When the system wakes up from suspend or sleep mode to normal active mode, pending DMA0 interrupts will be serviced according to priority of channels. DMA0 stalls when CPU is in debug mode.

11.4. Transfer Configuration

The following steps are required to configure one of the DMA0 channels for operation:

1. Select the channel to be configured by writing DMA0SEL.
2. Specify the data transfer function by writing DMA0NCF. This register also specifies the endian-ness of the data in XRAM and enables full or mid-point interrupts.
3. Configure the wrapping mode by writing to DMA0NMD. Setting this bit will automatically reset the address offset after each completed transfer.
4. Specify the base address in XRAM for the transfer by writing DMA0NBAH:L.
5. Specify the size of the transfer in bytes by writing DMA0NSZH:L.
6. Reset the address offset counter by writing 0 to DMA0NAOH:L.
7. Enable the DMA0 channel by writing 1 to the appropriate bit in DMA0EN.

SFR Definition 11.1. DMA0EN: DMA0 Channel Enable

Bit	7	6	5	4	3	2	1	0
Name		CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD2

Bit	Name	Function
7	Unused	Read = 0b, Write = Don't Care
6	CH6_EN	**Channel 6 Enable.** 0: Disable DMA0 channel 6. 1: Enable DMA0 channel 6.
5	CH5_EN	**Channel 5 Enable.** 0: Disable DMA0 channel 5. 1: Enable DMA0 channel 5.
4	CH4_EN	**Channel 4 Enable.** 0: Disable DMA0 channel 4. 1: Enable DMA0 channel 4.
3	CH3_EN	**Channel 3 Enable.** 0: Disable DMA0 channel 3. 1: Enable DMA0 channel 3.
2	CH2_EN	**Channel 2 Enable.** 0: Disable DMA0 channel 2. 1: Enable DMA0 channel 2.
1	CH1_EN	**Channel 1 Enable.** 0: Disable DMA0 channel 1. 1: Enable DMA0 channel 1.
0	CH0_EN	**Channel 0 Enable.** 0: Disable DMA0 channel 0. 1: Enable DMA0 channel 0.

SFR Definition 11.2. DMA0INT: DMA0 Full-Length Interrupt

Bit	7	6	5	4	3	2	1	0
Name		CH6_INT	CH5_INT	CH4_INT	CH3_INT	CH2_INT	CH1_INT	CH0_INT
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD3

Bit	Name	Function
7	Unused	Read = 0b, Write = Don't Care
6	CH6_INT	**Channel 6 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 6. 1: Full-length interrupt has not occured on channel 6.
5	CH5_INT	0: Full-length interrupt has not occured on channel 5. 1: Full-length interrupt has not occured on channel 5.
4	CH4_INT	**Channel 4 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 4. 1: Full-length interrupt has not occured on channel 4.
3	CH3_INT	**Channel 3 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 3. 1: Full-length interrupt has not occured on channel 3.
2	CH2_INT	**Channel 2 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 2. 1: Full-length interrupt has not occured on channel 2.
1	CH1_INT	**Channel 1 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 1. 1: Full-length interrupt has not occured on channel 1.
0	CH0_INT	**Channel 0 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 0. 1: Full-length interrupt has not occured on channel 0.

Note: 1.Full-length interrupt flag is set when the offset address DMA0NAOH/L is equals to data transfer size DMA0NSZH/L minus 1. This flag must be cleared by software or system reset. The full-length interrupt is enabled by setting bit 7 of DMA0NCF with DMA0SEL configured for the corresponding channel.

SFR Definition 11.3. DMA0MINT: DMA0 Mid-Point Interrupt

Bit	7	6	5	4	3	2	1	0
Name		CH6_MINT	CH5_MINT	CH4_MINT	CH3_MINT	CH2_MINT	CH1_MINT	CH0_MINT
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD4

Bit	Name	Function
7	Unused	Read = 0b, Write = Don't Care
6	CH6_MINT	**Channel 6 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 6. 1: Mid-Point interrupt has not occured on channel 6.
5	CH5_MINT	**Channel 5 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 5. 1: Mid-Point interrupt has not occured on channel 5.
4	CH4_MINT	**Channel 4 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 4. 1: Mid-Point interrupt has not occured on channel 4**.**
3	CH3_MINT	**Channel 3 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 3. 1: Mid-Point interrupt has not occured on channel 3.
2	CH2_MINT	**Channel 2 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 2. 1: Mid-Point interrupt has not occured on channel 2.
1	CH1_MINT	**Channel 1 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 1. 1: Mid-Point interrupt has not occured on channel 1.
0	CH0_MINT	**Channel 0 Mid-Point Interrupt Flag.** 0: Mid-Point interrupt has not occured on channel 0. 1: Mid-Point interrupt has not occured on channel 0.

Note: Mid-point Interrupt flag is set when the offset address DMA0NAOH/L equals to half of data transfer size DMA0NSZH/L if the transfer size is an even number or half of data transfer size DMA0NSZH/L plus one if the transfer size is an odd number. This flag must be cleared by software or system reset.The mid-point interrupt is enabled by setting bit 6 of DMA0NCF with DMA0SEL configured for the corresponding channel.

SFR Definition 11.4. DMA0BUSY: DMA0 Busy

Bit	7	6	5	4	3	2	1	0
Name		CH6_BUSY	CH5_BUSY	CH4_BUSY	CH3_BUSY	CH2_BUSY	CH1_BUSY	CH0_BUSY
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD5

Bit	Name	Description	Write	Read
7	Unused		No effect.	Always Reads 0.
6	CH6_BUSY	**Channel 6 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 6.	0: DMA0 channel 6 Idle. 1: DMA0 transfer in progress on channel 6.
5	CH5_BUSY	**Channel 5 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 5.	0: DMA0 channel 5 Idle. 1: DMA0 transfer in progress on channel 5.
4	CH4_BUSY	**Channel 4 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 4.	0: DMA0 channel 4 Idle. 1: DMA0 transfer in progress on channel 4.
3	CH3_BUSY	**Channel 3 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 3.	0: DMA0 channel 3 Idle. 1: DMA0 transfer in progress on channel 3.
2	CH2_BUSY	**Channel 2 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 2.	0: DMA0 channel 2 Idle. 1: DMA0 transfer in progress on channel 2.
1	CH1_BUSY	**Channel 1 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 1.	0: DMA0 channel 1 Idle. 1: DMA0 transfer in progress on channel 1.
0	CH0_BUSY	**Channel 0 Busy.**	0: No effect. 1: Force DMA0 transfer to start on channel 0.	0: DMA0 channel 0 Idle. 1: DMA0 transfer in progress on channel 0.

SFR Definition 11.5. DMA0SEL: DMA0 Channel Select for Configuration

Bit	7	6	5	4	3	2	1	0
Name						DMA0SEL[2:0]		
Type	R	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD1

Bit	Name	Function
7:3	Unused	Read = 0b, Write = Don't Care
2:0	DMA0SEL[2:0]	**Channel Select for Configuration.** These bits select the channel for configuration of the DMA0 transfer. The first step to configure a channel for DMA0 transfer is to select the desired channel, and then write to channel specific registers DMA0NCF, DMA0NBAL/H, DMA0NAOL/H, DMA0NSZL/H. 000: Select channel 0 001: Select channel 1 010: Select channel 2 011: Select channel 3 100: Select channel 4 101: Select channel 5 110: Select channel 6 111: Invalid

SFR Definition 11.6. DMA0NMD: DMA Channel Mode

Bit	7	6	5	4	3	2	1	0
Name								WRAP
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD6

Bit	Name	Function
7:1	reserved	Read = 0, Write = 0
0	WRAP	**Wrap Enable.** Setting this bit will enable wrapping. The DMA0NSZ register sets the transfer size. Normally the DMA0AO value starts at zero in increases to the DMANSZ minus one. At this point the transfer is complete and the interrupt bit will be set. If the WRAP bit is set, the DMA0NAO will be reset to zero.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.7. DMA0NCF: DMA Channel Configuration

Bit	7	6	5	4	3	2	1	0
Name	INTEN	MINTEN	STALL	ENDIAN	PERIPH[3:0]			
Type	R/W	R/W	R/W	R/W	R	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xC9

Bit	Name	Function
7	INTEN	**Full-Length Interrupt Enable.** 0: Disable the full-length interrupt of the selected channel. 1: Enable the full-length interrupt of the selected channel.
6	MINTEN	**Mid-Point Interrupt Enable.** 0: Disable the mid-point interrupt of the selected channel. 1: Enable the mid-point interrupt of the selected channel.
5	STALL	**DMA0 Stall.** Setting this bit stalls the DMA0 transfer on the selected channel. After a Stall, this bit must be cleared by software to resume normal operation. 0: The DMA0 transfer of the selected channel is not being stalled. 1: The DMA0 transfer of the selected channel is stalled.
4	ENDIAN	**Data Transfer Endianness.** This bit sets the byte order for multi-byte transfers. This is only relevant for two or three byte transfers. The value of this bit does not matter for single byte transfers. 0: Little Endian 1: Big Endian
3:0	PERIPH[2:0]	**Peripheral Selection of The Selected Channel.** These bits choose one of the nine DMA0 transfer functions for the selected channel. 0000: XRAM to ENC0L/M/H 0001: ENC0L/M/H sfrs to XRAM 0010: XRAM to CRC1IN sfr 0011: XRAM to SPI1DAT sfr 0100: SPI1DAT sfr to XRAM 0101: XRAM to AES0KIN sfr 0110: XRAM to AES0BIN sfr 0111: XRAM to AES0XIN sfr 1000: AES0YOUT sfr to XRAM
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.8. DMA0NBAH: Memory Base Address High Byte

Bit	7	6	5	4	3	2	1	0
Name					NBAH[3:0]			
Type	R	R	R	R	R/W			
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xCB

Bit	Name	Function
7:4	Unused	Read = 0b, Write = Don't Care
3:0	NBAH[3:0]	**Memory Base Address High Byte.** Sets high byte of the memory base address which is the DMA0 XRAM starting address of the selected channel if the channel's address offset DMA0NAO is reset to 0.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.9. DMA0NBAL: Memory Base Address Low Byte

Bit	7	6	5	4	3	2	1	0
Name	NBAH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xCA

Bit	Name	Function
7:0	NBAL[7:0]	**Memory Base Address Low Byte.** Sets low byte of the memory base address which is the DMA0 XRAM starting address of the selected channel if the channel's address offset DMA0NAO is reset to 0.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.10. DMA0NAOH: Memory Address Offset High Byte

Bit	7	6	5	4	3	2	1	0
Name							NAOH[1:0]	
Type	R	R	R	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xCD

Bit	Name	Function
7:2	Unused	Read = 0b, Write = Don't Care
1:0	NAOH[1:0]	**Memory Address Offset High Byte.** Sets the high byte of the address offset of the selected channel which acts a counter during DMA0 transfer. The address offset auto-increments by one after one byte is transferred. When configuring a channel for DMA0 transfer, the address offset should be reset to 0.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.11. DMA0NAOL: Memory Address Offset Low Byte

Bit	7	6	5	4	3	2	1	0
Name	NACL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xCC

Bit	Name	Function
7:0	NACL[7:0]	**Memory Address Offset Low Byte.** Sets the low byte of the address offset of the selected channel which acts a counter during DMA0 transfer. The address offset auto-increments by one after one byte is transferred. When configuring a channel for DMA0 transfer, the address offset should be reset to 0.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.12. DMA0NSZH: Transfer Size High Byte

Bit	7	6	5	4	3	2	1	0
Name							NSZH[1:0]	
Type	R	R	R	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xCF

Bit	Name	Function
7:2	Unused	Read = 0b, Write = Don't Care
1:0	NSZH[1:0]	**Transfer Size High Byte.** Sets high byte of DMA0 transfer size of the selected channel. Transfer size sets the maximum number of bytes for the DMA0 transfer. When the address offset is equal to the transfer size, a full-length interrupt is generated on the channel.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.13. DMA0NSZL: Memory Transfer Size Low Byte

Bit	7	6	5	4	3	2	1	0
Name	NSZL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xCE

Bit	Name	Function
7:0	NSZL[7:0]	**Memory Transfer Size Low Byte.** Sets low byte of DMA0 transfer size of the selected channel. Transfer size sets the maximum number of bytes for the DMA0 transfer. When the address offset is equal to the transfer size, a full-length interrupt is generated on the channel.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

12. Cyclic Redundancy Check Unit (CRC0)

C8051F96x devices include a cyclic redundancy check unit (CRC0) that can perform a CRC using a 16-bit or 32-bit polynomial. CRC0 accepts a stream of 8-bit data written to the CRC0IN register. CRC0 posts the 16-bit or 32-bit result to an internal register. The internal result register may be accessed indirectly using the CRC0PNT bits and CRC0DAT register, as shown in Figure 12.1. CRC0 also has a bit reverse register for quick data manipulation.

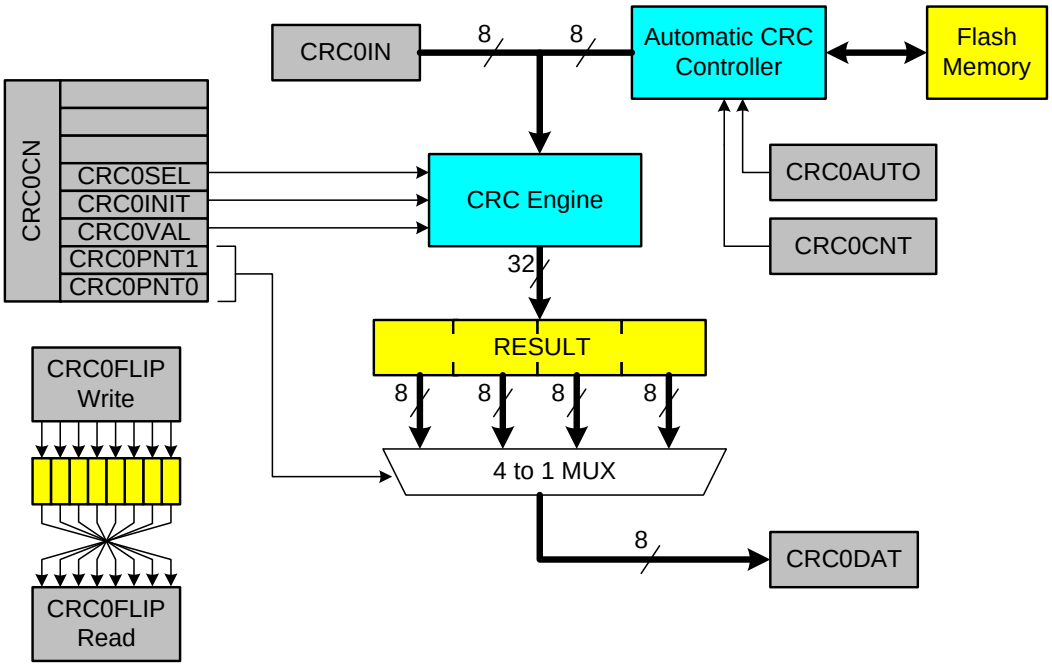


Figure 12.1. CRC0 Block Diagram

12.1. 16-bit CRC Algorithm

The C8051F96x CRC unit calculates the 16-bit CRC MSB-first, using a poly of 0x1021. The following describes the 16-bit CRC algorithm performed by the hardware:

1. XOR the most-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x0000 or 0xFFFF).
   a. If the MSB of the CRC result is set, left-shift the CRC result, and then XOR the CRC result with the polynomial (0x1021).
   b. If the MSB of the CRC result is not set, left-shift the CRC result.
2. Repeat at Step 2a for the number of input bits (8).

The 16-bit C8051F96x CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
   unsigned char i;                    // loop counter

   #define POLY 0x1021

   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x8000) == 0x8000)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc << 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc << 1;
      }
   }

   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

The following table lists several input values and the associated outputs using the 16-bit C8051F96x CRC algorithm:

Table 12.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

12.2. 32-bit CRC Algorithm

The C8051F41x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

Step 1. XOR the least-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x00000000 or 0xFFFFFFFF).
Step 2. Right-shift the CRC result.
Step 3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
Step 4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit 'F41x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input)
{
   unsigned char i; // loop counter
   #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)

   CRC_acc = CRC_acc ^ CRC_input;

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x00000001) == 0x00000001)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc >> 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc >> 1;
      }
   }
   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

The following table lists several input values and the associated outputs using the 32-bit 'F41x CRC algorithm (an initial value of 0xFFFFFFFF is used):

Table 12.2. Example 32-bit CRC Outputs

Input	Output
0x63	0xF9462090
0xAA, 0xBB, 0xCC	0x41B207B3
0x00, 0x00, 0xAA, 0xBB, 0xCC	0x78D129BC

12.3. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should select the desired polynomial and set the initial value of the result. Two polynomials are available: 0x1021 (16-bit) and 0x04C11DB7 (32-bit). The CRC0 result may be initialized to one of two values: 0x00000000 or 0xFFFFFFFF. The following steps can be used to initialize CRC0.

1. Select a polynomial (Set CRC0SEL to 0 for 32-bit or 1 for 16-bit).
2. Select the initial result value (Set CRC0VAL to 0 for 0x00000000 or 1 for 0xFFFFFFFF).
3. Set the result to its initial value (Write 1 to CRC0INIT).

12.4. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more flash sectors. The following steps can be used to automatically perform a CRC on flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. If necessary, set the IFBANK bits in the PSBANK for the desired code bank.
3. Write the index of the starting page to CRC0AUTO.
4. Set the AUTOEN bit in CRC0AUTO.
5. Write the number of flash sectors to perform in the CRC calculation to CRC0CNT.
   Note: Each flash sector is 1024 bytes.
6. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes.
7. Clear the AUTOEN bit in CRC0AUTO.
8. Read the CRC result using the procedure below.

Setting the IFBANK bits in the PSBANK SFR is only necessary when accessing the upper banks on 128 kB code bank devices (‘F960/1/2/3). Multiple CRCs are required to cover the entire 128 kB Flash array. When writing to the PSBANK SFR, the code initiating the auto CRC of flash must be executing from the common area.

12.5. Accessing the CRC0 Result

The internal CRC0 result is 32-bits (CRC0SEL = 0b) or 16-bits (CRC0SEL = 1b). The CRC0PNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

SFR Definition 12.1. CRC0CN: CRC0 Control

Bit	7	6	5	4	3	2	1	0
Name				CRC0SEL	CRC0INIT	CRC0VAL	CRC0PNT[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x92

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care.
4	CRC0SEL	**CRC0 Polynomial Select Bit.** This bit selects the CRC0 polynomial and result length (32-bit or 16-bit). 0: CRC0 uses the 32-bit polynomial 0x04C11DB7 for calculating the CRC result. 1: CRC0 uses the 16-bit polynomial 0x1021 for calculating the CRC result.
3	CRC0INIT	**CRC0 Result Initialization Bit.** Writing a 1 to this bit initializes the entire CRC result based on CRC0VAL.
2	CRC0VAL	**CRC0 Set Value Initialization Bit.** This bit selects the set value of the CRC result. 0: CRC result is set to 0x00000000 on write of 1 to CRC0INIT. 1: CRC result is set to 0xFFFFFFFF on write of 1 to CRC0INIT.
1:0	CRC0PNT[1:0]	**CRC0 Result Pointer.** Specifies the byte of the CRC result to be read/written on the next access to CRC0DAT. The value of these bits will auto-increment upon each read or write. For CRC0SEL = 0: 00: CRC0DAT accesses bits 7–0 of the 32-bit CRC result. 01: CRC0DAT accesses bits 15–8 of the 32-bit CRC result. 10: CRC0DAT accesses bits 23–16 of the 32-bit CRC result. 11: CRC0DAT accesses bits 31–24 of the 32-bit CRC result. For CRC0SEL = 1: 00: CRC0DAT accesses bits 7–0 of the 16-bit CRC result. 01: CRC0DAT accesses bits 15–8 of the 16-bit CRC result. 10: CRC0DAT accesses bits 7–0 of the 16-bit CRC result. 11: CRC0DAT accesses bits 15–8 of the 16-bit CRC result.

SFR Definition 12.2. CRC0IN: CRC0 Data Input

Bit	7	6	5	4	3	2	1	0
Name	CRC0IN[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x93

Bit	Name	Function
7:0	CRC0IN[7:0]	**CRC0 Data Input.** Each write to CRC0IN results in the written data being computed into the existing CRC result according to the CRC algorithm described in Section 12.1

SFR Definition 12.3. CRC0DAT: CRC0 Data Output

Bit	7	6	5	4	3	2	1	0
Name	CRC0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x91

Bit	Name	Function
7:0	CRC0DAT[7:0]	**CRC0 Data Output.** Each read or write performed on CRC0DAT targets the CRC result bits pointed to by the CRC0 Result Pointer (CRC0PNT bits in CRC0CN).

SFR Definition 12.4. CRC0AUTO: CRC0 Automatic Control

Bit	7	6	5	4	3	2	1	0
Name	AUTOEN	CRCDONE	CRC0ST[5:0]					
Type	R/W							R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x96

Bit	Name	Function
7	AUTOEN	**Automatic CRC Calculation Enable.** When AUTOEN is set to 1, any write to CRC0CN will initiate an automatic CRC starting at flash sector CRC0ST and continuing for CRC0CNT sectors.
6	CRCDONE	**CRCDONE Automatic CRC Calculation Complete.** Set to 0 when a CRC calculation is in progress. Note that code execution is stopped during a CRC calculation, therefore reads from firmware will always return 1.
5:0	CRC0ST[5:0]	**Automatic CRC Calculation Starting Flash Sector.** These bits specify the flash sector to start the automatic CRC calculation. The starting address of the first flash sector included in the automatic CRC calculation is CRC0ST x 1024. For 128 kB devices, pages 32–63 access the upper code bank as selected by the IFBANK bits in the PSBANK SFR.

SFR Definition 12.5. CRC0CNT: CRC0 Automatic Flash Sector Count

Bit	7	6	5	4	3	2	1	0
Name			CRC0CNT[5:0]					
Type	R/W							R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x97

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don’t Care.
5:0	CRC0CNT[5:0]	**Automatic CRC Calculation Flash Sector Count.** These bits specify the number of flash sectors to include in an automatic CRC calculation. The starting address of the last flash sector included in the automatic CRC calculation is (CRC0ST+CRC0CNT) x 1024. The last page should not exceed page 63. Setting both CRC0ST and CRC0CNT to 0 will perform a CRC over the 64kB banked memory space.

12.6. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 12.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

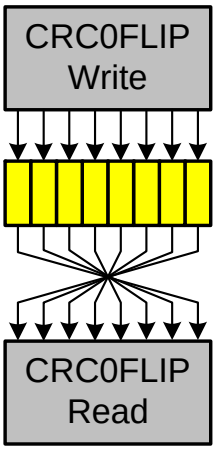


Figure 12.2. Bit Reverse Register

SFR Definition 12.6. CRC0FLIP: CRC0 Bit Flip

Bit	7	6	5	4	3	2	1	0
Name	CRC0FLIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0x94

Bit	Name	Function
7:0	CRC0FLIP[7:0]	**CRC0 Bit Flip.** Any byte written to CRC0FLIP is read back in a bit-reversed order, i.e. the written LSB becomes the MSB. For example: If 0xC0 is written to CRC0FLIP, the data read back will be 0x03. If 0x05 is written to CRC0FLIP, the data read back will be 0xA0.

13. DMA-Enabled Cyclic Redundancy Check Module (CRC1)

C8051F96x devices include a DMA-enabled cyclic redundancy check module (CRC1) that can perform a CRC of data using an arbitrary 16-bit polynomial. This peripheral can compute CRC results using direct DMA access to data in XRAM.

Using a DMA transfer provides much higher data throughput than using SFR access. Since the CPU can be in Idle mode while the CRC is calculated, CRC1 also provides substantial power savings. The CRC1 module is not restricted to a limited list of fixed polynomials. Instead, the user can specify any valid 16-bit polynomial.

CRC1 accepts a stream of 8-bit data written to the CRC1IN register. A DMA transfer can be used to autonomously transfer data from XRAM to the CRC1IN SFR. The CRC1 module may also be used with SFR access by writing directly to the CRC1IN SFR. After each byte is written, the CRC resultant is updated on the CRC1OUTH:L SFRs. After writing all data bytes, the final CRC results are available from the CRC1OUTH:L registers. The final results may be flipped or inverted using the FLIP and INV bits in the CRC1CN SFR. The initial seed value can be reset to 0x0000 or seeded with 0xFFFF.

13.1. Polynomial Specification

The arbitrary polynomial should be written to the CRC1POLH:L SFRs before writing data to the CRCIN SFR.

A valid 16-bit CRC polynomial must have an x^{16} term and an x^0 term. Theoretically, a 16-bit polynomial might have 17 terms total. However, the polynomial SFR is only 16-bits wide. The convention used is to omit the x^{16} term. The polynomial should be written in big endian bit order. The most significant bit corresponds to the highest order term. Thus, the most significant bit in the CRC1POLH SFR represents the x^{15} term, and the least significant bit in the CRC1POLL SFR represents the x^0 term. The least significant bit of CRC1POLL should always be set to one. The CRC results are undefined if this bit is cleared to a zero.

Figure 13.1 depicts the polynomial representation for the CRC-16-CCIT polynomial $x^{16} + x^{12} + x^5 + 1$, or 0x1021.

CRC1POLH:L = 0x1021

	CRC1POLH								CRC1POLL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
	↓			↓							↓					↓
	$x^{16}+$			$x^{12}+$							x^5+					1

Figure 13.1. Polynomial Representation

13.2. Endianness

The CRC1 module is optimized to process big endian data. Data written to the CRC1IN SFR should be in the normal bit order with the most significant bit stored in bit 7 and the least significant bit stored in bit 0. The input data is shifted left into the CRC engine. The CRC1 module will process one byte at a time and update the results for each byte. When used with the DMA, the first byte to be written should be stored in the lowest address.

Some communications systems may transmit data least significant bit first and may require calculation of a CRC in the transmission bit order. In this case, the bits must be flipped, using the CRC0FLIP SFR, before writing to the CRC1IN SFR. The final 16-bit result may be flipped using the flip bit in the CRC1CN SFR. Note that the polynomial is always written in big endian bit order.

13.3. CRC Seed Value

Normally, the initial value or the CRC results is cleared to 0x0000. However, a CRC might be specified with an initial value preset to all ones (0xFFFF).

The steps to preset the CRC with all ones is as follows:

1. Set the SEED bit to 1.
2. Reset the CRC1 module by setting the CLR bit to 1 in CRC1CN.
3. Clear the SEED bit to 0.

The CRC1 module is now ready to calculate a CRC using a CRC seed value of 0xFFFF.

13.4. Inverting the Final Value

Sometimes it is necessary to invert the final value. This will take the ones complement of the final result.

The steps to flip the final CRC results are as follows:

1. Clear the CRC module by setting the CLR bit in CRC1CN SFR.
2. Write the polynomial to CRC1POLH:L.
3. Write all data bytes to CRC1IN.
4. Set the INV bit in the CRC1CN SFR to invert the final results.
5. Read the final CRC results from CRC1OUTH:L.

Clear the FLIP bit in the CRC1CN SFR.

13.5. Flipping the Final Value

The steps to flip the final CRC results are as follows:

1. Clear the CRC module by setting the CLR bit in CRC1CN SFR.
2. Write the polynomial to CRC1POLH:L.
3. Write all data bytes to CRC1IN.
4. Set the FLIP bit in the CRC1CN SFR to flip the final results.
5. Read the final CRC results from CRC1OUTH:L.
6. Clear the FLIP bit in the CRC1CN SFR.

The flip operation will exchange bit 15 with bit 0, bit 14 with bit 1, bit 13 with bit 2, and so on.

13.6. Using CRC1 with SFR Access

The steps to perform a CRC using SFR access with the CRC1 module is as follow:

1. If desired, set the SEED bit in the CRC1CN SFR to seed with 0xFFFF.
2. Clear the CRC module by setting the CLR bit in the CRC1CN SFR.
3. Clear the SEED bit, if set previously in step 1.
4. Write the polynomial to CRC1POLH:L.
5. Write all data bytes to CRC1IN.
6. If desired, invert and/or flip the final results using the INV and FLIP bits.
7. Read the final CRC results from CRC1OUTH:L.
8. Clear the INV and/or FLIP bits, if set previously in step 6.

Note that all of the CRC1 SFRs are on SFR page 0x2.

13.7. Using the CRC1 module with the DMA

The steps to computing a CRC using the DMA are as follows.

1. If desired, set the SEED bit in CRC1CN to seed with 0xFFFF.
2. Clear the CRC module by setting the CLR bit in CRC1CN SFR.
3. Clear the SEED bit, if set previously in step 1.
4. Write the polynomial to CRC1POLH:L.
5. Configure the DMA for the CRC operation:
   a. Disable the desired DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the desired DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the CRC1IN peripheral request by writing 0x2 to DMA0NCF.
   d. Enable the DMA interrupt on the selected channel by setting bit 7 of DMA0NCF.
   e. Write 0 to DMA0NMD to disable wrapping.
   f. Write the address of the first byte of CRC data to DMA0NBAH:L.
   g. Write the size of the CRC data in bytes to DMA0NSZH:L.
   h. Clear the address offset SFRs DMA0A0H:L.
   i. Enable the interrupt on the desired channel by setting the corresponding bit in DMA0INT.
   j. Enable the desired channel by setting the corresponding bit in DMA0EN.
   k. Enable DMA interrupts by setting bit 5 of EIE2.
6. Set the DMA mode bit (bit 3) in the CRC1CN SFR to initiate the CRC operation.
7. Wait on the DMA interrupt.
8. If desired, invert and/or flip the final results using the INV and FLIP bits.
9. Read the final results from CRC1OUTH:L.
10. Clear the INV and/or FLIP bits, if set previously in step 8.

SFR Definition 13.1. CRC1CN: CRC1 Control

Bit	7	6	5	4	3	2	1	0
Name	CLR				DMA	FLIP	INV	SEED
Type	R/W	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xBE; Not Bit-Addressable

Bit	Name	Function
7	RST	**Reset.** Setting this bit to 1 will reset the CRC module and set the CRC results SFR to the seed value as specified by the SEED bit. The CRC module should be reset before starting a new CRC. This bit is self-clearing.
6:4	Reserved	
3	DMA	**DMA Mode.** Setting this bit will configure the CRC1 module for DMA mode. Once a DMA channel has been configured to use accept peripheral requests from CRC1, setting this bit will initiate a DMA CRC operation. This bit should be cleared after each CRC DMA transfer.
2	FLIP	**Flip.** Setting this bit will flip the contents of the 16-bit CRC result SFRs. (CRC0OUTH:CRC0OUTL) This operation is normally performed only on the final CRC results. This bit should be cleared before starting a new CRC computation.
1	INV	**Invert.** Setting this bit will invert the contents of the 16-bit CRC result SFR. (CRC0OUTH:CRC0OUTL) This operation is normally performed only on the final CRC results. This bit should be cleared before starting a new CRC computation.
0	SEED	**Seed Polarity.** If this bit is zero, a seed value or 0x0000 will be used. If this bit is 1, a seed value of 0xFFFF will be used. This bit should be set before setting the RST bit.

SFR Definition 13.2. CRC1IN: CRC1 Data IN

Bit	7	6	5	4	3	2	1	0
Name	CRC1IN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xB9; Not Bit-Addressable

Bit	Name	Function
7:0	CRC1IN[7:0]	**CRC1Data IN.** CRC Data should be sequentially written, one byte at a time, to the CRC1IN Data input SFR. When the CRC1 module is used with the DMA, the DMA will write directly to this SFR.

SFR Definition 13.3. CRC1POLL: CRC1 Polynomial LSB

Bit	7	6	5	4	3	2	1	0
Name	CRC1POLL[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xBC; Not Bit-Addressable

Bit	Name	Function
7:0	CRC1POLL[7:0]	**CRC1 Polynomial LSB.**

SFR Definition 13.4. CRC1POLH: CRC1 Polynomial MSB

Bit	7	6	5	4	3	2	1	0
Name	CRC1POLH[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xBD; Not Bit-Addressable

Bit	Name	Function
7:0	CRC1POLH[7:0]	**CRC1 Polynomial MSB.**

SFR Definition 13.5. CRC1OUTL: CRC1 Output LSB

Bit	7	6	5	4	3	2	1	0
Name	CRC1OUTL[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xBA; Not Bit-Addressable

Bit	Name	Function
7:0	CRC1OUTL[7:0]	**CRC1 Output LSB**

SFR Definition 13.6. CRC1OUTH: CRC1 Output MSB

Bit	7	6	5	4	3	2	1	0
Name	CRC1OUTH[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xBB; Not Bit-Addressable

Bit	Name	Function
7:0	CRC1OUTH[7:0]	**CRC1 Output MSB.**

14. Advanced Encryption Standard (AES) Peripheral

The C8051F96x includes a hardware implementation of the Advanced Encryption Standard Block Cipher as specified in NIST publication FIPS 197 “Advanced Encryption Standard (AES), November 2001. The Rijndael encryption algorithm was chosen by NIST for the AES block cipher. The AES block cipher can be used to encrypt data for wireless communications. Data can be encrypted before transmission and decrypted upon reception. This provides security for private networks.

The AES block cipher is a Symmetric key encryption algorithm. Symmetric Key encryption relies on secret keys that are known by both the sender and receiver. The decryption key may be obtained using a simple transformation of the encryption key. AES is not a public key encryption algorithm.

The AES block Cipher uses a fixed 16 byte block size. So data less than 16 bytes must be padded with zeros to fill the entire block. Wireless data must be padded and transmitted in 16-byte blocks. The entire 16-byte block must be transmitted to successfully decrypt the information.

The AES engine supports key lengths of 128-bits, 192-bits, or 256-bits. A key size of 128-bits is sufficient to protect the confidentiality of classified secret information. The Advanced Encryption Standard was designed to be secure for at least 20 to 30 years. The 128-bit key provides fastest encryption. The 192-bit and 256-bit key lengths may be used to protect highly sensitive classified top secret information.

Since symmetric key encryption relies on secret keys, the security of the data can only be protected if the key remains secret. If the encryption key is stored in flash memory, then the entire flash should be locked to ensure the encryption key cannot be discovered. (See flash security.)

The basic AES block cipher is implemented in hardware. This hardware accelerator provides performance that may be 1000 times faster than a software implementation. The higher performance translates to a power savings for low-power wireless applications.

The AES block cipher, or block cipher modes based on the AES block cipher, is used in many wireless standards. These include several IEEE standards in the wireless PAN (802.15) and wireless LAN (802.11) working groups.

14.1. Hardware Description

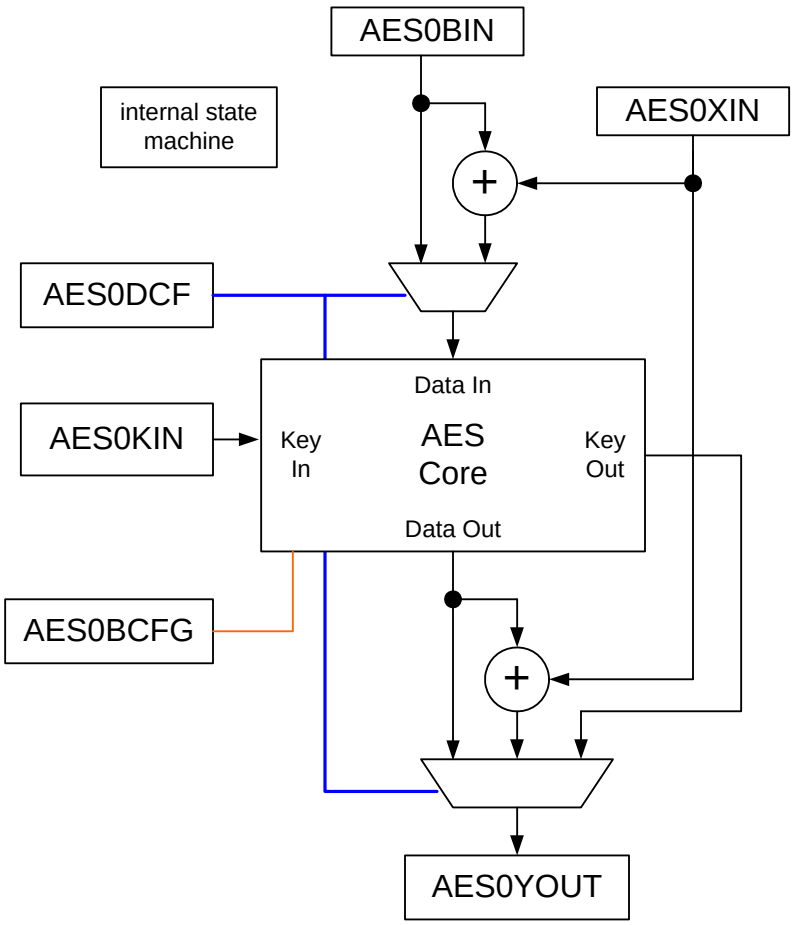


Figure 14.1. AES Peripheral Block Diagram

The AES Encryption module consists of these elements.

- AES Encryption/Decryption Core
- Configuration sfrs
- Key input sfr
- Data sfrs
- Input Multiplexer
- Output Multiplexer
- Input Exclusive OR block
- Output Exclusive OR block
- Internal State Machine

14.1.1. AES Encryption/Decryption Core

The AES Encryption/Decryption Core is a digital implementation of the Advanced Encryption Standard block cipher. The core may be used for either encryption or decryption. Encryption may be selected by setting bit 5 in the AES0BCFG sfr. When configured for encryption, plaintext is written to the AES Core data input and the encrypted ciphertext is read from the Data Output. Conversely, when configured for Decryption, encrypted ciphertext is written to the data input and decrypted plaintext is read form the Data Output.

When configured for Encryption, the encryption key must be written to the Key Input. When configured for decryption, the decryption key must be written to the Key Input.

The AES core may also be used to generate a decryption key from a known encryption key. To generate a decryption key, the core must be configured for encryption, the encryption key is written to the Key Input, and the Decryption Key may be read from the Key output.

AES is a symmetric key encryption algorithm. This means that the decryption key may be generated from an encryption key using a simple algorithm. Both keys must remain secret. If security of the encryption key is compromised, one can easily generate the decryption key.

Since it is easy to generate the decryption key, only the encryption key may be stored in Flash memory.

14.1.2. Data SFRs

The data sfrs are used for the data flow into and out of the AES module. When used with the DMA, the DMA itself will write to and read from the data sfrs. When used in manual mode, the data must be written to the data sfrs one byte at a time in the proper sequence.

The AES0KIN sfr provides a data path for the AES core Key input. For an encryption operation, the encryption key is written to the AES0KIN sfr, either by the DMA or direct sfr access. For a decryption operation, the decryption key must be written to the AES0KIN sfr.

The AES0BIN is the direct data input sfr for the AES block. For a simple encryption operation, the plaintext is written to the AES0BIN sfr – either by the DMA or direct sfr access. For decryption, the ciphertext to be decrypted is written to the AES0BIN sfr. The AES0BIN sfr is also used together with the AES0XIN when an exclusive OR operation is required on the input data path.

The AES0XIN sfr provides an input data path to the exclusive OR operator. The AES0XIN is not used for simple AES block cipher encryption or decryption. It is only use for block cipher modes that require an exclusive OR operator on the input or output data.

The AES core requires that the input data bytes are written in a specific order. When used with the DMA, this is managed by the internal state machine. When using direct sfr access, each of input data must be written one byte at a time to each sfr in this particular order.

1. Write AES0BIN
2. Write AES0XIN (optional)
3. Write AES0KIN

This sequence is repeated 16 times. When using a 192-bit or 256-bit key length, the remaining additional key bytes are written after writing all sixteen of the AES0BIN and AES0XIN bytes.

After encryption or decryption is completed, the resulting data may be read from the AES0YOUT. Optionally, exclusive OR data may be written to the AES0XIN sfr before reading the AES0YOUT sfr.

1. Write AES0XIN (optional)
2. Read AES0YOUT

14.1.3. Configuration sfrs

The AES Module has two configuration sfrs. The AES0BCFG sfr is used to configure the AES core. Bits 0 and 1 are used to select the Key size. The AES core supports 128-bit, 192-bit and 256-bit encryption. Bit 2 selects encrypt or decrypt. The AES enable bit (bit 3) is used to enable the AES module and start and new encryption operation. The AES DONE bit (bit 5) is the AES interrupt flag that signals a block of data has been completely encrypted or decrypted and is ready to be read from the AES0YOUT sfr. Note that the AES DONE interrupt is not normally used when the AES module is used with the DMA. Instead the DMA interrupt is used to signal that the encrypted or decrypted data has been transferred completely to memory. The DMA done interrupt is normally only used with direct sfr access.

The AES0DCFG sfr is used to select the data path for the AES module. Bits 0 through 2 are used to select the input and output multiplexer configuration. The AES data path should be configured prior to initiating a new encryption or decryption operation.

14.1.4. Input Multiplexer

The input multiplexer is used to select either the contents of the AES0BIN sfr or the contents of the AES0BIN sfr exclusive ORed with the contents of the AES0XIN sfr. The exclusive OR input data path provides support for CBC encryption.

14.1.5. Output Multiplexer

The output multiplexer selects the data source for the AES0YOUT sfr. The three possible sources are the AES Core data output, the AES Core Key output, and the AES core data output exclusive ORed with the AES0XIN sfr.

The AES core data output is used for simple encryption and decryption.

The exclusive OR output data path provides support for CBC mode decryption and CTR mode encryption/decryption. The AES0XIN is the source for both input and output exclusive OR data. When the AES0XIN is used with the input exclusive OR data path, the AEXIN data is written in sequence with the AES0BIN data. When used with the output XRO data path, the AES0XIN data is written after the encryption or decryption operation is complete.

The Key output is used to generate an inverse key. To generate a decryption key from an encryption key, the AES core should be configured for an encryption operation. To generate an encryption key from a decryption key, the AES core should be configured for a decryption operation.

14.1.6. Internal State Machine

The AES Module has an internal state machine that manages the data flow. The internal state machine accommodated the two different usage scenarios. When using the DMA, the internal state machine will send peripheral requests to the DMA requesting the DMA to transfer data from xram to the AES module input sfrs. Upon the completion of one block of data, the AES module will send peripheral requests requesting data to be transferred from the AES0YOUT sfr to xram. These peripheral requests and are managed by the internal state machine.

When not using the DMA, data must be written and read in a specific order. The DMA state machine will advance with each byte written or read.

The internal state machine may be reset by clearing the enable bit in the AESBGFG sfr. Clearing the enable bit before encryption or decryption operation will ensure that the state machine starts at the proper starting state.

When encrypting or decrypting multiple blocks it is not necessary to disable the AES module between blocks, as long as the proper sequence of events is obeyed.

14.2. Key Inversion

The Key output is used to generate an inverse key. To generate a decryption key from an encryption key, the AES core should be configured for an encryption operation. Dummy data and the encryption key are written to the AES0BIN and AES0KIN sfrs respectively. The output multiplexer should be configured to output the decryption key to the AES0YOUT SFR.

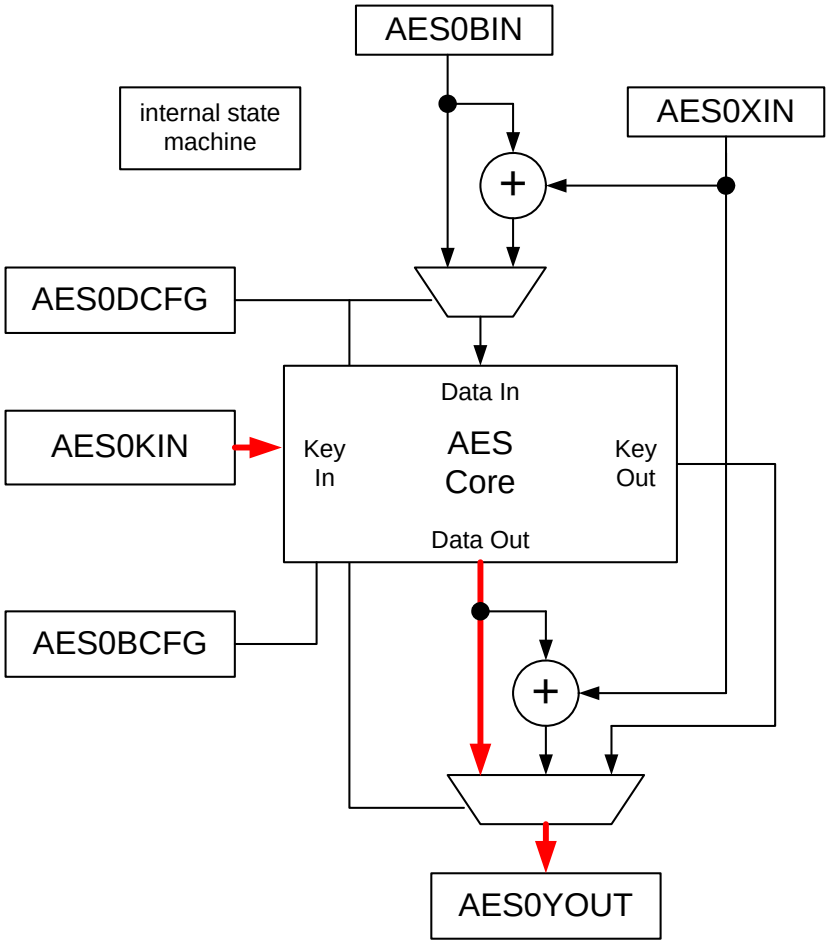


Figure 14.2. Key Inversion Data Flow

The dummy data may be zeros or arbitrary data. The content of the dummy data does not matter. But sixteen bytes of data must be written to the AES0BIN sfr to generate the inverse key.

14.2.1. Key Inversion using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

Steps to generate the Decryption Key from Encryption Key

- Prepare encryption key and dummy data in xram.
- Reset AES module by clearing bit 3 of AES0BCFG.
- Disable the first three DMA channels by clearing bits 0 to 2 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr.
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr.
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr.
  - Clear DMA0NSZH
  - Clear DMA0NAOH and DMA0NAOL.
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of dummy data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write 0x10 (16) to the DMA0NSZL sfr.
  - Clear DMA0NSZH
  - Clear DMA0NAOH and DMA0NAOL
- Configure the third DMA channel for the AES0YOUT sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr.
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address for the decryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr.
  - Clear DMA0NSZH.
  - Clear DMA0NAOH and DMA0NAOL.
- Clear first three DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first three DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for inverse key generation by writing 0x04 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG.
- Configure the AES core for encryption by setting the bit 2 of AES0BCFG.
- Initiate the encryption operation by setting bit 3 of AES0BCFG.
- Wait on the DMA interrupt from DMA channel 2.
- Disable the AES Module by clearing bit 2 of AES0BCFG.
- Disable the DMA by writing 0x00 to DMA0EN.

The key and data to be encrypted should be stored as an array with the first byte to be encrypted at the lowest address. The value of the big endian bit of the DMACF0 sfr does not matter. The AES block uses only one byte transfers, so there is no particular endianness associated with a one byte transfer.

The dummy data can be zeros or any value. The encrypted data is discarded, so the value of the dummy data does not mater.

It is not strictly required to use DMA channels 0, 1, and 2. Any three DMA channels may be used. The internal state machine of the AES module will send the peripheral requests in the required order.

If the other DMA channels are going to be used concurrently with encryption, then only the bits corresponding to the encryption channels should be manipulated in DM0AEN and DMA0NT sfrs.

14.2.2. Key Inversion using SFRs

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. However, it is also possible to use the DMA with direct SFR access. The steps are documented in the datasheet for completeness.

Steps to generate the Decryption Key from Encryption Key using SFR.

- First configure the AES block for Key inversion:
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for inverse key generation by writing 0x04 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for encryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Write the dummy data alternating with Key data:
  - Write the first dummy byte to AES0BIN
  - Write the first key byte to AES0KIN
  - Repeat until all dummy data bytes are written
- If using 192-bit and 256-bit key, write remaining key bytes to AES0KIN:
- Wait on AES done interrupt or poll bit 5 of AES0BCFG
- Read first byte of the decryption key from the AES0YOUT sfr

14.2.3. Extended Key Output Byte Order

When using a key length of 128-bits, the key output is in the same order as the bytes were written. When using an extended key of 192-bits or 256-bits. The extended portion of the key comes out first, before the first 16-bytes of the extended key.This is illustrated in Table 14.1.

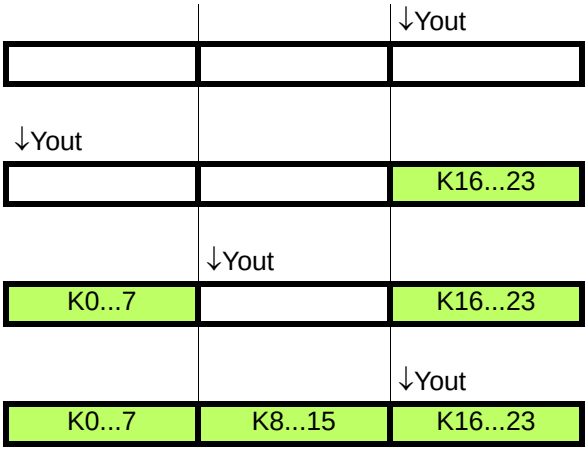
Table 14.1. Extended Key Output Byte Order

Size		Input	Output		
Bits	**Bytes**	**Order**	**Order**		
128	16	K0...15			K0...15
192	24	K0...23		K16...23	K0...15
256	32	K0...31	K16...23	K24...31	K0...15

14.2.4. Using the DMA to unwrap the extended Key

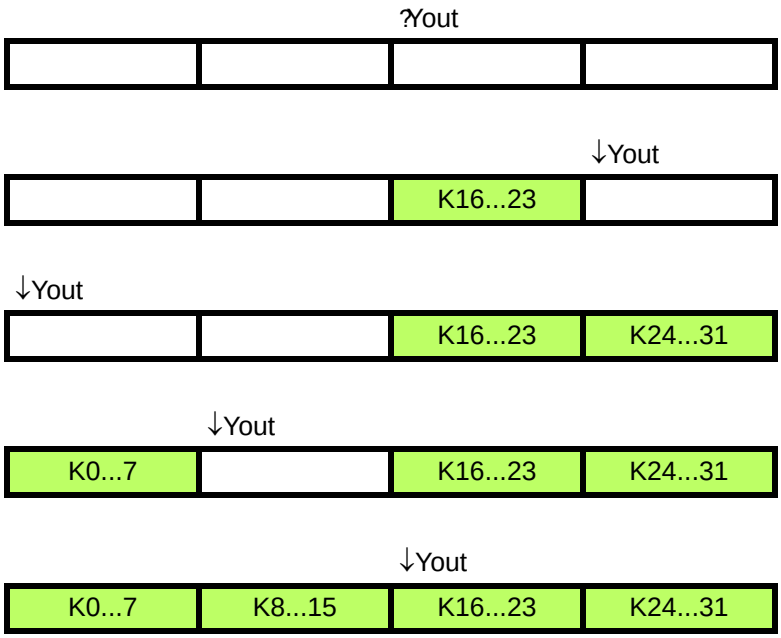
When used with the DMA, the address offset sfr DMANAOH/L may be manipulated to store the extended key in the desired order. This requires two DMA transfers for the AES0YOUT channel. When using a 192-bit key, the DMA0NSZ can be set to 24 bytes and the DMA0NA0 set to 16. This will place the last 8 bytes of the 192-bit key in the desired location as shown in Table 14.2. The Yout arrow indicated the address offset position after each 8-bytes are transferred. Enabling the WRAP bit in DMA0NMD will reset the DMA0NAO value after byte 23. Then the DMA0NZ can be reset to 16 for the remaining sixteen bytes.

Table 14.2. 192-Bit Key DMA Usage



When using a 256-bit key, the DMA0NSZ can be set to 32 and the DMA0NAOL set to 16 This will place the last16 bytes of the 256-bit key in the desired location as shown in Table 14.3.Enabling the WRAP bit in DMA0NMD will reset the DMA0NAO value after byte 31. Then the DMA0NZ can be set to 16 for the remaining sixteen bytes.

Table 14.3. 256-bit Key DMA Usage



14.3. AES Block Cipher

The basic AES Block Cipher is the basic encryption/decryption algorithm as defined by the NIST standard. A clock cipher mode is a method of encrypting and decrypting one block of data. The input data and output data are not manipulated, chained, or exclusive ORed with other data. This simple block cipher mode is sometimes called the Electronic Code Book (ECB) mode. The Electronic Codebook Mode is illustrated in Figure 14.3

Each operation represents one block (sixteen bytes) of data. The Plaintext is the plain unencrypted data. The Ciphertext is the encrypted data. The encryption key and decryption keys are symmetric. The decryption key is the inverse key of the decryption key. Note that the Encryption operation is not the same as the decryption operation. The two operations are different and the AES core operates differently depending on whether encryption or decryption is selected.

Note that each encryption or decryption operation is independent of other operations. Also note that the same key is used over and over again for each operation.

14.4. AES Block Cipher Data Flow

The AES0 module data flow for AES Block Cipher encryption and decryption shown in Figure 14.3. The data flow is the same for encryption and decryption. The AES0DCF sfr is always configured to route the AES core output to AES0YOUT. The XOR on the input and output paths are not used.

For an encryption operation, the core is configured for an encryption cipher, the encryption key is written to AES0KIN, the plaintext is written to the AES0BIN sfr. and the ciphertext is read from AES0YOUT.

For a decryption operation, the core is configured for an decryption cipher, the decryption key is written to AES0KIN, the ciphertext is written to the AES0BIN sfr. and the plaintext is read from AES0YOUT.

The key size is set to the desired key size.

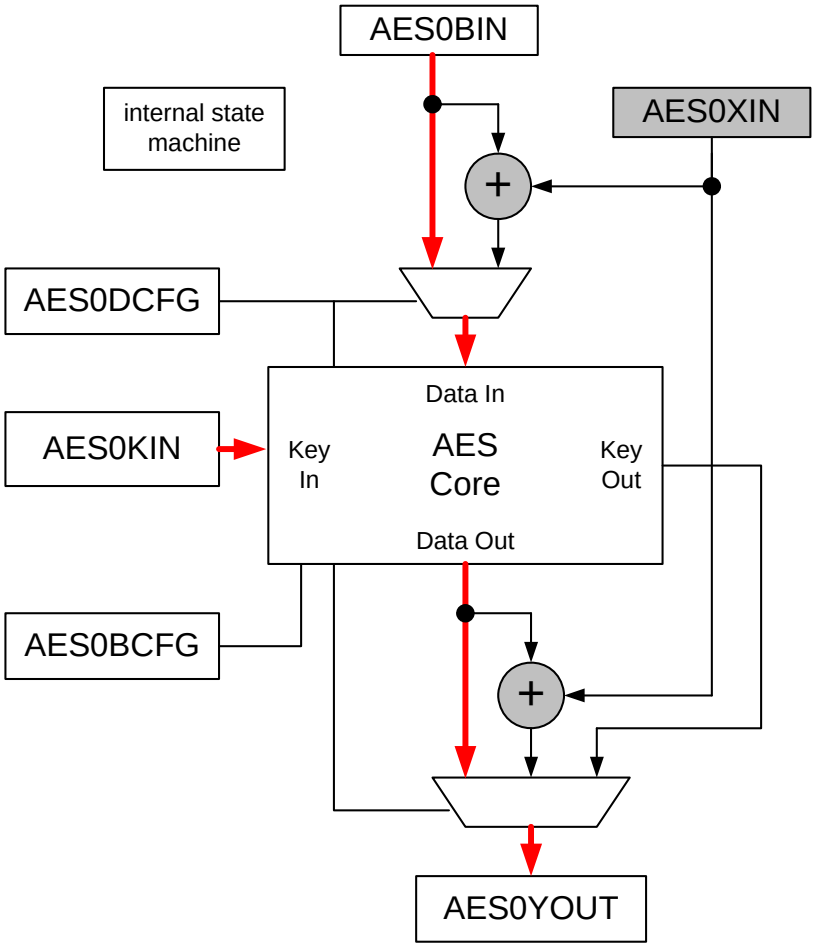


Figure 14.3. AES Block Cipher Data Flow

14.4.1. AES Block Cipher Encryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

Steps to encrypt data using Simple AES block encryption (ECB mode)

- Prepare encryption Key and data to be encrypted in xram.
- Reset AES module by clearing bit 2 of AES0BCFG.
- Disable the first three DMA channels by clearing bits 0 to 2 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the second DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Write 0x01 tDMA0NMD to enable wrapping
  - Write the xram location of encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to the DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to the AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be encrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0YOUT sfr
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr
  - Configure the third DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr.
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address for encrypted data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first three DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first three DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for AES Block Cipher by writing 0x00 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG
- Configure the AES core for encryption by setting the bit 2 of AES0BCFG
- Initiate the encryption operation be setting bit 3 of AES0BCFG
- Wait on the DMA interrupt from DMA channel 2
- Disable the AES Module by clearing bit 2 of AES0BCFG
- Disable the DMA by writing 0x00 to DMA0EN

14.4.2. AES Block Cipher Encryption using SFRs

- First Configure AES Module for AES Block Cipher
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for AES Block Cipher by writing 0x00 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for encryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write plaintext byte to AES0BIN.
  - Write encryption key byte to AES0KIN.
- Write remaining encryption key bytes to AES0KIN for 192-bit and 256-bit encryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Read 16 encrypted bytes from the AES0YOUT sfr.

If encrypting multiple blocks, this process may be repeated. It is not necessary reconfigure the AES module for each block.

14.5. AES Block Cipher Decryption

14.5.1. AES Block Cipher Decryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

- Prepare decryption key and data to be decryption in xram.
- Reset AES module by clearing bit 2 of AES0BCFG.
- sable the first three DMA channels by clearing bits 0 to 2 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Write 0x01 to DMA0NMD to enable wrapping
  - Write the xram location of decryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be decrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0YOUT sfr
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr
  - Configure the third DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping
  - Write the xram address for decrypted data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first three DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first three DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for AES Block Cipher by writing 0x00 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG
- Configure the AES core for decryption by clearing bit 2 of AES0BCFG
- Initiate the encryption operation be setting bit 3 of AES0BCFG
- Wait on the DMA interrupt from DMA channel 2
- Disable the AES Module by clearing bit 2 of AES0BCFG
- Disable the DMA by writing 0x00 to DMA0EN

14.5.2. AES Block Cipher Decryption using SFRs

- First Configure AES Module for AES Block Cipher
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for AES Block Cipher by writing 0x00 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for decryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write ciphertext byte to AES0BIN.
  - Write decryption key byte to AES0KIN.
- Write remaining decryption key bytes to AES0KIN for 192-bit and 256-bit decryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Read 16 plaintext bytes from the AES0YOUT sfr.

If decrypting multiple blocks, this process may be repeated. It is not necessary reconfigure the AES module for each block.

14.6. Block Cipher Modes

14.6.1. Cipher Block Chaining Mode

The Cipher Block Chaining (CBC) Mode algorithm significantly improves the strength of basic AES encryption by making each block encryption be a function of the previous block in addition to the current Plaintext and key. This algorithm is shown inFigure 14.4

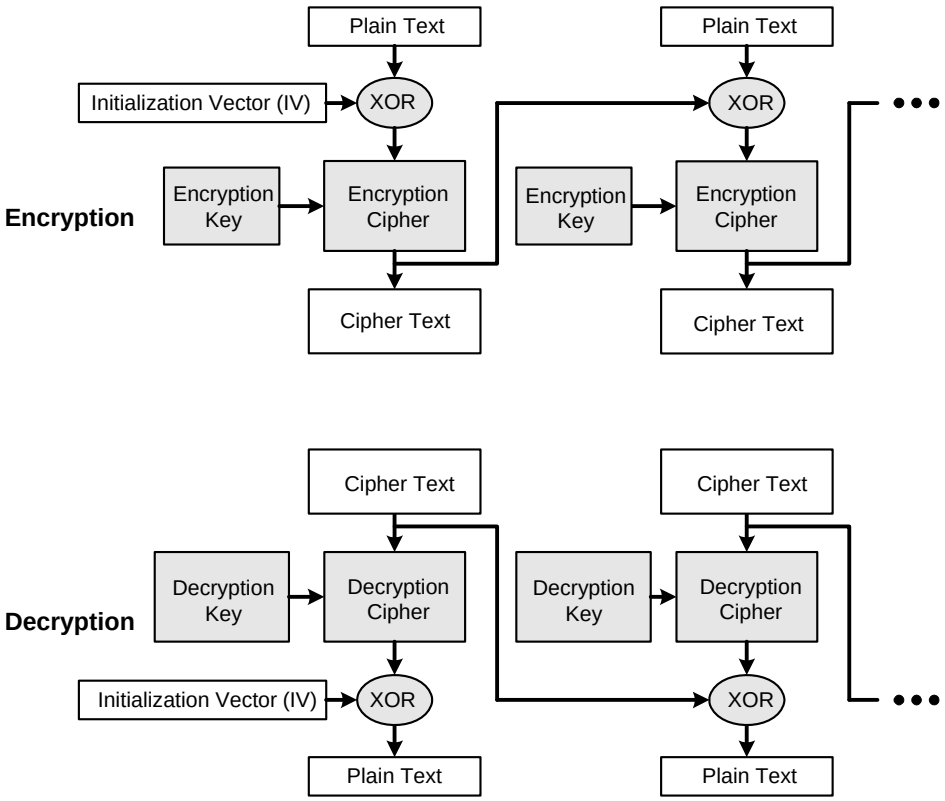


Figure 14.4. Cipher Block Chaining Mode

14.6.1.1. CBC Encryption Data Flow

The AES0 module data flow for CBC encryption is shown in Figure 14.5. The plaintext is written to the AES0BIN sfr. For the first block, the initialization vector is written to the AES0XIN sfr. For subsequent blocks, the previous block ciphertext is written to the AES0XIN sfr. The AES0DCF sfr is configured to XOR AES0XIN with AES0BIN for the AES core data input. The XOR on the output is not used. The AES core is configured for an encryption operation. The encryption key is written to AES0KIN. The key size is set to the desired key size.

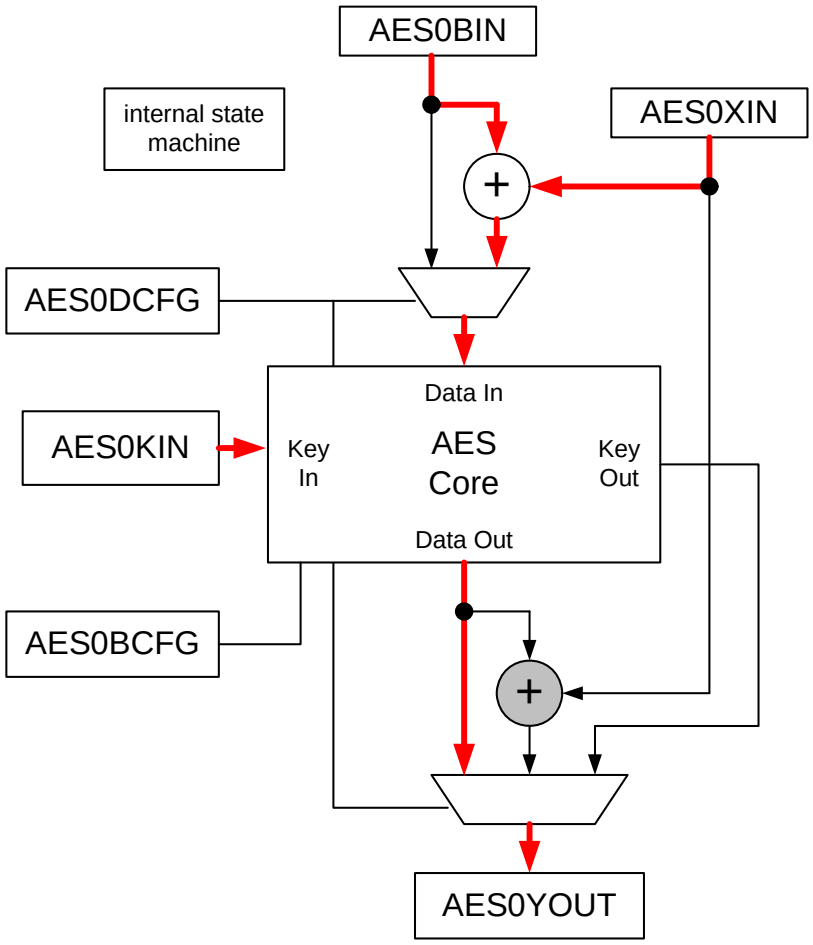


Figure 14.5. CBC Encryption Data Flow

14.6.2. CBC Encryption Initialization Vector Location

The first block to be encrypted uses the initialization vector for the AES0XIN data. Subsequent blocks will use the encrypted ciphertext from the previous block. The DMA is capable of encrypting multiple bocks. If the initialization is located at an arbitrary location in xram, the DMA base address location will need to be changed to the start of the encrypted ciphertext after encrypting the first block. However, if the initialization vector explicitly located in xram immediately before the encrypted ciphertext, the pointer will be advanced to the start of the encrypted ciphertext naturally and multiple blocks can be encrypted autonomously.

14.6.3. CBC Encryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

- Prepare encryption Key, initialization vector, and data to be encrypted in xram.

(The initialization vector should be located immediately before the data to be encrypted to encrypt multiple blocks.)

- Reset AES module by clearing bit 2 of AES0BCFG.
- Disable the first four DMA channels by clearing bits 0 to 3 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Write 0x01 to DMA0NMD to enable wrapping
  - Write the xram location of encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be encrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0XIN sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move xram to AES0XIN sfr by writing 0x07 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of initialization vector to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- * Configure the forth DMA channel for the AES0YOUT sfr
  - Select the forth channel by writing 0x03 to the DMA0SEL sfr
  - Configure the forth DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping
  - Write the xram address for encrypted data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be encrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first four DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.

- Enable first four DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for XOR on input data by writing 0x01 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG
- Configure the AES core for encryption by setting the bit 2 of AES0BCFG
- Initiate the encryption operation be setting bit 3 of AES0BCFG
- Wait on the DMA interrupt from DMA channel 3
- Disable the AES Module by clearing bit 2 of AES0BCFG
- Disable the DMA by writing 0x00 to DMA0EN

14.6.3.1. CBC Encryption using SFRs

- First Configure AES Module for CBC Block Cipher Mode Encryption
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for XOR on input data by writing 0x01 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for encryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write plaintext byte to AES0BIN.
  - Write initialization vector to AES0XIN
  - Write encryption key byte to AES0KIN.
- Write remaining encryption key bytes to AES0KIN for 192-bit and 256-bit decryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Read 16 encrypted bytes from the AES0YOUT sfr.

If encrypting multiple blocks, this process may be repeated. It is not necessary reconfigure the AES module for each block. When using Cipher Block Chaining the initialization vector is written to the AES0XIN sfr for the first block only, as described. Additional blocks will chain the encrypted data from the previous block.

14.6.4. CBC Decryption

The AES0 module data flow for CBC encryption is shown in Figure 14.6. The ciphertext is written to the AES0BIN sfr. For the first block, the initialization vector is written to the AES0XIN sfr. For subsequent blocks, the previous block ciphertext is written to the AES0XIN sfr. The AES0DCF sfr is configured to XOR AES0XIN with AES0BIN for the AES core data input. The XOR on the output is not used. The AES core is configured for an encryption operation. The encryption key is written to AES0KIN. The key size is set to the desired key size.

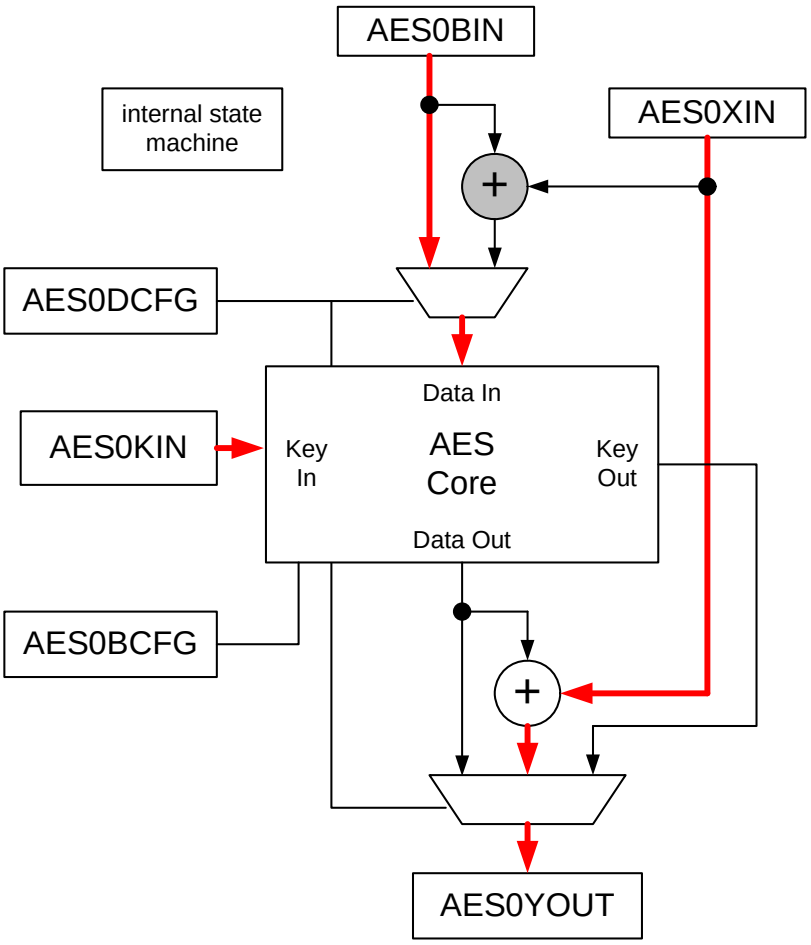


Figure 14.6. CBC Decryption Data Flow

14.6.4.1. CBC Decryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

- Prepare decryption Key, initialization vector, and data to be decrypted in xram.
- The initialization vector should be located immediately before the data to be decrypted to decrypt multiple blocks.
- Reset AES module by clearing bit 2 of AES0BCFG.
- Disable the first four DMA channels by clearing bits 0 to 3 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Write 0x01 to DMA0NMD to enable wrapping
  - Write the xram location of decryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be decrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0XIN sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move xram to AES0XIN sfr by writing 0x07 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of initialization vector to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the forth DMA channel for the AES0YOUT sfr
  - Select the forth channel by writing 0x03 to the DMA0SEL sfr
  - Configure the forth DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping
  - Write the xram address for decrypted data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first four DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first four DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for XOR on output data by writing 0x02 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG
- Configure the AES core for decryption by clearing bit 2 of AES0BCFG
- Initiate the decryption operation be setting bit 3 of AES0BCFG
- Wait on the DMA interrupt from DMA channel 3
- Disable the AES Module by clearing bit 2 of AES0BCFG
- Disable the DMA by writing 0x00 to DMA0EN

14.6.4.2. CBC Decryption using SFRs

- First Configure AES Module for CBC Block Cipher Mode Decryption
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for XOR on output data by writing 0x02 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for decryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write plaintext byte to AES0BIN.
  - Write encryption key byte to AES0KIN.
- Write remaining encryption key bytes to AES0KIN for 192-bit and 256-bit decryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Repeat alternating write read sequence 16 times
  - Write initialization vector to AES0XIN
  - Read decrypted data from AES0YOUT

If decrypting multiple blocks, this process may be repeated. It is not necessary reconfigure the AES module for each block. When using Cipher Block Chaining the initialization vector is written to the AES0XIN sfr for the first block only, as described. Additional blocks will chain the ciphertext data from the previous block.

14.6.5. Counter Mode

The Counter (CTR) Mode uses a sequential counter which is incremented after each block. This turns the block cipher into a stream cipher. This algorithm is shown inFigure 14.4. Note that the decryption operation actually uses the encryption key and encryption block cipher. The XOR operation is always on the output of the Cipher. The counter is a 16-byte block. Often the several bytes of the counter are initialized to a nonce (number used once). The last byte of the counter is incremented and propagated. Thus, the counter is treated as a 16-byte big endian integer.

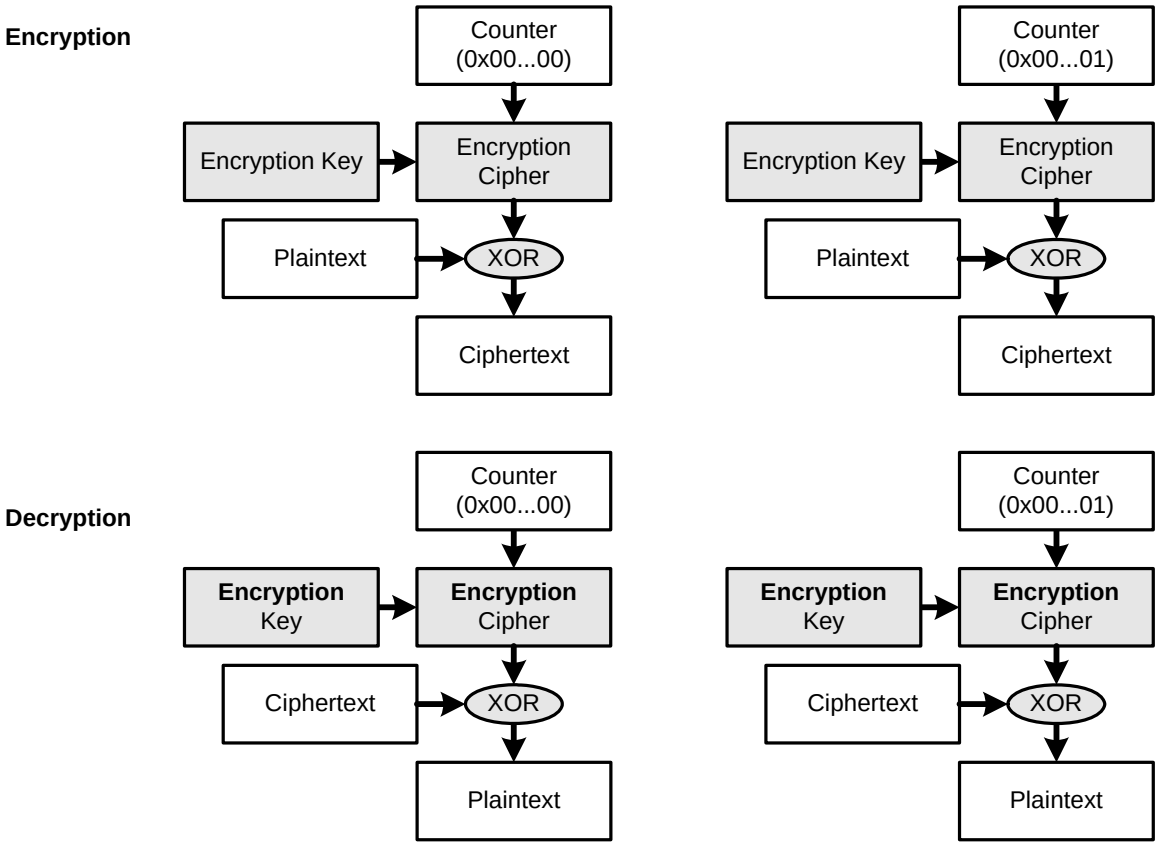


Figure 14.7. Counter Mode

14.6.5.1. CTR Data Flow

The AES0 module data flow for CTR encryption and decryption shown in Figure 14.5. The data flow is the same for encryption and decryption. The AES0DCF sfr is always configured to XOR AES0XIN with the AES Core output.The XOR on the input is not used. The AES core is configured for an encryption operation. The encryption key is written to AES0KIN. The key size is set to the desired key size.

For an encryption operation, the plaintext is written to the AES0BIN sfr and the ciphertext is read from AES0YOUT. For decryption, the ciphertext is written to AES0BIN and the plaintext is read from AES0YOUT.

Note the counter must be incremented after each block using software.

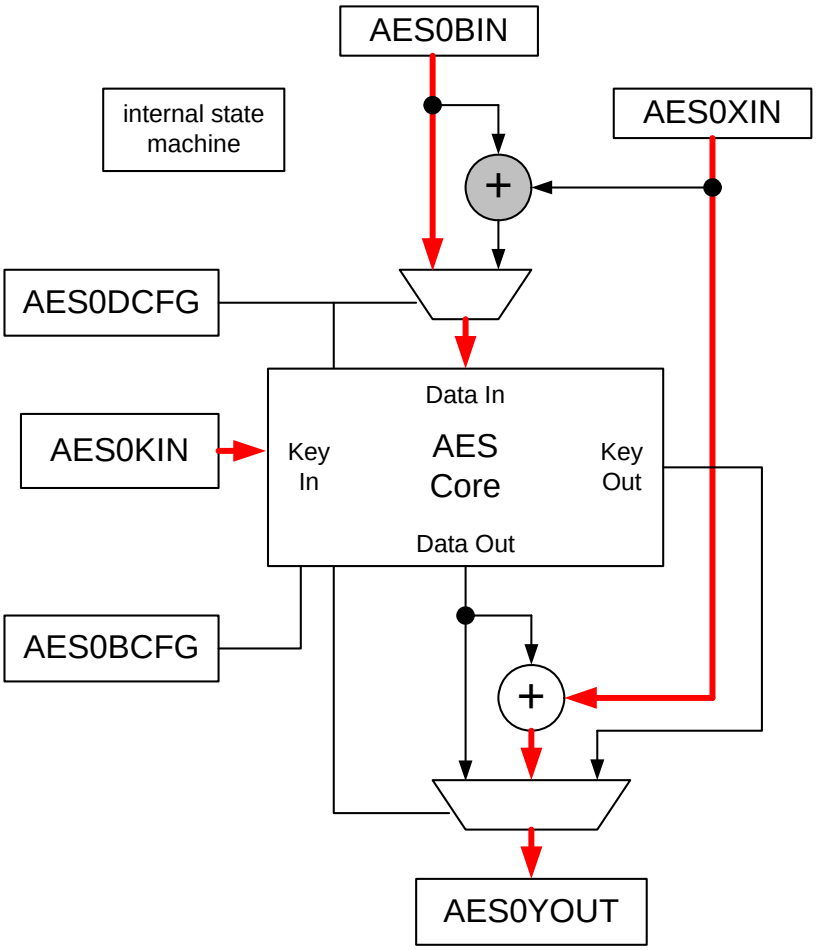


Figure 14.8. Counter Mode Data Flow

14.6.6. CTR Encryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the data sheet for completeness.

- Prepare encryption Key, counter, and data to be encrypted in xram.
- Reset AES module by clearing bit 2 of AES0BCFG.
- Disable the first four DMA channels by clearing bits 0 to 3 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping.
  - Write the xram location of encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be encrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write 16 DMA0NSZL sfr fro one block of 16 bytes.
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0XIN sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move xram to AES0XIN sfr by writing 0x07 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of counter to the DMA0NBAH and DMA0NBAL sfrs.
  - Write 16 DMA0NSZL sfr fro one block of 16 bytes.
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- * Configure the forth DMA channel for the AES0YOUT sfr
  - Select the forth channel by writing 0x03 to the DMA0SEL sfr
  - Configure the forth DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping
  - Write 16 DMA0NSZL sfr fro one block of 16 bytes.
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first four DMA interrupts by clearing bits 0 to 3 in the DMA0INT sfr.
- Enable first four DMA channels setting bits 0 to 3 in the DMA0EN sfr
- Configure the AES Module data flow for XOR on output data by writing 0x02 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG
- Configure the AES core for encryption by setting the bit 2 of AES0BCFG
- Initiate the encryption operation be setting bit 3 of AES0BCFG
- Wait on the DMA interrupt from DMA channel 3
- Disable the AES Module by clearing bit 2 of AES0BCFG

- Disable the DMA by writing 0x00 to DMA0EN
- Increment counter and repeat all steps for additional blocks

14.6.6.1. CTR Encryption using SFRs

- First Configure AES Module for CTR Block Cipher Mode Encryption
  - Reset AES module by writing 0x00 to AES0BCFG.
  - Configure the AES Module data flow for XOR on output data by writing 0x02 to the AES0DCFG sfr.
  - Write key size to bits 1 and 0 of the AES0BCFG.
  - Configure the AES core for encryption by setting bit 2 of AES0BCFG.
  - Enable the AES core by setting bit 3 of AES0BCFG.
- Repeat alternating write sequence 16 times
  - Write plaintext byte to AES0BIN.
  - Write counter byte to AES0XIN
  - Write encryption key byte to AES0KIN.
- Write remaining encryption key bytes to AES0KIN for 192-bit and 256-bit decryption only.
- Wait on AES done interrupt or poll bit 5 of AES0BCFG.
- Read 16 encrypted bytes from the AES0YOUT sfr.

If encrypting multiple blocks, increment the counter and repeat this process. It is not necessary reconfigure the AES module for each block.

SFR Definition 14.1. AES0BCFG: AES Block Configuration

Bit	7	6	5	4	3	2	1	0
Name			DONE	BUSY	EN	ENC	KSIZE	
Type	R	R	R/W	R	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE9; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
5	DONE	**Done Flag.** This bit is set upon completion of an encryption operation. When used with the DMA, the DONE bit signals the start of the out transfer. When used without the DMA, the done flag indicates data is ready to be read from AES0YOUT. The DONE bit is not cleared by hardware and must be cleared to zero by software at the start of the next encryption operation.
4	BUSY	**AES BUSY.** This bit is set while the AES block is engaged in an encryption or decryption operation. This bit is read only.
3	EN	**AES Enable.** This bit should be set to 1 to initiate an encryption or decryption operation. Clearing this bit to 0 will reset the AES module.
2	ENC	**Encryption/Decryption Select.** This is set to 1 to select an encryption operation. Clearing this bit to 0 will select a decryption operation.
1:0	SIZE[1:0]	**AES Key Size.** These bits select the key size for encryption/decryption. 00: 128-bits (16-bytes) 01: 198-bits (24-bytes) 10: 256-bits (32-bytes) 11: Reserved

SFR Definition 14.2. AES0DCFG: AES Data Configuration

Bit	7	6	5	4	3	2	1	0
Name			DONE	BUSY	EN	OUTSEL[1:0]		XORIN
Type	R	R	R/W	R	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xEA; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
2:0	OUTSEL[1:0]	**DATA Select.** These bits select the output data source for the AES0YOUT sfr. 00: Direct AES Data 01: AES Data XOR with AES0XIN 10: Inverse Key 11: reserved
1	XORIN	**XOR Input Enable.** Setting this bit with enable the XOR data path on the AES input. If enabled, AES0BIN will be XORed with the AES0XIN and the results will feed into the AES data input. Clearing this bit to 0 will disable the XOR gate on the input. The contents of AES0BIN will go directly into the AES data input.

SFR Definition 14.3. AES0BIN: AES Block Input

Bit	7	6	5	4	3	2	1	0
Name	AES0BIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xEB; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
7:0	AES0BIN[7:0]	**AES Block Input.** During an encryption operation, the plaintext is written to the AES0BIN sfr. During an decryption operation, the ciphertext is written to the AES0BIN sfr. During a key inversion the encryption key is written to AES0BIN. When used with the DMA, the DMA will write directly to this sfr. The AES0BIN may be used in conjunction with the AES0XIN sfr for some cipher block modes. When used without the DMA, AES0BIN, AES0XIN, and AES0KIN must be written in sequence. Reading this register will yield the last value written. This can be used for debug purposes.

SFR Definition 14.4. AES0XIN: AES XOR Input

Bit	7	6	5	4	3	2	1	0
Name	AES0XIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xEC; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
7:0	AES0XIN[7:0]	**AES XOR Input.** The AES0XIN may be used in conjunction with the AES0BIN sfr for some cipher block modes. When used with the DMA, the DMA will write directly to this sfr. When used without the DMA - AES0BIN, AES0XIN, and AES0KIN must be written in sequence. Reading this register will yield the last value written. This can be used for debug purposes.

SFR Definition 14.5. AES0KIN: AES Key Input

Bit	7	6	5	4	3	2	1	0
Name	AES0KIN[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xED; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
7:0	AES0KIN[7:0]	**AES Key Input.** During an encryption operation, the plaintext is written to the AES0BIN sfr. During an decryption operation, the ciphertext is written to the AES0BIN sfr. During a key inversion the encryption key is written to AES0BIN. When used with the DMA, the DMA will write directly to this sfr. The AES0BIN may be used in conjunction with the AES0XIN sfr for some cipher block modes. When used without the DMA - AES0BIN, AES0XIN, and AES0KIN must be written in sequence.

SFR Definition 14.6. AES0YOUT: AES Y Output

Bit	7	6	5	4	3	2	1	0
Name	AES0YOUT[7:0]							
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF5; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
7:0	AES0YOUT[7:0]	**AES Y Output.** Upon completion of an encryption/decryption operation The output data may be read, one byte at a time, from the AES0YOUT SFR. When used with the DMA, the DMA will read directly from this SFR. The AES0YOUT SFR may be used in conjunction with the AES0XIN SFR for some cipher block modes. When used without the DMA, the firmware should wait on the DONE flag before reading from the AES0YOUT SFR. When used without the DMA and using XOR on the output, one byte should be written to AES0XIN before reading each byte from AES0YOUT. Reading this register over the C2 interface will not increment the output data.

15. Encoder/Decoder

The Encoder/Decoder consists of three 8-bit data registers, a control register and an encoder/decoder logic block.

The size of the input data depends on the mode. The input data for Manchester encoding is one byte. For Manchester decoding it is two bytes. Three-out-of-Six encoding is two bytes. Three-out-of six decoding is three bytes.

The output size also depends on the mode selected. The input and output data size are shown below:

Table 15.1. Encoder Input and Output Data Sizes

	Input Data Size	Output Data Size
Operation	**Bytes**	**Bytes**
Manchester Encode	1	2
Manchester Decode	2	1
Three out of Six Encode	2	3
Three out of Six Decode	3	2

The input and output data is always right justified. So for Manchester mode the input uses only ENC0L and the output data is only in ENC0M and ENC0L. ENC0H is not used for Manchester mode

15.1. Manchester Encoding

To encode Manchester Data, first clear the MODE bit for Manchester encoding or decoding.

To encode, one byte of data is written to the data register ENC0L.

Setting the ENC bit will initiate encoding. After encoding, the encoded data will be in ENC0M and ENC0L. The upper nibble of the input data is encoded and placed in ENC0M. The lower nibble is encoded and placed in ENC0L.

Note that the input data should be readable in the data register until the encode bit is set. Once the READY bit is set, the input data has been replaced by the output data.

The ENC and DEC bits are self clearing. The READY bit is not cleared by hardware and must be cleared manually. The control register does not need to be bit addressable. The READY bit can be cleared while setting the ENC or DEC bit using a direct or immediate SFR mov instruction.

Table 15.2. Manchester Encoding

Input Data			Encoded Output		
nibble			byte		
dec	hex	bin	bin	hex	dec
0	0	0000	10101010	AA	170
1	1	0001	10101001	A9	169
2	2	0010	10100110	A6	166
3	3	0011	10100101	A5	165
4	4	0100	10011010	9A	154
5	5	0101	10011001	99	153
6	6	0110	10010110	96	150
7	7	0111	10010101	95	149
8	8	1000	01101010	6A	106
9	9	1001	01101001	69	105
10	A	1010	01100110	66	102
11	B	1011	01100101	65	101
12	C	1100	01011010	5A	90
13	D	1101	01011001	59	89
14	E	1110	01010110	56	86
15	F	1111	01010101	55	85

15.2. Manchester Decoding

Two bytes of Manchester data are written to ENC0M and ENC0L sfrs. Then the DEC bit is set to initiate decoding. After decoding the READY bit will be set. If the data is not a valid encoded Manchester data, the ERROR bit will be set, and the output will be all FFs.

The encoding and decoding process should be symmetric. Data can be written to the ENC0L sfr, then encoded, then decoding will give the original data.

Table 15.3. Manchester Decoding

Input			Decoded Output		
Byte			Nibble		
bin	hex	dec	dec	hex	bin
01010101	55	85	15	F	1111
01010110	56	86	14	E	1110
01011001	59	89	13	D	1101
01011010	5A	90	12	C	1100
01100101	65	101	11	B	1011
01100110	66	102	10	A	1010
01101001	69	105	9	9	1001
01101010	6A	106	8	8	1000
10010101	95	149	7	7	0111
10010110	96	150	6	6	0110
10011001	99	153	5	5	0101
10011010	9A	154	4	4	0100
10100101	A5	165	3	3	0011
10100110	A6	166	2	2	0010
10101001	A9	169	1	1	0001
10101010	AA	170	0	0	0000

15.3. Three-out-of-Six Encoding

Three out of six encoding is similar to Manchester encoding. In Three-out-of-Six encoding a nibble is encoded as a six-bit symbol. Four nibbles are encoded as 24-bits (three bytes).

Two bytes of data to be encoded are written to ENC0M and ENC0L. The MODE bit is set to 1 for Three-out-of-Six encoding. Setting the ENC bit will initiate encoding.

After encoding, the three encoded bytes are in ENC2-0.

Table 15.4. Three-out-of-Six Encoding Nibble

Input			**Encoded Output**			
nibble			**symbol**			
dec	hex	bin	bin	dec	hex	octal
0	0	0000	010110	22	16	26
1	1	0001	001101	13	0D	15
2	2	0010	001110	14	0E	16
3	3	0011	001011	11	0B	13
4	4	0100	011100	28	1C	34
5	5	0101	011001	25	19	31
6	6	0110	011010	26	1A	32
7	7	0111	010011	19	13	23
8	8	1000	101100	44	2C	54
9	9	1001	100101	37	25	45
10	A	1010	100110	38	26	46
11	B	1011	100011	35	23	43
12	C	1100	110100	52	34	64
13	D	1101	110001	49	31	61
14	E	1110	110010	50	32	62
15	F	1111	101001	41	29	51

15.4. Three-out-of-Six Decoding

Three-out-of-Six decoding is a similar inverse process. Three bytes of encoded data are written to ENC2-0. The DEC bit is set to initiate decoding. The READY bit will be set when decoding is complete. The ERROR bit will be set if the input date is not valid Three-out-of-Six data.

The Three-out-of-Six encoder decode process is also symmetric. Two bytes of arbitrary data may be written to ENC0M-ENC0L, then encoded, then decoding will yield the original data.

Table 15.5. Three-out-of-Six Decoding

Input			**Decoded Output**		
Symbol			**Nibble**		
bin	octal	dec	dec	hex	bin
001011	13	11	3	3	0011
001101	15	13	1	1	0001
001110	16	14	2	2	0010
010011	23	19	7	7	0111
010110	26	22	0	0	0000
011001	31	25	5	5	0101
011010	32	26	6	6	0110
011100	34	28	4	4	0100
100011	43	35	11	B	1011
100101	45	37	9	9	1001
100110	46	38	10	A	1010
101001	51	41	15	F	1111
101100	54	44	8	8	1000
110001	61	49	13	D	1101
110010	62	50	14	E	1110
110100	64	52	12	C	1100

15.5. Encoding/Decoding with SFR Access

The steps to perform a Encode/Decode operation using SFR access with the ENC0 module are as follow:

1. Clear ENC0CN by writing 0x00.
2. Write the input data to ENC0H:M:L.
3. Write the operation value to ENC0CN setting ENC, DEC, and MODE bits as desired and clearing all other bits.
   a. Write 0x10 for Manchester Decode operation.
   b. Write 0x11 for Three-out-of-Six Decode operation.
   c. Write 0x20 for Manchester Encode operation.
   d. Write 0x21 for Three-out-of-Six Encode operation.
4. Wait on the READY bit in ENC0CN.
5. For a decode operation only, check the ERROR bit in ENC0CN for a decode error.
6. Read the results from ENC0H:M:L.
7. Repeat steps 2-6 for all remaining data.

Note that all of the ENC0 SFRs are on SFR page 0x2. The READY and ERROR must be cleared in ENC0CN with each operation.

15.6. Decoder Error Interrupt

The Encoder/Decoder peripheral is capable of generating an interrupt on a decoder error. Normally, when used with the DMA, the DMA will transfer the entire specified transfer size to and from the Encoder/Decoder peripheral. If a decoder error occurs, decoding will continue until all data has been decoded. The error bit in the ENC0CN SFR will indicate if an error has occurred anywhere in the DMA transfer. Some applications will discard the entire packet after a single decoder error. Aborting the decoder operation at the first decoder error will conserve energy and minimize packet receiver turn-around time. The decoder interrupt service routine should first stall the ENC0 DMA channels by selecting the ENC0 DMA channels and then setting the STALL bit. Then disable the DMA channels by clearing the relevant DMA0EN bits. In addition, clear any ENC DMA channel interrupts by clearing the respective bits in DMA0NINT.

15.7. Using the ENC0 module with the DMA

The steps for Encoding/Decoding using the DMA are as follows.

1. Clear the ENC module by writing 0x00 to the ENC0CN SFR.
2. Configure the first DMA channel for the XRAM-to-ENC0 input transfer:
   a. Disable the first DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the first DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the XRAM-to-ENC0 input peripheral request by writing 0x00 to DMA0NCF.
   d. Set the ENDIAN bit in DMA0NCF to enable big-endian multi-byte DMA transfers.
   e. Write 0 to DMA0NMD to disable wrapping.
   f. Write the address of the first byte of input data DMA0NBAH:L.
   g. Write the size of the input data transfer in bytes to DMA0NSZH:L.
   h. Clear the address offset SFRs DMA0A0H:L.
3. Configure the second DMA channel for the ENC0-to-XRAM output transfer:
   a. Disable the second DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the second DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the SPI1DAT-to-XRAM output peripheral request by writing 0x01 to DMA0NCF.
   d. Set the ENDIAN bit in DMA0NCF to enable big-endian multi-byte DMA transfers.
   e. Enable DMA interrupts for the second channel by setting bit 7 of DMA0NCF.
   f. Write 0 to DMA0NMD to disable wrapping.
   g. Write the address for the first byte of the output data to DMA0NBAH:L.
   h. Write the size of the output data transfer in bytes to DMA0NSZH:L.
   i. Clear the address offset SFRs DMA0A0H:L.
   j. Enable the interrupt on the second channel by setting the corresponding bit in DMA0INT.
4. Clear the interrupt bits in DMA0INT for both channels.
5. Enable DMA interrupts by setting bit 5 of EIE2.
6. If desired for a decode operation, enable the ERROR interrupt bit by setting bit 6 of EIE2.
7. Write the operation value to ENC0CN setting ENC, DEC, and MODE bits for the desired operation. The DMA bit and ENDIAN bits must be set. The READY bits and ERROR bits must be cleared.
   a. Write 0x16 for Manchester Decode operation.
   b. Write 0x17 for Three-out-of-Six Decode operation.
   c. Write 0x26 for Manchester Encode operation.
   d. Write 0x27 for Three-out-of-Six Encode operation.
8. Wait on the DMA interrupt.
9. Clear the DMA enables in the DMA0EN SFR.
10. Clear the DMA interrupts in the DMA0INT SFR.
11. For a decode operation only, check the ERROR bit in ENC0CN for a decode error.

Note that the encoder and all DMA channels should be configured for Big-Endian mode.

SFR Definition 15.1. ENC0CN: Encoder Decoder 0 Control

Bit	7	6	5	4	3	2	1	0
Name	READY	ERROR	ENC	DEC		DMA	ENDIAN	MODE
Type	R	R	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC5; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
7	READY	**Ready Flag.**
6	ERROR	**Error Flag.**
5	ENC	**Encode.** Setting this bit will initiate an Encode operation.
4	DEC	**Decode.** Setting this bit will initiate a Decode operation.
2	DMA	**DMA Mode Enable.** This bit should be set when using the encoder/decoder with the DMA.
1	ENDIAN	**Big-Endian DMA Mode Select.** This bit should be set when using the DMA with big-endian multiple byte DMA transfers. The DMA must also be configured for the same endian mode.
0	MODE	**Mode.** 0: Select Manchester encoding or decoding. 1:Select Three-out-of-Six encoding or decoding.

SFR Definition 15.2. ENC0L: ENC0 Data Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ENC0L[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xC2; Bit-Addressable

Bit	Name	Function
7:0	ENC0L[7:0]	**ENC0 Data Low Byte.**

SFR Definition 15.3. ENC0M: ENC0 Data Middle Byte

Bit	7	6	5	4	3	2	1	0
Name	ENC0M[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xC3; Bit-Addressable

Bit	Name	Function
7:0	ENC0M[7:0]	**ENC0 Data Middle Byte.**

SFR Definition 15.4. ENC0H: ENC0 Data High Byte

Bit	7	6	5	4	3	2	1	0
Name	ENC0H[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xC4; Bit-Addressable

Bit	Name	Function
7:0	ENC0H[7:0]	**ENC0 Data High Byte.**

16. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the C8051F96x's resources and peripherals. The CIP-51 controller core duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the C8051F96x. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 16.3 lists the SFRs implemented in the C8051F96x device family.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g., P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing unoccupied addresses in the SFR space will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 16.3, for a detailed description of each register.

16.1. SFR Paging

The CIP-51 features SFR paging, allowing the device to map many SFRs into the 0x80 to 0xFF memory address space. The SFR memory space has 256 *pages*. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFRs. The C8051F96x family of devices utilizes two SFR pages: 0x00 and 0x0F. SFR pages are selected using the Special Function Register Page Selection register, SFRPAGE (see SFR Definition 11.3). The procedure for reading and writing an SFR is as follows:

1. Select the appropriate SFR page number using the SFRPAGE register.
2. Use direct accessing mode to read or write the special function register (MOV instruction).

16.2. Interrupts and SFR Paging

When an interrupt occurs, the current SFRPAGE is pushed onto the SFR page stack. Upon execution of the RETI instruction, the SFR page is automatically restored to the SFR Page in use prior to the interrupt. This is accomplished via a three-byte *SFR Page Stack*. The top byte of the stack is SFRPAGE, the current SFR Page. The second byte of the SFR Page Stack is SFRNEXT. The third, or bottom byte of the SFR Page Stack is SFRLAST. Upon an interrupt, the current SFRPAGE value is pushed to the SFRNEXT byte, and the value of SFRNEXT is pushed to SFRLAST. On a return from interrupt, the SFR Page Stack is popped resulting in the value of SFRNEXT returning to the SFRPAGE register, thereby restoring the SFR page context without software intervention. The value in SFRLAST (0x00 if there is no SFR Page value in the bottom of the stack) of the stack is placed in SFRNEXT register. If desired, the values stored in SFRNEXT and SFRLAST may be modified during an interrupt, enabling the CPU to return to a different SFR Page upon execution of the RETI instruction (on interrupt exit). Modifying registers in the SFR Page Stack does not cause a push or pop of the stack. Only interrupt calls and returns will cause push/pop operations on the SFR Page Stack.

On the C8051F96x devices, the SFRPAGE must be explicitly set in the interrupt service routine.

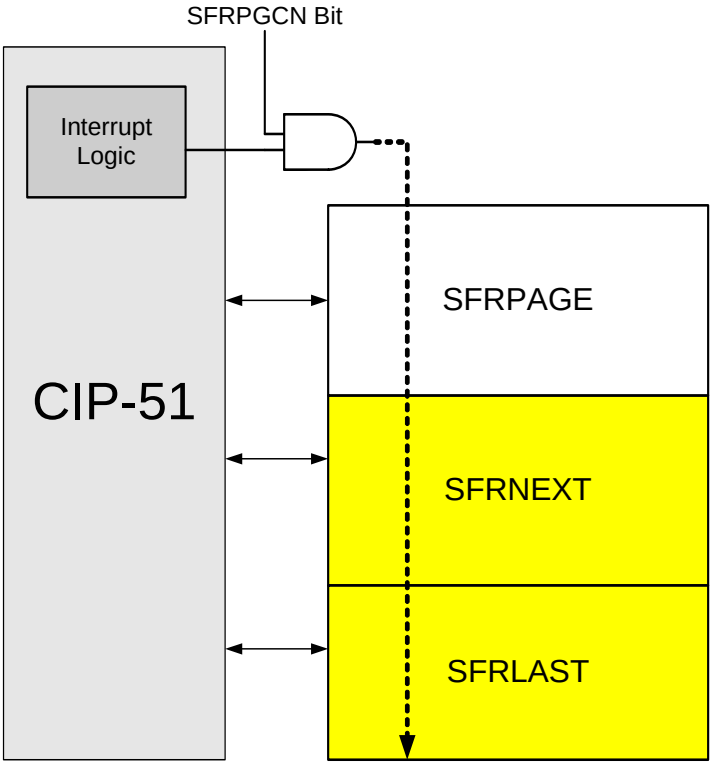


Figure 16.1. SFR Page Stack

Automatic hardware preserving and restoring of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFR0CN). This function defaults to “enabled” upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) are provided in Table 16.3 in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Certain SFRs are accessible from ALL SFR pages, and are denoted by the “(ALL PAGES)” designation. For example, the Port I/O registers P0, P1, P2, and P3 all have the “(ALL PAGES)” designation, indicating these SFRs are accessible from all SFR pages regardless of the SFRPAGE register value.

SFR Definition 16.1. SFRPGCN: SFR Page Control

Bit	7	6	5	4	3	2	1	0
Name								SFRPGEN
Type	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	1

;SFR Page = 0xF; SFR Address = 0x8E

Bit	Name	Function
7:1	Unused	Read = 0000000b; Write = Don't Care
0	SFRPGEN	**SFR Automatic Page Control Enable.** Upon interrupt, the C8051 Core will vector to the specified interrupt service routine. This bit controls the automatic preservation and restoration of the SRFPAGE by hardware. 0: SFR Automatic Paging disabled. The C8051 core will neither preserve the SRFPAGE upon entering an interrupt service routine, nor restore the SFRPAGE upon exiting the interrupt service routine. The interrupt service routine should preserve and restore the active SFRPAGE in firmware. 1: SFR Automatic Paging enabled. The C8051 core will preserve the SRFPAGE upon entering an interrupt service routine and restore the SFRPAGE upon exiting the Interrupt service routine. The firmware does not need to preserve and restore the SFRPAGE in the interrupt service routing. However, firmware must set the SFRPAGE within the interrupt service routine before accessing SFRs.

SFR Definition 16.2. SFRPAGE: SFR Page

Bit	7	6	5	4	3	2	1	0
Name	SFRPAGE[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xA7

Bit	Name	Function
7:0	SFRPAGE[7:0]	**SFR Page Bits.** Represents the SFR Page the C8051 core uses when reading or modifying SFRs. Write: Sets the SFR Page. Read: Byte is the SFR page the C8051 core is using. When enabled in the SFR Page Control Register (SFR0CN), the C8051 core will automatically switch to the SFR Page that contains the SFRs of the corresponding peripheral/function that caused the interrupt, and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt). SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and not by reading/writing to the SFRPAGE register)

SFR Definition 16.3. SFRNEXT: SFR Next

Bit	7	6	5	4	3	2	1	0
Name	SFRNEXT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

;SFR Page = All Pages; SFR Address = 0x85

Bit	Name	Function
7:0	SFRNEXT[7:0]	**SFR Page Bits.** This is the value that will go to the SFR Page register upon a return from interrupt. Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt. Read: Returns the value of the SFR page contained in the second byte of the SFR stack. SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to “push” or “pop”. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

SFR Definition 16.4. SFRLAST: SFR Last

Bit	7	6	5	4	3	2	1	0
Name	SFRLAST[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

;SFR Page = All Pages; SFR Address = 0x86

Bit	Name	Function
7:0	SFRLAST[7:0]	**SFR Page Stack Bits.** This is the value that will go to the SFRNEXT register upon a return from interrupt. Write: Sets the SFR Page in the last entry of the SFR Stack. This will cause the SFRNEXT SFR to have this SFR page value upon a return from interrupt. Read: Returns the value of the SFR page contained in the last entry of the SFR stack. SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to “push” or “pop”. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Table 16.1. SFR Map (0xC0–0xFF)

Addr.	Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xF8	0x0	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
	0x2	SPI1CN	PC0DCL	PC0DCH	PC0INT0	PC0INT1	DC0RDY		
	0xF		P4MDOUT	P5MDOUT	P6MDOUT	P7MDOUT	CLKMODE	PCLKEN	
0xF0	0x0		P0MDIN	P1MDIN	P2MDIN	SMB0ADR	SMB0ADM	EIP1	EIP2
	0x2		PC0CMP1L	PC0CMP1M	PC0CMP1H	PC0HIST	AES0YOUT		
	0xF		P3MDIN	P4MDIN	P5MDIN	P6MDIN	PCLKACT		
0xE8	0x0	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
	0x2		AES0BCFG	AES0DCFG	AES0BIN	AES0XIN	AES0KIN		
	0xF		DEVICEID	REVID					
0xE0	0x0	ACC	XBR0	XBR1	XBR2	IT01CF		EIE1	EIE2
	0x2		PC0CMP0L	PC0CMP0M	PC0CMP0H	PC0TH			
	0xF		XBR0	XBR1	XBR2	IT01CF			
0xD8	0x0	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	PCA0PWM
	0x2		PC0MD	PC0CTR0L	PC0TRML	PC0CTR0H	PC0CTR1L	PC0TRMH	PC0CTR1H
	0xF		P4	P5	P6	P7			
0xD0	0x0	PSW	REF0CN	PCA0CPL5	PCA0CPH5	P0SKIP	P1SKIP	P2SKIP	P0MAT
	0x2		DMA0SEL	DMA0EN	DMA0INT	DMA0MINT	DMA0BUSY	DMA0NMD	PC0PCF
	0xF								
0xC8	0x0	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	PCA0CPM5	P1MAT
	0x2		DMA0NCF	DMA0NBAL	DMA0NBAH	DMA0NAOL	DMA0NAOH	DMA0NSZL	DMA0NSZH
	0xF								
0xC0	0x0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P0MASK
	0x2		PC0STAT	ENC0L	ENC0M	ENC0H	ENC0CN	VREGINSDL	VREGINSDH
	0xF								

Table 16.2. SFR Map (0x80–0xBF)

Addr.	Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
0xB8	0x0	IP	IREF0CN	ADC0AC	ADC0MX	ADC0CF	ADC0L	ADC0H	P1MASK
	0x2		CRC1IN	CRC1OUTL	CRC1OUTH	CRC1POLL	CRC1POLH	CRC1CN	
	0xF		IREF0CF	ADC0PWR	ADC0TK		TOFFL	TOFFH	
0xB0	0x0	P3	OSCXCN	OSCICN	PMU0MD		PMU0CF	PMU0FL	FLKEY
	0x2		DC0CN	DC0CF	DC0MD		LCD0CHPCN	LCD0BUFMD	
	0xF		P3MDOUT	OSCIFL	OSCICL			FLSCL	
0xA8	0x0	IE	CLKSEL	EMI0CN	EMI0CF	RTC0ADR	RTC0DAT	RTC0KEY	EMI0TC
	0x2		LCD0CLKDIVL	LCD0CLKDIVH	LCD0MSCN	LCD0MSCF	LCD0CHPCF	LCD0CHPMD	LCD0VBMCF
	0xF		CLKSEL	P6DRV	P7DRV	LCD0BUFCF			
0xA0	0x0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	SFRPAGE
	0x2		SPI1CFG	SPI1CKR	SPI1DAT	LCD0PWR	LCD0CF	LCD0VBMCN	
	0xF		P3DRV	P4DRV	P5DRV	P0DRV	P1DRV	P2DRV	
0x98	0x0	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
	0x2		LCD0DD	LCD0DE	LCD0DF	LCD0CNTRST	LCD0CN	LCD0BLINK	LCD0TOGR
	0xF					LCD0BUFCN			
0x90	0x0	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H		
	0x2		LCD0D6	LCD0D7	LCD0D8	LCD0D9	LCD0DA	LCD0DB	LCD0DC
	0xF		CRC0DAT	CRC0CN	CRC0IN	CRC0FLIP		CRC0AUTO	CRC0CNT
0x88	0x0	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
	0x2		LCD0D0	LCD0D1	LCD0D2	LCD0D3	LCD0D4	LCD0D5	
	0xF							SFRPGCN	
0x80	0x0	P0	SP	DPL	DPH	PSBANK	SFRNEXT	SFRLAST	PCON
	0x2								
	0xF								

Table 16.3. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
ADC0AC	0xBA	0x0	ADC0 Accumulator Configuration	88
ADC0CF	0xBC	0x0	ADC0 Configuration	87
ADC0CN	0xE8	All pages	ADC0 Control	86
ADC0GTH	0xC4	0x0	ADC0 Greater-Than Compare High	92
ADC0GTL	0xC3	0x0	ADC0 Greater-Than Compare Low	92
ADC0H	0xBE	0x0	ADC0 High	91
ADC0L	0xBD	0x0	ADC0 Low	91
ADC0LTH	0xC6	0x0	ADC0 Less-Than Compare Word High	93
ADC0LTL	0xC5	0x0	ADC0 Less-Than Compare Word Low	93
ADC0MX	0xBB	0x0	ADC0 MUX	96
ADC0PWR	0xBA	0xF	ADC0 Burst Mode Power-Up Time	89
ADC0TK	0xBB	0xF	ADC0 Tracking Control	90
AES0BCFG	0xE9	0x2	AES0 Block Configuration	202
AES0BIN	0xEB	0x2	AES0 Block Input	204
AES0DCFG	0xEA	0x2	AES0 Data Configuration	203
AES0KIN	0xED	0x2	AES0 Key Input	205
AES0XIN	0xEC	0x2	AES0 XOR Input	205
AES0YOUT	0xF5	0x2	AES Y Out	206
CKCON	0x8E	0x0	Clock Control	445
CLKMODE	0xFD	0xF	Clock Mode	262
CLKSEL	0xA9	0x0 and 0xF	Clock Select	291
CPT0CN	0x9B	0x0	Comparator0 Control	108
CPT0MD	0x9D	0x0	Comparator0 Mode Selection	109
CPT0MX	0x9F	0x0	Comparator0 Mux Selection	113
CPT1CN	0x9A	0x0	Comparator1 Control	110
CPT1MD	0x9C	0x0	Comparator1 Mode Selection	111
CPT1MX	0x9E	0x0	Comparator1 Mux Selection	114
CRC0AUTO	0x96	0xF	CRC0 Automatic Control	166
CRC0CNT	0x97	0xF	CRC0 Automatic Flash Sector Count	166
CRC0CN	0x92	0xF	CRC0 Control	164
CRC0DAT	0x91	0xF	CRC0 Data	165
CRC0FLIP	0x94	0xF	CRC0 Flip	167
CRC0IN	0x93	0xF	CRC0 Input	165

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
CRC1CN	0xBE	0x2	CRC1 Control	172
CRC1IN	0xB9	0x2	CRC1 In	173
CRC1OUTH	0xBB	0x2	CRC1 Out High	174
CRC1OUTL	0xBA	0x2	CRC1 Out Low	174
CRC1POLH	0xBD	0x2	CRC1 Polynomial High	173
CRC1POLL	0xBC	0x2	CRC1 Polynomial Low	173
DC0CF	0xB2	0x2	DC0 Configuration	274
DC0CN	0xB1	0x2	DC0 Control	273
DC0MD	0xB3	0x2	DC0 Mode	275
DC0RDY	0xFD	0x2	DC0 Ready	276
DEVICEID	0xE9	0xF	Device ID	249
DMA0BUSY	0xD5	0x2	DMA0 Busy	153
DMA0EN	0xD2	0x2	DMA0 Enable	150
DMA0INT	0xD3	0x2	DMA0 Interrupt	151
DMA0MINT	0xD4	0x2	DMA0 Middle Interrupt	152
DMA0NAOH	0xCD	0x2	DMA0 Address Offset High (Selected Channel)	158
DMA0NAOL	0xCC	0x2	DMA0 Address Offset Low (Selected Channel)	158
DMA0NBAH	0xCB	0x2	DMA0 Base Address High (Selected Channel)	157
DMA0NBAL	0xCA	0x2	DMA0 Base Address Low (Selected Channel)	157
DMA0NCF	0xC9	0x2	DMA0 Configuration	156
DMA0NMD	0xD6	0x2	DMA0 Mode (Selected Channel)	155
DMA0NSZH	0xCF	0x2	DMA0 Size High (Selected Channel)	159
DMA0NSZL	0xCE	0x2	DMA0 Size Low (Selected Channel)	159
DMA0SEL	0xD1	0x2	DMA0 Channel Select	154
DPH	0x83	All Pages	Data Pointer High	121
DPL	0x82	All Pages	Data Pointer Low	121
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	238
EIE2	0xE7	All Pages	Extended Interrupt Enable 2	240
EIP1	0xF6	All Pages	Extended Interrupt Priority 1	239
EIP2	0xF7	All Pages	Extended Interrupt Priority 2	241
EMI0CF	0xAB	0x0	EMIF Configuration	133
EMI0CN	0xAA	0x0	EMIF Control	132
EMI0TC	0xAF	0x0	EMIF Timing Control	138
ENC0CN	0xC5	0x2	ENC0 Control	214

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
ENC0H	0xC4	0x2	ENC0 High	215
ENC0L	0xC2	0x2	ENC0 Low	215
ENC0M	0xC3	0x2	ENC0 Middle	215
FLKEY	0xB7	All Pages	Flash Lock And Key	254
FLSCL	0xB6	0xF	Flash Scale Register	255
FLWR	0xE5	0x0	Flash Write Only	255
FRBCN	0xB5	0xF	Flash Read Buffer Control	256
IE	0xA8	All Pages	Interrupt Enable	236
IP	0xB8	All Pages	Interrupt Priority	237
IREF0CF	0xB9	0xF	Current Reference IREF0 Configuration	104
IREF0CN	0xB9	0x0	Current Reference IREF0 Configuration	103
IT01CF	0xE4	0x0 and 0xF	INT0/INT1 Configuration	243
LCD0BLINK	0x9E	0x2	LCD0 Blink Mask	346
LCD0BUFCF	0xAC	0xF	LCD0 Buffer Configuration	350
LCD0BUFCN	0x9C	0xF	LCD0 Buffer Control	349
LCD0BUFMD	0xB6	0x2	LCD0 Buffer Mode	350
LCD0CF	0xA5	0x2	LCD0 Configuration	348
LCD0CHPCF	0xAD	0x2	LCD0 Charge Pump Configuration	349
LCD0CHPCN	0xB5	0x2	LCD0 Charge Pump Control	348
LCD0CHPMD	0xAE	0x2	LCD0 Charge Pump Mode	349
LCD0CLKDIVH	0xAA	0x2	LCD0 Clock Divider High	345
LCD0CLKDIVL	0xA9	0x2	LCD0 Clock Divider Low	345
LCD0CN	0x9D	0x2	LCD0 Control	337
LCD0CNTRST	0x9C	0x2	LCD0 Contrast	341
LCD0D0	0x89	0x2	LCD0 Data 0	335
LCD0D1	0x8A	0x2	LCD0 Data 1	335
LCD0D2	0x8B	0x2	LCD0 Data 2	335
LCD0D3	0x8C	0x2	LCD0 Data 3	335
LCD0D4	0x8D	0x2	LCD0 Data 4	335
LCD0D5	0x8E	0x2	LCD0 Data 5	335
LCD0D6	0x91	0x2	LCD0 Data 6	335
LCD0D7	0x92	0x2	LCD0 Data 7	335
LCD0D8	0x93	0x2	LCD0 Data 8	335

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
LCD0D9	0x94	0x2	LCD0 Data 9	335
LCD0DA	0x95	0x2	LCD0 Data A	335
LCD0DB	0x96	0x2	LCD0 Data B	335
LCD0DC	0x97	0x2	LCD0 Data C	335
LCD0DD	0x99	0x2	LCD0 Data D	335
LCD0DE	0x9A	0x2	LCD0 Data E	335
LCD0DF	0x9B	0x2	LCD0 Data F	335
LCD0MSCF	0xAC	0x2	LCD0 Master Configuration	343
LCD0MSCN	0xAB	0x2	LCD0 Master Control	342
LCD0PWR	0xA4	0x2	LCD0 Power	343
LCD0TOGR	0x9F	0x2	LCD0 Toggle Rate	347
LCD0VBMCF	0xAF	0x2	LCD0 VBAT Monitor Configuration	350
LCD0VBMCN	0xA6	0x2	LCD0 VBAT Monitor Control	344
OSCICL	0xB3	0xF	Internal Oscillator Calibration	293
OSCICN	0xB2	0x0	Internal Oscillator Control	292
OSCXCN	0xB1	0x0	External Oscillator Control	294
P0DRV	0xA4	0xF	Port 0 Drive Strength	366
P0MASK	0xC7	0x0	Port 0 Mask	361
P0MAT	0xD7	0x0	Port 0 Match	361
P0MDIN	0xF1	0x0	Port 0 Input Mode Configuration	365
P0MDOUT	0xA4	0x0	Port 0 Output Mode Configuration	365
P0SKIP	0xD4	0x0	Port 0 Skip	364
P0	0x80	All Pages	Port 0 Latch	364
P1DRV	0xA5	0xF	Port 1 Drive Strength	368
P1MASK	0xBF	0x0	Port 1 Mask	362
P1MAT	0xCF	0x0	Port 1 Match	362
P1MDIN	0xF2	0x0	Port 1 Input Mode Configuration	367
P1MDOUT	0xA5	0x0	Port 1 Output Mode Configuration	368
P1SKIP	0xD5	0x0	Port 1 Skip	367
P1	0x90	All Pages	Port 1 Latch	366
P2DRV	0xA6	0xF	Port 2 Drive Strength	371
P2MDIN	0xF3	0x0	Port 2 Input Mode Configuration	370
P2MDOUT	0xA6	0x0	Port 2 Output Mode Configuration	370
P2SKIP	0xD6	0x0	Port 2 Skip	369

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
P2	0xA0	All Pages	Port 2 Latch	369
P3DRV	0xA1	0xF	Port 3 Drive Strength	373
P3MDIN	0xF1	0xF	Port 3 Input Mode Configuration	372
P3MDOUT	0xB1	0xF	P3 Mode Out	372
P3	0xB0	All Pages	Port 3	371
P4DRV	0xA2	0xF	Port 4 Drive Strength	375
P4MDIN	0xF2	0xF	Port 4 Input Mode Configuration	374
P4MDOUT	0xF9	0xF	P4 Mode Out	374
P4	0xD9	0xF	Port 4 Latch	373
P5DRV	0xA3	0xF	Port 5 Drive Strength	377
P5MDIN	0xF3	0xF	Port 5 Input Mode Configuration	376
P5MDOUT	0xFA	0xF	P5 Mode Out	376
P5	0xDA	0xF	Port 5 Latch	375
P6DRV	0xAA	0xF	Port 6 Drive Strength	379
P6MDIN	0xF4	0xF	Port 6 Input Mode Configuration	378
P6MDOUT	0xFB	0xF	P6 Mode Out	378
P6	0xDB	0xF	Port 6 Latch	377
P7DRV	0xAB	0xF	Port 7 Drive Strength	380
P7MDOUT	0xFC	0xF	P7 Mode Out	380
P7	0xDC	0xF	Port 7 Latch	379
PC0CMP0H	0xE3	0x2	PC0 Comparator 0 High	329
PC0CMP0L	0xE1	0x2	PC0 Comparator 0 Low	329
PC0CMP0M	0xE2	0x2	PC0 Comparator 0 Middle	329
PC0CMP1H	0xF3	0x2	PC0 Comparator 1 High	330
PC0CMP1L	0xF1	0x2	PC0 Comparator 1 Low	330
PC0CMP1M	0xF2	0x2	PC0 Comparator 1 Middle	330
PC0CTR0H	0xDC	0x2	PC0 Counter 0 High	327
PC0CTR0L	0xDA	0x2	PC0 Counter 0 Low	327
PC0CTR0M	0xD8	0x2	PC0 Counter 0 Middle	327
PC0CTR1H	0xDF	0x2	PC0 Counter 1 High	328
PC0CTR1L	0xDD	0x2	PC0 Counter 1 Low	328
PC0DCH	0xFA	0x2	PC0 Debounce Configuration High	325
PC0DCL	0xF9	0x2	PC0 Debounce Configuration Low	326
PC0HIST	0xF4	0x2	PC0 History	331

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
PC0INT0	0xFB	0x2	PC0 Interrupt 0	332
PC0INT1	0xFC	0x2	PC0 Interrupt 1	333
PC0MD	0xD9	0x2	PC0 Mode	321
PC0PCF	0xD7	0x2	PC0 Pull-up Configuration	322
PC0STAT	0xC1	0x2	PC0 Status	324
PC0TH	0xE4	0x2	PC0 Threshold	323
PCA0CN	0xD8	All Pages	PCA0 Control	480
PCA0CPH0	0xFC	0x0	PCA0 Capture 0 High	485
PCA0CPH1	0xEA	0x0	PCA0 Capture 1 High	485
PCA0CPH2	0xEC	0x0	PCA0 Capture 2 High	485
PCA0CPH3	0xEE	0x0	PCA0 Capture 3 High	485
PCA0CPH4	0xFE	0x0	PCA0 Capture 4 High	485
PCA0CPH5	0xD3	0x0	PCA0 Capture 5 High	485
PCA0CPL0	0xFB	0x0	PCA0 Capture 0 Low	485
PCA0CPL1	0xE9	0x0	PCA0 Capture 1 Low	485
PCA0CPL2	0xEB	0x0	PCA0 Capture 2 Low	485
PCA0CPL3	0xED	0x0	PCA0 Capture 3 Low	485
PCA0CPL4	0xFD	0x0	PCA0 Capture 4 Low	485
PCA0CPL5	0xD2	0x0	PCA0 Capture 5 Low	485
PCA0CPM0	0xDA	0x0	PCA0 Module 0 Mode Register	483
PCA0CPM1	0xDB	0x0	PCA0 Module 1 Mode Register	483
PCA0CPM2	0xDC	0x0	PCA0 Module 2 Mode Register	483
PCA0CPM3	0xDD	0x0	PCA0 Module 3 Mode Register	483
PCA0CPM4	0xDE	0x0	PCA0 Module 4 Mode Register	483
PCA0CPM5	0xCE	0x0	PCA0 Module 5 Mode Register	483
PCA0H		0x0	PCA0 Counter High	484
PCA0L	0xF9	0x0	PCA0 Counter Low	484
PCA0MD	0xD9	0x0	PCA0 Mode	481
PCA0PWM	0xDF	0x0	PCA0 PWM Configuration	482
PCLKACT	0xF5	0xF	Peripheral Clock Enable Active Mode	260
PCLKEN	0xFE	0xF	Peripheral Clock Enables (LP Idle)	261
PCON	0x87	All Pages	Power Control	268
PMU0CF	0xB5	0x0	PMU0 Configuration 0	265
PMU0FL	0xB6	0x0	PMU0 flag	266

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
PMU0MD	0xB3	0x0	Internal Oscillator Calibration	267
PSBANK	0x84	All Pages	Flash Page Switch Bank SFR	127
PSCTL	0x8F	All Pages	Program Store R/W Control	253
PSW	0xD0	All Pages	Program Status Word	123
REF0CN	0xD1	0x0	Voltage Reference Control	102
REG0CN	0xC9	0x0	Voltage Regulator (REG0) Control	277
REVID	0xEA	0xF	Revision ID	249
RSTSRC	0xEF	0x0	Reset Source Configuration/Status	285
RTC0ADR	0xAC	0x0	RTC0 Address	298
RTC0DAT	0xAD	0x0	RTC0 Data	299
RTC0KEY	0xAE	0x0	RTC0 Key	298
SBUF0	0x99	0x0	UART0 Data Buffer	408
SCON0	0x98	All Pages	UART0 Control	407
SFRLAST	0x86	All Pages	SFR Page Stack Last	221
SFRNEXT	0x85	All Pages	SFR Page Stack Next	220
SFRPAGE	0xA7	All Pages	SFR Page	219
SFRPGCN	0x8E	0xF	SFR Page Control	218
SMB0ADM	0xF5	0x0	SMBus Slave Address Mask	392
SMB0ADR	0xF4	0x0	SMBus Slave Address	391
SMB0CF	0xC1	0x0	SMBus0 Configuration	387
SMB0CN	0xC0	All Pages	SMBus0 Control	389
SMB0DAT	0xC2	0x0	SMBus0 Data	393
SPI0CFG	0xA1	0x0	SPI0 Configuration	418
SPI0CKR	0xA2	0x0	SPI0 Clock Rate Control	420
SPI0CN	0xF8	0x0	SPI0 Control	419
SPI0DAT	0xA3	0x0	SPI0 Data	420
SPI1CFG	0xA1	0x2	SPI1 Configuration	438
SPI1CKR	0xA2	0x2	SPI1 Clock Rate Control	440
SPI1CN	0xF8	0x2	SPI1 Control	439
SPI1DAT	0xA3	0x2	SPI1 Data	440
SP	0x81	All Pages	Stack Pointer	122
TCON	0x88	All Pages	Timer/Counter Control	450
TH0	0x8C	0x0	Timer/Counter 0 High	453
TH1	0x8D	0x0	Timer/Counter 1 High	453

Table 16.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
TL0	0x8A	0x0	Timer/Counter 0 Low	452
TL1	0x8B	0x0	Timer/Counter 1 Low	452
TMOD	0x89	0x0	Timer/Counter Mode	451
TMR2CN	0xC8	All Pages	Timer/Counter 2 Control	457
TMR2H	0xCD	0x0	Timer/Counter 2 High	459
TMR2L	0xCC	0x0	Timer/Counter 2 Low	459
TMR2RLH	0xCB	0x0	Timer/Counter 2 Reload High	458
TMR2RLL	0xCA	0x0	Timer/Counter 2 Reload Low	458
TMR3CN	0x91	0x0	Timer/Counter 3 Control	463
TMR3H	0x95	0x0	Timer/Counter 3 High	465
TMR3L	0x94	0x0	Timer/Counter 3 Low	465
TMR3RLH	0x93	0x0	Timer/Counter 3 Reload High	464
TMR3RLL	0x92	0x0	Timer/Counter 3 Reload Low	464
TOFFH	0xBB	0xF	Temperature Offset High	99
TOFFL	0xBD	0xF	Temperature Offset Low	99
VDM0CN	0xFF	All Pages	VDD Monitor Control	282
XBR0	0xE1	0x0 and 0xF	Port I/O Crossbar Control 0	358
XBR1	0xE2	0x0 and 0xF	Port I/O Crossbar Control 1	359
XBR2	0xE3	0x0 and 0xF	Port I/O Crossbar Control 2	360

17. Interrupt Handler

The C8051F96x microcontroller family includes an extended interrupt system supporting multiple interrupt sources and two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Refer to Table 17.1, “Interrupt Summary,” on page 234 for a detailed listing of all interrupt sources supported by the device. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR or an indirect register. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1. If both global interrupts and the specific interrupt source is enabled, a CPU interrupt request is generated when the interrupt-pending flag is set.

As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Some interrupt-pending flags are automatically cleared by hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

17.1. Enabling Interrupt Sources

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable and Extended Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic 0 will be held in a pending state, and will not be serviced until the EA bit is set back to logic 1.

17.2. MCU Interrupt Sources and Vectors

The CPU services interrupts by generating an LCALL to a predetermined address (the interrupt vector address) to begin execution of an interrupt service routine (ISR). The interrupt vector addresses associated with each interrupt source are listed in Table 17.1 on page 234. Software should ensure that the interrupt vector for each enabled interrupt source contains a valid interrupt service routine.

Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag.

17.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. If a high priority interrupt preempts a low priority interrupt, the low priority interrupt will finish execution after the high priority interrupt completes. Each interrupt has an associated interrupt priority bit in in the Interrupt Priority and Extended Interrupt Priority registers used to configure its priority level. Low priority is the default.

If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate. See Table 17.1 on page 234 to determine the fixed priority order used to arbitrate between simultaneously recognized interrupts.

17.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 7 system clock cycles: 1 clock cycle to detect the interrupt, 1 clock cycle to execute a single instruction, and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 19 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

Table 17.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 ($\overline{INT0}$)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 ($\overline{INT1}$)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
SmaRTClock Alarm	0x0043	8	ALRM (RTC0CN.2)*	N	N	EARTC0 (EIE1.1)	PARTC0 (EIP1.1)
ADC0 Window Comparator	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 End of Conversion	0x0053	10	AD0INT (ADC0STA.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
Comparator1	0x006B	13	CP1FIF (CPT1CN.4) CP1RIF (CPT1CN.5)	N	N	ECP1 (EIE1.6)	PCP1 (EIP1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
VDD/VBAT Supply Monitor Early Warning	0x007B	15	VDDOK (VDM0CN.5)[1] VBOK (VDM0CN.2)[1]			EWARN (EIE2.0)	PWARN (EIP2.0)
Port Match	0x0083	16	None			EMAT (EIE2.1)	PMAT (EIP2.1)

Table 17.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
SmaRTClock Oscillator Fail	0x008B	17	OSCFAIL (RTC0CN.5)[2]	N	N	ERTC0F (EIE2.2)	PFRTC0F (EIP2.2)
SPI1	0x0093	18	SPIF (SPI1CN.7) WCOL (SPI1CN.6) MODF (SPI1CN.5) RXOVRN (SPI1CN.4)	N	N	ESPI1 (EIE2.3)	PSPI1 (EIP2.3)
Pulse Counter	0x009B	19	C0ZF (PC0CN.4) C1ZF (PC0CN.6)	N	N	EPC0 (EIE2.4)	PPC0 (EIP2.4)
DMA0	0x00A3	20	DMAINT0...7 DMAMINT0...7	N	N	EDMA0 (EIE2.5)	PDMA0 (EIP2.5)
Encoder0	0x00AB	21	ENCERR(ENCCN.6)	N	N	EENC0 (EIE2.6)	PENC0 (EIP2.6)
AES	0x00B3	22	AESDONE (AESBCF.5)	N	N	EAES0 (EIE2.7)	PAES0 (EIP2.7)

Notes:
1. Indicates a read-only interrupt pending flag. The interrupt enable may be used to prevent software from vectoring to the associated interrupt service routine.
2. Indicates a register located in an indirect memory space.

17.5. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in the following register descriptions. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 17.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xA8; Bit-Addressable

Bit	Name	Function
7	EA	**Enable All Interrupts.** Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	**Enable Serial Peripheral Interface (SPI0) Interrupt.** This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	**Enable Timer 2 Interrupt.** This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	**Enable UART0 Interrupt.** This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	**Enable Timer 1 Interrupt.** This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	**Enable External Interrupt 1.** This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	**Enable Timer 0 Interrupt.** This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	**Enable External Interrupt 0.** This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

SFR Definition 17.2. IP: Interrupt Priority

Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xB8; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1b, Write = don't care.
6	PSPI0	**Serial Peripheral Interface (SPI0) Interrupt Priority Control.** This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	**Timer 2 Interrupt Priority Control.** This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	**UART0 Interrupt Priority Control.** This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	**Timer 1 Interrupt Priority Control.** This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	**External Interrupt 1 Priority Control.** This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	**Timer 0 Interrupt Priority Control.** This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	**External Interrupt 0 Priority Control.** This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

SFR Definition 17.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	ERTC0A	ESMB0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xE6

Bit	Name	Function
7	ET3	**Enable Timer 3 Interrupt.** This bit sets the masking of the Timer 3 interrupt. 0: Disable Timer 3 interrupts. 1: Enable interrupt requests generated by the TF3L or TF3H flags.
6	ECP1	**Enable Comparator1 (CP1) Interrupt.** This bit sets the masking of the CP1 interrupt. 0: Disable CP1 interrupts. 1: Enable interrupt requests generated by the CP1RIF or CP1FIF flags.
5	ECP0	**Enable Comparator0 (CP0) Interrupt.** This bit sets the masking of the CP0 interrupt. 0: Disable CP0 interrupts. 1: Enable interrupt requests generated by the CP0RIF or CP0FIF flags.
4	EPCA0	**Enable Programmable Counter Array (PCA0) Interrupt.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3	EADC0	**Enable ADC0 Conversion Complete Interrupt.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the AD0INT flag.
2	EWADC0	**Enable Window Comparison ADC0 Interrupt.** This bit sets the masking of ADC0 Window Comparison interrupt. 0: Disable ADC0 Window Comparison interrupt. 1: Enable interrupt requests generated by ADC0 Window Compare flag (AD0WINT).
1	ERTC0A	**Enable SmaRTClock Alarm Interrupts.** This bit sets the masking of the SmaRTClock Alarm interrupt. 0: Disable SmaRTClock Alarm interrupts. 1: Enable interrupt requests generated by a SmaRTClock Alarm.
0	ESMB0	**Enable SMBus (SMB0) Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

SFR Definition 17.4. EIP1: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PRTC0A	PSMB0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xF6

Bit	Name	Function
7	PT3	**Timer 3 Interrupt Priority Control.** This bit sets the priority of the Timer 3 interrupt. 0: Timer 3 interrupts set to low priority level. 1: Timer 3 interrupts set to high priority level.
6	PCP1	**Comparator1 (CP1) Interrupt Priority Control.** This bit sets the priority of the CP1 interrupt. 0: CP1 interrupt set to low priority level. 1: CP1 interrupt set to high priority level.
5	PCP0	**Comparator0 (CP0) Interrupt Priority Control.** This bit sets the priority of the CP0 interrupt. 0: CP0 interrupt set to low priority level. 1: CP0 interrupt set to high priority level.
4	PPCA0	**Programmable Counter Array (PCA0) Interrupt Priority Control.** This bit sets the priority of the PCA0 interrupt. 0: PCA0 interrupt set to low priority level. 1: PCA0 interrupt set to high priority level.
3	PADC0	**ADC0 Conversion Complete Interrupt Priority Control.** This bit sets the priority of the ADC0 Conversion Complete interrupt. 0: ADC0 Conversion Complete interrupt set to low priority level. 1: ADC0 Conversion Complete interrupt set to high priority level.
2	PWADC0	**ADC0 Window Comparator Interrupt Priority Control.** This bit sets the priority of the ADC0 Window interrupt. 0: ADC0 Window interrupt set to low priority level. 1: ADC0 Window interrupt set to high priority level.
1	PRTC0A	**SmaRTClock Alarm Interrupt Priority Control.** This bit sets the priority of the SmaRTClock Alarm interrupt. 0: SmaRTClock Alarm interrupt set to low priority level. 1: SmaRTClock Alarm interrupt set to high priority level.
0	PSMB0	**SMBus (SMB0) Interrupt Priority Control.** This bit sets the priority of the SMB0 interrupt. 0: SMB0 interrupt set to low priority level. 1: SMB0 interrupt set to high priority level.

SFR Definition 17.5. EIE2: Extended Interrupt Enable 2

Bit	7	6	5	4	3	2	1	0
Name	EAES0	EENC0	EDMA0	EPC0	ESPI1	ERTC0F	EMAT	EWARN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages;SFR Address = 0xE7

Bit	Name	Function
7	EAES0	**Enable AES0 Interrupt.** This bit sets the masking of AES0 interrupts. 0: Disable all AES0 interrupts. 1: Enable interrupt requests generated by AES0.
6	EENC0	**Enable Encoder (ENC0) Interrupt.** This bit sets the masking of ENC0 interrupts. 0: Disable all ENC0 interrupts. 1: Enable interrupt requests generated by ENC0.
5	EDMA0	**Enable DMA0 Interrupt.** This bit sets the masking of DMA0 interrupts. 0: Disable all DMA0 interrupts. 1: Enable interrupt requests generated by DMA0.
4	EPC0	**Enable Pulse Counter (PC0) Interrupt.** This bit sets the masking of PC0 interrupts. 0: Disable all PC0 interrupts. 1: Enable interrupt requests generated by PC0.
3	ESPI1	**Enable Serial Peripheral Interface (SPI1) Interrupt.** This bit sets the masking of the SPI1 interrupts. 0: Disable all SPI1 interrupts. 1: Enable interrupt requests generated by SPI1.
2	ERTC0F	**Enable SmaRTClock Oscillator Fail Interrupt.** This bit sets the masking of the SmaRTClock Alarm interrupt. 0: Disable SmaRTClock Alarm interrupts. 1: Enable interrupt requests generated by SmaRTClock Alarm.
1	EMAT	**Enable Port Match Interrupts.** This bit sets the masking of the Port Match Event interrupt. 0: Disable all Port Match interrupts. 1: Enable interrupt requests generated by a Port Match.
0	EWARN	**Enable VDD/DC+ Supply Monitor Early Warning Interrupt.** This bit sets the masking of the VDD/DC+ Supply Monitor Early Warning interrupt. 0: Disable the VDD/DC+ Supply Monitor Early Warning interrupt. 1: Enable interrupt requests generated by VDD/DC+ Supply Monitor.

SFR Definition 17.6. EIP2: Extended Interrupt Priority 2

Bit	7	6	5	4	3	2	1	0
Name	PAES0	PENC0	PDMA0	PPC0	PSPI1	PRTC0F	PMAT	PWARN
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xF7

Bit	Name	Function
7	PAES0	**AES0 Interrupt Priority Control.** This bit sets the priority of the AES0 interrupt. 0: AES0 interrupt set to low priority level. 1: AES0 interrupt set to high priority level.
6	PENC0	**Encoder (ENC0) Interrupt Priority Control.** This bit sets the priority of the ENC0 interrupt. 0: ENC0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PDMA0	**DMA0 Interrupt Priority Control.** This bit sets the priority of the DMA0 interrupt. 0: DMA0 interrupt set to low priority level. 1: DMA0 interrupt set to high priority level.
4	PPC0	**Pulse Counter (PC0) Interrupt Priority Control.** This bit sets the priority of the PC0 interrupt. 0: PC0 interrupt set to low priority level. 1: PC0 interrupt set to high priority level.
3	PSPI0	**Serial Peripheral Interface (SPI1) Interrupt Priority Control.** This bit sets the priority of the SPI0 interrupt. 0: SPI1 interrupt set to low priority level. 1: SPI1 interrupt set to high priority level.
2	PRTC0F	**SmaRTClock Oscillator Fail Interrupt Priority Control.** This bit sets the priority of the SmaRTClock Alarm interrupt. 0: SmaRTClock Alarm interrupt set to low priority level. 1: SmaRTClock Alarm interrupt set to high priority level.
1	PMAT	**Port Match Interrupt Priority Control.** This bit sets the priority of the Port Match Event interrupt. 0: Port Match interrupt set to low priority level. 1: Port Match interrupt set to high priority level.
0	PWARN	**VDD/DC+ Supply Monitor Early Warning Interrupt Priority Control.** This bit sets the priority of the VDD/DC+ Supply Monitor Early Warning interrupt. 0: VDD/DC+ Supply Monitor Early Warning interrupt set to low priority level. 1: VDD/DC+ Supply Monitor Early Warning interrupt set to high priority level.

17.6. External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “32.1. Timer 0 and Timer 1” on page 446) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 17.7). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT0}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “27.3. Priority Crossbar Decoder” on page 355 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

SFR Definition 17.7. IT01CF: $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Page = 0x0 and 0xF; SFR Address = 0xE4

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P1.6 111: Select P1.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P1.6 111: Select P1.7

18. Flash Memory

On-chip, re-programmable flash memory is included for program code and non-volatile data storage. The flash memory can be programmed in-system, a single byte at a time, through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during flash write/erase operations. Refer to Table 4.8 for complete flash memory electrical characteristics.

18.1. Programming the Flash Memory

The simplest means of programming the flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program flash memory, see Section “34. C2 Interface” on page 486.

The flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before programming flash memory using MOVX, flash programming operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target flash memory); and (2) Writing the flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software. For detailed guidelines on programming flash from firmware, please see Section “18.5. Flash Write and Erase Guidelines” on page 250.

To ensure the integrity of the flash contents, the on-chip VDD Monitor must be enabled and enabled as a reset source in any system that includes code that writes and/or erases flash memory from software. Furthermore, there should be no delay between enabling the V_{DD} Monitor and enabling the V_{DD} Monitor as a reset source. Any attempt to write or erase flash memory while the V_{DD} Monitor is disabled, or not enabled as a reset source, will cause a Flash Error device reset.

18.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a flash write or erase is attempted before the key codes have been written properly. The flash lock resets after each write or erase; the key codes must be written again before a following flash operation can be performed. The FLKEY register is detailed in SFR Definition 18.4.

18.1.2. Flash Erase Procedure

The flash memory is organized in 1024-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 1024-byte page, perform the following steps:

1. Save current interrupt state and disable interrupts.
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank.
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a data byte to any location within the 1024-byte page to be erased.
8. Clear the PSWE and PSEE bits.

9. Restore previous interrupt state.

Steps 4–7 must be repeated for each 1024-byte page to be erased.

Notes:

1. Flash security settings may prevent erasure of some flash pages, such as the reserved area and the page containing the lock bytes. For a summary of flash security settings and restrictions affecting flash erase operations, please see Section “18.3. Security Options” on page 247.
2. 8-bit MOVX instructions cannot be used to erase or write to flash memory at addresses higher than 0x00FF.

18.1.3. Flash Write Procedure

A write to flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in flash. **A byte location to be programmed should be erased before a new value is written.**

The recommended procedure for writing a single byte in flash is as follows:

1. Save current interrupt state and disable interrupts.
2. Set the PSWE bit (register PSCTL).
3. Clear the PSEE bit (register PSCTL).
4. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank.
5. Ensure that the flash byte has been erased (has a value of 0xFF).
6. Write the first key code to FLKEY: 0xA5.
7. Write the second key code to FLKEY: 0xF1.
8. Using the MOVX instruction, write a single data byte to the desired location within the 1024-byte sector.
9. Clear the PSWE bit.
10. Restore previous interrupt state.

Steps 2–8 must be repeated for each byte to be written.

Notes:

1. Flash security settings may prevent writes to some areas of flash, such as the reserved area. For a summary of flash security settings and restrictions affecting flash write operations, please see Section “18.3. Security Options” on page 247.
2. 8-bit MOVX instructions cannot be used to erase or write to flash memory at addresses higher than 0x00FF.

18.1.4. Flash Write Optimization

The flash write procedure includes a block write option to optimize the time to perform consecutive byte writes. When block write is enabled by setting the CHBLKW bit (FLRBCN.0), writes to flash will occur in blocks of 4 bytes and require the same amount of time as a single byte write. This is performed by caching the bytes whose address end in 00b, 01b, and 10b that is written to flash and then committing all four bytes to flash when the byte with address 11b is written. When block writes are enabled, if the write to the byte with address 11b does not occur, the other three data bytes written is not committed to flash.

A write to flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in flash. **The Flash Block to be programmed should be erased before a new value is written.**

The recommended procedure for writing a 4-byte flash block is as follows:

1. Save current interrupt state and disable interrupts.
2. Set the CHBLKW bit (register FLRBCN).
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank
6. Write the first key code to FLKEY: 0xA5.
7. Write the second key code to FLKEY: 0xF1.
8. Using the MOVX instruction, write the first data byte to the desired location within the 1024-byte sector whose address ends in 00b.
9. Write the first key code to FLKEY: 0xA5.
10. Write the second key code to FLKEY: 0xF1.
11. Using the MOVX instruction, write the second data byte to the next higher flash address ending in 01b.
12. Write the first key code to FLKEY: 0xA5.
13. Write the second key code to FLKEY: 0xF1.
14. Using the MOVX instruction, write the third data byte to the next higher flash address ending in 10b.
15. Write the first key code to FLKEY: 0xA5.
16. Write the second key code to FLKEY: 0xF1.
17. Using the MOVX instruction, write the final data byte to the next higher flash address ending in 11b.
18. Clear the PSWE bit.
19. Clear the CHBLKW bit.
20. Restore previous interrupt state.

Steps 5–17 must be repeated for each flash block to be written.

Notes:

1. Flash security settings may prevent writes to some areas of flash, such as the reserved area. For a summary of flash security settings and restrictions affecting flash write operations, please see Section “18.3. Security Options” on page 247.
2. 8-bit MOVX instructions cannot be used to erase or write to flash memory at addresses higher than 0x00FF.

18.2. Non-volatile Data Storage

The flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

18.3. Security Options

The CIP-51 provides security options to protect the flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the flash memory; both PSWE and PSEE must be set to 1 before software can erase flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of flash user space offers protection of the flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The flash security mechanism allows the user to lock *n* 1024-byte flash pages, starting at page 0 (addresses 0x0000 to 0x03FF), where *n* is the 1s complement number represented by the Security Lock Byte. **The page containing the Flash Security Lock Byte is unlocked when no other flash pages are locked (all bits of the Lock Byte are 1) and locked when any other flash pages are locked (any bit of the Lock Byte is 0).** See example in Figure 18.1

The 128 kB flash devices (C8051F960/1/2/3) do not have a reserved area. The lock byte is at the top of the flash area (0x1FFFF). Writing 0x80 to the lock byte of the 128 kB devices will lock the entire flash. *Writing 0x00 to the 128 kB flash devices will prevent future erases.*

Security Lock Byte:	11111101b
ones Complement:	00000010b
Flash pages locked:	3 (First two flash pages + Lock Byte Page)

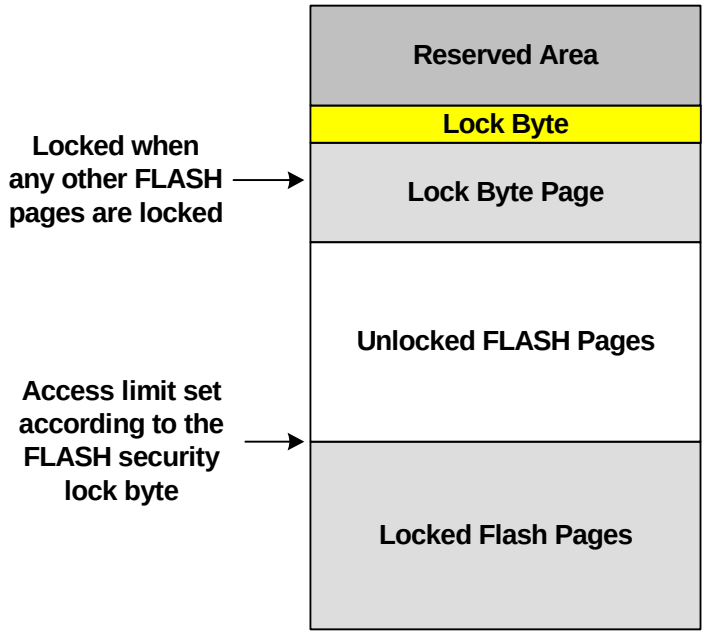


Figure 18.1. Flash Security Example

The level of flash security depends on the flash access method. The three flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 18.1 summarizes the flash security features of the C8051F96x devices.

Table 18.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	Flash Error Reset	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	Flash Error Reset	Flash Error Reset
Erase page containing Lock Byte—Unlock all pages (if any page is locked)	C2 Device Erase Only	Flash Error Reset	Flash Error Reset
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Read, Write or Erase Reserved Area	Not Permitted	Flash Error Reset	Flash Error Reset

C2 Device Erase—Erases all flash pages including the page containing the Lock Byte.

Flash Error Reset—Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is '1' after reset).

- All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).

- Locking any flash page also locks the page containing the Lock Byte.

- Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.

- If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.

18.4. Determining the Device Part Number at Run Time

In many applications, user software may need to determine the MCU part number at run time in order to determine the hardware capabilities. The part number can be determined by reading the value of the DEVICEID Special Function Register.

The value of the DEVICEID register can be decoded as follows:

0xD0—C8051F960
0xD1—C8051F961
0xD2—C8051F962
0xD3—C8051F963
0xD4—C8051F964
0xD5—C8051F965
0xD6—C8051F966
0xD7—C8051F967
0xD8—C8051F968

SFR Definition 18.1. DEVICEID: Device Identification

Bit	7	6	5	4	3	2	1	0
Name	DEVICEID[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xE9

Bit	Name	Function
7:0	DEVICEID[7:0]	**Device Identification.** These bits contain a value that can be decoded to determine the device part number.

SFR Definition 18.2. REVID: Revision Identification

Bit	7	6	5	4	3	2	1	0
Name	REVID[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xEA

Bit	Name	Function
7:0	REVID[7:0]	**Revision Identification.** These bits contain a value that can be decoded to determine the silicon revision. For example, 0x00 for Rev A and 0x01 for Rev B, etc.

18.5. Flash Write and Erase Guidelines

Any system which contains routines which write or erase flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of VDD, system clock frequency, or temperature. This accidental execution of flash modifying code can result in alteration of flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

To help prevent the accidental modification of flash by firmware, the VDD Monitor must be enabled and enabled as a reset source on C8051F96x devices for the flash to be successfully modified. **If either the VDD Monitor or the VDD Monitor reset source is not enabled, a Flash Error Device Reset will be generated when the firmware attempts to modify the flash.**

The following guidelines are recommended for any system that contains routines which write or erase flash from code.

18.5.1. VDD Maintenance and the VDD Monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum VDD rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external VDD brownout circuit to the $\overline{\text{RST}}$ pin of the device that holds the device in reset until VDD reaches the minimum device operating voltage and re-asserts $\overline{\text{RST}}$ if VDD drops below the minimum device operating voltage.
3. Keep the on-chip VDD Monitor enabled and enable the VDD Monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the 'C' compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware," available from the Silicon Laboratories web site.

Notes:
On C8051F96x devices, both the VDD Monitor and the VDD Monitor reset source must be enabled to write or erase flash without generating a Flash Error Device Reset.

On C8051F96x devices, both the VDD Monitor and the VDD Monitor reset source are enabled by hardware after a power-on reset.

4. As an added precaution, explicitly enable the VDD Monitor and enable the VDD Monitor as a reset source inside the functions that write and erase flash memory. The VDD Monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct, but "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a '1'. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

18.5.2. PSWE Maintenance

1. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write flash bytes and one routine in code that sets both PSWE and PSEE both to a 1 to erase flash pages.
2. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in “AN201: Writing to Flash from Firmware,” available from the Silicon Laboratories web site.
3. Disable interrupts prior to setting PSWE to a 1 and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the flash write or erase operation will be serviced in priority order after the flash operation has been completed and interrupts have been re-enabled by software.
4. Make certain that the flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
5. Add address bounds checking to the routines that write or erase flash memory to ensure that a routine called with an illegal address does not result in modification of the flash.

18.5.3. System Clock

1. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
2. If operating from the external oscillator, switch to the internal oscillator during flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the flash operation has completed.

Additional flash recommendations and example code can be found in “AN201: Writing to Flash from Firmware," available from the Silicon Laboratories website.

18.6. Minimizing Flash Read Current

The flash memory in the C8051F96x devices is responsible for a substantial portion of the total digital supply current when the device is executing code. Below are suggestions to minimize flash read current.

1. Use idle, low power idle, suspend, or sleep modes while waiting for an interrupt, rather than polling the interrupt flag. Idle mode and low power idle mode is particularly well-suited for use in implementing short pauses, since the wake-up time is no more than three system clock cycles. See the Power Management chapter for details on the various low-power operating modes.
2. The flash memory is organized in 4-byte words starting with a byte with address ending in 00b and ending with a byte with address ending in 11b. A 4-byte pre-fetch buffer is used to read 4 bytes of flash in a single read operation. Short loops that straddle word boundaries or have an instruction byte with address ending in 11b should be avoided when possible. If a loop executes in 20 or more clock cycles, any resulting increase in operating current due to mis-alignment will be negligible.
3. To minimize the power consumption of small loops, it is best to locate them such that the number of 4-byte words to be fetched from flash is minimized. Consider a 2-byte, 3-cycle loop (e.g., SJMP $, or while(1);). The flash read current of such a loop will be minimized if both address bytes are contained in the first 3 bytes of a single 4-byte word. Such a loop should be manually located at an address ending in 00b or the number of bytes in the loop should be increased (by padding with NOP instructions) in order to minimize flash read current.

SFR Definition 18.3. PSCTL: Program Store R/W Control

Bit	7	6	5	4	3	2	1	0
Name							PSEE	PSWE
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page =All Pages; SFR Address = 0x8F

Bit	Name	Function
7:2	Unused	Read = 000000b, Write = don't care.
1	PSEE	**Program Store Erase Enable.** Setting this bit (in combination with PSWE) allows an entire page of flash program memory to be erased. If this bit is logic 1 and flash writes are enabled (PSWE is logic 1), a write to flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter. 0: Flash program memory erasure disabled. 1: Flash program memory erasure enabled.
0	PSWE	**Program Store Write Enable.** Setting this bit allows writing a byte of data to the flash program memory using the MOVX write instruction. The flash location should be erased before writing data. 0: Writes to flash program memory disabled. 1: Writes to flash program memory enabled; the MOVX write instruction targets flash memory.

SFR Definition 18.4. FLKEY: Flash Lock and Key

Bit	7	6	5	4	3	2	1	0
Name	FLKEY[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xB7

Bit	Name	Function
7:0	FLKEY[7:0]	**Flash Lock and Key Register.** Write: This register provides a lock and key function for flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a flash write or erase operation is attempted while these operations are disabled, the flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to flash, it can intentionally lock the flash by writing a non-0xA5 value to FLKEY from software. Read: When read, bits 1–0 indicate the current flash lock state. 00: Flash is write/erase locked. 01: The first key code has been written (0xA5). 10: Flash is unlocked (writes/erases allowed). 11: Flash writes/erases disabled until the next reset.

SFR Definition 18.5. FLSCL: Flash Scale

Bit	7	6	5	4	3	2	1	0
Name		BYPASS						
Type	R	R/W	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xB7

Bit	Name	Function
7	Reserved	Always Write to 0.
6	BYPASS	**Flash Read Timing One-Shot Bypass.** 0: The one-shot determines the flash read time. 1: The system clock determines the flash read time. Leaving the one-shot enabled will provide the lowest power consumption up to 25 MHz.
5:0	Reserved	Always Write to 000000.

Note: Operations which clear the BYPASS bit do not need to be immediately followed by a benign 3-byte instruction. For code compatibility with C8051F930/31/20/21 devices, a benign 3-byte instruction whose third byte is a don't care should follow the clear operation. See the C8051F93x-C8051F92x data sheet for more details.

SFR Definition 18.6. FLWR: Flash Write Only

Bit	7	6	5	4	3	2	1	0
Name	FLWR[7:0]							
Type	W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xE5

Bit	Name	Function
7:0	FLWR[7:0]	**Flash Write Only.** All writes to this register have no effect on system operation.

SFR Definition 18.7. FRBCN: Flash Read Buffer Control

Bit	7	6	5	4	3	2	1	0
Name							FRBD	CHBLKW
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	1	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xB5

Bit	Name	Function
7:2	Unused	Read = 000000b. Write = don't care.
1	FRBD	**Flash Read Buffer Disable Bit.** 0: Flash read buffer is enabled and being used. 1: Flash read buffer is disabled and bypassed.
0	CHBLKW	**Block Write Enable Bit.** This bit allows block writes to flash memory from firmware. 0: Each byte of a software flash write is written individually. 1: Flash bytes are written in groups of four.

19. Power Management

C8051F96x devices support 6 power modes: Normal, Idle, Stop, Low Power Idle, Suspend, and Sleep. The power management unit (PMU0) allows the device to enter and wake-up from the available power modes. A brief description of each power mode is provided in Table 19.1. Detailed descriptions of each mode can be found in the following sections.

Table 19.1. Power Modes

Power Mode	Description	Wake-Up Sources	Power Savings
Normal	Device fully functional	N/A	Excellent MIPS/mW
Idle	All peripherals fully functional. Very easy to wake up.	Any Interrupt.	Good No Code Execution
Stop	Legacy 8051 low power mode. A reset is required to wake up.	Any Reset.	Good No Code Execution Precision Oscillator Disabled
Low Power Idle	Improved Idle mode that uses clock gating to save power.	Any Interrupt	Very Good No Code Execution Selective Clock Gating
Suspend	Similar to Stop Mode, but very fast wake-up time and code resumes execution at the next instruction.	SmaRTClock, Port Match, Comparator0, $\overline{\text{RST}}$ pin, Pulse Counter VBAT Monitor.	Very Good No Code Execution All Internal Oscillators Disabled System Clock Gated
Sleep	Ultra Low Power and flexible wake-up sources. Code resumes execution at the next instruction.	SmaRTClock, Port Match, Comparator0, $\overline{\text{RST}}$ pin, Pulse Counter VBAT Monitor.	Excellent Power Supply Gated All Oscillators except SmaRT-Clock Disabled

In battery powered systems, the system should spend as much time as possible in sleep mode in order to preserve battery life. When a task with a fixed number of clock cycles needs to be performed, the device should switch to normal mode, finish the task as quickly as possible, and return to sleep mode. Idle mode, low power idle mode, and suspend mode provide a very fast wake-up time; however, the power savings in these modes will not be as much as in sleep Mode. Stop Mode is included for legacy reasons; the system will be more power efficient and easier to wake up when idle, low power idle, suspend, or sleep mode is used.

Although switching power modes is an integral part of power management, enabling/disabling individual peripherals as needed will help lower power consumption in all power modes. Each analog peripheral can be disabled when not in use or placed in a low power mode. Digital peripherals such as timers or serial busses draw little power whenever they are not in use. Digital peripherals draw no power in Sleep Mode.

19.1. Normal Mode

The MCU is fully functional in Normal Mode. Figure 19.1 shows the on-chip power distribution to various peripherals. There are three supply voltages powering various sections of the chip: VBAT, DCOUT, and the 1.8 V internal core supply (output of VREG0). All analog peripherals are directly powered from the VBAT pin. All digital peripherals and the CIP-51 core are powered from the 1.8 V internal core supply (output of VREG0). The Pulse counter, RAM, PMU0, and the SmaRTClock are powered from the internal core supply when the device is in normal mode. The input to VREG0 is controlled by software and depends on the settings of the power select switch. The power select switch may be configured to power VREG0 from VBAT or from the output of the DC0.

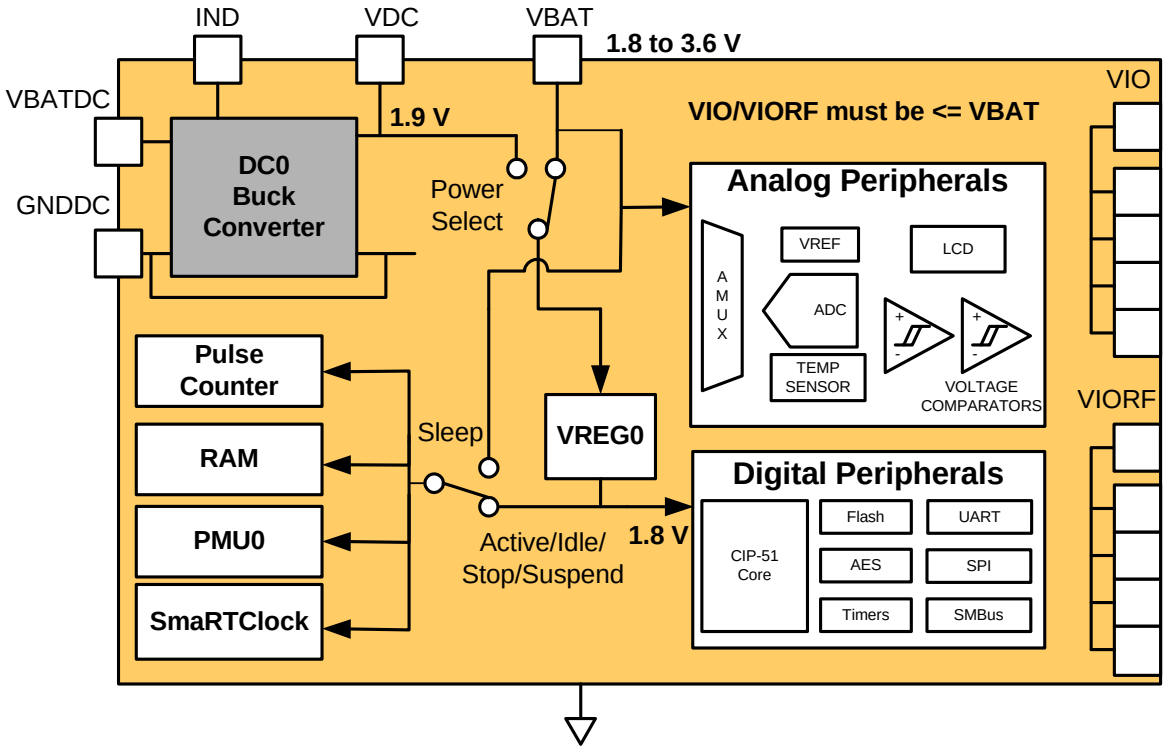


Figure 19.1. C8051F96x Power Distribution

19.2. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event

of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section “22.6. PCA Watchdog Timer Reset” on page 283 for more information on the use and configuration of the WDT.

19.3. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the precision internal oscillator and CPU are stopped; the state of the low power oscillator and the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout.

Stop Mode is a legacy 8051 power mode; it will not result in optimal power savings. Sleep, Suspend, or Low Power Idle mode will provide more power savings if the MCU needs to be inactive for a long period of time.

19.4. Low Power Idle Mode

Low Power Idle Mode uses clock gating to reduce the supply current when the device is placed in Idle mode. This mode is enabled by configuring the clock tree gates using the PCLKEN register, setting the LPMEN bit in the CLKMODE register, and placing the device in Idle mode. The clock is automatically gated from the CPU upon entry into Idle mode when the LPMEN bit is set. This mode provides substantial power savings over the standard Idle Mode especially at high system clock frequencies.

The clock gating logic may also be used to reduce power when executing code. Low Power Active Mode is enabled by configuring the PCLKACT and PCLKEN registers, then setting the LPMEN bit. The PCLKACT register provides the ability to override the PCLKEN setting to force a clock to certain peripherals in Low Power Active mode. If the PCLKACT register is left at its default value, then PCLKEN determines which perpherals will be clocked in this mode. The CPU is always clocked in Low Power Active Mode.

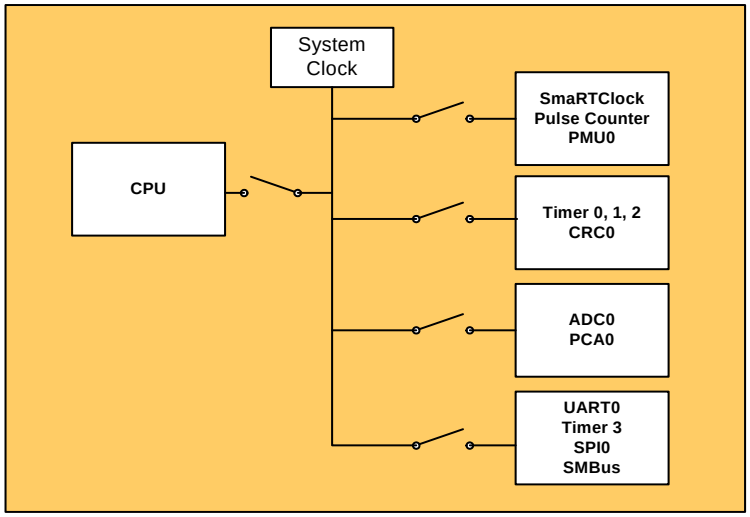


Figure 19.2. Clock Tree Distribution

SFR Definition 19.1. PCLKACT: Peripheral Active Clock Enable

Bit	7	6	5	4	3	2	1	0
Name					PCLKACT[3:0]			
Type	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xF5

Bit	Name	Function
7:4	Unused	Read = 0b; Write = don't care.
3	PCLKACT3	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to the SmaRTClock, Pulse Counter, and PMU0 revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Active Mode.
2	PCLKACT2	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to Timer 0, Timer 1, Timer 2, and CRC0 revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Active Mode.
1	PCLKACT1	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to ADC0 and PCA0 revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to ADC0 and PCA0 in Low Power Active Mode.
0	PCLKACT0	**Clock Enable Controls for Peripherals in Low Power Active Mode.** 0: Clocks to UART0, Timer 3, SPI0, and the SMBus revert to the PCLKEN setting in Low Power Active Mode. 1: Enable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Active Mode.

SFR Definition 19.2. PCLKEN: Peripheral Clock Enable

Bit	7	6	5	4	3	2	1	0
Name					PCLKEN[3:0]			
Type	R/W	R/W	R/W	R/W	R/W			
Reset								

SFR Page = 0xF; SFR Address = 0xFE

Bit	Name	Function
7:4	Unused	Read = 0b; Write = don't care.
3	PCLKEN3	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Idle Mode. 1: Enable clocks to the SmaRTClock, Pulse Counter, and PMU0 in Low Power Idle Mode.
2	PCLKEN2	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Idle Mode. 1: Enable clocks to Timer 0, Timer 1, Timer 2, and CRC0 in Low Power Idle Mode.
1	PCLKEN1	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disableclocks to ADC0 and PCA0 in Low Power Idle Mode. 1: Enable clocks to ADC0 and PCA0 in Low Power Idle Mode.
0	PCLKEN0	**Clock Enable Controls for Peripherals in Low Power Idle Mode.** 0: Disable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Idle Mode. 1: Enable clocks to UART0, Timer 3, SPI0, and the SMBus in Low Power Idle Mode.

SFR Definition 19.3. CLKMODE: Clock Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	LPMEN	Reserved	Reserved
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xFD; Bit-Addressable

Bit	Name	Function
7:3	Reserved	Read = 0b; Write = Must write 00000b.
2	LPMEN	**Low Power Mode Enable.** Setting this bit allows the device to enter Low Power Active or Idle Mode.
1	Reserved	Read = 0b; Must write 0b.
0	Reserved	Read = 0b; Must write 0b.

19.5. Suspend Mode

Setting the Suspend Mode Select bit (PMU0CF.6) causes the system clock to be gated off and all internal oscillators disabled. The system clock source must be set to the low power internal oscillator or the precision oscillator prior to entering Suspend Mode. All digital logic (timers, communication peripherals, interrupts, CPU, etc.) stops functioning until one of the enabled wake-up sources occurs.

The following wake-up sources can be configured to wake the device from Suspend Mode:

- Pulse Counter Count Reached Event
- VBAT Monitor (part of LCD logic)
- SmaRTClock Oscillator Fail
- SmaRTClock Alarm
- Port Match Event
- Comparator0 Rising Edge

Note: Upon wake-up from suspend mode, PMU0 requires two system clocks in order to update the PMU0CF wake-up flags. All flags will read back a value of '0' during the first two system clocks following a wake-up from suspend mode.

In addition, a noise glitch on $\overline{\text{RST}}$ that is not long enough to reset the device will cause the device to exit suspend. In order for the MCU to respond to the pin reset event, software must not place the device back into suspend mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the /RST pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into suspend mode. A 4.7 kW pullup resistor to VDD is recommend for RST to prevent noise glitches from waking the device.

19.6. Sleep Mode

Setting the Sleep Mode Select bit (PMU0CF.7) turns off the internal 1.8 V regulator (VREG0) and switches the power supply of all on-chip RAM to the VBAT pin (see Figure 19.1). Power to most digital logic on the chip is disconnected; only PMU0, LCD, Power Select Switch, Pulse Counter, and the SmaRTClock remain powered. Analog peripherals remain powered; however, only the Comparators remain functional when the device enters Sleep Mode. All other analog peripherals (ADC0, IREF0, External Oscillator, etc.) should be disabled prior to entering Sleep Mode. The system clock source must be set to the low power internal oscillator or the precision oscillator prior to entering Sleep Mode.

GPIO pins configured as digital outputs will retain their output state during sleep mode. In two-cell mode, they will maintain the same current drive capability in sleep mode as they have in normal mode.

GPIO pins configured as digital inputs can be used during sleep mode as wakeup sources using the port match feature. In two-cell mode, they will maintain the same input level specs in sleep mode as they have in normal mode.

‘C8051F96x devices support a wakeup request for external devices. Upon exit from sleep mode, the wake-up request signal is driven low, allowing other devices in the system to wake up from their low power modes.

RAM and SFR register contents are preserved in sleep mode as long as the voltage on VBAT does not fall below V_{POR}. The PC counter and all other volatile state information is preserved allowing the device to resume code execution upon waking up from Sleep mode.

The following wake-up sources can be configured to wake the device from sleep mode:

- Pulse Counter Count Reached Event
- VBAT Monitor (part of LCD logic)
- SmaRTClock Oscillator Fail
- SmaRTClock Alarm
- Port Match Event
- Comparator0 Rising Edge

The comparator requires a supply voltage of at least 1.8 V to operate properly. On C8051F96x devices, the POR supply monitor can be disabled to save power by writing 1 to the MONDIS (PMU0MD.5) bit. When the POR supply monitor is disabled, all reset sources will trigger a full POR and will re-enable the POR supply monitor.

In addition, any falling edge on $\overline{\text{RST}}$ (due to a pin reset or a noise glitch) will cause the device to exit sleep mode. In order for the MCU to respond to the pin reset event, software must not place the device back into sleep mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the $\overline{\text{RST}}$ pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into sleep mode. A 4.7 kΩ pullup resistor to VDD is recommend for $\overline{\text{RST}}$ to prevent noise glitches from waking the device.

19.7. Configuring Wakeup Sources

Before placing the device in a low power mode, one or more wakeup sources should be enabled so that the device does not remain in the low power mode indefinitely. For idle mode, this includes enabling any interrupt. For stop mode, this includes enabling any reset source or relying on the $\overline{\text{RST}}$ pin to reset the device.

Wake-up sources for suspend and sleep modes are configured through the PMU0CF register. Wake-up sources are enabled by writing 1 to the corresponding wake-up source enable bit. Wake-up sources must be re-enabled each time the device is placed in Suspend or Sleep mode, in the same write that places the device in the low power mode.

The reset pin is always enabled as a wake-up source. On the falling edge of $\overline{\text{RST}}$, the device will be awaken from sleep mode. The device must remain awake for more than 15 µs in order for the reset to take place.

19.8. Determining the Event that Caused the Last Wakeup

When waking from idle mode, the CPU will vector to the interrupt which caused it to wake up. When waking from stop mode, the RSTSRC register may be read to determine the cause of the last reset.

Upon exit from suspend or sleep mode, the wake-up flags in the PMU0CF register can be read to determine the event which caused the device to wake up. After waking up, the wake-up flags will continue to be updated if any of the wake-up events occur. Wake-up flags are always updated, even if they are not enabled as wake-up sources.

All wake-up flags enabled as wake-up sources in PMU0CF must be cleared before the device can enter suspend or sleep mode. After clearing the wake-up flags, each of the enabled wake-up events should be checked in the individual peripherals to ensure that a wake-up event did not occur while the wake-up flags were being cleared.

SFR Definition 19.4. PMU0CF: Power Management Unit Configuration[1,2,3]

Bit	7	6	5	4	3	2	1	0
Name	SLEEP	SUSPEND	CLEAR	RSTWK	RTCFWK	RTCAWK	PMATWK	CPT0WK
Type	W	W	W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	Varies	Varies	Varies	Varies	Varies

SFR Page = 0x0; SFR Address = 0xB5

Bit	Name	Description	Write	Read
7	SLEEP	**Sleep Mode Select**	Writing 1 places the device in Sleep Mode.	N/A
6	SUSPEND	**Suspend Mode Select**	Writing 1 places the device in Suspend Mode.	N/A
5	CLEAR	**Wake-up Flag Clear**	Writing 1 clears all wake-up flags.	N/A
4	RSTWK	**Reset Pin Wake-up Flag**	N/A	Set to 1 if a falling edge has been detected on RST.
3	RTCFWK	**SmaRTClock Oscillator Fail Wake-up Source Enable and Flag**	0: Disable wake-up on SmaRTClock Osc. Fail. 1: Enable wake-up on SmaRTClock Osc. Fail.	Set to 1 if the SmaRTClock Oscillator has failed.
2	RTCAWK	**SmaRTClock Alarm Wake-up Source Enable and Flag**	0: Disable wake-up on SmaRTClock Alarm. 1: Enable wake-up on SmaRTClock Alarm.	Set to 1 if a SmaRTClock Alarm has occurred.
1	PMATWK	**Port Match Wake-up Source Enable and Flag**	0: Disable wake-up on Port Match Event. 1: Enable wake-up on Port Match Event.	Set to 1 if a Port Match Event has occurred.
0	CPT0WK	**Comparator0 Wake-up Source Enable and Flag**	0: Disable wake-up on Comparator0 rising edge. 1: Enable wake-up on Comparator0 rising edge.	Set to 1 if Comparator0 rising edge has occurred.

Notes:

1. Read-modify-write operations (ORL, ANL, etc.) should not be used on this register. Wake-up sources must be re-enabled each time the SLEEP or SUSPEND bits are written to 1.
2. The Low Power Internal Oscillator cannot be disabled and the MCU cannot be placed in Suspend or Sleep Mode if any wake-up flags are set to 1. Software should clear all wake-up sources after each reset and after each wake-up from Suspend or Sleep Modes.
3. PMU0 requires two system clocks to update the wake-up source flags after waking from Suspend mode. The wake-up source flags will read ‘0’ during the first two system clocks following the wake from Suspend mode.

SFR Definition 19.5. PMU0FL: Power Management Unit Flag[1,2]

Bit	7	6	5	4	3	2	1	0
Name						BATMWK	Reserved	PC0WK
Type	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	Varies

SFR Page = 0x0; SFR Address = 0xB6

Bit	Name	Description	Write	Read
7:3	Unused	**Unused**	Don't Care.	0000000
2	BATMWK	**VBAT Monitor (inside LCD Logic) Wake-up Source Enable and Flag**	0: Disable wake-up on VBAT Monitor event. 1: Enable wake-up on CS0 event.	Set to 1 if VBAT Monitor event caused the last wake-up.
1	Reserved	**Reserved**	Must write 0.	Always reads 0.
0	CS0WK	**Pulse Counter Wake-up Source Enable and Flag**	0: Disable wake-up on PC0 event. 1: Enable wake-up on PC0 event.	Set to 1 if PC0 event caused the last wake-up.

Notes:

1. The Low Power Internal Oscillator cannot be disabled and the MCU cannot be placed in suspend or sleep mode if any wake-up flags are set to 1. Software should clear all wake-up sources after each reset and after each wake-up from Suspend or Sleep Modes.
2. PMU0 requires two system clocks to update the wake-up source flags after waking from suspend mode. The wake-up source flags will read 0 during the first two system clocks following the wake from suspend mode.

SFR Definition 19.6. PMU0MD: Power Management Unit Mode

Bit	7	6	5	4	3	2	1	0
Name	RTCOE	WAKEOE	MONDIS					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xB6

Bit	Name	Function
7	RTCOE	**Buffered SmaRTClock Output Enable.** Enables the buffered SmaRTClock oscillator output on P0.2. 0: Buffered SmaRTClock output not enabled. 1: Buffered SmaRTClock output not enabled.
6	WAKEOE	**Wakeup Request Output Enable.** Enables the Sleep Mode wake-up request signal on P0.3. 0: Wake-up request signal is not enabled. 1: Wake-up request signal is enabled.
5	MONDIS	**POR Supply Monitor Disable.** Writing a 1 to this bit disables the POR supply monitor.
4:0	Unused	Read = 00000b. Write = Don't Care.

SFR Definition 19.7. PCON: Power Management Control Register

Bit	7	6	5	4	3	2	1	0
Name	GF[4:0]					PWRSEL	STOP	IDLE
Type	R/W					R/W	W	W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0x87

Bit	Name	Description	Write	Read
7:3	GF[5:0]	**General Purpose Flags**	Sets the logic value.	Returns the logic value.
2	PWRSEL	**Power Select**	0: VBAT is selected as the input to VREG0. 1: VDC is selected as the input to VREG0.	
1	STOP	**Stop Mode Select**	Writing 1 places the device in Stop Mode.	N/A
0	IDLE	**Idle Mode Select**	Writing 1 places the device in Idle Mode.	N/A

19.9. Power Management Specifications

See Table 4.7 on page 69 for detailed Power Management Specifications.

20. On-Chip DC-DC Buck Converter (DC0)

C8051F96x devices include an on-chip step down dc-dc converter to efficiently utilize the energy stored in the battery, thus extending the operational life time. The dc-dc converter is a switching buck converter with an input supply of 1.8 to 3.8 V and an output that is programmable from 1.8 to 3.5 V in steps of 0.1 V. The battery voltage should be at least 0.4 V higher than the programmed output voltage. The programmed output voltage has a default value of 1.9 V. The dc-dc converter can supply up to 250 mW. The dc-dc converter can be used to power the MCU and/or external devices in the system (e.g., an RF transceiver).

The dc-dc converter has a built in voltage reference and oscillator, and will automatically limit or turn off the switching activity in case the peak inductor current rises beyond a safe limit or the output voltage rises above the programmed target value. This allows the dc-dc converter output to be safely overdriven by a secondary power source (when available) in order to preserve battery life. When enabled, the dc-dc converter can source current into the output capacitor, but cannot sink current. The dc-dc converter's settings can be modified using SFR registers described in Section 20.8.

Figure 20.1 shows a block diagram of the buck converter.

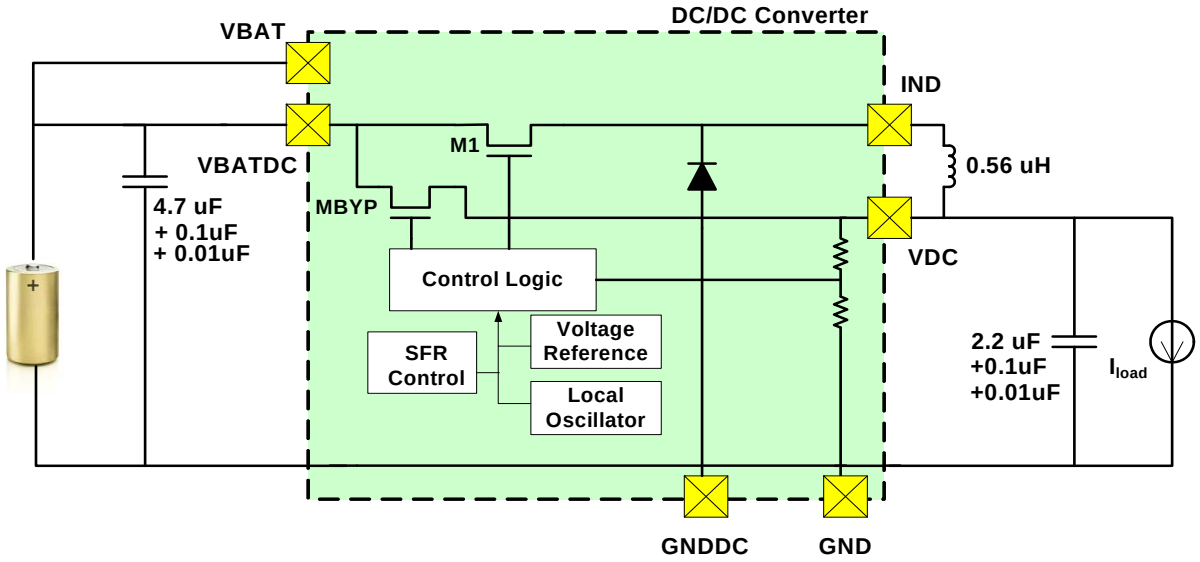


Figure 20.1. Step Down DC-DC Buck Converter Block Diagram

20.1. Startup Behavior

The dc-dc converter is enabled by setting bit DC0EN (DC0MD.0) to logic 1. When first enabled, the M1 switch turns on and continues to supply current into the output capacitor through the inductor until the VDC output voltage reaches the programmed level set by by the VSEL bits (DC0CF.[6:3]).

The peak transient current in the inductor is limited for safe operation. The peak inductor current is programmable using the ILIMIT bits (DC0MD.[6:4]). The peak inductor current, size of the output capacitor and the amount of dc load current present during startup will determine the length of time it takes to charge the output capacitor. The RDYH and RDYL bits (DC0RDY.7 and DC0DRY.6) may be used to determine when the output voltage is within approximately 100 mV of the programmed voltage.

In order to ensure reliable startup of the dc-dc converter, the following restrictions have been imposed:

- The maximum dc load current allowed during startup is given in Table 4.20 on page 77. If the dc-dc converter is powering external sensors or devices through the VDC pin, then the current supplied to these sensors or devices is counted towards this limit. The in-rush current into capacitors does not count towards this limit.

- The maximum total output capacitance is given in Table 4.20 on page 77. This value includes the required 2.2 µF ceramic output capacitor and any additional capacitance connected to the VDC pin.

The peak inductor current limit is programmable by software as shown in Table 20.1. Limiting the peak inductor current can allow the dc-dc converter to start up using a high impedance power source (such as when a battery is near its end of life) or allow inductors with a low current rating to be utilized. By default, the peak inductor current is set to 500 mA.

.

Table 20.1. IPeak Inductor Current Limit Settings

ILIMIT	Peak Current (mA)
001	200
010	300
011	400
100	500
101	600

The peak inductor current is dependent on several factors including the dc load current and can be estimated using following equation:

$$I_{PK} = \frac{2 \times I_{LOAD} \times (VDC - VBATDC)}{efficiency \times inductance \times frequency}$$

efficiency = 0.80

inductance = 0.68 µH

frequency = 2.4 MHz

20.2. High Power Applications

The dc-dc converter is designed to provide the system with 150 mW of output power. At high output power, an inductor with low dc resistance should be chosen in order to minimize power loss and maximize efficiency. At load currents higher than 20 mA, efficiency improvents may be achieved by placing a schottky diode (e.g. MBR052LT1) between the IND pin and GND in parallel with the internal diode (see Figure 20.1).

20.3. Pulse Skipping Mode

The dc-dc converter allows the user to set the minimum pulse width such that if the duty cycle needs to decrease below a certain width in order to maintain regulation, an entire "clock pulse" will be skipped.

Pulse skipping can provide substantial power savings, particularly at low values of load current. The converter will continue to maintain a minimum output voltage at its programmed value when pulse skipping is employed, though the output voltage ripple can be higher. Another consideration is that the dc-dc will operate with pulse-frequency modulation rather than pulse-width modulation, which makes the switching frequency spectrum less predictable; this could be an issue if the dc-dc converter is used to power a radio.

20.4. Optimizing Board Layout

The PCB layout does have an effect on the overall efficiency. The following guidelines are recommended to acheive the optimum layout:

- Place the input capacitor stack as close as possible to the VBATDC pin. The smallest capacitors in the stack should be placed closest to the VBATDC pin.
- Place the output capacitor stack as close as possible to the VDC pin. The smallest capacitors in the stack should be placed closest to the VDC pin.
- Minimize the trace length between the IND pin, the inductor, and the VDC pin.

20.5. Selecting the Optimum Switch Size

The dc-dc converter provides the ability to change the size of the built-in switches. To maximize efficiency, one of two switch sizes may be selected. The large switches are ideal for carrying high currents and the small switches are ideal for low current applications. The ideal switchover point to switch from the small switches to the large switches is at approximately 5 mA total output current.

20.6. DC-DC Converter Clocking Options

The dc-dc converter may be clocked from its internal oscillator, or from any system clock source, selectable by the CLKSEL bit (DC0CF.0). The dc-dc converter internal oscillator frequency is approximately 2.4 MHz. For a more accurate clock source, the system clock, or a divided version of the system clock may be used as the dc-dc clock source. The dc-dc converter has a built in clock divider (configured using DC0CF[6:5]) which allows any system clock frequency over 1.6 MHz to generate a valid clock in the range of 1.9 to 3.8 MHz.

When the precision internal oscillator is selected as the system clock source, the OSCICL register may be used to fine tune the oscillator frequency and the dc-dc converter clock. The oscillator frequency should only be decreased since it is factory calibrated at its maximum frequency. The minimum frequency which can be reached by the oscillator after taking into account process variations is approximately 16 MHz. The system clock routed to the dc-dc converter clock divider also may be inverted by setting the CLKINV bit (DC0CF.3) to logic 1. These options can be used to minimize interference in noise sensitive applications.

20.7. Bypass Mode

The dc-dc converter has a bypass switch (MBYP), see Figure 20.1, which allows the output voltage (VDC) to be directly tied to the input supply (VBATDC), bypassing the dc-dc converter. The bypass switch may be used independently from the dc-dc converter. For example, applications that need to power the VDC supply in the lowest power Sleep mode can turn on the bypass switch prior to turning off the dc-dc converter in order to avoid powering down the external circuitry connected to VDC.

There are two ways to close the bypass switch. Using the first method, Forced Bypass Mode, the FORBYP bit is set to a logic 1 forcing the bypass switch to close. Clearing the FORBYP bit to logic 0 will allow the switch to open if it is not being held closed using Automatic Bypass Mode.

The Automatic Bypass Mode, enabled by setting the AUTOBYP to logic 1, closes the bypass switch when the difference between VBATDC and the programmed output voltage is less than approximately 0.4 V. Once the difference exceeds approximately 0.5 V, the bypass switch is opened unless being held closed by Forced Bypass Mode. In most systems, Automatic Bypass Mode will be left enabled, and the Forced Bypass Mode will be used to close the switch as needed by the system.

20.8. DC-DC Converter Register Descriptions

The SFRs used to configure the dc-dc converter are described in the following register descriptions.

SFR Definition 20.1. DC0CN: DC-DC Converter Control

Bit	7	6	5	4	3	2	1	0
Name	CLKSEL	CLKDIV[1:0]		AD0CKINV	CLKINV	ILIMIT	MIN_PW[1:0]	
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	1

SFR Page = 0x2; SFR Address = 0xB1

Bit	Name	Function
7	CLKSEL	**DC-DC Converter Clock Source Select.** Specifies the dc-dc converter clock source. 0: The dc-dc converter is clocked from its local oscillator. 1: The dc-dc converter is clocked from the system clock.
6:5	CLKDIV[1:0]	**DC-DC Clock Divider.** Divides the dc-dc converter clock when the system clock is selected as the clock source for dc-dc converter. Ignored all other times. 00: The dc-dc converter clock is system clock divided by 1. 01: The dc-dc converter clock is system clock divided by 2. 10: The dc-dc converter clock is system clock divided by 4. 11: The dc-dc converter clock is system clock divided by 8.
4	AD0CKINV	**ADC0 Clock Inversion (Clock Invert During Sync).** Inverts the ADC0 SAR clock derived from the dc-dc converter clock when the SYNC bit (DC0CN.3) is enabled. This bit is ignored when the SYNC bit is set to zero. 0: ADC0 SAR clock is inverted. 1: ADC0 SAR clock is not inverted.
3	CLKINV	**DC-DC Converter Clock Invert.** Inverts the system clock used as the input to the dc-dc clock divider. 0: The dc-dc converter clock is not inverted. 1: The dc-dc converter clock is inverted.
2	SYNC	**ADC0 Synchronization Enable.** When synchronization is enabled, the ADC0SC[4:0] bits in the ADC0CF register must be set to 00000b. 0: The ADC is not synchronized to the dc-dc converter. 1: The ADC is synchronized to the dc-dc converter. ADC0 tracking is performed during the longest quiet time of the dc-dc converter switching cycle and ADC0 SAR clock is also synchronized to the dc-dc converter switching cycle.
1:0	MINPW[1:0]	**DC-DC Converter Minimum Pulse Width.** Specifies the minimum pulse width. 00: Minimum pulse detection logic is disabled (no pulse skipping). 01: Minimum pulse width is 10 ns. 10: Minimum pulse width is 20 ns. 11: Minimum pulse width is 40 ns.

SFR Definition 20.2. DC0CF: DC-DC Converter Configuration

Bit	7	6	5	4	3	2	1	0
Name	BYPASS	VSEL[3:0]				OSCDIS	SWSEL[1:0]	
Type	R	R/W				R/W		
Reset	0	0	0	0	1	0	1	1

SFR Page = 0x2; SFR Address = 0xB2

Bit	Name	Function
7	BYPASS	**DC-DC Converter Bypass Switch Active Indicator.** 0: The bypass switch is open. 1: The bypass switch is closed (VDC is connected to VBATDC).
6:3	VSEL[3:0]	**DC-DC Converter Output Voltage Select.** Specifies the target output voltage. 0000: Target output voltage is 1.8 V. 1000: Target output voltage is 2.6 V. 0001: Target output voltage is 1.9 V. 1001: Target output voltage is 2.7 V. 0010: Target output voltage is 2.0 V. 1010: Target output voltage is 2.8 V. 0011: Target output voltage is 2.1 V. 1011: Target output voltage is 2.9 V. 0100: Target output voltage is 2.2 V. 1100: Target output voltage is 3.0 V. 0101: Target output voltage is 2.3 V. 1101: Target output voltage is 3.1 V. 0110: Target output voltage is 2.4 V. 1110: Target output voltage is 3.3 V. 0111: Target output voltage is 2.5 V. 1111: Target output voltage is 3.5 V.
2	VSEL[2:0]	**DC-DC Converter Local Oscillator Disabled.** 0: The local oscillator inside the dc-dc converter is enabled. 1: The local oscillator inside the dc-dc converter is disabled.
1:0	SWSEL[1:0]	**DC-DC Converter Power Switch Select.** Selects the size of the power switches (M1, M2). Using smaller switches will resut in higher efficiency at low supply currents. 00: Minimum switch size, optimized for load currents smaller than 5 mA. 01: Reserved. 10: Reserved. 11: Maximum switch size, optimized for load currents greater than 5 mA.

SFR Definition 20.3. DC0MD: DC-DC Converter Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved	ILIMIT			FORBYP	AUTOBYP	Reserved	DC0EN
Type	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xB3

Bit	Name	Function
7	Reserved	Read = 0b; Must write 0b.
6:4	ILIMIT	**Peak Current Limit Threshold.** 000: Reserved 001: Peak Inductor current is limited to 200 mA 010: Peak Inductor current is limited to 300 mA 011: Peak Inductor current is limited to 400 mA 100: Peak Inductor current is limited to 500 mA 101: Peak Inductor current is limited to 600 mA 110: Reserved 111: Reserved
3	FORBYP	**Enable Forced Bypass Mode.** 0: Forced bypass mode is disabled. 1: Forced bypass mode is enabled.
2	AUTOBYP	**Enable Automatic Bypass Mode.** 0: Automatic Bypass mode is disabled. 1: Automatic bypass mode is enabled.
1	Reserved	Read = 1b; Must write 1b.
0	DC0EN	**DC-DC Converter Enable.** 0: DC-DC converter is disabled. 1: DC-DC converter is enabled.

SFR Definition 20.4. DC0RDY: DC-DC Converter Ready Indicator

Bit	7	6	5	4	3	2	1	0
Name	RDYH	RDYL	Reserved					
Type	R	R	R/W					
Reset	0	0	0	1	1	1	1	1

SFR Page = 0x2; SFR Address = 0xFD

Bit	Name	Function
7	RDYH	**DC0 Ready Indicator (High Threshold).** Indicates when VDC is 100 mV higher than the target output value. 0: VDC pin voltage is less than the DC0 High Threshold. 1: VDC pin voltage is higher than the DC0 High Threshold.
6	RDYL	**DC0 Ready Indicator (Low Threshold).** Indicates when VDC is 100 mV lower than the target output value. 0: VDC pin voltage is less than the DC0 Low Threshold. 1: VDC pin voltage is higher than the DC0 Low Threshold.
5:0	Reserved	Read = 011111b; Must write 011111b.

20.9. DC-DC Converter Specifications

See Table 4.20 on page 77 for a detailed listing of dc-dc converter specifications.

21. Voltage Regulator (VREG0)

C8051F96x devices include an internal voltage regulator (VREG0) to regulate the internal core supply to 1.8 V from a VDD/DC+ supply of 1.8 to 3.6 V. Electrical characteristics for the on-chip regulator are specified in the Electrical Specifications chapter.

The REG0CN register allows the Precision Oscillator Bias to be disabled, reducing supply current in all non-sleep power modes. This bias should only be disabled when the precision oscillator is not being used.

The internal regulator (VREG0) is disabled when the device enters sleep mode and remains enabled when the device enters suspend mode. See Section “19. Power Management” on page 257 for complete details about low power modes.

SFR Definition 21.1. REG0CN: Voltage Regulator Control

Bit	7	6	5	4	3	2	1	0
Name				OSCBIAS				
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC9

Bit	Name	Function
7:5	Reserved	Read = 000b. Must Write 000b.
4	OSCBIAS	**Precision Oscillator Bias.** When set to 1, the bias used by the precision oscillator is forced on. If the precision oscillator is not being used, this bit may be cleared to 0 to to save supply current in all non-Sleep power modes.
3:0	Reserved	Read = 0000b. Must Write 0000b.

21.1. Voltage Regulator Electrical Specifications

See Table 4.17 on page 75 for detailed Voltage Regulator Electrical Specifications.

22. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled

All SFRs are reset to the predefined values noted in the SFR descriptions. The contents of RAM are unaffected during a reset; any previously stored data is preserved as long as power is not lost. Since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to an internal oscillator. Refer to Section “23. Clocking Sources” on page 286 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source (Section “33.4. Watchdog Timer Mode” on page 477 details the use of the Watchdog Timer). Program execution begins at location 0x0000.

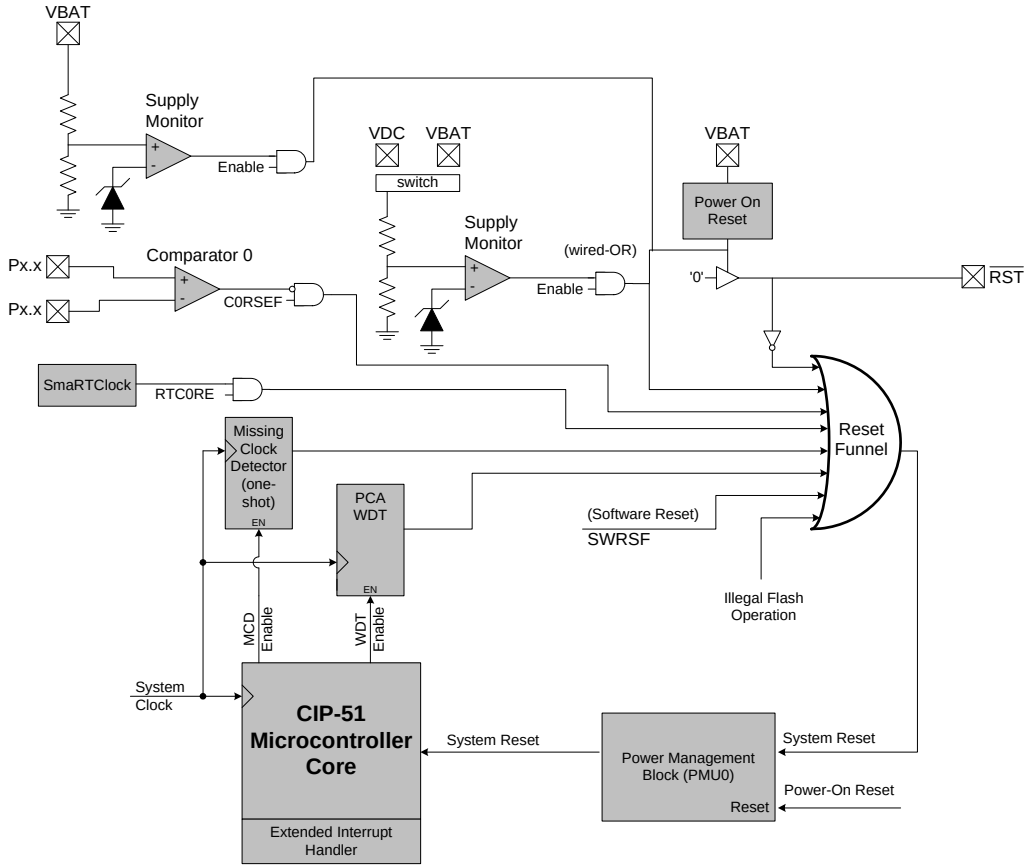


Figure 22.1. Reset Sources

22.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin voltage tracks the supply voltage (through a weak pull-up) until the device is released from reset. After the supply settles above VPOR, a delay occurs before the device is released from reset; the delay decreases as the supply ramp time increases (ramp time is defined as how fast the supply ramps from 0 V to V_{POR}). Figure 22.2 plots the power-on and supply monitor reset timing. For valid ramp times (less than 3 ms), the power-on reset delay ($T_{PORDelay}$) is typically 7 ms (V_{DD} = 1.8 V) or 15 ms (V_{DD} = 3.6 V).

Note: The maximum supply ramp time is 3 ms; slower ramp times may cause the device to be released from reset before the supply reaches the V_{POR} level.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000), software can read the PORSF flag to determine if a power-up was the cause of reset. The contents of internal data memory should be assumed to be undefined after a power-on reset.

The POR supply monitor will continue to monitor the VBAT supply, even in Sleep Mode, to reset the system if the supply voltage drops below VPOR. It can be disabled to save power by writing 1 to the MONDIS (PMU0MD.5) bit. When the POR supply monitor is disabled, all reset sources will trigger a full POR and will re-enable the POR supply monitor.

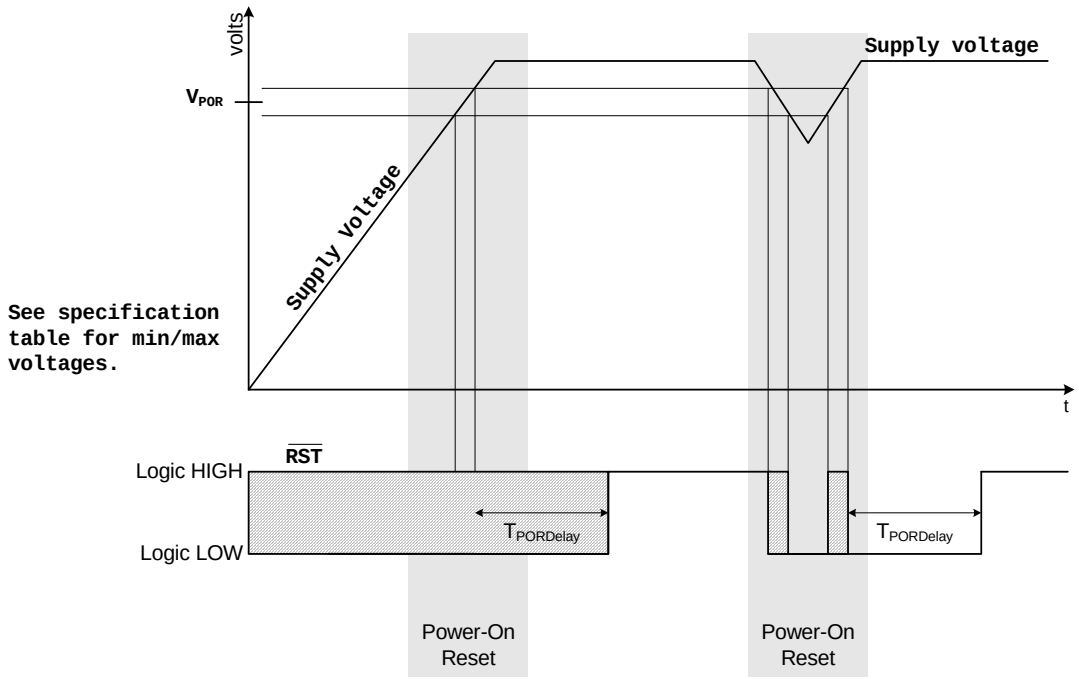


Figure 22.2. Power-On Reset Timing Diagram

22.2. Power-Fail Reset

C8051F96x devices have two Active Mode Supply Monitors that can hold the system in reset if the supply voltage drops below V_{RST}. The first of the two identical supply monitors is connected to the output of the supply select switch (which chooses the VBAT or VDC pin as the source of the digital supply voltage) and is enabled and selected as a reset source after each power-on or power-fail reset. This supply monitor will be referred to as the digital supply monitor. The second supply monitor is connected directly to the VBAT pin an is disabled after each power-on or power-fail reset. This supply monitor will be referred to as the analog supply monitor. The analog supply monitor should be enabled any time the supply select switch is set to the VDC pin to ensure that the VBAT supply does not drop below V_{RST} .

When enabled and selected as a reset source, any power down transition or power irregularity that causes the monitored supply voltage to drop below V_{RST} will cause the $\overline{RST}$ pin to be driven low and the CIP-51 will be held in a reset state (see Figure 22.2). When the supply voltage returns to a level above V_{RST}, the CIP-51 will be released from the reset state.

After a power-fail reset, the PORSF flag reads 1, the contents of RAM invalid, and the digital supply monitor is enabled and selected as a reset source. The enable state of either supply monitor and its selection as a reset source is only altered by power-on and power-fail resets. For example, if the supply monitor is de-selected as a reset source and disabled by software, then a software reset is performed, the supply monitor will remain disabled and de-selected after the reset.

In battery-operated systems, the contents of RAM can be preserved near the end of the battery's usable life if the device is placed in Sleep Mode prior to a power-fail reset occurring. When the device is in Sleep Mode, the power-fail reset is automatically disabled, both active mode supply monitors are turned off, and the contents of RAM are preserved as long as the supply does not fall below V_{POR}. A large capacitor can be used to hold the power supply voltage above V_{POR} while the user is replacing the battery. Upon waking from Sleep mode, the enable and reset source select state of the V_{DD} supply monitor are restored to the value last set by the user.

To allow software early notification that a power failure is about to occur, the VDDOK bit is cleared when the supply falls below the V_{WARN} threshold. The VDDOK bit can be configured to generate an interrupt. Each of the active mode supply montiors have their independent VDDOK and V_{WARN} flags. See Section "17. Interrupt Handler" on page 232 for more details.

Important Note: To protect the integrity of Flash contents, **the active mode supply monitor(s) must be enabled and selected as a reset source if software contains routines which erase or write Flash memory.** If the digital supply monitor is not enabled, any erase or write performed on Flash memory will cause a Flash Error device reset.

Important Notes:

- The Power-on Reset (POR) delay is not incurred after a supply monitor reset. See Section “4. Electrical Characteristics” on page 56 for complete electrical characteristics of the active mode supply monitors.
- Software should take care not to inadvertently disable the supply monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to 1 to keep the supply monitor enabled as a reset source.
- The supply monitor must be enabled before selecting it as a reset source**.** Selecting the supply monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the supply monitor and selecting it as a reset source. See Section “4. Electrical Characteristics” on page 56 for minimum supply monitor turn-on time. **No delay should be introduced in systems where software contains routines that erase or write Flash memory.** The procedure for enabling the V_{DD} supply monitor and selecting it as a reset source is shown below:
  1. Enable the Supply Monitor (VDMEN bit in VDM0CN = 1).
  2. Wait for the Supply Monitor to stabilize (optional).
  3. Select the Supply Monitor as a reset source (PORSF bit in RSTSRC = 1).

SFR Definition 22.1. VDM0CN: VDD Supply Monitor Control

Bit	7	6	5	4	3	2	1	0
Name	VDMEN	VDDSTAT	VDDOK	VDDOKIE	VBMEN	VBSTAT	VBOK	VBOKIE
Type	R/W	R	R	R/W	R/W	R	R	R/W
Reset	1	Varies	Varies	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xFF

Bit	Name	Function
7	VDMEN	**Digital Supply Monitor Enable (Power Select Switch Output).** 0: Digital Supply Monitor Disabled. 1: Digital Supply Monitor Enabled.
6	VDDSTAT	**Digital Supply Status.** This bit indicates the current digital power supply status. 0: Digital supply is at or below the V_{RST} threshold. 1: Digital supply is above the V_{RST} threshold.
5	VDDOK	**Digital Supply Status (Early Warning).** This bit indicates the current digital power supply status. 0: Digital supply is at or below the VDD_{WARN} threshold. 1: Digital supply is above the VDD_{WARN} threshold.
4	VDDOKIE	**Digital Early Warning Interrupt Enable.** Enables the V_{DD} Early Warning Interrupt. 0: V_{DD} Early Warning Interrupt is disabled. 1: V_{DD} Early Warning Interrupt is enabled.
3	VBMEN	**Analog Supply Monitor Enable (VBAT Pin).** 0: Analog Supply Monitor Disabled. 1: Analog Supply Monitor Enabled.
2	VBSTAT	**Analog Supply Status.** This bit indicates the analog (VBAT) power supply status. 0: VBAT is at or below the V_{RST} threshold. 1: VBAT is above the V_{RST} threshold.
1	VBOK	**Analog Supply Status (Early Warning).** This bit indicates the current VBAT power supply status. 0: VBAT is at or below the VDD_{WARN} threshold. 1: VBAT is above the VDD_{WARN} threshold.
0	VBOKIE	**Analog Early Warning Interrupt Enable.** Enables the VBAT Early Warning Interrupt. 0: VBAT Early Warning Interrupt is disabled. 1: VBAT Early Warning Interrupt is enabled.

22.3. External Reset

The external RST pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the RST pin generates a reset; an external pullup and/or decoupling of the RST pin may be necessary to avoid erroneous noise-induced resets. See Table 4.6 for complete RST pin specifications. The external reset remains functional even when the device is in the low power suspend and sleep modes. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

22.4. Missing Clock Detector Reset

The missing clock detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than 100 µs, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read 1, signifying the MCD as the reset source; otherwise, this bit reads 0. Writing a 1 to the MCDRSF bit enables the Missing Clock Detector; writing a 0 disables it. The missing clock detector reset is automatically disabled when the device is in the low power suspend or sleep mode. Upon exit from either low power state, the enabled/disabled state of this reset source is restored to its previous value. The state of the RST pin is unaffected by this reset.

22.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a 1 to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0–), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read 1 signifying Comparator0 as the reset source; otherwise, this bit reads 0. The Comparator0 reset source remains functional even when the device is in the low power suspend and sleep states as long as Comparator0 is also enabled as a wake-up source. The state of the RST pin is unaffected by this reset.

22.6. PCA Watchdog Timer Reset

The programmable watchdog timer (WDT) function of the programmable counter array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in Section “33.4. Watchdog Timer Mode” on page 477; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to 1. The PCA Watchdog Timer reset source is automatically disabled when the device is in the low power suspend or sleep mode. Upon exit from either low power state, the enabled/disabled state of this reset source is restored to its previous value.The state of the RST pin is unaffected by this reset.

22.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to 1 and a MOVX write operation targets an address above the Lock Byte address.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above the Lock Byte address.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above the Lock Byte address.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section "18.3. Security Options" on page 247).
- A Flash write or erase is attempted while the V_{DD} Monitor is disabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{RST}$ pin is unaffected by this reset.

22.8. SmaRTClock (Real Time Clock) Reset

The SmaRTClock can generate a system reset on two events: SmaRTClock Oscillator Fail or SmaRTClock Alarm. The SmaRTClock Oscillator Fail event occurs when the SmaRTClock Missing Clock Detector is enabled and the SmaRTClock clock is below approximately 20 kHz. A SmaRTClock alarm event occurs when the SmaRTClock Alarm is enabled and the SmaRTClock timer value matches the ALARMn registers. The SmaRTClock can be configured as a reset source by writing a 1 to the RTC0RE flag (RSTSRC.7). The SmaRTClock reset remains functional even when the device is in the low power Suspend or Sleep mode. The state of the $\overline{RST}$ pin is unaffected by this reset.

22.9. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit (RSTSRC.4). The SWRSF bit will read 1 following a software forced reset. The state of the $\overline{RST}$ pin is unaffected by this reset.

SFR Definition 22.2. RSTSRC: Reset Source

Bit	7	6	5	4	3	2	1	0
Name	RTC0RE	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF
Type	R/W	R	R/W	R/W	R	R/W	R/W	R
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Page = 0x0; SFR Address = 0xEF.

Bit	Name	Description	Write	Read
7	RTC0RE	**SmaRTClock Reset Enable and Flag**	0: Disable SmaRTClock as a reset source. 1: Enable SmaRTClock as a reset source.	Set to 1 if SmaRTClock alarm or oscillator fail caused the last reset.
6	FERROR	**Flash Error Reset Flag.**	N/A	Set to 1 if Flash read/write/erase error caused the last reset.
5	C0RSEF	**Comparator0 Reset Enable and Flag.**	0: Disable Comparator0 as a reset source. 1: Enable Comparator0 as a reset source.	Set to 1 if Comparator0 caused the last reset.
4	SWRSF	**Software Reset Force and Flag.**	Writing a 1 forces a system reset.	Set to 1 if last reset was caused by a write to SWRSF.
3	WDTRSF	**Watchdog Timer Reset Flag.**	N/A	Set to 1 if Watchdog Timer overflow caused the last reset.
2	MCDRSF	**Missing Clock Detector (MCD) Enable and Flag.**	0: Disable the MCD. 1: Enable the MCD. The MCD triggers a reset if a missing clock condition is detected.	Set to 1 if Missing Clock Detector timeout caused the last reset.
1	PORSF	**Power-On / Power-Fail Reset Flag, and Power-Fail Reset Enable.**	0: Disable the VDD Supply Monitor as a reset source. 1: Enable the VDD Supply Monitor as a reset source.[3]	Set to 1 anytime a power-on or V_{DD} monitor reset occurs.[2]
0	PINRSF	**HW Pin Reset Flag.**	N/A	Set to 1 if $\overline{\text{RST}}$ pin caused the last reset.

Notes:
1. It is safe to use read-modify-write operations (ORL, ANL, etc.) to enable or disable specific interrupt sources.
2. If PORSF read back 1, the value read from all other bits in this register are indeterminate.
3. Writing a 1 to PORSF before the VDD Supply Monitor is stabilized may generate a system reset.

23. Clocking Sources

C8051F96x devices include a programmable precision internal oscillator, an external oscillator drive circuit, a low power internal oscillator, and a SmaRTClock real time clock oscillator. The precision internal oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 23.1. The external oscillator can be configured using the OSCXCN register. The low power internal oscillator is automatically enabled and disabled when selected and deselected as a clock source. SmaRTClock operation is described in the SmaRTClock oscillator chapter.

The system clock (SYSCLK) can be derived from the precision internal oscillator, external oscillator, low power internal oscillator, low power internal oscillator divided by 8, or SmaRTClock oscillator. The global clock divider can generate a system clock that is 1, 2, 4, 8, 16, 32, 64, or 128 times slower that the selected input clock source. Oscillator electrical specifications can be found in the Electrical Specifications Chapter.

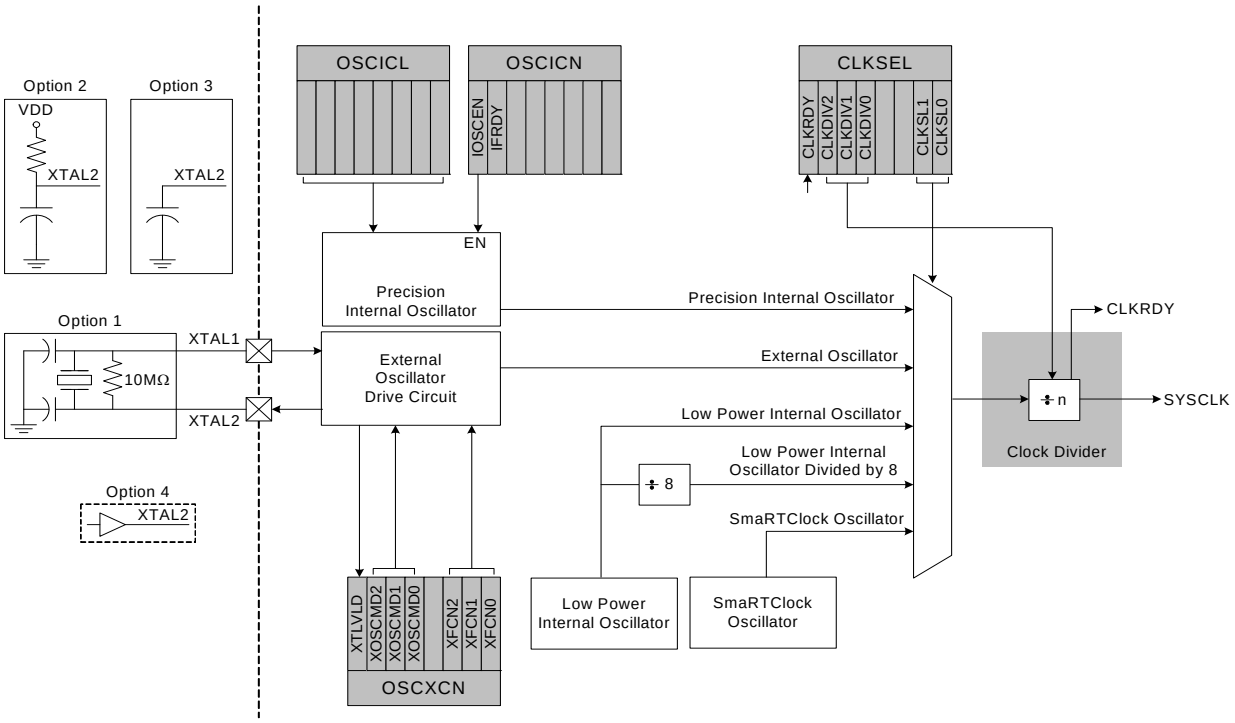


Figure 23.1. Clocking Sources Block Diagram

The proper way of changing the system clock when both the clock source and the clock divide value are being changed is as follows:

If switching from a fast “undivided” clock to a slower “undivided” clock:

1. Change the clock divide value.
2. Poll for CLKRDY > 1.
3. Change the clock source.

If switching from a slow “undivided” clock to a faster “undivided” clock:

1. Change the clock source.
2. Change the clock divide value.
3. Poll for CLKRDY > 1.

23.1. Programmable Precision Internal Oscillator

All C8051F96x devices include a programmable precision internal oscillator that may be selected as the system clock. OSCICL is factory calibrated to obtain a 24.5 MHz frequency. See Section “4. Electrical Characteristics” on page 56 for complete oscillator specifications.

The precision oscillator supports a spread spectrum mode which modulates the output frequency in order to reduce the EMI generated by the system. When enabled (SSE = 1), the oscillator output frequency is modulated by a stepped triangle wave whose frequency is equal to the oscillator frequency divided by 384 (63.8 kHz using the factory calibration). The deviation from the nominal oscillator frequency is +0%, –1.6%, and the step size is typically 0.26% of the nominal frequency. When using this mode, the typical average oscillator frequency is lowered from 24.5 MHz to 24.3 MHz.

23.2. Low Power Internal Oscillator

All C8051F96x devices include a low power internal oscillator that defaults as the system clock after a system reset. The low power internal oscillator frequency is 20 MHz ± 10% and is automatically enabled when selected as the system clock and disabled when not in use. See Section “4. Electrical Characteristics” on page 56 for complete oscillator specifications.

23.3. External Oscillator Drive Circuit

All C8051F96x devices include an external oscillator circuit that may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. Figure 23.1 shows a block diagram of the four external oscillator options. The external oscillator is enabled and configured using the OSCXCN register.

The external oscillator output may be selected as the system clock or used to clock some of the digital peripherals (e.g., Timers, PCA, etc.). See the data sheet chapters for each digital peripheral for details. See Section “4. Electrical Characteristics” on page 56 for complete oscillator specifications.

23.3.1. External Crystal Mode

If a crystal or ceramic resonator is used as the external oscillator, the crystal/resonator and a 10 MΩ resistor must be wired across the XTAL1 and XTAL2 pins as shown in Figure 23.1, Option 1. Appropriate loading capacitors should be added to XTAL1 and XTAL2, and both pins should be configured for analog I/O with the digital output drivers disabled.

Figure 23.2 shows the external oscillator circuit for a 20 MHz quartz crystal with a manufacturer recommended load capacitance of 12.5 pF. Loading capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins. The total value of the each loading capacitor and the stray capacitance of each XTAL pin should equal 12.5 pF x 2 = 25 pF. With a stray capacitance of 10 pF per pin, the 15 pF capacitors yield an equivalent series capacitance of 12.5 pF across the crystal.

Note: The recommended load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

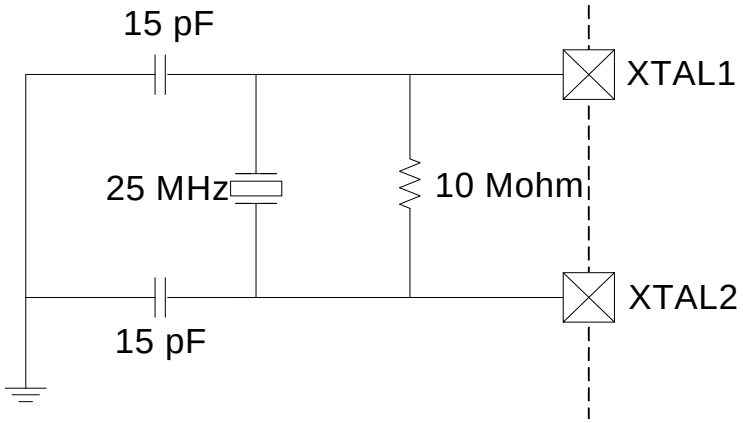


Figure 23.2. 25 MHz External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency. The selection should be based on Table 23.1. For example, a 25 MHz crystal requires an XFCN setting of 111b.

Table 23.1. Recommended XFCN Settings for Crystal Mode

XFCN	Crystal Frequency	Bias Current	Typical Supply Current (VDD = 2.4 V)
000	f ≤ 20 kHz	0.5 µA	3.0 µA, f = 32.768 kHz
001	20 kHz < f ≤ 58 kHz	1.5 µA	4.8 µA, f = 32.768 kHz
010	58 kHz < f ≤ 155 kHz	4.8 µA	9.6 µA, f = 32.768 kHz
011	155 kHz < f ≤ 415 kHz	14 µA	28 µA, f = 400 kHz
100	415 kHz < f ≤ 1.1 MHz	40 µA	71 µA, f = 400 kHz
101	1.1 MHz < f ≤ 3.1 MHz	120 µA	193 µA, f = 400 kHz
110	3.1 MHz < f ≤ 8.2 MHz	550 µA	940 µA, f = 8 MHz
111	8.2 MHz < f ≤ 25 MHz	2.6 mA	3.9 mA, f = 25 MHz

When the crystal oscillator is first enabled, the external oscillator valid detector allows software to determine when the external system clock has stabilized. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure for starting the crystal is as follows:

1. Configure XTAL1 and XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD => 1.
4. Switch the system clock to the external oscillator.

23.3.2. External RC Mode

If an RC network is used as the external oscillator, the circuit should be configured as shown in Figure 23.1, Option 2. The RC network should be added to XTAL2, and XTAL2 should be configured for analog I/O with the digital output drivers disabled. XTAL1 is not affected in RC mode.

The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. The resistor should be no smaller than 10 kΩ. The oscillation frequency can be determined by the following equation:

$$f = \frac{1.23 \times 10^3}{R \times C}$$

where
f = frequency of clock in MHzR = pull-up resistor value in kΩ
V_{DD} = power supply voltage in VoltsC = capacitor value on the XTAL2 pin in pF

To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation. For example, if the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

$$f = \frac{1.23 \times 10^3}{R \times C} = \frac{1.23 \times 10^3}{246 \times 50} = 100 \text{ kHz}$$

where
f = frequency of clock in MHz R = pull-up resistor value in kΩ
V_{DD} = power supply voltage in Volts C = capacitor value on the XTAL2 pin in pF

Referencing Table 23.2, the recommended XFCN setting is 010.

Table 23.2. Recommended XFCN Settings for RC and C modes

XFCN	Approximate Frequency Range (RC and C Mode)	K Factor (C Mode)	Typical Supply Current/ Actual Measured Frequency (C Mode, VDD = 2.4 V)
000	f ≤ 25 kHz	K Factor = 0.87	3.0 µA, f = 11 kHz, C = 33 pF
001	25 kHz < f ≤ 50 kHz	K Factor = 2.6	5.5 µA, f = 33 kHz, C = 33 pF
010	50 kHz < f ≤ 100 kHz	K Factor = 7.7	13 µA, f = 98 kHz, C = 33 pF
011	100 kHz < f ≤ 200 kHz	K Factor = 22	32 µA, f = 270 kHz, C = 33 pF
100	200 kHz < f ≤ 400 kHz	K Factor = 65	82 µA, f = 310 kHz, C = 46 pF
101	400 kHz < f ≤ 800 kHz	K Factor = 180	242 µA, f = 890 kHz, C = 46 pF
110	800 kHz < f ≤ 1.6 MHz	K Factor = 664	1.0 mA, f = 2.0 MHz, C = 46 pF
111	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590	4.6 mA, f = 6.8 MHz, C = 46 pF

When the RC oscillator is first enabled, the external oscillator valid detector allows software to determine when oscillation has stabilized. The recommended procedure for starting the RC oscillator is as follows:

1. Configure XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD $\geq$ 1.
4. Switch the system clock to the external oscillator.

23.3.3. External Capacitor Mode

If a capacitor is used as the external oscillator, the circuit should be configured as shown in Figure 23.1, Option 3. The capacitor should be added to XTAL2, and XTAL2 should be configured for analog I/O with the digital output drivers disabled. XTAL1 is not affected in RC mode.

The capacitor should be no greater than 100 pF; however, for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. The oscillation frequency and the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register can be determined by the following equation:

$$f = \frac{KF}{C \times V_{DD}}$$

where
f = frequency of clock in MHzR = pull-up resistor value in kΩ
V_{DD} = power supply voltage in VoltsC = capacitor value on the XTAL2 pin in pF

Below is an example of selecting the capacitor and finding the frequency of oscillation Assume V_{DD} = 3.0 V and f = 150 kHz:

$$f = \frac{KF}{C \times V_{DD}}$$

$$0.150\ \text{MHz} = \frac{KF}{C \times 3.0}$$

Since a frequency of roughly 150 kHz is desired, select the K Factor from Table 23.2 as KF = 22:

$$0.150\ \text{MHz} = \frac{22}{C \times 3.0\ \text{V}}$$

$$C = \frac{22}{0.150\ \text{MHz} \times 3.0\ \text{V}}$$

$$C = 48.8\ \text{pF}$$

Therefore, the XFCN value to use in this example is 011 and C is approximately 50 pF.

The recommended startup procedure for C mode is the same as RC mode.

23.3.4. External CMOS Clock Mode

If an external CMOS clock is used as the external oscillator, the clock should be directly routed into XTAL2. The XTAL2 pin should be configured as a digital input. XTAL1 is not used in external CMOS clock mode.

The external oscillator valid detector will always return zero when the external oscillator is configured to External CMOS Clock mode.

23.4. Special Function Registers for Selecting and Configuring the System Clock

The clocking sources on C8051F96x devices are enabled and configured using the OSCICN, OSCICL, OSCXCN and the SmaRTClock internal registers. See Section “24. SmaRTClock (Real Time Clock)” on page 295 for SmaRTClock register descriptions. The system clock source for the MCU can be selected using the CLKSEL register. To minimize active mode current, the oneshot timer which sets Flash read time should by bypassed when the system clock is greater than 10 MHz. See the FLSCL register description for details.

The clock selected as the system clock can be divided by 1, 2, 4, 8, 16, 32, 64, or 128. When switching between two clock divide values, the transition may take up to 128 cycles of the undivided clock source. The CLKRDY flag can be polled to determine when the new clock divide value has been applied. The clock divider must be set to "divide by 1" when entering Suspend or Sleep Mode.

The system clock source may also be switched on-the-fly. The switchover takes effect after one clock period of the slower oscillator.

SFR Definition 23.1. CLKSEL: Clock Select

Bit	7	6	5	4	3	2	1	0
Name	CLKRDY	CLKDIV[2:0]				CLKSEL[2:0]		
Type	R	R/W			R/W	R/W		
Reset	0	0	0	1	0	0	1	0

SFR Page = 0x0 and 0xF; SFR Address = 0xA9

Bit	Name	Function
7	CLKRDY	**System Clock Divider Clock Ready Flag.** 0: The selected clock divide setting has not been applied to the system clock. 1: The selected clock divide setting has been applied to the system clock.
6:4	CLKDIV[2:0]	**System Clock Divider Bits.** Selects the clock division to be applied to the undivided system clock source. 000: System clock is divided by 1. 001: System clock is divided by 2. 010: System clock is divided by 4. 011: System clock is divided by 8. 100: System clock is divided by 16. 101: System clock is divided by 32. 110: System clock is divided by 64. 111: System clock is divided by 128.
3	Unused	Read = 0b. Must Write 0b.
2:0	CLKSEL[2:0]	**System Clock Select.** Selects the oscillator to be used as the undivided system clock source. 000: Precision Internal Oscillator. 001: External Oscillator. 010: Low Power Oscillator divided by 8. 011: SmaRTClock Oscillator. 100: Low Power Oscillator. All other values reserved.

SFR Definition 23.2. OSCICN: Internal Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY						
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	Varies	Varies	Varies	Varies	Varies	Varies

SFR Page = 0x0; SFR Address = 0xB2

Bit	Name	Function
7	IOSCEN	**Internal Oscillator Enable.** 0: Internal oscillator disabled. 1: Internal oscillator enabled.
6	IFRDY	**Internal Oscillator Frequency Ready Flag.** 0: Internal oscillator is not running at its programmed frequency. 1: Internal oscillator is running at its programmed frequency.
5:0	Reserved	Must perform read-modify-write.

Notes:

1. Read-modify-write operations such as ORL and ANL must be used to set or clear the enable bit of this register.
2. OSCBIAS (REG0CN.4) must be set to 1 before enabling the precision internal oscillator.

SFR Definition 23.3. OSCICL: Internal Oscillator Calibration

Bit	7	6	5	4	3	2	1	0
Name	SSE	OSCICL[6:0]						
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Page = 0xF; SFR Address = 0xB3

Bit	Name	Function
7	SSE	**Spread Spectrum Enable.** 0: Spread Spectrum clock dithering disabled. 1: Spread Spectrum clock dithering enabled.
6:0	OSCICL	**Internal Oscillator Calibration.** Factory calibrated to obtain a frequency of 24.5 MHz. Incrementing this register decreases the oscillator frequency and decrementing this register increases the oscillator frequency. The step size is approximately 1% of the calibrated frequency. The recommended calibration frequency range is between 16 and 24.5 MHz.

Note: If the Precision Internal Oscillator is selected as the system clock, the following procedure should be used when changing the value of the internal oscillator calibration bits.
1. Switch to a different clock source.
2. Disable the oscillator by writing OSCICN.7 to 0.
3. Change OSCICL to the desired setting.
4. Enable the oscillator by writing OSCICN.7 to 1.

SFR Definition 23.4. OSCXCN: External Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	XCLKVLD	XOSCMD[2:0]				XFCN[2:0]		
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xB1

Bit	Name	Function
7	XCLKVLD	**External Oscillator Valid Flag.** Provides External Oscillator status and is valid at all times for all modes of operation except External CMOS Clock Mode and External CMOS Clock Mode with divide by 2. In these modes, XCLKVLD always returns 0. 0: External Oscillator is unused or not yet stable. 1: External Oscillator is running and stable.
6:4	XOSCMD	**External Oscillator Mode Bits.** Configures the external oscillator circuit to the selected mode. 00x: External Oscillator circuit disabled. 010: External CMOS Clock Mode. 011: External CMOS Clock Mode with divide by 2 stage. 100: RC Oscillator Mode. 101: Capacitor Oscillator Mode. 110: Crystal Oscillator Mode. 111: Crystal Oscillator Mode with divide by 2 stage.
3	Reserved	Read = 0b. Must Write 0b.
2:0	XFCN	**External Oscillator Frequency Control Bits.** Controls the external oscillator bias current. 000-111: See Table 23.1 on page 288 (Crystal Mode) or Table 23.2 on page 289 (RC or C Mode) for recommended settings.

24. SmaRTClock (Real Time Clock)

C8051F96x devices include an ultra low power 32-bit SmaRTClock Peripheral (Real Time Clock) with alarm. The SmaRTClock has a dedicated 32 kHz oscillator that can be configured for use with or without a crystal. No external resistor or loading capacitors are required. The on-chip loading capacitors are programmable to 16 discrete levels allowing compatibility with a wide range of crystals. The SmaRTClock can operate directly from a 1.8–3.6 V battery voltage and remains operational even when the device goes into its lowest power down mode. The SmaRTClock output can be buffered and routed to a GPIO pin to provide an accurate, low frequency clock to other devices while the MCU is in its lowest power down mode (see "PMU0MD: Power Management Unit Mode" on page 267 for more details). C8051F96x devices also support an ultra low power internal LFO that reduces sleep mode current.

The SmaRTClock allows a maximum of 36 hour 32-bit independent time-keeping when used with a 32.768 kHz Watch Crystal. The SmaRTClock provides an Alarm and Missing SmaRTClock events, which could be used as reset or wakeup sources. See Section "22. Reset Sources" on page 278 and Section "19. Power Management" on page 257 for details on reset sources and low power mode wake-up sources, respectively.

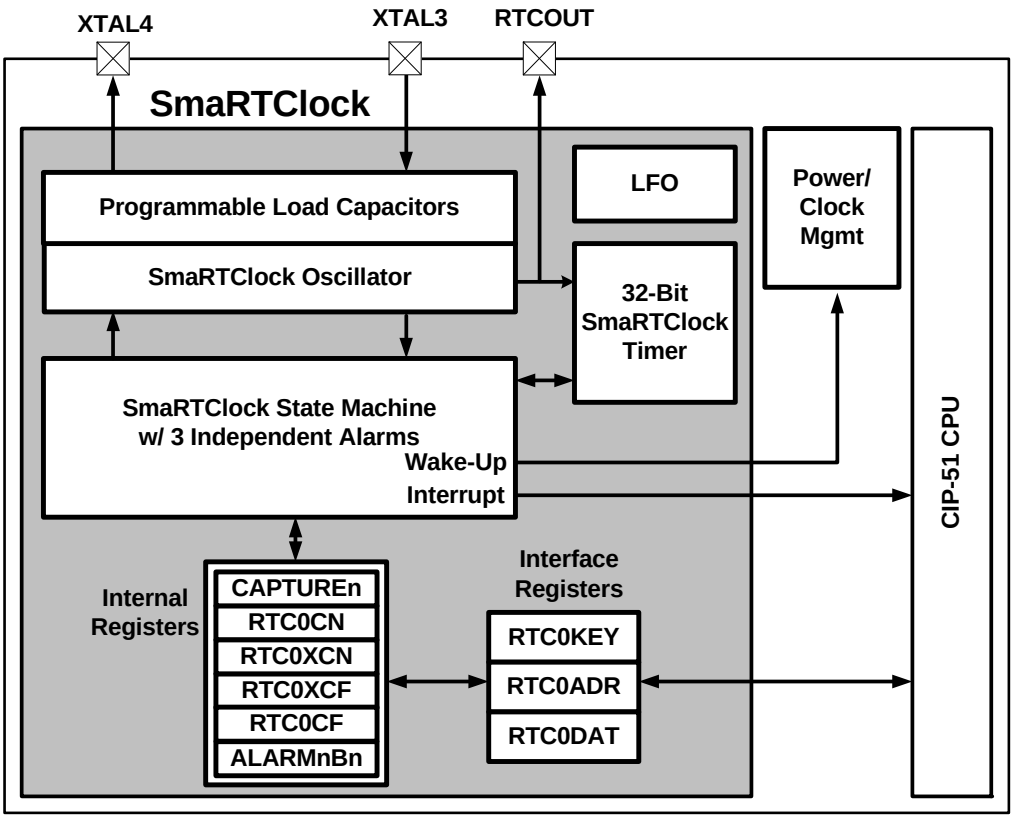


Figure 24.1. SmaRTClock Block Diagram

24.1. SmaRTClock Interface

The SmaRTClock Interface consists of three registers: RTC0KEY, RTC0ADR, and RTC0DAT. These interface registers are located on the CIP-51's SFR map and provide access to the SmaRTClock internal registers listed in Table 24.1. The SmaRTClock internal registers can only be accessed indirectly through the SmaRTClock Interface.

Table 24.1. SmaRTClock Internal Registers

SmaRTClock Address	SmaRTClock Register	Register Name	Description
0x00–0x03	CAPTUREn	SmaRTClock Capture Registers	Four Registers used for setting the 32-bit SmaRTClock timer or reading its current value.
0x04	RTC0CN	SmaRTClock Control Register	Controls the operation of the SmaRTClock State Machine.
0x05	RTC0XCN	SmaRTClock Oscillator Control Register	Controls the operation of the SmaRTClock Oscillator.
0x06	RTC0XCF	SmaRTClock Oscillator Configuration Register	Controls the value of the progammable oscillator load capacitance and enables/disables AutoStep.
0x07	RTC0CF	SmaRTClock Configuration Register	Contains an alarm enable and flag for each SmaRTClock alarm.
0x08–0x0B	ALARM0Bn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.
0x0C–0x0F	ALARM1Bn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.
0x10–0x13	ALARM2Bn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.

24.1.1. SmaRTClock Lock and Key Functions

The SmaRTClock Interface has an RTC0KEY register for legacy reasons, however, all writes to this register are ignored. The SmaRTClock interface is always unlocked on C8051F96x.

24.1.2. Using RTC0ADR and RTC0DAT to Access SmaRTClock Internal Registers

The SmaRTClock internal registers can be read and written using RTC0ADR and RTC0DAT. The RTC0ADR register selects the SmaRTClock internal register that will be targeted by subsequent reads or writes. A SmaRTClock Write operation is initiated by writing to the RTC0DAT register. Below is an example of writing to a SmaRTClock internal register.

1. Write 0x05 to RTC0ADR. This selects the internal RTC0CN register at SmaRTClock Address 0x05.
2. Write 0x00 to RTC0DAT. This operation writes 0x00 to the internal RTC0CN register.

A SmaRTClock Read operation is initiated by writing the register address to RTC0ADR and reading from RTC0DAT. Below is an example of reading a SmaRTClock internal register.

1. Write 0x05 to RTC0ADR. This selects the internal RTC0CN register at SmaRTClock Address 0x05.
2. Read data from RTC0DAT. This data is a copy of the RTC0CN register.

24.1.3. SmaRTClock Interface Autoread Feature

When Autoread is enabled, each read from RTC0DAT initiates the next indirect read operation on the SmaRTClock internal register selected by RTC0ADR. Software should set the register address once at the beginning of each series of consecutive reads. Autoread is enabled by setting AUTORD (RTC0ADR.6) to logic 1.

24.1.4. RTC0ADR Autoincrement Feature

For ease of reading and writing the 32-bit CAPTURE and ALARM values, RTC0ADR automatically increments after each read or write to a CAPTUREn or ALARMn register. This speeds up the process of setting an alarm or reading the current SmaRTClock timer value. Autoincrement is always enabled.

Recommended Instruction Timing for a multi-byte register read with auto read enabled:

```
mov RTC0ADR, #040h
mov A, RTC0DAT
mov A, RTC0DAT
mov A, RTC0DAT
mov A, RTC0DAT
```

Recommended Instruction Timing for a multi-byte register write:

```
mov RTC0ADR, #010h
mov RTC0DAT, #05h
mov RTC0DAT, #06h
mov RTC0DAT, #07h
mov RTC0DAT, #08h
```

SFR Definition 24.1. RTC0KEY: SmaRTClock Lock and Key

Bit	7	6	5	4	3	2	1	0
Name	RTC0ST[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAE

Bit	Name	Function
7:0	RTC0ST	**SmaRTClock Interface Status.** Provides lock status when read. **Read:** **0x02:** SmaRTClock Interface is unlocked. **Write:** Writes to RTC0KEY have no effect.

SFR Definition 24.2. RTC0ADR: SmaRTClock Address

Bit	7	6	5	4	3	2	1	0
Name		AUTORD		ADDR[4:0]				
Type	R	R/W	R	R/W				
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xAC

Bit	Name	Function
7	Reserved	Read = 0; Write = don't care.
6	AUTORD	**SmaRTClock Interface Autoread Enable.** Enables/disables Autoread. 0: Autoread Disabled. 1: Autoread Enabled.
5	Unused	Read = 0b; Write = Don't Care.
4:0	ADDR[4:0]	**SmaRTClock Indirect Register Address.** Sets the currently selected SmaRTClock register. See Table 24.1 for a listing of all SmaRTClock indirect registers.
Note: The ADDR bits increment after each indirect read/write operation that targets a CAPTUREn or ALARMnBn internal SmaRTClock register.		

SFR Definition 24.3. RTC0DAT: SmaRTClock Data

Bit	7	6	5	4	3	2	1	0
Name	RTC0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0xAD

Bit	Name	Function
7:0	RTC0DAT	**SmaRTClock Data Bits.** Holds data transferred to/from the internal SmaRTClock register selected by RTC0ADR.
Note: Read-modify-write instructions (orl, anl, etc.) should not be used on this register.		

24.2. SmaRTClock Clocking Sources

The SmaRTClock peripheral is clocked from its own timebase, independent of the system clock. The SmaRTClock timebase can be derived from an external CMOS clock, the internal LFO, or the SmaRT-Clock oscillator circuit, which has two modes of operation: Crystal Mode, and Self-Oscillate Mode. The oscillation frequency is 32.768 kHz in Crystal Mode and can be programmed in the range of 10 kHz to 40 kHz in Self-Oscillate Mode. The internal LFO frequency is 16.4 kHz ±20%. The frequency of the SmaRTClock oscillator can be measured with respect to another oscillator using an on-chip timer. See Section “32. Timers” on page 444 for more information on how this can be accomplished.

Note: The SmaRTClock timebase can be selected as the system clock and routed to a port pin. See Section “23. Clocking Sources” on page 286 for information on selecting the system clock source and Section “27. Port Input/Output” on page 351 for information on how to route the system clock to a port pin. The SmaRTClock timebase can also be routed to a port pin while the device is in its ultra low power sleep mode. See the PMU0MD register description for details.

24.2.1. Using the SmaRTClock Oscillator with a Crystal or External CMOS Clock

When using Crystal Mode, a 32.768 kHz crystal should be connected between XTAL3 and XTAL4. No other external components are required. The following steps show how to start the SmaRTClock crystal oscillator in software:

1. Configure the XTAL3 and XTAL4 pins for Analog I/O.
2. Set SmaRTClock to Crystal Mode (XMODE = 1).
3. Disable Automatic Gain Control (AGCEN) and enable Bias Doubling (BIASX2) for fast crystal startup.
4. Set the desired loading capacitance (RTC0XCF).
5. Enable power to the SmaRTClock oscillator circuit (RTC0EN = 1).
6. Wait 20 ms.
7. Poll the SmaRTClock Clock Valid Bit (CLKVLD) until the crystal oscillator stabilizes.
8. Poll the SmaRTClock Load Capacitance Ready Bit (LOADRDY) until the load capacitance reaches its programmed value.
9. Enable Automatic Gain Control (AGCEN) and disable Bias Doubling (BIASX2) for maximum power savings.
10. Enable the SmaRTClock missing clock detector.
11. Wait 2 ms.
12. Clear the PMU0CF wake-up source flags.

In Crystal Mode, the SmaRTClock oscillator may be driven by an external CMOS clock. The CMOS clock should be applied to XTAL3. XTAL34 should be left floating. In this mode, the external CMOS clock is ac coupled into the SmaRTClock and should have a minimum voltage swing of 400 mV. The CMOS clock signal voltage should not exceed VDD or drop below GND. Bias levels closer to VDD will result in lower I/O power consumption because the XTAL3 pin has a built-in weak pull-up. The SmaRTClock oscillator should be configured to its lowest bias setting with AGC disabled. The CLKVLD bit is indeterminate when using a CMOS clock, however, the OSCFAIL bit may be checked 2 ms after SmaRTClock oscillator is powered on to ensure that there is a valid clock on XTAL3. The CLKVLD bit is forced low when BIASX2 is disabled.

24.2.2. Using the SmaRTClock Oscillator in Self-Oscillate Mode

When using Self-Oscillate Mode, the XTAL3 and XTAL4 pins are internally shorted together. The following steps show how to configure SmaRTClock for use in Self-Oscillate Mode:

1. Configure the XTAL3 and XTAL4 pins for analog I/O and disable the digital driver.
2. Set SmaRTClock to Self-Oscillate Mode (XMODE = 0).
3. Set the desired oscillation frequency:
   For oscillation at about 20 kHz, set BIASX2 = 0.
   For oscillation at about 40 kHz, set BIASX2 = 1.
4. The oscillator starts oscillating instantaneously.
5. Fine tune the oscillation frequency by adjusting the load capacitance (RTC0XCF).

24.2.3. Using the Low Frequency Oscillator (LFO)

The low frequency oscillator provides an ultra low power, on-chip clock source to the SmaRTClock. The typical frequency of oscillation is 16.4 kHz ±20%. No external components are required to use the LFO and the XTAL3 and XTAL4 pins may be used for general purpose I/O without any effect on the LFO.

The following steps show how to configure SmaRTClock for use with the LFO:

1. Enable and select the Low Frequency Oscillator (LFOEN = 1).
2. The LFO starts oscillating instantaneously.

When the LFO is enabled, the SmaRTClock oscillator increments bit 1 of the 32-bit timer (instead of bit 0). This effectively multiplies the LFO frequency by 2, making the RTC timebase behave as if a 32.768 kHz crystal is connected at the output.

24.2.4. Programmable Load Capacitance

The programmable load capacitance has 16 values to support crystal oscillators with a wide range of recommended load capacitance. If Automatic Load Capacitance Stepping is enabled, the crystal load capacitors start at the smallest setting to allow a fast startup time, then slowly increase the capacitance until the final programmed value is reached. The final programmed loading capacitor value is specified using the LOADCAP bits in the RTC0XCF register. The LOADCAP setting specifies the amount of on-chip load capacitance and does not include any stray PCB capacitance. Once the final programmed loading capacitor value is reached, the LOADRDY flag will be set by hardware to logic 1.

When using the SmaRTClock oscillator in Self-Oscillate mode, the programmable load capacitance can be used to fine tune the oscillation frequency. In most cases, increasing the load capacitor value will result in a decrease in oscillation frequency.Table 24.2 shows the crystal load capacitance for various settings of LOADCAP.

Table 24.2. SmaRTClock Load Capacitance Settings

LOADCAP	Crystal Load Capacitance	Equivalent Capacitance seen on XTAL3 and XTAL4
0000	4.0 pF	8.0 pF
0001	4.5 pF	9.0 pF
0010	5.0 pF	10.0 pF
0011	5.5 pF	11.0 pF
0100	6.0 pF	12.0 pF
0101	6.5 pF	13.0 pF
0110	7.0 pF	14.0 pF
0111	7.5 pF	15.0 pF
1000	8.0 pF	16.0 pF
1001	8.5 pF	17.0 pF
1010	9.0 pF	18.0 pF
1011	9.5 pF	19.0 pF
1100	10.5 pF	21.0 pF
1101	11.5 pF	23.0 pF
1110	12.5 pF	25.0 pF
1111	13.5 pF	27.0 pF

24.2.5. Automatic Gain Control (Crystal Mode Only) and SmaRTClock Bias Doubling

Automatic Gain Control allows the SmaRTClock oscillator to trim the oscillation amplitude of a crystal in order to achieve the lowest possible power consumption. Automatic Gain Control automatically detects when the oscillation amplitude has reached a point where it safe to reduce the drive current, therefore, it may be enabled during crystal startup. It is recommended to enable Automatic Gain Control in most systems which use the SmaRTClock oscillator in Crystal Mode. The following are recommended crystal specifications and operating conditions when Automatic Gain Control is enabled:

- ESR < 50 kΩ
- Load Capacitance < 10 pF
- Supply Voltage < 3.0 V
- Temperature > –20 °C

When using Automatic Gain Control, it is recommended to perform an oscillation robustness test to ensure that the chosen crystal will oscillate under the worst case condition to which the system will be exposed. The worst case condition that should result in the least robust oscillation is at the following system conditions: lowest temperature, highest supply voltage, highest ESR, highest load capacitance, and lowest bias current (AGC enabled, Bias Double Disabled).

To perform the oscillation robustness test, the SmaRTClock oscillator should be enabled and selected as the system clock source. Next, the SYSCLK signal should be routed to a port pin configured as a push-pull digital output. The positive duty cycle of the output clock can be used as an indicator of oscillation robust-

ness. As shown in Figure 24.2, duty cycles less than 65% indicate a robust oscillation. As the duty cycle approaches 68%, oscillation becomes less reliable and the risk of clock failure increases. Increasing the bias current (by disabling AGC) will always improve oscillation robustness and will reduce the output clock's duty cycle. This test should be performed at the worst case system conditions, as results at very low temperatures or high supply voltage will vary from results taken at room temperature or low supply voltage.

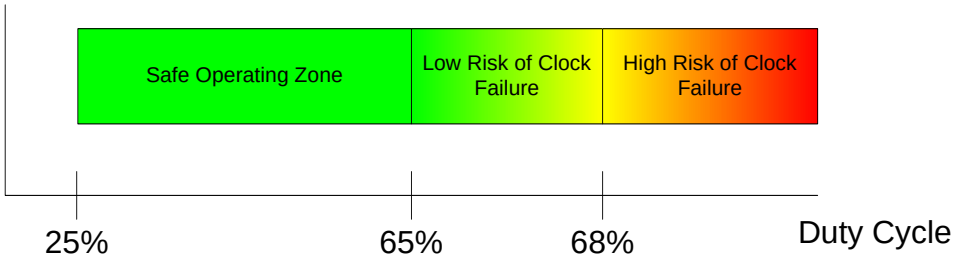


Figure 24.2. Interpreting Oscillation Robustness (Duty Cycle) Test Results

As an alternative to performing the oscillation robustness test, Automatic Gain Control may be disabled at the cost of increased power consumption (approximately 200 nA). Disabling Automatic Gain Control will provide the crystal oscillator with higher immunity against external factors which may lead to clock failure. Automatic Gain Control must be disabled if using the SmaRTClock oscillator in self-oscillate mode.

Table 24.3 shows a summary of the oscillator bias settings. The SmaRTClock Bias Doubling feature allows the self-oscillation frequency to be increased (almost doubled) and allows a higher crystal drive strength in crystal mode. High crystal drive strength is recommended when the crystal is exposed to poor environmental conditions such as excessive moisture. SmaRTClock Bias Doubling is enabled by setting BIASX2 (RTC0XCN.5) to 1.

.

Table 24.3. SmaRTClock Bias Settings

Mode	Setting	Power Consumption
Crystal	Bias Double Off, AGC On	Lowest
	Bias Double Off, AGC Off	Low
	Bias Double On, AGC On	High
	Bias Double On, AGC Off	Highest
Self-Oscillate	Bias Double Off	Low
	Bias Double On	High

24.2.6. Missing SmaRTClock Detector

The missing SmaRTClock detector is a one-shot circuit enabled by setting MCLKEN (RTC0CN.6) to 1. When the SmaRTClock Missing Clock Detector is enabled, OSCFAIL (RTC0CN.5) is set by hardware if SmaRTClock oscillator remains high or low for more than 100 µs.

A SmaRTClock Missing Clock detector timeout can trigger an interrupt, wake the device from a low power mode, or reset the device. See Section “17. Interrupt Handler” on page 232, Section “19. Power Management” on page 257, and Section “22. Reset Sources” on page 278 for more information.

Note: The SmaRTClock Missing Clock Detector should be disabled when making changes to the oscillator settings in RTC0XCN.

24.2.7. SmaRTClock Oscillator Crystal Valid Detector

The SmaRTClock oscillator crystal valid detector is an oscillation amplitude detector circuit used during crystal startup to determine when oscillation has started and is nearly stable. The output of this detector can be read from the CLKVLD bit (RTX0XCN.4).

Notes:

1. The CLKVLD bit has a blanking interval of 2 ms. During the first 2 ms after turning on the crystal oscillator, the output of CLKVLD is not valid.
2. This SmaRTClock crystal valid detector (CLKVLD) is not intended for detecting an oscillator failure. The missing SmaRTClock detector (CLKFAIL) should be used for this purpose.
3. The CLKVLD bit output is driven low when BIASX2 is disabled.

24.3. SmaRTClock Timer and Alarm Function

The SmaRTClock timer is a 32-bit counter that, when running (RTC0TR = 1), is incremented every SmaRTClock oscillator cycle. The timer has an alarm function that can be set to generate an interrupt, wake the device from a low power mode, or reset the device at a specific time. See Section “17. Interrupt Handler” on page 232, Section “19. Power Management” on page 257, and Section “22. Reset Sources” on page 278 for more information.

The SmaRTClock timer includes an Auto Reset feature, which automatically resets the timer to zero one SmaRTClock cycle after the alarm 0 signal is deasserted. When using Auto Reset, the Alarm match value should always be set to 2 counts less than the desired match value. When using the LFO in combination with Auto Reset, the right-justified Alarm match value should be set to 4 counts less than the desired match value. Auto Reset can be enabled by writing a 1 to ALRM (RTC0CN.2).

24.3.1. Setting and Reading the SmaRTClock Timer Value

The 32-bit SmaRTClock timer can be set or read using the six CAPTUREn internal registers. Note that the timer does not need to be stopped before reading or setting its value. The following steps can be used to set the timer value:

1. Write the desired 32-bit set value to the CAPTUREn registers.
2. Write 1 to RTC0SET. This will transfer the contents of the CAPTUREn registers to the SmaRTClock timer.
3. Operation is complete when RTC0SET is cleared to 0 by hardware.

The following steps can be used to read the current timer value:

1. Write 1 to RTC0CAP. This will transfer the contents of the timer to the CAPTUREn registers.
2. Poll RTC0CAP until it is cleared to 0 by hardware.
3. A snapshot of the timer value can be read from the CAPTUREn registers

Notes:

1. If the system clock is faster than 4x the SmaRTClock, then the HSMODE bit should be set to allow the set and capture operations to be concluded quickly (system clock used for transfers).
2. If the system clock is slower than 4x the SmaRTClock, then HSMODE should be set to zero, and RTC must be

running (RTC0TR = 1) in order to set or capture the main timer. The transfer can take up to 2 smaRTClock cycles to complete.

24.3.2. Setting a SmaRTClock Alarm

The SmaRTClock alarm function compares the 32-bit value of SmaRTClock Timer to the value of the ALARMnBn registers. An alarm event is triggered if the SmaRTClock timer is **equal to** the ALARMnBn registers. If Auto Reset is enabled, the 32-bit timer will be cleared to zero one SmaRTClock cycle after the alarm 0 event.

The SmaRTClock alarm event can be configured to reset the MCU, wake it up from a low power mode, or generate an interrupt. See Section “17. Interrupt Handler” on page 232, Section “19. Power Management” on page 257, and Section “22. Reset Sources” on page 278 for more information.

The following steps can be used to set up a SmaRTClock Alarm:

1. Disable SmaRTClock Alarm Events (RTC0AEN = 0).
2. Set the ALARMn registers to the desired value.
3. Enable SmaRTClock Alarm Events (RTC0AEN = 1).

Notes:

1. The ALRM bit, which is used as the SmaRTClock Alarm Event flag, is cleared by disabling SmaRTClock Alarm Events (RTC0AEN = 0).
2. If AutoReset is disabled, disabling (RTC0AEN = 0) then Re-enabling Alarm Events (RTC0AEN = 1) after a SmaRTClock Alarm without modifying ALARMn registers will automatically schedule the next alarm after 2^32 SmaRTClock cycles (approximately 36 hours using a 32.768 kHz crystal).

24.3.3. Software Considerations for using the SmaRTClock Timer and Alarm

The SmaRTClock timer and alarm have two operating modes to suit varying applications. The two modes are described below:

Mode 1:
The first mode uses the SmaRTClock timer as a perpetual timebase which is never reset to zero. Every 36 hours, the timer is allowed to overflow without being stopped or disrupted. The alarm interval is software managed and is added to the ALRMnBn registers by software after each alarm. This allows the alarm match value to always stay ahead of the timer by one software managed interval. If software uses 32-bit unsigned addition to increment the alarm match value, then it does not need to handle overflows since both the timer and the alarm match value will overflow in the same manner.

This mode is ideal for applications which have a long alarm interval (e.g., 24 or 36 hours) and/or have a need for a perpetual timebase. An example of an application that needs a perpetual timebase is one whose wake-up interval is constantly changing. For these applications, software can keep track of the number of timer overflows in a 16-bit variable, extending the 32-bit (36 hour) timer to a 48-bit (272 year) perpetual timebase.

Mode 2:
The second mode uses the SmaRTClock timer as a general purpose up counter which is auto reset to zero by hardware after each alarm 0 event. The alarm interval is managed by hardware and stored in the ALRM0Bn registers. Software only needs to set the alarm interval once during device initialization. After each alarm 0 event, software should keep a count of the number of alarms that have occurred in order to keep track of time. Alarm 1 and alarm 2 events do not trigger the auto reset.

This mode is ideal for applications that require minimal software intervention and/or have a fixed alarm interval. This mode is the most power efficient since it requires less CPU time per alarm.

Internal Register Definition 24.4. RTC0CN: SmaRTClock Control

Bit	7	6	5	4	3	2	1	0
Name	RTC0EN	MCLKEN	OSCFAIL	RTC0TR		HSMODE	RTC0SET	RTC0CAP
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	Varies	0	0	0	0	0

SmaRTClock Address = 0x04

Bit	Name	Function
7	RTC0EN	**SmaRTClock Enable.** Enables/disables the SmaRTClock oscillator and associated bias currents. 0: SmaRTClock oscillator disabled. 1: SmaRTClock oscillator enabled.
6	MCLKEN	**Missing SmaRTClock Detector Enable.** Enables/disables the missing SmaRTClock detector. 0: Missing SmaRTClock detector disabled. 1: Missing SmaRTClock detector enabled.
5	OSCFAIL	**SmaRTClock Oscillator Fail Event Flag.** Set by hardware when a missing SmaRTClock detector timeout occurs. Must be cleared by software. The value of this bit is not defined when the SmaRTClock oscillator is disabled.
4	RTC0TR	**SmaRTClock Timer Run Control.** Controls if the SmaRTClock timer is running or stopped (holds current value). 0: SmaRTClock timer is stopped. 1: SmaRTClock timer is running.
3	Reserved	Read = 0b; Must write 0b.
2	HSMODE	**High Speed Mode Enable.** Should be set to 1 if the system clock is faster than 4x the SmaRTClock frequency. 0: High Speed Mode is disabled. 1: High Speed Mode is enabled.
1	RTC0SET	**SmaRTClock Timer Set.** Writing 1 initiates a SmaRTClock timer set operation. This bit is cleared to 0 by hardware to indicate that the timer set operation is complete.
0	RTC0CAP	**SmaRTClock Timer Capture.** Writing 1 initiates a SmaRTClock timer capture operation. This bit is cleared to 0 by hardware to indicate that the timer capture operation is complete.

Internal Register Definition 24.5. RTC0XCN: SmaRTClock Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	AGCEN	XMODE	BIASX2	CLKVLD	LFOEN			
Type	R/W	R/W	R/W	R	R/W	R	R	R
Reset	0	0	0	0	0	0	0	0

SmaRTClock Address = 0x05

Bit	Name	Function
7	AGCEN	**SmaRTClock Oscillator Automatic Gain Control (AGC) Enable.** 0: AGC disabled. 1: AGC enabled.
6	XMODE	**SmaRTClock Oscillator Mode.** Selects Crystal or Self Oscillate Mode. 0: Self-Oscillate Mode selected. 1: Crystal Mode selected.
5	BIASX2	**SmaRTClock Oscillator Bias Double Enable.** **Enables/disables the Bias Double feature.** 0: Bias Double disabled. 1: Bias Double enabled.
4	CLKVLD	**SmaRTClock Oscillator Crystal Valid Indicator.** Indicates if oscillation amplitude is sufficient for maintaining oscillation. This bit always reads 0 when BIASX2 is disabled. 0: Oscillation has not started or oscillation amplitude is too low to maintain oscillation. 1: Sufficient oscillation amplitude detected.
3	LFOEN	**Low Frequency Oscillator Enable and Select.** Overrides XMODE and selects the internal low frequency oscillator (LFO) as the SmaRTClock oscillator source. 0: XMODE determines SmaRTClock oscillator source. 1: LFO enabled and selected as SmaRTClock oscillator source.
2:0	Unused	Read = 000b; Write = Don't Care.

Internal Register Definition 24.6. RTC0XCF: SmaRTClock Oscillator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AUTOSTP	LOADRDY			LOADCAP			
Type	R/W	R	R	R	R/W			
Reset	0	0	0	0	Varies	Varies	Varies	Varies

SmaRTClock Address = 0x06

Bit	Name	Function
7	AUTOSTP	**Automatic Load Capacitance Stepping Enable.** Enables/disables automatic load capacitance stepping. 0: Load capacitance stepping disabled. 1: Load capacitance stepping enabled.
6	LOADRDY	**Load Capacitance Ready Indicator.** Set by hardware when the load capacitance matches the programmed value. 0: Load capacitance is currently stepping. 1: Load capacitance has reached it programmed value.
5:4	Unused	Read = 00b; Write = Don't Care.
3:0	LOADCAP	**Load Capacitance Programmed Value.** Holds the user's desired value of the load capacitance. See Table 24.2 on page 302.

Internal Register Definition 24.7. RTC0CF: SmaRTClock Configuration

Bit	7	6	5	4	3	2	1	0
Name		ALRM2	ALRM1	ALRM0	AUTORST	RTC2EN	RTC1EN	RTC0EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SmaRTClock Address = 0x07

Bit	Name	Function
7	Reserved	Read = 0b; Must write 0b.
6	ALRM2	**Event Flag for Alarm 2.** This bit must be cleared by software. Writing a '1' to this bit has no effect. 0: An Alarm 2 event has not occured since the last time the flag was cleared. 1: An Alarm 2 event has occured.
5	ALRM1	**Event Flag for Alarm 1.** This bit must be cleared by software. Writing a '1' to this bit has no effect. 0: An Alarm 1 event has not occured since the last time the flag was cleared. 1: An Alarm 1 event has occured.
4	ALRM0	**Event Flag for Alarm 0.** This bit must be cleared by software. Writing a '1' to this bit has no effect. 0: An Alarm 0 event has not occured since the last time the flag was cleared. 1: An Alarm 0 event has occured.
3	AUTORST	**Auto Reset Enable.** Enables the Auto Reset function to clear the counter when an Alarm 0 event occurs. 0: Auto Reset is disabled 1: Auto Reset is enabled.
2	ALRM2EN	**Alarm 2 Enable.** 0: Alarm 2 is disabled. 1: Alarm 2 is enabled.
1	ALRM1EN	**Alarm 1 Enable.** 0: Alarm 1 is disabled. 1: Alarm 1 is enabled.
0	ALRM0EN	**Alarm 0 Enable.** 0: Alarm 0 is disabled. 1: Alarm 0 is enabled.

Internal Register Definition 24.8. CAPTUREn: SmaRTClock Timer Capture

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE[31:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SmaRTClock Addresses: CAPTURE0 = 0x00; CAPTURE1 = 0x01; CAPTURE2 =0x02; CAPTURE3: 0x03.

Bit	Name	Function
7:0	CAPTURE[31:0]	**SmaRTClock Timer Capture.** These 4 registers (CAPTURE3–CAPTURE0) are used to read or set the 32-bit SmaRTClock timer. Data is transferred to or from the SmaRTClock timer when the RTC0SET or RTC0CAP bits are set.
Note: The least significant bit of the timer capture value is CAPTURE0.0.		

Internal Register Definition 24.9. ALARM0Bn: SmaRTClock Alarm 0 Match Value

Bit	7	6	5	4	3	2	1	0
Name	ALARM0[31:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SmaRTClock Address: ALARM0B0 = 0x08; ALARM0B1 = 0x09; ALARM0B2 = 0x0A; ALARM0B3 = 0x0B

Bit	Name	Function
7:0	ALARM0[31:0]	**SmaRTClock Alarm 0 Programmed Value.** These 4 registers (ALARM0B3–ALARM0B0) are used to set an alarm event for the SmaRTClock timer. The SmaRTClock alarm should be disabled (ALRM0EN=0) when updating these registers.
Note: The least significant bit of the alarm programmed value is ALARM0B0.0.		

Internal Register Definition 24.10. ALARM1Bn: SmaRTClock Alarm 1 Match Value

Bit	7	6	5	4	3	2	1	0
Name	ALARM1[31:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SmaRTClock Address: ALARM1B0 = 0x0C; ALARM1B1 = 0x0D; ALARM1B2 = 0x0E; ALARM1B3 = 0x0F

Bit	Name	Function
7:0	ALARM1[31:0]	**SmaRTClock Alarm 1 Programmed Value.** These 4 registers (ALARM1B3–ALARM1B0) are used to set an alarm event for the SmaRTClock timer. The SmaRTClock alarm should be disabled (ALRM1EN=0) when updating these registers.
Note: The least significant bit of the alarm programmed value is iALARM1B0.0.		

Internal Register Definition 24.11. ALARM2Bn: SmaRTClock Alarm 2 Match Value

Bit	7	6	5	4	3	2	1	0
Name	ALARM2[31:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SmaRTClock Address: ALARM2B0 = 0x10; ALARM2B1 = 0x11; ALARM2B2 = 0x12; ALARM2B3 = 0x13

Bit	Name	Function
7:0	ALARM2[31:0]	**SmaRTClock Alarm 2 Programmed Value.** These 4 registers (ALARM2B3–ALARM2B0) are used to set an alarm event for the SmaRTClock timer. The SmaRTClock alarm should be disabled (ALRM2EN=0) when updating these registers.
Note: The least significant bit of the alarm programmed value is ALARM2B0.0.		

25. Low-Power Pulse Counter

The C8051F960x family of microcontrollers contains a low-power Pulse Counter module with advanced features, such as ultra low power input comparators, a wide range of pull up values with a self calibration engine, asymmetrical integrators for low pass filtering and switch debounce, single, dual, and quadrature modes of operation, two 24-bit counters, threshold comparators, and a variety of interrupt and sleep wake up capabilities. This combination of features provides water, gas, and heat metering system designers with an optimal tool for saving power while collecting meter usage data.

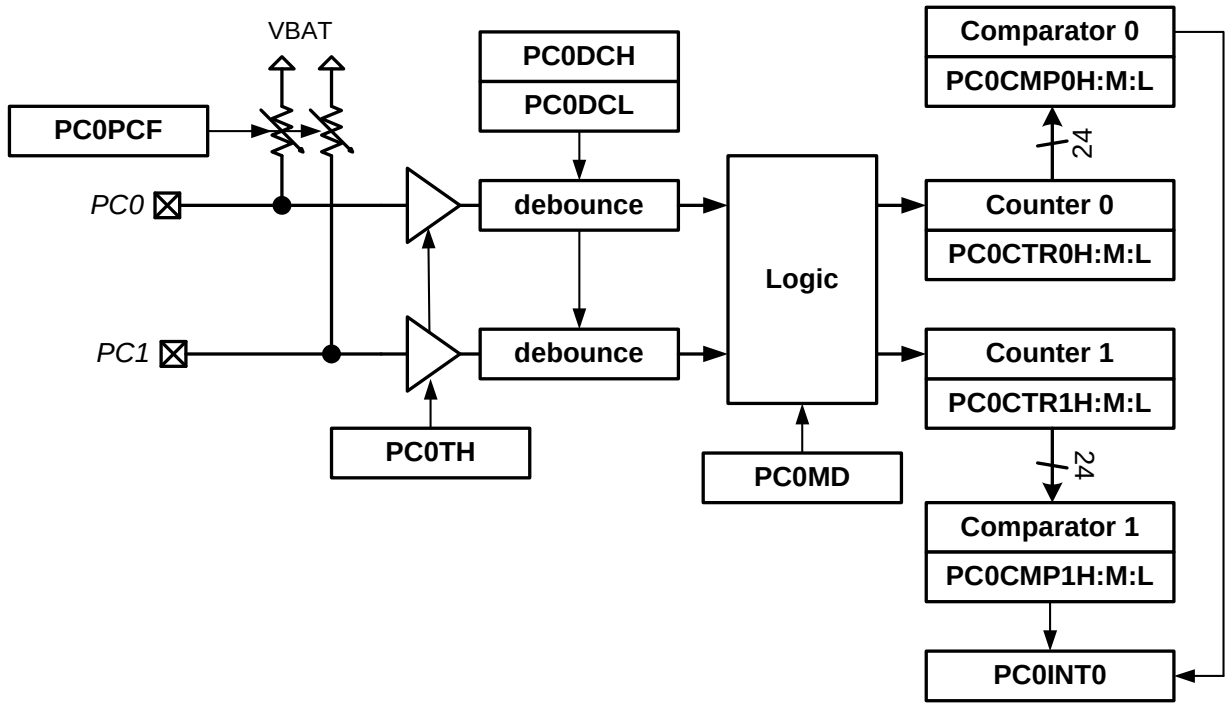


Figure 25.1. Pulse Counter Block Diagram

The low-power Pulse Counter is a low-power sleep-mode peripheral designed primarily to work meters using reed switches, including water and gas meters. The Pulse Counter is very flexible and can count pulses from many different types of sources.

The Pulse Counter operates in sleep mode to enable ultra-low power metering systems. The MCU does not have to wake up on every edge or transition and can remain in sleep mode while the Pulse Counter counts pulses for an extended period of time. The Pulse Counter includes two 24-bit counters. These counters can count up to 16,777,215 (2^{24}-1) transitions in sleep mode before overflowing. The Pulse Counter can wake up the MCU when one of the counters overflows. The Pulse Counter also has two 24-bit comparators. The comparators have the ability to wake up the MCU when the one of the counters reaches a predetermined threshold.

The Pulse Counter uses the RTC clock for sampling, de-bouncing, and managing the low-power pull-up resistors. The RTC must be enabled when counting pulses. The RTC alarms can wake up the MCU periodically to read the pulse counters, instead of using the Pulse Counter comparators. For example, the RTC can wake up the MCU every five minutes. The MCU can then read the Pulse Counter and transmit the information using the UART or a wireless transceiver.

25.1. Counting Modes

The Pulse Counter supports three different counting modes: single counter mode, dual counter mode, and quadrature counter mode. Figure 25.2 illustrates the three counter modes.

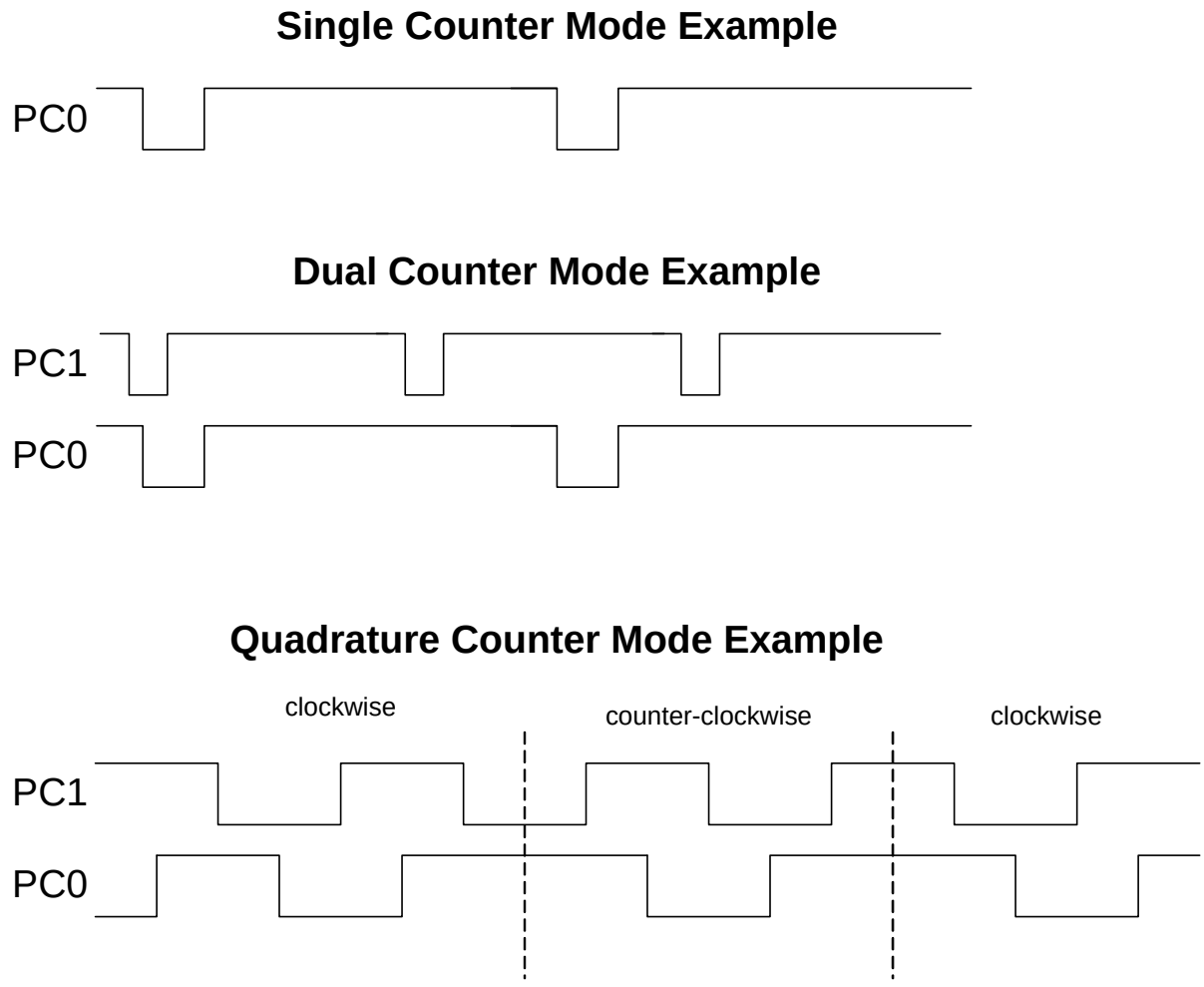


Figure 25.2. Mode Examples

The single counter mode uses only one Pulse Counter pin PC0 (P1.0) to count pulses from a single input channel. This mode uses only counter 0 and comparator. (Counter 1 and comparator 1 are not used.) The single counter mode supports only one meter-encoder with a single-channel output. A single-channel encoder is an effective solution when the metered fluid flows only in one direction. A single-channel encoder does not provide any direction information and does not support bidirectional fluid metering.

The dual counter mode supports two independent single-channel meters. Each meter has its own independent counter and comparator. Some of the global configuration settings apply to both channels, such as pull-up current, sampling rate, and debounce time. The dual mode may also be used for a redundant count using a two-channel non-quadrature encoder.

Quadrature counter mode supports a single two-channel quadrature meter encoder. The quadrature counter mode supports bidirectional encoders and applications with bidirectional fluid flow. In quadrature counter mode, clock-wise counts will increment counter 0, while counter clock-wise counts will increment counter 1. Subtracting counter 1 from counter 0 will yield the net position. If the normal fluid flow is clock-

wise, then the counter clockwise counter 1 value represents the cumulative back-flow. Firmware may use the back-flow counter with the corresponding comparator to implement a back-flow alarm. The clock-wise sequence is (LL-HL-HH-LH), and the counter clock-wise sequence is (LL-LH-HH-HL). (For this sequence LH means PC1 = Low and PC0 = High.)

Firmware cannot write to the counters. The counters are reset when PC0MD is written and have their counting enabled when the PC0MD[7:6] mode bits are set to either single, dual, or quadrature modes. The counters only increment and will roll over to 0x000000 after reaching 0xFFFFFF. For single mode, the PC0 input connects to counter 0. In dual mode, the PC0 input connects to counter 0 while the PC1 input connects to counter 1. In Quadrature mode, clock-wise counts are sent to counter 0 while counter clock-wise counts are sent to counter 1.

25.2. Reed Switch Types

The Pulse Counter works with both Form-A and Form-C reed switches. A Form-A switch is a Normally-Open Single-Pole Single-Throw (NO SPST) switch. A Form-C reed switch is a Single-Pole Double-Throw (SPDT) switch. Figure 25.3 illustrates some of the common reed switch configurations for a single-channel meter.

The Form-A switch requires a pull-up resistor. The energy used by the pull-up resistor may be a substantial portion of the energy budget. To minimize energy usage, the Pulse Counter has a programmable pull-up resistance and an automatic calibration engine. The calibration engine can automatically determine the smallest usable pull-up strength setting. A Form-C switch does not require a pull-up resistor and will provide a lower power solution. However, the Form-C switches are more expensive and require an additional wire for VBAT.

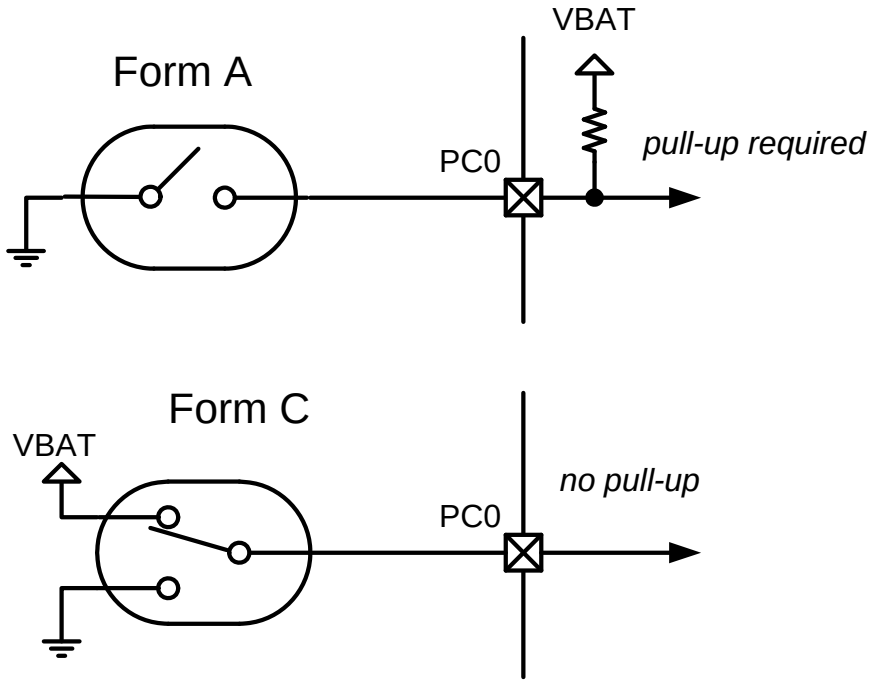


Figure 25.3. Reed Switch Configurations

25.3. Programmable Pull-Up Resistors

The Pulse Counter features low-power pull-up resistors with a programmable resistance and duty-cycle. The average pull-up current will depend on the selected resistor, sample rate, and pull-up duty-cycle multiplier. Example code is available that will calculate the values for the Pull-Up configuration SFR (PC0PCF).

Table 25.1through Table 25.3 are used with Equation 25.1 to calculate the average pull-up resistor current. Table 25.4through Table 25.7 give the average current for all combinations.

$$I_{pull\text{-}up} = I_R \times D_{SR} \times D_{PU}$$

Equation 25.1. Average Pull-Up Current

Where:

I_R = Pull-up Resistor current selected by PC0PCF[4:2].

D_{SR} = Sample Rate Duty Cycle Multiplier selected by PC0MD[5:4].

D_{PU} = Pull-Up Duty Cycle Multiplier selected by PC0PCF[4:2].

Table 25.1. Pull-Up Resistor Current

PC0PCF[4:2]	I_R
000	0
001	1 μA
010	4 μA
011	16 μA
100	64 μA
101	256 μA
110	1 mA
111	4 mA

Table 25.2. Sample Rate Duty-Cycle Multiplier

PC0MD[5:4]	D_{SR}
000	1
001	1/2
010	1/4
011	1/8

Table 25.3. Pull-Up Duty-Cycle Multiplier

PC0PCF[4:2]	D_{PU}
000	1/4
001	3/8
010	1/2
011	3/4

Table 25.4. Average Pull-Up Current (Sample Rate = 250 µs)

	PC0PCF[4:2]								Duty Cycle
PC0PCF[1:0]	000	001	010	011	100	101	110	111	
00	disabled	250 nA	1.0 µA	4.0 µA	16 µA	64 µA	250 µA	1000 µA	25%
01	disabled	375 nA	1.5 µA	6.0 µA	24 µA	96 µA	375 µA	1500 µA	37.5%
10	disabled	500 nA	2.0 µA	8.0 µA	32 µA	128 µA	500 µA	2000 µA	50%
11	disabled	750 nA	3.0 µA	12.0 µA	48 µA	192 µA	750 µA	3000 µA	75%

Table 25.5. Average Pull-Up Current (Sample Rate = 500 µs)

	PC0PCF[4:2]								Duty Cycle
PC0PCF[1:0]	000	001	010	011	100	101	110	111	
00	disabled	125 nA	0.50 µA	2.0 µA	8 µA	32 µA	125 µA	500 µA	12.5%
01	disabled	188 nA	0.75 µA	3.0 µA	12 µA	48 µA	188 µA	750 µA	18.8%
10	disabled	250 nA	1.0 µA	4.0 µA	16 µA	64 µA	250 µA	1000 µA	25%
11	disabled	375 nA	1.5 µA	6.0 µA	24 µA	96 µA	375 µA	1500 µA	37.5%

Table 25.6. Average Pull-Up Current (Sample Rate = 1 ms)

	PC0PCF[4:2]								Duty Cycle
PC0PCF[1:0]	000	001	010	011	100	101	110	111	
00	disabled	63 nA	250 nA	1.0 µA	4 µA	16 µA	63 µA	250 µA	6.3%
01	disabled	94 nA	375 nA	1.5 µA	6 µA	24 µA	94 µA	375 µA	9.4%
10	disabled	125 nA	500 nA	2.0 µA	8 µA	32 µA	125 µA	500 µA	12.5%
11	disabled	188 nA	750 nA	3.0 µA	12 µA	48 µA	188 µA	750 µA	18.8%

Table 25.7. Average Pull-Up Current (Sample Rate = 2 ms)

	PC0PCF[4:2]								Duty Cycle
PC0PCF[1:0]	000	001	010	011	100	101	110	111	
00	disabled	31 nA	125 nA	0.50 µA	2.0 µA	8 µA	31 µA	125 µA	3.1%
01	disabled	47 nA	188 nA	0.75 µA	3.0 µA	12 µA	47 µA	188 µA	4.7%
10	disabled	63 nA	250 nA	1.0 µA	4.0 µA	16 µA	63 µA	250 µA	6.3%
11	disabled	94 nA	375 nA	1.5 µA	6.0 µA	24 µA	94 µA	375 µA	9.4%

25.4. Automatic Pull-Up Resistor Calibration

The Pulse Counter includes an automatic calibration engine which can automatically determine the minimum pull-up current for a particular application. The automatic calibration is especially useful when the load capacitance of field wiring varies from one installation to another.

The automatic calibration uses one of the Pulse Counter inputs (PC0 or PC1) for calibration. The CALPORT bit in the PC0PCF SFR selects either PC0 or PC1 for calibration. The reed switch on the selected input should be in the open state to allow the signal to charge during calibration. The calibration engine can calibrate the pull-ups with the meter connected normally, provided that the reed switch is open during calibration. During calibration, the integrators will ignore the input comparators, and the counters will not be incremented. Using a 250 µs sample rate and a 32 kHz RTCCLK, the calibration time will be 21 ms (28 tests @ 750 µs each) or shorter depending on the pull up strength selected. The calibration will fail if the reed switch remains closed during this entire period. If the reed switch is both opened and closed during the calibration period, the value written into PCCF[4:0] may be larger than what is actually required. The transition flag in the PC0INT1 can detect when the reed switch opens, and most systems with a wheel rotation of 10 Hz or slower should have sufficient high time for the calibration to complete before the next closing of the reed switch. Slowing the sample rate will also increase the calibration time. The same drive strength will used for both PC0 and PC1.

The example code for the Pulse Counter includes code for managing the automatic calibration engine.

25.5. Sample Rate

The Pulse Counter has a programmable sampling rate. The Pulse Counter samples the state of the reed switches at discrete time intervals based on the RTC clock. The PC0MD SFR sets the sampling rate. The system designer should carefully consider the maximum pulse rate for the particular application when setting the sampling rate and debounce time. Sample rates from 250 µs to 2 ms can be selected to either further reduce power consumption or work with shorter pulse widths. The slowest sampling rate (2 ms) will provide the lowest possible power consumption.

25.6. Debounce

Like most mechanical switches, reed switches exhibit switch bouncing that could potentially result in false counts or quadrature errors. The Pulse Counter includes digital debounce logic using a digital integrator that can eliminate false counts due to switch bounce. The input of the integrator connects to the Pulse Counter inputs with the programmable pull-ups. The output connects to the counters.

The debounce integrator has two independent programmable thresholds: one for the rising edge (Debounce High) and one for the falling edge (Debounce Low). The PC0DCH (PC0 Debounce Config High) SFR sets the threshold for the rising edge. This SFR sets the number of cumulative high samples required to output a logic high to the counter. The PC0DCL (PC0 Debounce Config Low) SFR sets the threshold for the falling edge. This SFR sets the number of cumulative high samples required to output a logic low to the counter.

Note that the debounce does count consecutive samples. Requiring consecutive samples would be susceptible to noise. The digital integrator inherently filters out noise.

The system designer should carefully consider the maximum anticipated counter frequency and duty-cycle when setting the debounce time. If the debounce configuration is set too large, the Pulse Counter will not count short pulses. The debounce-high configuration should be set to less than one-half the minimum input pulse high-time. Similarly, the debounce-low configuration should be set to less than one-half the minimum input pulse low-time.

The Debounce Timing diagram (Figure 25.4) illustrates the operation of the debounce integrator. The top waveform is the representation of the reed switch (high: open, low: closed) which shows some random switch bounce. The bottom waveform is the final signal that goes into the counter which has the switch bounce removed. Based on the actual reed switch used and sample rate, the switch bounce time may appear shorter in duration than the example in Figure 25.4. The second waveform is the pull-up resistor

enable signal. The enable signal enables the pull-up resistor when high and disables when low. PC0 is the line to the reed switch. On the right side of PC0 waveform, the line voltage is decreasing towards ground when the pull-up resistors are disabled. Beneath the charging waveform, the arrows represent the sample points. The pulse counter samples the PC0 voltage once the charging completes. The sensed ones and zeros are the sampled data. Finally the integrator waveform illustrates the output of the digital integrator. The integrator is set to 4 initially and counts to down to 0 before toggling the output low. Once the integrator reaches the low state, it needs to count up to 4 before toggling its output to the high state. The debounce logic filters out switch bounce or noise that appears for a short duration.

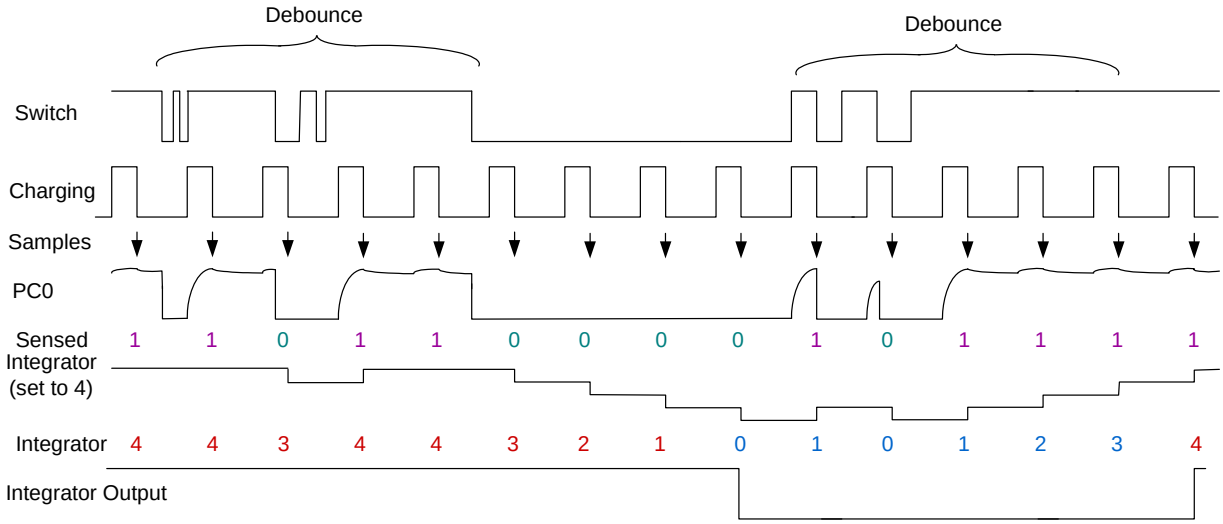


Figure 25.4. Debounce Timing

25.7. Reset Behavior

Unlike most MCU peripherals, an MCU reset does not completely reset the Pulse Counter. This includes a power on reset and all other reset sources. An MCU reset does not clear the counter values. The Pulse Counter SFRs do not reset to a default value upon reset. The 24-bit counter values are persistent unless cleared manually by writing to the PC0MD SFR. Note that if the VBAT voltage ever drops below the minimum operating voltage, this may compromise contents of the counters.

The PC0MD register should normally be written only once after reset. The PC0MD SFR is the master mode register. This register sets the counter mode and sample rate. Writing to the PC0MD SFR also resets the other PC0xxx SFRs.

Note that the RTC clock will reset on an MCU reset, so counting cannot resume until the RTC clock has been re-started.

Firmware should read the reset sources SFR RSTSRC to determine the source of the last reset and initialize the Pulse Counter accordingly.

When the pulse counter resets, it takes some time (typically two RTC clock cycles) to synchronize between internal clock domains. The counters do not increment during this synchronization time.

25.8. Wake up and Interrupt Sources

The Pulse Counter has multiple interrupt and wake-up source conditions. To enable an interrupt, enable the source in the PC0INT0/1 SFRs and enable the Pulse Counter interrupt using bit 4 of the EIE2 bit register. The Pulse Counter interrupt service routine should read the interrupt flags in PC0INT0/1 to determine the source of the interrupt and clear the interrupt flags.

To enable the Pulse Counter as a wake up source, enable the source in the PC0INT0/1 SFRs and enable the Pulse Counter as a wake-up source by setting bit 0 (PC0WK) to 1 in the PMU0FL SFR. Upon waking, firmware should read the PMCU0CF and PMU0FL SFRs to determine the wake-up source. If the PC0WK bit is set indicating that the Pulse Counter has woke the MCU, firmware should read the flag bits PC0INT0/1 SFRs to determine the Pulse Counter wake-up source and clear the flag bits before going back to sleep.

PC0INT0 includes the more common interrupt and wake-up sources. These include comparator match, counter overflow, and quadrature direction change. PC0INT1 includes interrupt and wake-up sources for the advanced features, including flutter detection and quadrature error.

25.9. Real-Time Register Access

Several of the Pulse Counter registers values change in real-time synchronous to the RTC clock. Hardware synchronization between the RTC clock domain and the system clock domain hardware would result in long delays when reading real-time registers. Instead, real-time register values are available instantaneously, but the read must be qualified using the read valid bit (PC0TH bit 0). If the register value does not change during the read access, the read valid bit will be set indicating the last was valid. If the value of the real-time register changes during the read access, the read valid bit is 0, indicating the read was invalid. After an invalid read, firmware must read the register and check the read valid bit again.

These 8-bit counter registers need to be qualified using the read valid bit:

- PC0STAT
- PC0HIST
- PC0INT0
- PC0INT1
- PC0CTR0L
- PC0CTC1L

The 24-bit counters are three-byte real-time read-only registers that require a special access method for reading. Firmware must read the low-byte (PC0CTR0L and PC0CTR1L) first and qualify using the read valid bit. Reading the low-byte latches the middle and high bytes. If the read valid bit is 0, the read is invalid and firmware must read the low-byte and check the read valid bit again. If the read valid bit is set, the read is valid and the middle and high bytes are also safe to read. Firmware should read the middle and high bytes only after reading the low byte and qualifying with the read valid bit.

The 24-bit compators are three-byte real-time read-write registers that require a special access method for writing. Firmware must write the low-byte last. After writing the low-byte, it might take up to two RTC clock cycles for the new comparator value to take effect. System designers should consider the synchronization delay when setting the comparator value. The counter may be incremented before new comparator value takes effect. Setting the comparator to at least 2 counts above the current count will eliminate the chance of missing the comparator match during synchronization.

Example code is provided with accessor functions for all the real-time Pulse Counter registers.

25.10. Advanced Features

25.10.1. Quadrature Error

The quadrature encoder must only send valid quadrature codes. A valid quadrature sequence consists of four valid states. The quadrature codes are only permitted to transition to one of the adjacent states, and an invalid transition will result in a quadrature error. Note that a quadrature error is likely to occur when first enabling the quadrature counter mode, since the Pulse Counter state machine starts at the LL state and the initial state of the quadrature is arbitrary. It is safe to ignore the first quadrature error immediately after initialization.

25.10.2. Flutter Detection

The flutter detection can be used with either quadrature counter mode or dual counter mode when the two inputs are expected to be in step. Flutter refers to the case where one input continues toggling while the other input stops toggling. This may indicate a broken reed switch or a pressure oscillation when the wheel magnet stops at just the right distance from the reed switch. If a pressure oscillation causes a slight rotational oscillation in the wheel, it could cause a number of pulses on one of the inputs, but not on the other. All four edges are checked by the flutter detection feature (PC1 positive, PC1 negative, PC0 positive, and PC0 negative).When enabled, Flutter detection may be used as an interrupt or wake-up source.

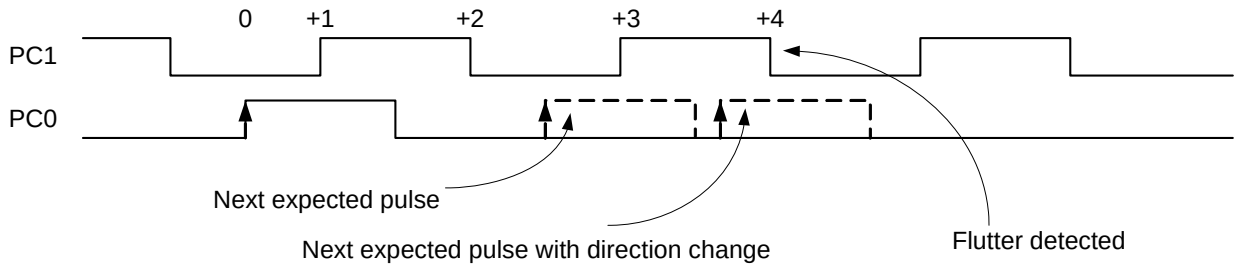


Figure 25.5. Flutter Example

For example, flutter detected on the PC0 positive edge means that 4 edges (positive or negative) were detected on PC1 since the last PC0 positive edge. Each PC0 positive edge resets the flutter detection counter while either PC1 edge increments the counter. There are similar counters for all four edges.

The flutter detection circuit provides interrupts or wake-up sources, but firmware must also read the Pulse Counter registers to determine what corrective action, if any, must be taken.

On the start of flutter event, the firmware should save both counter values and the PC0HIST register. Once the end of flutter event occurs the firmware should also save both counter values and the PC0HIST register. The stop count on flutter, STPCNTFLTR (PCMD[2]), be used to stop the counters when flutter is occurring (quadrature mode only). For quadrature mode, the opposite counter should be decremented by one. In other words, if the direction was clock-wise, the counter clock-wise counter (counter 1) should be decremented by one to correct for one increment before flutter was detected. For dual mode, two reed switches can be used to get a redundant count. If flutter starts during dual mode, both counters should be saved by firmware. After flutter stops, both counters should be read again. The counter that incremented the most was the one that picked up the flutter. There is also a mode to switch from quadrature to dual (PC0MD[1]) when flutter occurs. This changes the counter style from quadrature (count on any edge of PC1 or PC0) to dual to allow all counts to be recorded. Once flutter ends, this mode switches the counters back to quadrature mode. STPCNTFLTR does not function when PC0MD[1] is set.

SFR Definition 25.1. PC0MD: PC0 Mode Configuration

Bit	7	6	5	4	3	2	1	0
Name	PCMODE[1:0]		PCRATE[1:0]		DUALCMPL	STPCNTFLTR	DUALSTCH	
Type	R/W		R/W		R/W	R	R/W	R
Reset	0	0	0	0	0	1	0	0

SFR Address = 0xD9; SFR Page = 0x2

Bit	Name	Function
7:6	PCMODE[1:0]	**Counter Mode** 00: Pulse Counter disabled. 01: Single Counter mode. 10: Dual Counter mode. 11: Quadrature Counter mode.
5:4	PCRATE[1:0]	**PC Sample Rate** 00: 250 µs 01: 500 µs 10: 1 ms 11: 2 ms
3	Reserved	
2	STPCNTFLTR	**Stop Counting on Flutter** (Only valid for quadrature counter mode and DUALSTCH off.) 0: Disabled. 1: Enabled.
1	DUALSTCH	**Dual Mode Switch During Flutter** (Only valid for quadrature counter mode.) 0: Disabled—quadrature mode remains set during flutter. 1: Enabled—quadrature mode changes to dual during flutter.
0	Reserved	

Note that writing to this register will clear the counter registers PC0CTR0H:M:L and PC0CTR1H:M:L.

SFR Definition 25.2. PC0PCF: PC0 Mode Pull-Up Configuration

Bit	7	6	5	4	3	2	1	0
Name	PUCAL	CALRES	CALPORT	RES[2:0]			DUTY[1:0]	
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

SFR Address = 0xD7; SFR Page = 0x2

Bit	Name	Function
7	PUCAL	**Pull-Up Driver Calibration** 0: Calibration complete or not running. 1: Start calibration of pull up (Self clearing). Calibration determines the lowest usable pull-up strength.
6	CALRES	**Calibration Result** 0: Fail (switch may be closed preventing detection of pull ups). Writes value of 0x11111 to PC0PCF[4:0] 1: Pass (writes calibrated value into PC0PCF[4:0]).
5	CALPORT	**Calibration Port** 0: Calibration on PC0 only. 1: Calibration on PC1 only.
4:2	RES[2:0]	**Pull-Up Resistor Select** Current with force pull-up on bit set (PC0TH.2=1) and VBAT=3.6V. 000: Pull-up disabled. 001: 1 μA.* 010: 4 μA.* 011: 16 μA.* 100: 64 μA.* 101: 256 μA.* 110: 1 mA.* 111: 4 mA.* *The effective average pull-up current depends on selected resistor, pull-up resistor duty-cycle multiplier, and sample rate duty-cycle multiplier.
1:0	DUTY[1:0]	**Pull-Up Resistor Duty Cycle Multiplier** 000: 1/4 (25%)* 001: 3/8 (37.5%)* 010: 1/2 (50%)* 011: 3/4 (75%)* *The final pull-up resistor duty cycle is the sample rate duty-cycle multiplier times the pull-up duty-cycle multiplier.

SFR Definition 25.3. PC0TH: PC0 Threshold Configuration

Bit	7	6	5	4	3	2	1	0
Name	PCTTHRESHI[1:0]		PCTHRESLO[1:0]		PDOWN	PUP		RDVALID
Type	R/W		R/W		R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	1

SFR Address = 0xE4; SFR Page = 0x2

Bit	Name	Function
7:6	PCTTHRESHI[1:0]	**Pulse Counter Input Comparator VIH Threshold** (Percentage of VIO.) 10: 50% 11: 55% 00: 59% 01: 63%
5:4	PCTHRESLO[1:0]	**Pulse Counter Input Comparator VIL Threshold** (percentage of VIO.) 10: 34% 11: 38% 00: 42% 01: 46%
3	PDOWN	**Force Pull-Down On** 0: PC0 and PC1 pull-down not forced on. 1: PC0 and PC1 grounded.
2	PUP	**Force Pull-Up** 0: PC0 and PC1 pull-up not forced on continuously. See PC0PCF[1:0] for duty cycle. 1: PC0 and PC1 pulled high continuously to the PC0PCF[4:2] setting. PDOWN overrides PUP setting.
1	Reserved	
0	RDVALID	**Read Valid** Holds the status of the last read for real-time registers PC0STAT, PC0HIST, PC0CTR0L, PC0CTR1L, PC0INT0, and PC0INT1. 0: The last read was invalid. 1: The last read was valid. RDVALID is set back to 1 upon reading.

SFR Definition 25.4. PC0STAT: PC0 Status

Bit	7	6	5	4	3	2	1	0
Name	FLUTTER	DIRECTION	STATE[1:0]		PC1PREV	PC0PREV	PC1	PC0
Type	RO	RO	RO		RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0x2

Bit	Name	Function
7	FLUTTER	**Flutter** During quadrature mode, a disparity may occur between the number of negative edges of PC1 and PC0 or the number of positive edges of PC1 and PC0. This could indicate flutter on one reed switch or one reed switch may be faulty. 0: No flutter detected. 1: Flutter detected.
6	DIRECTION	**Direction** Only applicable for quadrature mode. (First letter is PC1; second letter is PC0) 0: Counter clock-wise - (LL-LH-HH-HL) 1: Clock-wise - (LL-HL-HH-LH)
5:4	STATE[1:0]	**PC0 State** Current State of Internal State Machine.
3	PC1PREV	**PC1 Previous** Previous Output of PC1 Integrator.
2	PC0PREV	**PC0 Previous** Previous Output of PC0 Integrator.
1	PC1	**PC1** Current Output of PC1 Integrator.
0	PC0	**PC0** Current Output of PC0 Integrator.

SFR Definition 25.5. PC0DCH: PC0 Debounce Configuration High

Bit	7	6	5	4	3	2	1	0
Name	PC0DCH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	1	0	0

SFR Address = 0xFA; SFR Page = 0x2

Bit	Name	Function
7:0	PC0DCH[7:0]	**Pulse Counter Debounce High** Number of cumulative good samples seen by the integrator before recognizing the input as high. Sampling a low will decrement the count while sampling a high will increment the count. The actual value used is PC0DCH plus one. Switch bounce produces a random looking signal. The worst case would be to bounce low at each sample point and not start incrementing the integrator until the switch bounce settled. Therefore, minimum pulse width should account for twice the debounce time. For example, using a sample rate of 250 µs and a PC0DCH value of 0x13 will look for 20 cumulative highs before recognizing the input as high (250 µs x (16+3+1) = 5 ms).

SFR Definition 25.6. PC0DCL: PC0 Debounce Configuration Low

Bit	7	6	5	4	3	2	1	0
Name	PC0DCL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	1	0	0

SFR Address = 0xF9; SFR Page = 0x2

Bit	Name	Function
7:0	PC0DCL[7:0]	**Pulse Counter Debounce Low** Number of cumulative good samples seen by the integrator before recognizing the input as low. Setting PC0DCL to 0x00 will disable integrators on both PC0 and PC1. The actual value used is PC0DCL plus one. Sampling a low decrements while sampling a high increments the count. Switch bounce produces a random looking signal. The worst case would be to bounce high at each sample point and not start decrementing the integrator until the switch bounce settled. Therefore, minimum pulse width should account for twice the debounce time. For example, using a sample rate of 1 ms and a PC0DCL value of 0x09 will look for 10 cumulative lows before recognizing the input as low (1 ms x 10 = 10 ms). The minimum pulse width should be 20 ms or greater for this example. If PC0DCL has a value of 0x03 and the sample rate is 500 µs, the integrator would need to see 4 cumulative lows before recognizing the low (500 µs x 4 = 2 ms). The minimum pulse width should be 4 ms for this example.

SFR Definition 25.7. PC0CTR0H: PC0 Counter 0 High (MSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR0H[23:16]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDC; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR0H[23:16]	**PC0 Counter 0 High Byte** Bits 23:16 of Counter 0.

SFR Definition 25.8. PC0CTR0M: PC0 Counter 0 Middle

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR0M[15:8]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD8; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR0M[15:8]	**PC0 Counter 0 Middle Byte** Bits 15:8 of Counter 0.

SFR Definition 25.9. PC0CTR0L: PC0 Counter 0 Low (LSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR0L[7:0]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDA; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR0L[7:0]	**PC0 Counter 0 Low Byte** Bits 7:0 of Counter 0.

Note: PC0CTR0L must be read before PC0CTR0M and PC0CTR0H to latch the count for reading. PC0CTRL must be qualified using the RDVALID bit (PC0TH[0]).

SFR Definition 25.10. PC0CTR1H: PC0 Counter 1 High (MSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR1H[23:16]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDF; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR1H[23:16]	**PC0 Counter 1 High Byte** Bits 23:16 of Counter 1.

SFR Definition 25.11. PC0CTR1M: PC0 Counter 1 Middle

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR1M[15:8]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDE; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR1M[15:8]	**PC0 Counter 1 Middle Byte** Bits 15:8 of Counter 1.

SFR Definition 25.12. PC0CTR1L: PC0 Counter 1 Low (LSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CTR1L[7:0]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xDD; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CTR1L[7:0]	**PC0 Counter 1 Low Byte** Bits 7:0 of Counter 1.

Note: PC0CTR1L must be read before PC0CTR1M and PC0CTR1H to latch the count for reading.

SFR Definition 25.13. PC0CMP0H: PC0 Comparator 0 High (MSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CMP0H[23:16]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE3; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CMP0H[23:16]	**PC0 Comparator 0 High Byte** Bits 23:16 of Counter 0.

SFR Definition 25.14. PC0CMP0M: PC0 Comparator 0 Middle

Bit	7	6	5	4	3	2	1	0
Name	PC0CMP0M[15:8]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CMP0M[15:8]	**PC0 Comparator 0 Middle Byte** Bits 15:8 of Counter 0.

SFR Definition 25.15. PC0CMP0L: PC0 Comparator 0 Low (LSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CMP0L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE1; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CMP0L[7:0]	**PC0 Comparator 0 Low Byte** Bits 7:0 of Counter 0.

Note: PC0CMP0L must be written last after writing PC0CMP0M and PC0CMP0H. After writing PC0CMP0L, the synchronization into the PC clock domain can take 2 RTC clock cycles.

SFR Definition 25.16. PC0CMP1H: PC0 Comparator 1 High (MSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CMP1H[23:16]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF3; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CMP1H[23:16]	**PC0 Comparator 1 High Byte** Bits 23:16 of Counter 0.

SFR Definition 25.17. PC0CMP1M: PC0 Comparator 1 Middle

Bit	7	6	5	4	3	2	1	0
Name	PC0CMP1M[15:8]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF2; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CMP1M[15:8]	**PC0 Comparator 1 Middle Byte** Bits 15:8 of Counter 0.

SFR Definition 25.18. PC0CMP1L: PC0 Comparator 1 Low (LSB)

Bit	7	6	5	4	3	2	1	0
Name	PC0CMP1L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF1; SFR Page = 0x2

Bit	Name	Function
7:0	PC0CMP1L[7:0]	**PC0 Comparator 1 Low Byte** Bits 7:0 of Counter 0.

Note: PC0CMP1L must be written last after writing PC0CMP1M and PC0CMP1H. After writing PC0CMP1L the synchronization into the PC clock domain can take 2 RTC clock cycles.

SFR Definition 25.19. PC0HIST: PC0 History

Bit	7	6	5	4	3	2	1	0
Name	PC0HIST[7:0]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4; SFR Page = 0x2

Bit	Name	Function
7:0	PC0HIST[7:0]	**PC0 History.** Contains the last 8 recorded directions (1: clock-wise, 0: counter clock-wise) on the previous 8 counts. Values of 0x55 or 0xAA may indicate flutter during quadrature mode.

SFR Definition 25.20. PC0INT0: PC0 Interrupt 0

Bit	7	6	5	4	3	2	1	0
Name	CMP1F	CMP1EN	CMP0F	CMP0EN	OVRF	OVREN	DIRCHGF	DIRCHGEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xFB; SFR Page = 0x2

Bit	Name	Function
7	CMP1F	**Comparator 1 Flag** 0: Counter 1 did not match comparator 1 value. 1: Counter 1 matched comparator 1 value.
6	CMP1EN	**Comparator 1 Interrupt/Wake-up Source Enable** 0:CMP1F not enabled as interrupt or wake-up source. 1:CMP1F enabled as interrupt or wake-up source.
5	CMP0F	**Comparator 0 Flag** 0: Counter 0 did not match comparator 0 value. 1: Counter 0 matched comparator 0 value.
4	CMP0EN	**Comparator 0 Interrupt/Wake-up Source Enable** 0:CMP0F not enabled as interrupt or wake-up source. 1:CMP0F enabled as interrupt or wake-up source.
3	OVRF	**Counter Overflow Flag** 1:Neither of the counters has overflowed. 1:One of the counters has overflowed.
2	OVREN	**Counter Overflow Interrupt/Wake-up Source Enable** 0:OVRF not enabled as interrupt or wake-up source. 1:OVRF enabled as interrupt or wake-up source.
1	DIRCHGF	**Direction Change Flag** Direction changed for quadrature mode only. 0:No change in direction detected. 1:Direction Change detected.
0	DIRCHGEN	**Direction Change Interrupt/Wake-up Source Enable** 0:DIRCHGF not enabled as interrupt or wake-up source. 1:DIRCHGF enabled as interrupt or wake-up source.

SFR Definition 25.21. PC0INT1: PC0 Interrupt 1

Bit	7	6	5	4	3	2	1	0
Name	FLTRSTRF	FLTRSTREN	FLTRSTPF	FLTRSTPEN	ERRORF	ERROREN	TRANSF	TRANSEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xFC; SFR Page = 0x2

Bit	Name	Function
7	FLTRSTRF	**Flutter Start Flag** Flutter detection for quadrature mode or dual mode only. 0: No flutter detected. 1: Start of flutter detected.
6	FLTRSTREN	**Flutter Start Interrupt/Wake-up Source Enable** 0:FLTRSTRF not enabled as interrupt or wake-up source. 1:FLTRSTRF enabled as interrupt or wake-up source.
5	FLTRSTPF	**Flutter Stop Flag** Flutter detection for quadrature mode or dual mode only. 0: No flutter stop detected. 1: Flutter stop detected.
4	FLTRSTPEN	**Flutter Stop Interrupt/Wake-up Source Enable** 0:FLTRSTPF not enabled as interrupt or wake-up source. 1:FLTRSTPF enabled as interrupt or wake-up source.
3	ERRORF	**Quadrature Error Flag** 0: No Quadrature Error detected. 1: Quadrature Error detected.
2	ERROREN	**Quadrature Error Interrupt/Wake-up Source Enable** 0:ERRORF not enabled as interrupt or wake-up source. 1:ERRORF enabled as interrupt or wake-up source.
1	TRANSF	**Transition Flag** 0: No transition detected. 1: Transition detected on PC0 or PC1.
0	TRANSEN	**Transition Interrupt/Wake-up Source Enable** 0: TRANSF not enabled as interrupt or wake-up source. 1: TRANSF enabled as interrupt or wake-up source.

26. LCD Segment Driver

C8051F96x devices contain an LCD segment driver and on-chip bias generation that supports static, 2-mux, 3-mux and 4-mux LCDs with 1/2 or 1/3 bias. The on-chip charge pump with programmable output voltage allows software contrast control which is independent of the supply voltage. LCD timing is derived from the SmaRTClock oscillator to allow precise control over the refresh rate.

The C8051F96x uses special function registers (SFRs) to store the enabled/disabled state of individual LCD segments. All LCD waveforms are generated on-chip based on the contents of the LCD0Dn registers An LCD blinking function is also supported. A block diagram of the LCD segment driver is shown in Figure 26.1.

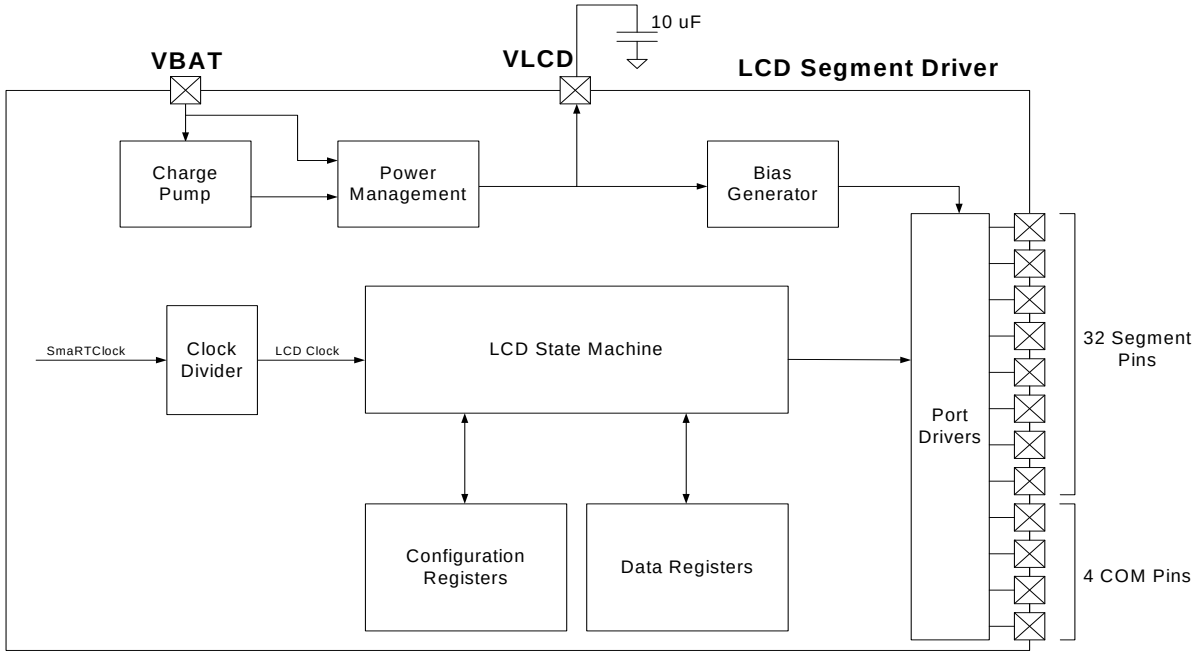


Figure 26.1. LCD Segment Driver Block Diagram

26.1. Configuring the LCD Segment Driver

The LCD segment driver supports multiple mux options: static, 2-mux, 3-mux, and 4-mux mode. It also supports 1/2 and 1/3 bias options. The desired mux mode and bias is configured through the LCD0CN register. A divide value may also be applied to the SmaRTClock output before being used as the LCD0 clock source.

The following procedure is recommended for using the LCD Segment Driver:

1. Initialize the SmaRTClock and configure the LCD clock divide settings in the LCD0CN register.
2. Determine the GPIO pins which will be used for the LCD function.
3. Configure the Port I/O pins to be used for LCD as Analog I/O.
4. Configure the LCD size, mux mode, and bias using the LCD0CN register.
5. Enable the LCD bias and clock gate by writing 0x50 to the LCD0MSCN register.
6. Configure the device into the desired Contrast Control Mode.
7. If VIO is internally or externally shorted to VBAT, disable the VLCD/VIO Supply Comparator using the

LCD0CF Register.

8. Set the LCD contrast using the LCD0CNTRST register.
9. Set the desired threshold for the VBAT Supply Monitor.
10. Set the LCD refresh rate using the LCD0DIVH:LCD0DIVL registers.
11. Write a pattern to the LCD0Dn registers.
12. Enable the LCD by setting bit 0 of LCD0MSCN to logic 1 (LCD0MSCN |= 0x01).

26.2. Mapping Data Registers to LCD Pins

The LCD0 data registers are organized as 16 byte-wide special function registers (LCD0Dn), each half-byte or nibble in these registers controls 1 LCD output pin. There are 32 nibbbles used to control the 32 segment pins.

Each LCD0 segment pin can control 1, 2, 3, or 4 LCD segments depending on the selected mux mode. The least significant bit of each nibble controls the segment connected to the backplane signal COM0. The next to least significant bit controls the segment associated with COM1, the next bit controls the segment associated with COM2, and the most significant bit in the 4-bit nibble controls the segment associated with COM3.

In static mode, only the least significant bit in each nibble is used and the three remaining bits in each nibble are ignored. In 2-mux mode, only the two least significant bits are used; in 3-mux mode, only the three least significant bits are used, and in 4-mux mode, each of the 4 bits in the nibble controls one LCD segment. Bits with a value of 1 turn on the associated segment and bits with a value of 0 turn off the associated segment.

SFR Definition 26.1. LCD0Dn: LCD0 Data

Bit	7	6	5	4	3	2	1	0
Name	LCD0Dn							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page: 0x2

Addresses: LCD0D0 = 0x89, LCD0D1 = 0x8A, LCD0D2 = 0x8B, LCD0D3 = 0x8C,
LCD0D4 = 0x8D, LCD0D5 = 0x8E, LCD0D6 = 0x91, LCD0D7 = 0x92,
LCD0D8 = 0x93, LCD0D9 = 0x94, LCD0DA = 0x95, LCD0DB = 0x96,
LCD0DC = 0x97, LCD0DD = 0x99, LCD0DE = 0x9A, LCD0DF = 0x9B.

Bit	Name	Function
7:0	LCD0Dn	**LCD Data.** Each nibble controls one LCD pin. See “Mapping Data Registers to LCD Pins” on page 335 for additonal informatoin.

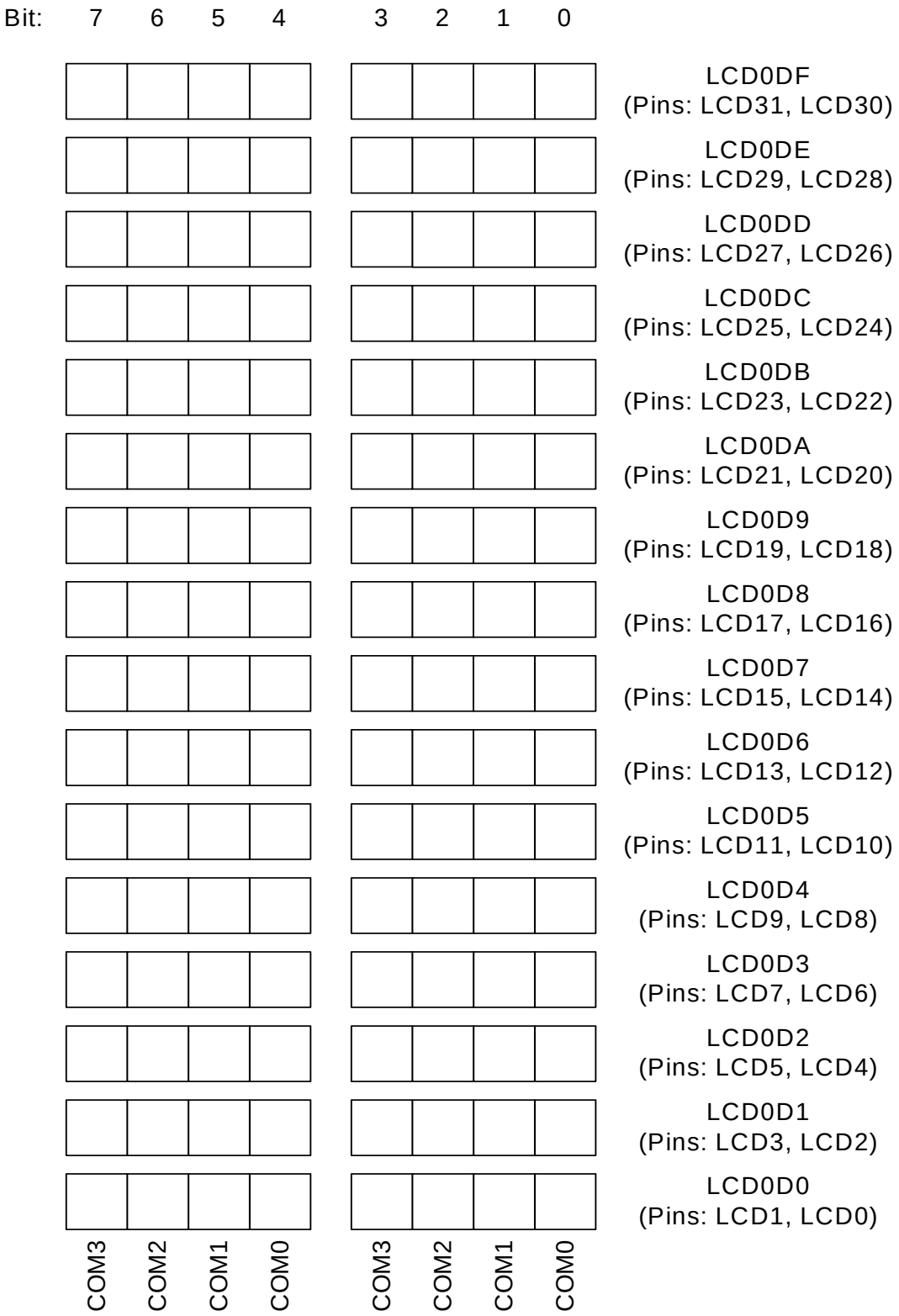


Figure 26.2. LCD Data Register to LCD Pin Mapping

SFR Definition 26.2. LCD0CN: LCD0 Control Register

Bit	7	6	5	4	3	2	1	0
Name		CLKDIV[1:0]		BLANK	SIZE	MUXMD[1:0]		BIAS
Type	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0x9D

Bit	Name	Function
7	Reserved	Read = 0. Must Write 0b.
6:5	CLKDIV[1:0]	**LCD0 Clock Divider.** Divides the SmaRTClock output for use by the LCD0 module. See Table 4.18 on page 76 for LCD clock frequency range. 00: The LCD clock is the SmaRTClock divided by 1. 01: The LCD clock is the SmaRTClock divided by 2. 10: The LCD clock is the SmaRTClock divded by 4. 11: Reserved.
4	BLANK	**Blank All Segments.** Blanks all LCD segments using a single bit. 0: All LCD segments are controlled by the LCD0Dn registers. 1: All LCD segments are blank (turned off).
3	SIZE	**LCD Size Select.** Selects whether 16 or 32 segment pins will be used for the LCD function. 0: P0 and P1 are used as LCD segment pins. 1: P0, P1, P2, and P3 are used as LCD segment pins.
2:1	MUXMD[1:0]	**LCD Bias Power Mode.** Selects the mux mode. 00: Static mode selected. 01: 2-mux mode selected. 10: 3-mux mode selected. 11: 4-mux mode selected.
0	BIAS	**Bias Select**. Selects between 1/2 Bias and 1/3 Bias. This bit is ignored if Static mode is selected. 0: LCD0 is configured for 1/3 Bias. 1: LCD0 is configured for 1/2 Bias.

26.3. LCD Contrast Adjustment

The LCD Bias voltages which determine the LCD contrast are generated using the VBAT supply voltage or the on-chip charge pump. There are four contrast control modes to accomodate a wide variety of applications and supply voltages. The target contrast voltage is programmable in 60 mV steps from 1.9 to 3.72 V. The LCD contrast voltage is controlled by the LCD0CNTRST register and the contrast control mode is selected by setting the appropriate bits in the LCD0MSCN, LCD0MSCF, LCD0PWR, and LCD0VBMCN registers.

Note: An external 10 µF decoupling capacitor is required on the VLCD pin to create a charge reservoir at the output of the charge pump.

Table 26.1. Bit Configurations to select Contrast Control Modes

Mode	LCD0MSCN.2	LCD0MSCF.0	LCD0PWR.3	LCD0VBMCN.7
1	0	1	0	0
2	0	1	1	1
3	1*	0	1	1
4	1*	0	0	1
*** May be set to 0 to support increased load currents.**				

26.3.1. Contrast Control Mode 1 (Bypass Mode)

In Contrast Control Mode 1, the contrast control circuitry is disabled and the VLCD voltage follows the VBAT supply voltage, as shown in Figure 26.3. This mode is useful in systems where the VBAT voltage always remains constant and will provide the lowest LCD power consumption. Bypass Mode is selected using the following procedure:

1. Clear Bit 2 of the LCD0MSCN register to 0b (LCD0MSCN &= ~0x04)
2. Set Bit 0 of the LCD0MSCF register to 1b (LCD0MSCF |= 0x01)
3. Clear Bit 3 of the LCD0PWR register to 0b (LCD0PWR &= ~0x08)
4. Clear Bit 7 of the LCD0VBMCN register to 0b (LCD0VBMCN &= ~0x80)

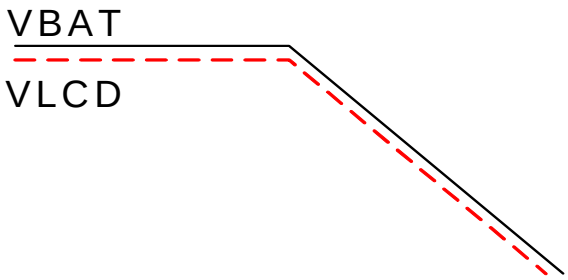


Figure 26.3. Contrast Control Mode 1

26.3.2. Contrast Control Mode 2 (Minimum Contrast Mode)

In Contrast Control Mode 2, a minimum contrast voltage is maintained, as shown in Figure 26.4. The VLCD supply is powered directly from VBAT as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to acheive the desired minimum contrast voltage on VLCD. Minimum Contrast Mode is selected using the following procedure:

1. Clear Bit 2 of the LCD0MSCN register to 0b (LCD0MSCN &= ~0x04)
2. Set Bit 0 of the LCD0MSCF register to 1b (LCD0MSCF |= 0x01)
3. Set Bit 3 of the LCD0PWR register to 1b (LCD0PWR |= 0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

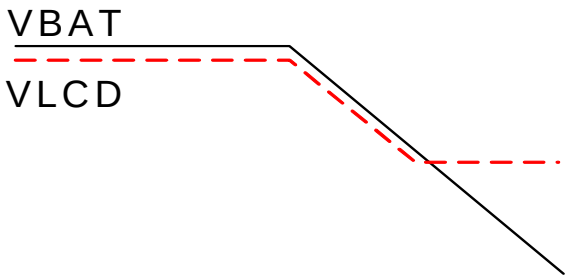


Figure 26.4. Contrast Control Mode 2

26.3.3. Contrast Control Mode 3 (Constant Contrast Mode)

In Contrast Control Mode 3, a constant contrast voltage is maintained. The VLCD supply is regulated to the programmed contrast voltage using a variable resistor between VBAT and VLCD as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to acheive the desired contrast voltage on VLCD. Constant Contrast Mode is selected using the following procedure:

1. Set Bit 2 of the LCD0MSCN register to 1b (LCD0MSCN |= 0x04)
2. Clear Bit 0 of the LCD0MSCF register to 0b (LCD0MSCF &= ~0x01)
3. Set Bit 3 of the LCD0PWR register to 1b (LCD0PWR |= 0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

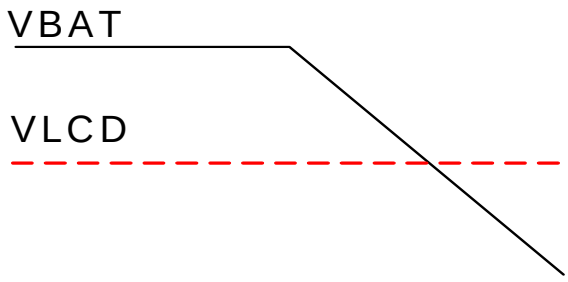


Figure 26.5. Contrast Control Mode 3

26.3.4. Contrast Control Mode 4 (Auto-Bypass Mode)

In Contrast Control Mode 4, behavior is identical to Constant Contrast Mode as long as VBAT is greater than the VBAT monitor threshold voltage. When VBAT drops below the programmed threshold, the device automatically enters bypass mode powering VLCD directly from VBAT. The charge pump is always disabled in this mode. Auto-Bypass Mode is selected using the following procedure:

1. Set Bit 2 of the LCD0MSCN register to 1b (LCD0MSCN |= 0x04)
2. Clear Bit 0 of the LCD0MSCF register to 0b (LCD0MSCF &= ~0x01)
3. Clear Bit 3 of the LCD0PWR register to 0b (LCD0PWR &= ~0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

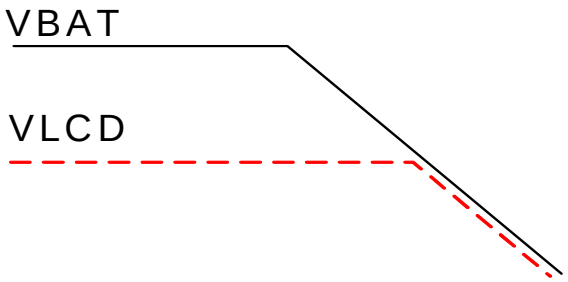


Figure 26.6. Contrast Control Mode 4

SFR Definition 26.3. LCD0CNTRST: LCD0 Contrast Adjustment

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	CNTRST				
Type	R/W	R/W	R/W	R/W				
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0x9C

Bit	Name	Function
7:5	Reserved	Read = 000. Write = Must write 000.
4:0	CNTRST	**Contrast Setpoint.** Determines the setpoint for the VLCD voltage necessary to achieve the desired contrast. 00000: 1.90 00001: 1.96 00010: 2.02 00011: 2.08 00100: 2.13 00101: 2.19 00110: 2.25 00111: 2.31 01000: 2.37 01001: 2.43 01010: 2.49 01011: 2.55 01100: 2.60 01101: 2.66 01110: 2.72 01111: 2.78 10000: 2.84 10001: 2.90 10010: 2.96 10011: 3.02 10100: 3.07 10101: 3.13 10110: 3.19 10111: 3.25 11000: 3.31 11001: 3.37 11010: 3.43 11011: 3.49 11100: 3.54 11101: 3.60 11110: 3.66 11111: 3.72

SFR Definition 26.4. LCD0MSCN: LCD0 Master Control

Bit	7	6	5	4	3	2	1	0
Name		BIASEN	DCBIASOE	CLKOE		LOWDRV	LCDRST	LCDEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAB

Bit	Name	Function
7	Reserved	Read = 0b. Must write 0b.
6	BIASEN	**LCD0 Bias Enable.** LCD0 bias may be disabled when using a static LCD (single backplane), contrast control mode 1 (Bypass Mode) is selected, and the VLCD/VIO Supply Comparator is disabled (LCD0CF.5 = 1). It is required for all other modes. 0: LCD0 Bias is disabled. 1: LCD0 Bias is enabled
5	DCBIASOE	**DCDC Converter Bias Output Enable. (Note 1)** 0: The bias for the DCDC converter is gated off. 1: LCD0 provides the bias for the DCDC converter.
4	CLKOE	**LCD Clock Output Enable.** 0: The clock signal to the LCD0 module is gated off. 1: The SmaRTClock provides the undivided clock to the LCD0 Module.
3	Reserved	Read = 0b. Must write 0b.
2	LOWDRV	**Charge Pump Reduced Drive Mode.** This bit should be set to 1 in Contrast Control Mode 3 and Mode 4 for minimum power consumption. This bit may be set to 0 in these modes to support higher load current requirements. 0: The charge pump operates at full power. 1: The charge pump operates at reduced power.
1	LCDRST	**LCD0 Reset.** Writing a 1 to this bit will clear all the LCD0Dn registers to 0x00. This bit must be cleared by software.
0	LCDEN	**LCD0 Enable.** 0: LCD0 is disabled. 1: LCD0 is enabled.
Note 1: To same bias generator is shared by the DCDC Converter and LCD0.		

SFR Definition 26.5. LCD0MSCF: LCD0 Master Configuration

Bit	7	6	5	4	3	2	1	0
Name							DCENSLP	CHPBYP
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	0

SFR Page = 0x2; SFR Address = 0xAC

Bit	Name	Function
7:2	Reserved	Read = 111111b. Must write 111111b.
1	DCENSLP	**DCDC Converter Enable in Sleep Mode** 0: DCDC is disabled in Sleep Mode. 1: DCDC is enabled in Sleep Mode.
0	CHPBYP	**LCD0 Charge Pump Bypass** This bit should be set to 1b in Contrast Control Mode 1 and Mode 2. 0: LCD0 Charge Pump is not bypassed. 1: LCD0 Charge Pump is bypassed.

SFR Definition 26.6. LCD0PWR: LCD0 Power

Bit	7	6	5	4	3	2	1	0
Name					MODE			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	1

SFR Page = 0x2; SFR Address = 0xA4

Bit	Name	Function
7:4	Unused	Read = 0000b. Write = don’t care.
3	MODE	**LCD0 Contrast Control Mode Selection.** 0: LCD0 Contrast Control Mode 1 or Mode 4 is selected. 1: LCD0 Contrast Control Mode 2 or Mode 3 is selected.
2:0	Reserved	Read = 001b. Must write 001b.

26.4. Adjusting the VBAT Monitor Threshold

The VBAT Monitor is used primarily for the contrast control function, to detect when VBAT has fallen below a specific threshold. The VBAT monitor threshold may be set independently of the contrast setting or it may be linked to the contrast setting. When the VBAT monitor threshold is linked to the contrast setting, an offset (in 60mV steps) may be configured so that the VBAT monitor generates a VBAT low condition prior to VBAT dropping below the programmed contrast voltage. The LCD0VBMCN register is used to enable and configure the VBAT Monitor. The VBAT monitor may be enabled as a wake-up source to wake up the device from Sleep mode when the battery is getting low. See the Power Management chapter for more details.

SFR Definition 26.7. LCD0VBMCN: LCD0 VBAT Monitor Control

Bit	7	6	5	4	3	2	1	0
Name	VBATMEN	OFFSET		THRLD[4:0]				
Type	R/W	R/W	R/W	R/W				
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xA6

Bit	Name	Function
7	VBATMEN	**VBAT Monitor Enable** The VBAT Monitor should be enabled in Contrast Control Mode 2, Mode 3, and Mode 4. 0: The VBAT Monitor is disabled. 1: The VBAT Monitor is enabled.
6	OFFSET	**VBAT Monitor Offset Enable** 0: The VBAT Monitor Threshold is independent of the contrast setting. 1: The VBAT Monitor Threshold is linked to the contrast setting.
5	Unused	Read = 0. Write = Don't Care.
4:0	THRLD[4:0]	**VBAT Monitor Threshold** If OFFSET is set to 0b, this bit field has the same defintion as the CNTRST bit field and can be programmed independently of the contrast. If OFFSET is set to 1b, this bit field is interpreted as an offset to the currently programmed contrast setting. **The LCD0CNTRST register should be written before setting OFFSET to logic 1 and should not be changed as long as VBAT Monitor Offset is enabled.** When THRLD[4:0] is set to 00000b, the VBAT monitor threshold is equal to the contrast voltage. When THRLD[4:0] is set to 00001b, the VBAT monitor threshold is one step higher than the contrast voltage. The step size is equal to the step size of the CNTRST bit field.

26.5. Setting the LCD Refresh Rate

The clock to the LCD0 module is derived from the SmaRTClock and may be divided down according to the settings in the LCD0CN register. The LCD refresh rate is derived from the LCD0 clock and can be programmed using the LCD0DIVH:LCD0DIVL registers. The LCD mux mode must be taken into account when determining the prescaler value. See the LCD0DIVH/LCD0DIVL register descriptions for more details. For maximum power savings, choose a slow LCD refresh rate and the minimum LCD0 clock frequency. For the least flicker, choose a fast LCD refresh rate.

SFR Definition 26.8. LCD0CLKDIVH: LCD0 Refresh Rate Prescaler High Byte

Bit	7	6	5	4	3	2	1	0
Name							LCD0DIV[9:8]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAA

Bit	Name	Function
7:2	Unused	Read = 000000. Write = Don't Care.
1:0	LCD0DIV[9:8]	**LCD Refresh Rate Prescaler**. Sets the LCD refresh rate according to the following equation: $\text{LCD Refresh Rate} = \frac{\text{LCD0 Clock Frequency}}{4 \times \text{mux_mode} \times (LCD0DIV + 1)}$

SFR Definition 26.9. LCD0CLKDIVL: LCD Refresh Rate Prescaler Low Byte

Bit	7	6	5	4	3	2	1	0
Name	LCD0DIV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xA9

Bit	Name	Function
7:0	LCD0DIV[7:0]	**LCD Refresh Rate Prescaler**. Sets the LCD refresh rate according to the following equation: $\text{LCD Refresh Rate} = \frac{\text{LCD0 Clock Frequency}}{4 \times \text{mux_mode} \times (LCD0DIV + 1)}$

26.6. Blinking LCD Segments

The LCD driver supports blinking LCD applications such as clock applications where the ":" separator toggles on and off once per second. If the LCD is only displaying the hours and minutes, then the device only needs to wake up once per minute to update the display. The once per second blinking is automatically handled by the C8051F96x.

The LCD0BLINK register can be used to enable blinking on any LCD segment connected to the LCD0 or LCD1 segment pin. In static mode, a maximum of 2 segments can blink. In 2-mux mode, a maximum of 4 segments can blink; in 3-mux mode, a maximum of 6 segments can blink; and in 4-mux mode, a maximum of 8 segments can blink. The LCD0BLINK mask register targets the same LCD segments as the LCD0D0 register. If an LCD0BLINK bit corresponding to an LCD segment is set to 1, then that segment will toggle at the frequency set by the LCD0TOGR register without any software intervention.

SFR Definition 26.10. LCD0BLINK: LCD0 Blink Mask

Bit	7	6	5	4	3	2	1	0
Name	LCD0BLINK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0x9E

Bit	Name	Function
7:0	LCD0BLINK[7:0]	**LCD0 Blink Mask.** Each bit maps to a specific LCD segment connected to the LCD0 and LCD1 segment pins. A value of 1 indicates that the segment is blinking. A value of 0 indicates that the segment is not blinking. This bit to segment mapping is the same as the LCD0D0 register.

SFR Definition 26.11. LCD0TOGR: LCD0 Toggle Rate

Bit	7	6	5	4	3	2	1	0
Name					TOGR[3:0]			
Type	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0x9F

Bit	Name	Function
7:4	Unused	Read = 0000. Write = Don't Care.
3:0	TOGR[3:0]	**LCD Toggle Rate Divider.** Sets the LCD Toggle Rate according to the following equation: $\text{LCD Toggle Rate} = \frac{\text{Refresh Rate} \times \text{mux_mode} \times 2}{\text{Toggle Rate Divider}}$ 0000: Reserved. 0001: Reserved. 0010: Toggle Rate Divider is set to divide by 2. 0011: Toggle Rate Divider is set to divide by 4. 0100: Toggle Rate Divider is set to divide by 8. 0101: Toggle Rate Divider is set to divide by 16. 0110: Toggle Rate Divider is set to divide by 32. 0111: Toggle Rate Divider is set to divide by 64. 1000: Toggle Rate Divider is set to divide by 128. 1001: Toggle Rate Divider is set to divide by 256. 1010: Toggle Rate Divider is set to divide by 512. 1011: Toggle Rate Divider is set to divide by 1024. 1100: Toggle Rate Divider is set to divide by 2048. 1101: Toggle Rate Divider is set to divide by 4096. All other values reserved.

26.7. Advanced LCD Optimizations

The special function registers described in this section should be left at their reset value for most systems. Some systems with specific low power or large load requirments will benefit from tweaking the values in these registers to achieve minimum power consumption or maximum drive level.

SFR Definition 26.12. LCD0CF: LCD0 Configuration

Bit	7	6	5	4	3	2	1	0
Name			CMPBYP					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xA5

Bit	Name	Function
7:6	Reserved	Read = 00b. Must write 00b.
5	CMPBYP	**VLCD/VIO Supply Comparator Disable.** Setting this bit to ‘1’ disables the supply voltage comparator which determines if the VIO supply is lower than VLCD. This comparator should only be disabled, as a power saving measure, if VIO is internally or externally shorted to VBAT.
4:0	Reserved	Read = 00b. Must write 00000b.

SFR Definition 26.13. LCD0CHPCN: LCD0 Charge Pump Control

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	0	1	1

SFR Page = 0x2; SFR Address = 0xB5

Bit	Name	Function
7:0	Reserved	Must write 0x4B.

SFR Definition 26.14. LCD0CHPCF: LCD0 Charge Pump Configuration

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAD

Bit	Name	Function
7:0	Reserved	Must write 0x60.

SFR Definition 26.15. LCD0CHPMD: LCD0 Charge Pump Mode

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	0	0	1

SFR Page = 0x2; SFR Address = 0xAE

Bit	Name	Function
7:0	Reserved	Must write 0xE9.

SFR Definition 26.16. LCD0BUFCN: LCD0 Buffer Control

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

SFR Page = 0xF; SFR Address = 0x9C

Bit	Name	Function
7:0	Reserved	Must write 0x44.

SFR Definition 26.17. LCD0BUFCF: LCD0 Buffer Configuration

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	0

SFR Page = 0xF; SFR Address = 0xAC

Bit	Name	Function
7:0	Reserved	Must write 0x32.

SFR Definition 26.18. LCD0BUFMD: LCD0 Buffer Mode

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	0	1	0

SFR Page = 0x2; SFR Address = 0xB6

Bit	Name	Function
7:0	Reserved	Must write 0x4A.

SFR Definition 26.19. LCD0VBMCF: LCD0 VBAT Monitor Configuration

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	1

SFR Page = 0x2; SFR Address = 0xAF

Bit	Name	Function
7:0	Reserved	Must write 0x0B.

27. Port Input/Output

Digital and analog resources are available through 57 I/O pins (C8051F960/2/4/6/8) or 34 I/O pins (C8051F961/3/5/7/9). Port pins are organized as eight byte-wide ports. Port pins can be defined as digital or analog I/O. Digital I/O pins can be assigned to one of the internal digital resources or used as general purpose I/O (GPIO). Analog I/O pins are used by the internal analog resources. P7.0 can be used as GPIO and is shared with the C2 Interface Data signal (C2D). See Section “34. C2 Interface” on page 486 for more details.

The designer has complete control over which digital and analog functions are assigned to individual port pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. See Section 27.3 for more information on the Crossbar.

For Port I/Os configured as push-pull outputs, current is sourced from the VIO or VIORF supply pin. On 40-pin devices, the VIO and VIORF supply pins are internally tied to VBAT. See Section 27.1 for more information on Port I/O operating modes and the electrical specifications chapter for detailed electrical specifications.

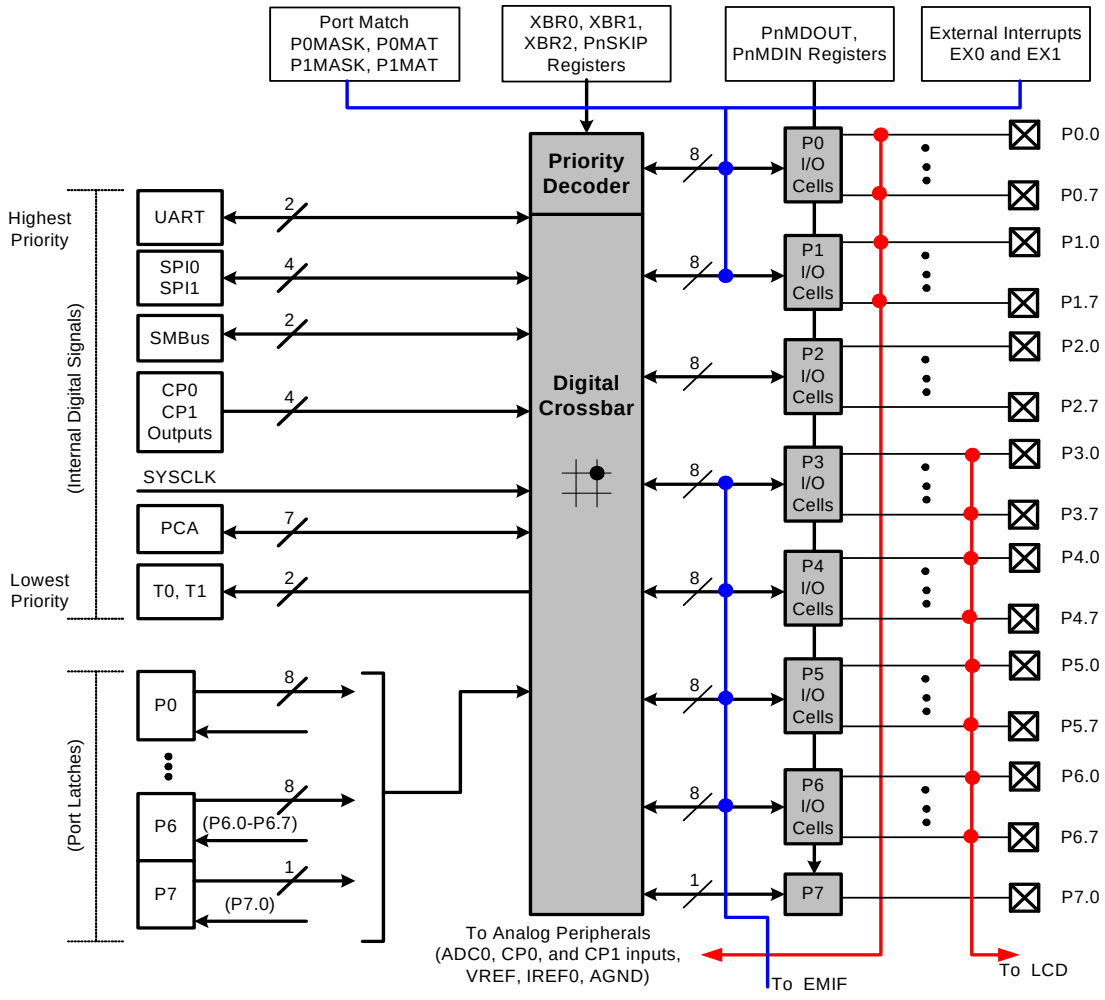


Figure 27.1. Port I/O Functional Block Diagram

27.1. Port I/O Modes of Operation

Port pins P0.0–P6.7 use the Port I/O cell shown in Figure 27.2. The supply pin for P1.4 - P2.3 is VIORF and the supply for all other GPIOs is VIO. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the PnMDIN registers. P7.0 can only be used for digital functtons and is shared with the C2D signal. On reset, all Port I/O cells default to a digital high impedance state with weak pull-ups enabled.

27.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC input, external oscillator input/output, or AGND, VREF, or Current Reference output should be configured for analog I/O (PnMDIN.n = 0). When a pin is configured for analog I/O, its weak pullup and digital receiver are disabled. In most cases, software should also disable the digital output drivers. Port pins configured for analog I/O will always read back a value of 0 regardless of the actual voltage on the pin.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

27.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the supply or GND rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high and low drivers turned off) when the output logic value is 1.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption and may be globally disabled by setting WEAKPUD to 1. The user must ensure that digital I/O are always internally or externally pulled or driven to a valid logic state. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

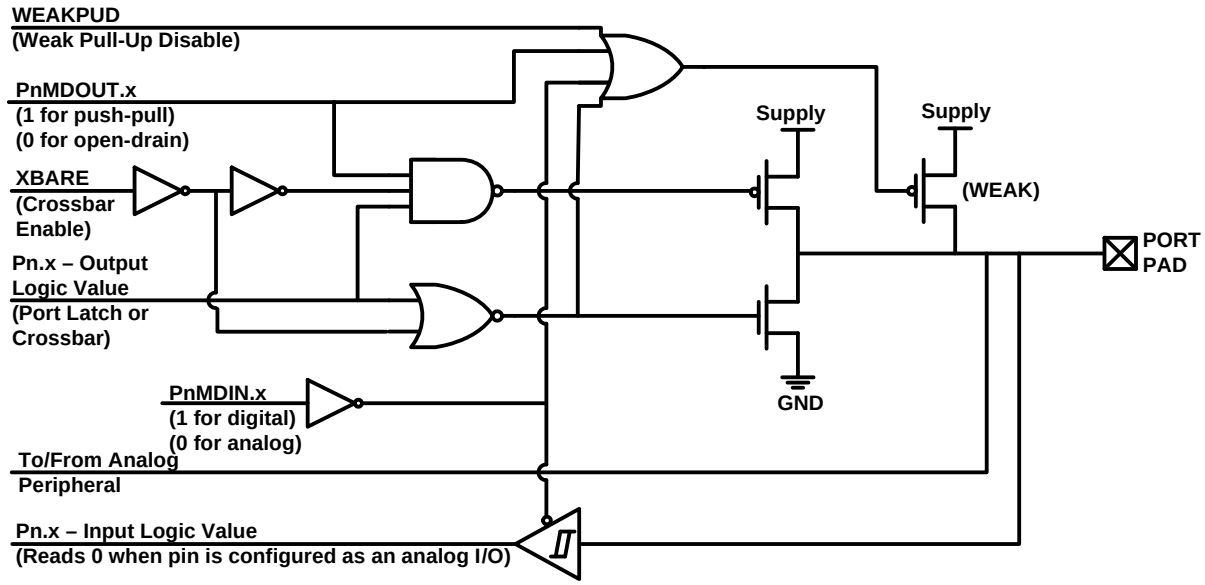


Figure 27.2. Port I/O Cell Block Diagram

27.1.3. Interfacing Port I/O to High Voltage Logic

All Port I/O configured for digital, open-drain operation are capable of interfacing to digital logic operating at a supply voltage up to VBAT + TBD. An external pull-up resistor to the higher supply voltage is typically required for most systems.

27.1.4. Increasing Port I/O Drive Strength

Port I/O output drivers support a high and low drive strength; the default is low drive strength. The drive strength of a Port I/O can be configured using the PnDRV registers. See Section “4. Electrical Characteristics” on page 56 for the difference in output drive strength between the two modes.

27.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P2.6 can be assigned to various analog, digital, and external interrupt functions. The Port pins assigned to analog functions should be configured for analog I/O and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

27.2.1. Assigning Port I/O Pins to Analog Functions

Table 27.1 shows all available analog functions that need Port I/O assignments. **Port pins selected for these analog functions should have their digital drivers disabled (PnMDOUT.n = 0 and Port Latch = 1) and their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 27.1 shows the potential mapping of Port I/O to each analog function.

Table 27.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P0.7, P1.4–P2.3	ADC0MX, PnSKIP
Comparator0 Input	P0.0–P0.7, P1.4–P2.3	CPT0MX, PnSKIP
Comparator1 Input	P0.0–P0.7, P1.4–P2.3	CPT1MX, PnSKIP
LCD Pins (LCD0)	P2.4–P6.7	PnMDIN, PnSKIP
Pulse Counter (PC0)	P1.0, P1.1	P1MDIN, PnSKIP
Voltage Reference (VREF0)	P0.0	REF0CN, PnSKIP
Analog Ground Reference (AGND)	P0.1	REF0CN, PnSKIP
Current Reference (IREF0)	P0.7	IREF0CN, PnSKIP
External Oscillator Input (XTAL1)	P0.2	OSCXCN, PnSKIP
External Oscillator Output (XTAL2)	P0.3	OSCXCN, PnSKIP
SmaRTClock Input (XTAL3)	P1.2	P1MDIN, PnSKIP
SmaRTClock Output (XTAL4)	P1.3	P1MDIN, PnSKIP

27.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 27.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 27.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SPI1, SMBus, CP0 and CP1 Outputs, System Clock Output, PCA0, Timer0 and Timer1 External Inputs.	Any Port pin available for assignment by the Crossbar. This includes P0.0–P2.7 pins which have their PnSKIP bit set to 0. **Note:** The Crossbar will always assign UART0 and SPI1 pins to fixed locations.	XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0–P7.0	P0SKIP, P1SKIP, P2SKIP
External Memory Interface	P3.6–P6.7	EMI0CF

27.2.3. Assigning Port I/O Pins to External Digital Event Capture Functions

External digital event capture functions can be used to trigger an interrupt or wake the device from a low power mode when a transition occurs on a digital I/O pin. The digital event capture functions do not require dedicated pins and will function on both GPIO pins (PnSKIP = 1) and pins in use by the Crossbar (PnSKIP = 0). External digital even capture functions cannot be used on pins configured for analog I/O. Table 27.3 shows all available external digital event capture functions.

Table 27.3. Port I/O Assignment for External Digital Event Capture Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
External Interrupt 0	P0.0–P0.5, P1.6, P1.7	IT01CF
External Interrupt 1	P0.0–P0.4, P1.6, P1.7	IT01CF
Port Match	P0.0–P1.7	P0MASK, P0MAT P1MASK, P1MAT

27.3. Priority Crossbar Decoder

The Priority Crossbar Decoder assigns a Port I/O pin to each software selected digital function using the fixed peripheral priority order shown in Figure 27.3. The registers XBR0, XBR1, and XBR2 defined in SFR Definition 27.1, SFR Definition 27.2, and SFR Definition 27.3 are used to select digital functions in the Crossbar. The Port pins available for assignment by the Crossbar include all Port pins (P0.0–P2.6) which have their corresponding bit in PnSKIP set to 0.

From Figure 27.3, the highest priority peripheral is UART0. If UART0 is selected in the Crossbar (using the XBRn registers), then P0.4 and P0.5 will be assigned to UART0. The next highest priority peripheral is SPI1. If SPI1 is selected in the Crossbar, then P2.0–P2.2 will be assigned to SPI1. P2.3 will be assigned if SPI1 is configured for 4-wire mode. The user should ensure that the pins to be assigned by the Crossbar have their PnSKIP bits set to 0.

For all remaining digital functions selected in the Crossbar, starting at the top of Figure 27.3 going down, the least-significant unskipped, unassigned Port pin(s) are assigned to that function. If a Port pin is already assigned (e.g., UART0 or SPI1 pins), or if its PnSKIP bit is set to 1, then the Crossbar will skip over the pin and find next available unskipped, unassigned Port pin. All Port pins used for analog functions, GPIO, or dedicated digital functions such as the EMIF should have their PnSKIP bit set to 1.

Figure 27.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP, P2SKIP = 0x00); Figure 27.4 shows the Crossbar Decoder priority with the External Oscillator pins (XTAL1 and XTAL2) skipped (P0SKIP = 0x0C).

Important Notes:

- The Crossbar must be enabled (XBARE = 1) before any Port pin is used as a digital output. Port output drivers are disabled while the Crossbar is disabled.
- When SMBus is selected in the Crossbar, the pins associated with SDA and SCL will automatically be forced into open-drain output mode regardless of the PnMDOUT setting.
- SPI0 can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1-NSSMD0 bits in register SPI0CN. The NSS signal is only routed to a Port pin when 4-wire mode is selected. When SPI0 is selected in the Crossbar, the SPI0 mode (3-wire or 4-wire) will affect the pinout of all digital functions lower in priority than SPI0.
- For given XBRn, PnSKIP, and SPInCN register settings, one can determine the I/O pin-out of the device using Figure 27.3 and Figure 27.4.

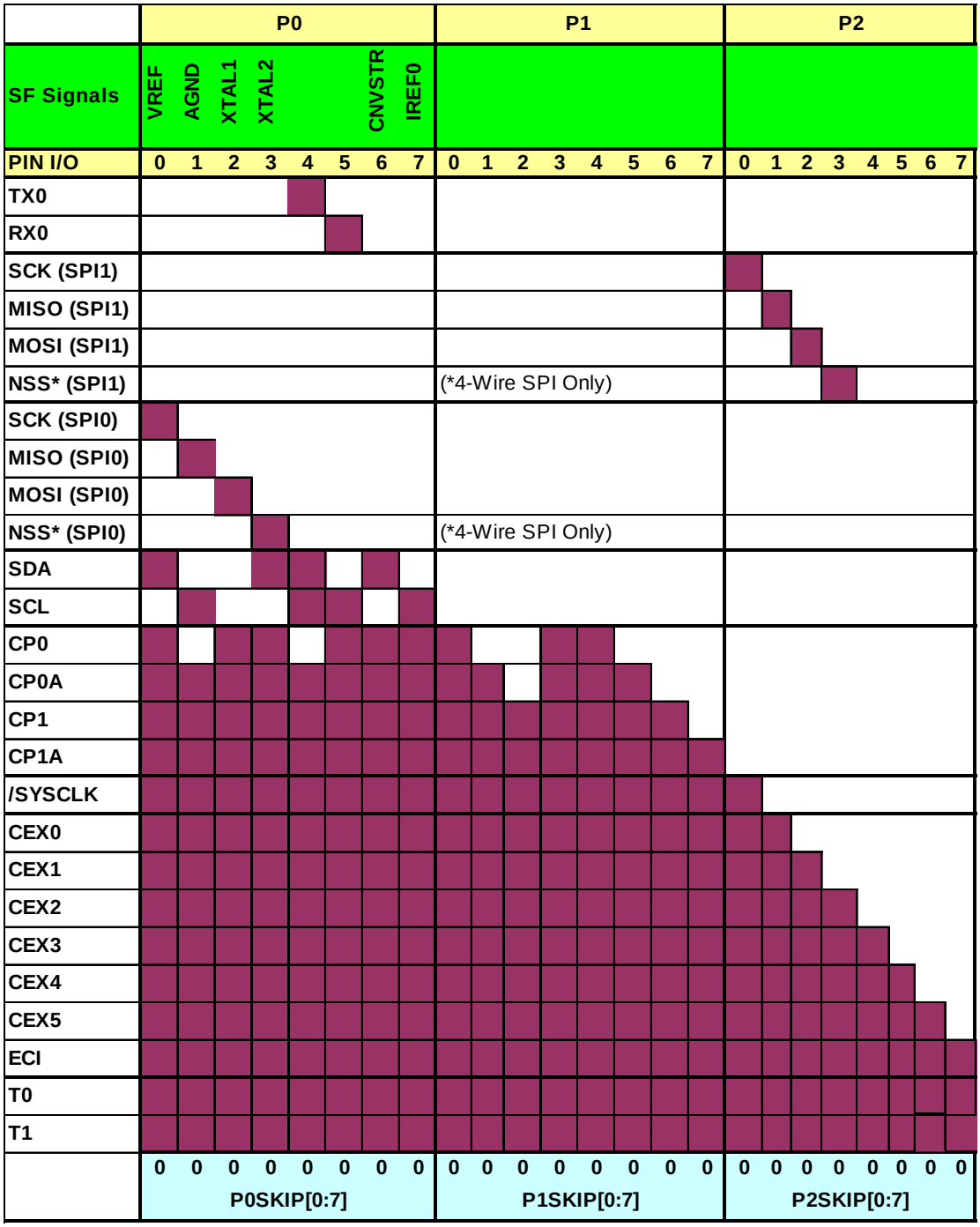


Figure 27.3. Crossbar Priority Decoder with No Pins Skipped

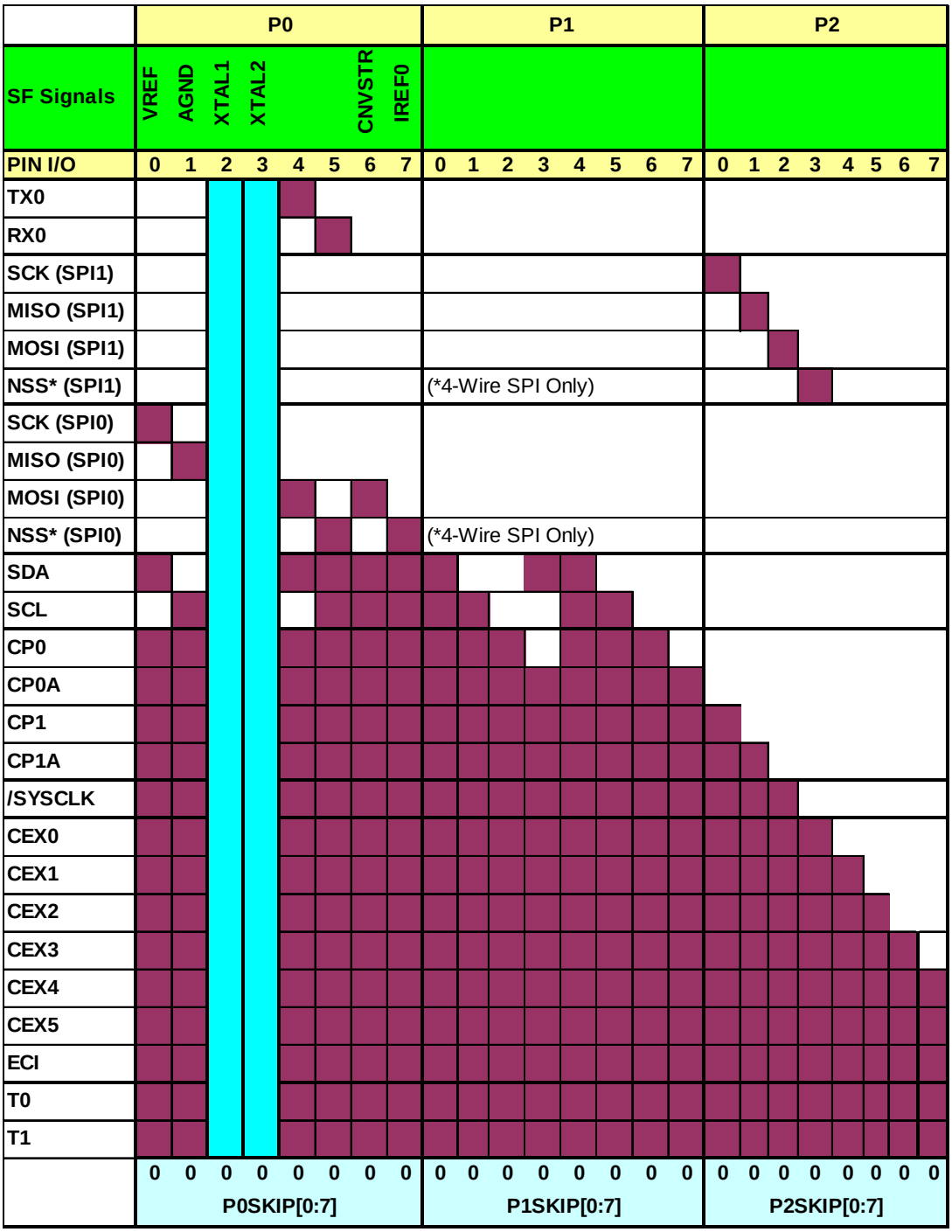


Figure 27.4. Crossbar Priority Decoder with Crystal Pins Skipped

SFR Definition 27.1. XBR0: Port I/O Crossbar Register 0

Bit	7	6	5	4	3	2	1	0
Name	CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0 and 0xF; SFR Address = 0xE1

Bit	Name	Function
7	CP1AE	**Comparator1 Asynchronous Output Enable.** 0: Asynchronous CP1 output unavailable at Port pin. 1: Asynchronous CP1 output routed to Port pin.
6	CP1E	**Comparator1 Output Enable.** 0: CP1 output unavailable at Port pin. 1: CP1 output routed to Port pin.
5	CP0AE	**Comparator0 Asynchronous Output Enable.** 0: Asynchronous CP0 output unavailable at Port pin. 1: Asynchronous CP0 output routed to Port pin.
4	CP0E	**Comparator0 Output Enable.** 0: CP1 output unavailable at Port pin. 1: CP1 output routed to Port pin.
3	SYSCKE	**$\overline{\text{SYSCLK}}$ Output Enable.** 0: $\overline{\text{SYSCLK}}$ output unavailable at Port pin. 1: $\overline{\text{SYSCLK}}$ output routed to Port pin.
2	SMB0E	**SMBus I/O Enable.** 0: SMBus I/O unavailable at Port pin. 1: SDA and SCL routed to Port pins.
1	SPI0E	**SPI0 I/O Enable** 0: SPI0 I/O unavailable at Port pin. 1: SCK, MISO, and MOSI (for SPI0) routed to Port pins. NSS (for SPI0) routed to Port pin only if SPI0 is configured to 4-wire mode.
0	URT0E	**UART0 Output Enable.** 0: UART I/O unavailable at Port pin. 1: TX0 and RX0 routed to Port pins P0.4 and P0.5.
Note: SPI0 can be assigned either 3 or 4 Port I/O pins.		

SFR Definition 27.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name		SPI1E	T1E	T0E	ECIE	PCA0ME[2:0]		
Type	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0 and 0xF; SFR Address = 0xE2

Bit	Name	Function
7	Unused	Read = 0b; Write = Don't Care.
6	SPI1E	**SPI0 I/O Enable.** 0: SPI1 I/O unavailable at Port pin. 1: SCK (for SPI1) routed to P2.0. MISO (for SPI1) routed to P2.1. MOSI (for SPI1) routed to P2.2. NSS (for SPI1) routed to P2.3 only if SPI1 is configured to 4-wire mode.
5	T1E	**Timer1 Input Enable.** 0: T1 input unavailable at Port pin. 1: T1 input routed to Port pin.
4	T0E	**Timer0 Input Enable.** 0: T0 input unavailable at Port pin. 1: T0 input routed to Port pin.
3	ECIE	**PCA0 External Counter Input (ECI) Enable.** 0: PCA0 external counter input unavailable at Port pin. 1: PCA0 external counter input routed to Port pin.
2:0	PCA0ME	**PCA0 Module I/O Enable.** 000: All PCA0 I/O unavailable at Port pin. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to Port pins. 011: CEX0, CEX1, CEX2 routed to Port pins. 100: CEX0, CEX1, CEX2 CEX3 routed to Port pins. 101: CEX0, CEX1, CEX2, CEX3, CEX4 routed to Port pins. 110: CEX0, CEX1, CEX2, CEX3, CEX4, CEX5 routed to Port pins. 111: Reserved.
Note: SPI1 can be assigned either 3 or 4 Port I/O pins.		

SFR Definition 27.3. XBR2: Port I/O Crossbar Register 2

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE						
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0 and 0xF; SFR Address = 0xE3

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable** 0: Weak Pullups enabled (except for Port I/O pins configured for analog mode).
6	XBARE	**Crossbar Enable** 0: Crossbar disabled. 1: Crossbar enabled.
5:0	Unused	Read = 000000b; Write = Don't Care.
Note: The Crossbar must be enabled (XBARE = 1) to use any Port pin as a digital output.		

27.4. Port Match

Port match functionality allows system events to be triggered by a logic value change on P0 or P1. A software controlled value stored in the PnMAT registers specifies the expected or normal logic values of P0 and P1. A Port mismatch event occurs if the logic levels of the Port's input pins no longer match the software controlled value. This allows Software to be notified if a certain change or pattern occurs on P0 or P1 input pins regardless of the XBRn settings.

The PnMASK registers can be used to individually select which P0 and P1 pins should be compared against the PnMAT registers. A Port mismatch event is generated if (P0 & P0MASK) does not equal (PnMAT & P0MASK) or if (P1 & P1MASK) does not equal (PnMAT & P1MASK).

A Port mismatch event may be used to generate an interrupt or wake the device from a low power mode. See Section "17. Interrupt Handler" on page 232 and Section "19. Power Management" on page 257 for more details on interrupt and wake-up sources.

SFR Definition 27.4. P0MASK: Port0 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P0MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0xC7

Bit	Name	Function
7:0	P0MASK[7:0]	**Port0 Mask Value.** Selects the P0 pins to be compared with the corresponding bits in P0MAT. 0: P0.n pin pad logic value is ignored and cannot cause a Port Mismatch event. 1: P0.n pin pad logic value is compared to P0MAT.n.

SFR Definition 27.5. P0MAT: Port0 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P0MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page= 0x0; SFR Address = 0xD7

Bit	Name	Function
7:0	P0MAT[7:0]	**Port 0 Match Value.** Match comparison value used on Port 0 for bits in P0MASK which are set to 1. 0: P0.n pin logic value is compared with logic LOW. 1: P0.n pin logic value is compared with logic HIGH.

SFR Definition 27.6. P1MASK: Port1 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P1MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0xBF

Bit	Name	Function
7:0	P1MASK[7:0]	**Port 1 Mask Value.** Selects P1 pins to be compared to the corresponding bits in P1MAT. 0: P1.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P1.n pin logic value is compared to P1MAT.n.
Note:		

SFR Definition 27.7. P1MAT: Port1 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P1MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xCF

Bit	Name	Function
7:0	P1MAT[7:0]	**Port 1 Match Value.** Match comparison value used on Port 1 for bits in P1MASK which are set to 1. 0: P1.n pin logic value is compared with logic LOW. 1: P1.n pin logic value is compared with logic HIGH.
Note:		

27.5. Special Function Registers for Accessing and Configuring Port I/O

All Port I/O are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable. When writing to a Port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the Port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the Crossbar, the Port register can always read its corresponding Port I/O pin). The exception to this is the execution of the read-modify-write instructions that target a Port Latch register as the destination. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a Port SFR. For these instructions, the value of the latch register (not the pin) is read, modified, and written back to the SFR.

Each Port has a corresponding PnSKIP register which allows its individual Port pins to be assigned to digital functions or skipped by the Crossbar. All Port pins used for analog functions, GPIO, or dedicated digital functions such as the EMIF should have their PnSKIP bit set to 1.

The Port input mode of the I/O pins is defined using the Port Input Mode registers (PnMDIN). Each Port cell can be configured for analog or digital I/O. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is P2.7, which can only be used for digital I/O.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings.

The drive strength of the output drivers are controlled by the Port Drive Strength (PnDRV) registers. The default is low drive strength. See Section “4. Electrical Characteristics” on page 56 for the difference in output drive strength between the two modes.

SFR Definition 27.8. P0: Port0

Bit	7	6	5	4	3	2	1	0
Name	P0[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = All Pages; SFR Address = 0x80; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P0[7:0]	**Port 0 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P0.n Port pin is logic LOW. 1: P0.n Port pin is logic HIGH.

SFR Definition 27.9. P0SKIP: Port0 Skip

Bit	7	6	5	4	3	2	1	0
Name	P0SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0xD4

Bit	Name	Function
7:0	P0SKIP[7:0]	**Port 0 Crossbar Skip Enable Bits.** These bits select Port 0 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P0.n pin is not skipped by the Crossbar. 1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 27.10. P0MDIN: Port0 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P0MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page= 0x0; SFR Address = 0xF1

Bit	Name	Function
7:0	P0MDIN[7:0]	**Analog Configuration Bits for P0.7–P0.0 (respectively).** Port pins configured for analog mode have their weak pullup, and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P0.n pin is configured for analog mode. 1: Corresponding P0.n pin is not configured for analog mode.

SFR Definition 27.11. P0MDOUT: Port0 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P0MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xA4

Bit	Name	Function
7:0	P0MDOUT[7:0]	**Output Configuration Bits for P0.7–P0.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P0MDIN is logic 0. 0: Corresponding P0.n Output is open-drain. 1: Corresponding P0.n Output is push-pull.

SFR Definition 27.12. P0DRV: Port0 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P0DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xA4

Bit	Name	Function
7:0	P0DRV[7:0]	**Drive Strength Configuration Bits for P0.7–P0.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P0.n Output has low output drive strength. 1: Corresponding P0.n Output has high output drive strength.

SFR Definition 27.13. P1: Port1

Bit	7	6	5	4	3	2	1	0
Name	P1[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = All Pages; SFR Address = 0x90; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P1[7:0]	**Port 1 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P1.n Port pin is logic LOW. 1: P1.n Port pin is logic HIGH.

SFR Definition 27.14. P1SKIP: Port1 Skip

Bit	7	6	5	4	3	2	1	0
Name	P1SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD5

Bit	Name	Function
7:0	P1SKIP[7:0]	**Port 1 Crossbar Skip Enable Bits.** These bits select Port 1 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P1.n pin is not skipped by the Crossbar. 1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 27.15. P1MDIN: Port1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xF2

Bit	Name	Function
7:0	P1MDIN[7:0]	**Analog Configuration Bits for P1.7–P1.0 (respectively).** Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.

SFR Definition 27.16. P1MDOUT: Port1 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xA5

Bit	Name	Function
7:0	P1MDOUT[7:0]	**Output Configuration Bits for P1.7–P1.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P1MDIN is logic 0. 0: Corresponding P1.n Output is open-drain. 1: Corresponding P1.n Output is push-pull.

SFR Definition 27.17. P1DRV: Port1 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P1DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xA5

Bit	Name	Function
7:0	P1DRV[7:0]	**Drive Strength Configuration Bits for P1.7–P1.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P1.n Output has low output drive strength. 1: Corresponding P1.n Output has high output drive strength.

SFR Definition 27.18. P2: Port2

Bit	7	6	5	4	3	2	1	0
Name	P2[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = All Pages; SFR Address = 0xA0; Bit-Addressable

Bit	Name	Description	Read	Write
7:0	P2[7:0]	**Port 2 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P2.n Port pin is logic LOW. 1: P2.n Port pin is logic HIGH.

SFR Definition 27.19. P2SKIP: Port2 Skip

Bit	7	6	5	4	3	2	1	0
Name	P2SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD6

Bit	Name	Description	Read	Write
7:0	P2SKIP[7:0]	**Port 1 Crossbar Skip Enable Bits.** These bits select Port 2 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P2.n pin is not skipped by the Crossbar. 1: Corresponding P2.n pin is skipped by the Crossbar.		

SFR Definition 27.20. P2MDIN: Port2 Input Mode

Bit	7	6	5	4	3	2	1	0
Name		P2MDIN[6:0]						
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0xF3

Bit	Name	Function
7	Reserved	Read = 1b; Must Write 1b.
6:0	P2MDIN[3:0]	**Analog Configuration Bits for P2.6–P2.0 (respectively).** Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P2.n pin is configured for analog mode. 1: Corresponding P2.n pin is not configured for analog mode.

SFR Definition 27.21. P2MDOUT: Port2 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P2MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xA6

Bit	Name	Function
7:0	P2MDOUT[7:0]	**Output Configuration Bits for P2.7–P2.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P2MDIN is logic 0. 0: Corresponding P2.n Output is open-drain. 1: Corresponding P2.n Output is push-pull.

SFR Definition 27.22. P2DRV: Port2 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P2DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0F; SFR Address = 0xA6

Bit	Name	Function
7:0	P2DRV[7:0]	**Drive Strength Configuration Bits for P2.7–P2.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P2.n Output has low output drive strength. 1: Corresponding P2.n Output has high output drive strength.

SFR Definition 27.23. P3: Port3

Bit	7	6	5	4	3	2	1	0
Name	P3[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = All Pages; SFR Address = 0xB0; Bit-Addressable

Bit	Name	Description	Read	Write
7:0	P3[7:0]	**Port 3 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P3.n Port pin is logic LOW. 1: P3.n Port pin is logic HIGH.

SFR Definition 27.24. P3MDIN: Port3 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xF1

Bit	Name	Function
7:0	P3MDIN[3:0]	**Analog Configuration Bits for P3.7–P3.0 (respectively).** Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P3.n pin is configured for analog mode. 1: Corresponding P3.n pin is not configured for analog mode.

SFR Definition 27.25. P3MDOUT: Port3 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xB1

Bit	Name	Function
7:0	P3MDOUT[7:0]	**Output Configuration Bits for P3.7–P3.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P3MDIN is logic 0. 0: Corresponding P3.n Output is open-drain. 1: Corresponding P3.n Output is push-pull.

SFR Definition 27.26. P3DRV: Port3 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P3DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xA1

Bit	Name	Function
7:0	P3DRV[7:0]	**Drive Strength Configuration Bits for P3.7–P3.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P3.n Output has low output drive strength. 1: Corresponding P3.n Output has high output drive strength.

SFR Definition 27.27. P4: Port4

Bit	7	6	5	4	3	2	1	0
Name	P4[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xD9

Bit	Name	Description	Read	Write
7:0	P4[7:0]	**Port 4 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P4.n Port pin is logic LOW. 1: P4.n Port pin is logic HIGH.

SFR Definition 27.28. P4MDIN: Port4 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P4MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xF2

Bit	Name	Function
7:0	P4MDIN[3:0]	**Analog Configuration Bits for P4.7–P4.0 (respectively).** Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P4.n pin is configured for analog mode. 1: Corresponding P4.n pin is not configured for analog mode.

SFR Definition 27.29. P4MDOUT: Port4 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P4MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xF9

Bit	Name	Function
7:0	P4MDOUT[7:0]	**Output Configuration Bits for P4.7–P4.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P4MDIN is logic 0. 0: Corresponding P4.n Output is open-drain. 1: Corresponding P4.n Output is push-pull.

SFR Definition 27.30. P4DRV: Port4 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P4DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xA2

Bit	Name	Function
7:0	P4DRV[7:0]	**Drive Strength Configuration Bits for P4.7–P4.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P4.n Output has low output drive strength. 1: Corresponding P4.n Output has high output drive strength.

SFR Definition 27.31. P5: Port5

Bit	7	6	5	4	3	2	1	0
Name	P5[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xDA

Bit	Name	Description	Read	Write
7:0	P5[7:0]	**Port 5 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P5.n Port pin is logic LOW. 1: P5.n Port pin is logic HIGH.

SFR Definition 27.32. P5MDIN: Port5 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P5MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xF3

Bit	Name	Function
7:0	P5MDIN[3:0]	**Analog Configuration Bits for P5.7–P5.0 (respectively).** Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P5.n pin is configured for analog mode. 1: Corresponding P5.n pin is not configured for analog mode.
Note:		

SFR Definition 27.33. P5MDOUT: Port5 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P5MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xFA

Bit	Name	Function
7:0	P5MDOUT[7:0]	**Output Configuration Bits for P5.7–P5.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P5MDIN is logic 0. 0: Corresponding P5.n Output is open-drain. 1: Corresponding P5.n Output is push-pull.
Note:		

SFR Definition 27.34. P5DRV: Port5 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P5DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xA3

Bit	Name	Function
7:0	P5DRV[7:0]	**Drive Strength Configuration Bits for P5.7–P5.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P5.n Output has low output drive strength. 1: Corresponding P5.n Output has high output drive strength.

SFR Definition 27.35. P6: Port6

Bit	7	6	5	4	3	2	1	0
Name	P6[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xDB

Bit	Name	Description	Read	Write
7:0	P6[7:0]	**Port 6 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P6.n Port pin is logic LOW. 1: P6.n Port pin is logic HIGH.

SFR Definition 27.36. P6MDIN: Port6 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P6MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xF4

Bit	Name	Function
7:0	P6MDIN[3:0]	**Analog Configuration Bits for P6.7–P6.0 (respectively).** Port pins configured for analog mode have their weak pullup and digital receiver disabled. The digital driver is not explicitly disabled. 0: Corresponding P6.n pin is configured for analog mode. 1: Corresponding P6.n pin is not configured for analog mode.

SFR Definition 27.37. P6MDOUT: Port6 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P6MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xFB

Bit	Name	Function
7:0	P6MDOUT[7:0]	**Output Configuration Bits for P6.7–P6.0 (respectively).** These bits control the digital driver even when the corresponding bit in register P6MDIN is logic 0. 0: Corresponding P6.n Output is open-drain. 1: Corresponding P6.n Output is push-pull.

SFR Definition 27.38. P6DRV: Port6 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P6DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xAA

Bit	Name	Function
7:0	P6DRV[7:0]	**Drive Strength Configuration Bits for P6.7–P6.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P6.n Output has low output drive strength. 1: Corresponding P6.n Output has high output drive strength.

SFR Definition 27.39. P7: Port7

Bit	7	6	5	4	3	2	1	0
Name								P7.0
Type								R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0xF; SFR Address = 0xDC

Bit	Name	Description	Read	Write
7:1	Unused	Read = 0000000b; Write = Don't Care.		
0	P7.0	**Port 7 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P7.0 Port pin is logic LOW. 1: P7.0 Port pin is logic HIGH.

SFR Definition 27.40. P7MDOUT: Port7 Output Mode

Bit	7	6	5	4	3	2	1	0
Name								P7MDOUT
Type								R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xFC

Bit	Name	Function
7:1	Unused	Read = 0000000b; Write = Don't Care.
0	P7MDOUT.0	**Output Configuration Bits for P7.0.** These bits control the digital driver. 0: P7.0 Output is open-drain. 1: P7.0 Output is push-pull.

SFR Definition 27.41. P7DRV: Port7 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name								P7DRV
Type								R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0xF; SFR Address = 0xAB

Bit	Name	Function
7:1	Unused	Read = 0000000b; Write = Don't Care.
0	P7DRV.0	**Drive Strength Configuration Bits for P7.0.** Configures digital I/O Port cells to high or low output drive strength. 0: P7.0 Output has low output drive strength. 1: P7.0 Output has high output drive strength.

28. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I^2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. The SMBus peripheral can be fully driven by software (i.e., software accepts/rejects slave addresses, and generates ACKs), or hardware slave address recognition and automatic ACK generation can be enabled to minimize software overhead. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 28.1.

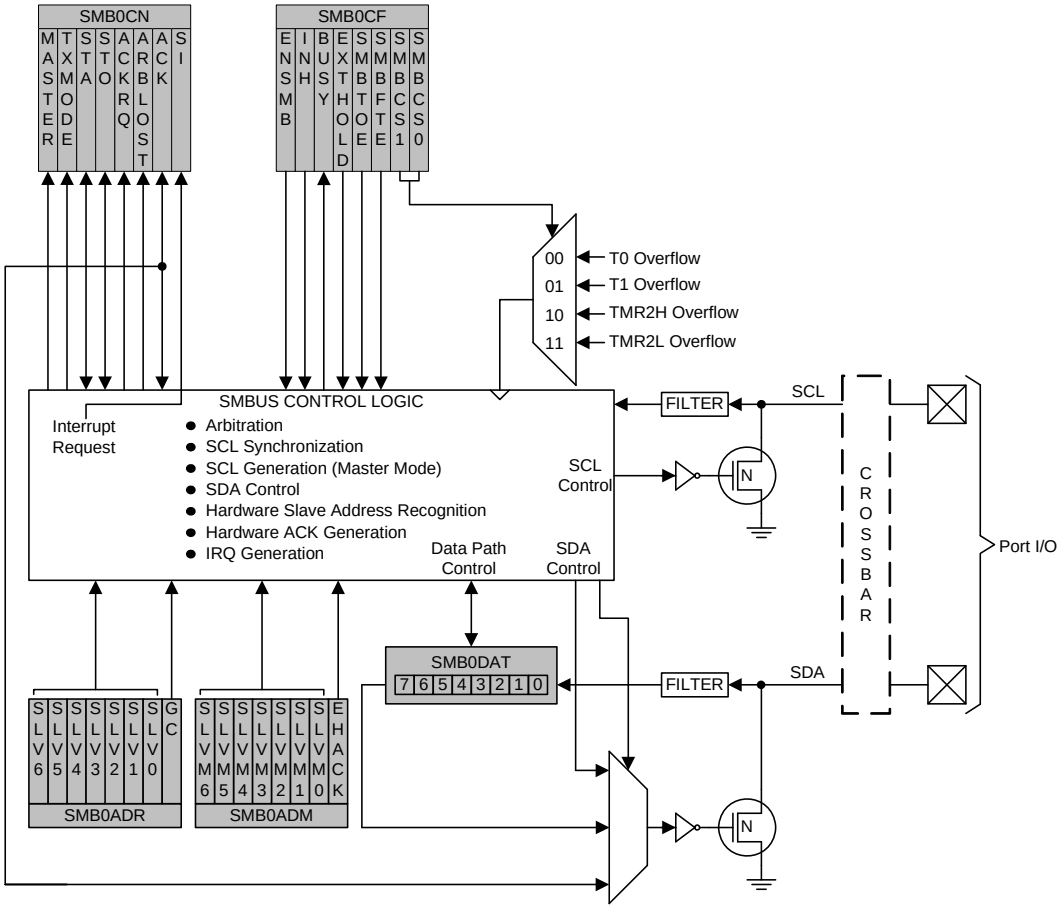


Figure 28.1. SMBus Block Diagram

28.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

28.2. SMBus Configuration

Figure 28.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

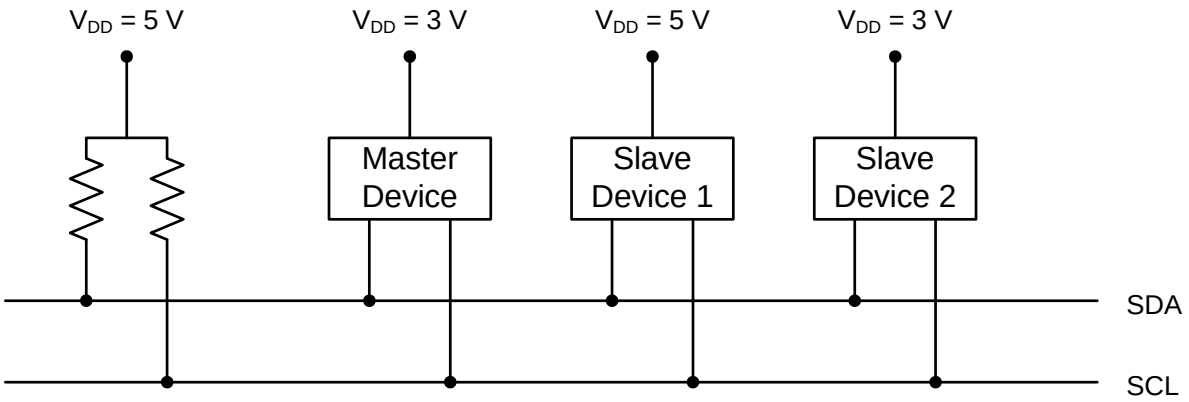


Figure 28.2. Typical SMBus Configuration

28.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. Note that it is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 28.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 28.3 illustrates a typical SMBus transaction.

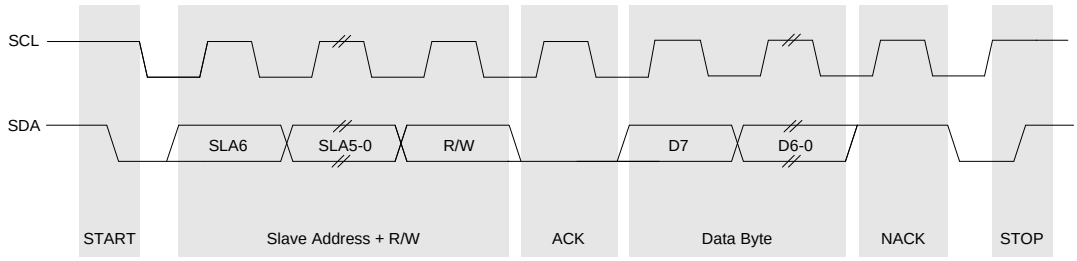


Figure 28.3. SMBus Transaction

28.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

28.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “28.3.5. SCL High (SMBus Free) Timeout” on page 384). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

28.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I^2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

28.3.4. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to

overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

28.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more that 50 µs, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. A clock source is required for free timeout detection, even in a slave-only implementation.

28.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information
- Optional hardware recognition of slave address and automatic acknowledgement of address/data

SMBus interrupts are generated for each data byte or slave address that is transferred. When hardware acknowledgement is disabled, the point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e., sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e., receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. If hardware acknowledgement is enabled, these interrupts are always generated after the ACK cycle. See Section 28.5 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in Section 28.4.2; Table 28.5 provides a quick SMB0CN decoding reference.

28.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

Table 28.1. SMBus Clock Source Selection

SMBCS1	SMBCS0	SMBus Clock Source
0	0	Timer 0 Overflow
0	1	Timer 1 Overflow
1	0	Timer 2 High Byte Overflow
1	1	Timer 2 Low Byte Overflow

The SMBCS1–0 bits select the SMBus clock source, which is used only when operating as a master or when the Free Timeout detection is enabled. When operating as a master, overflows from the selected source determine the absolute minimum SCL low and high times as defined in Equation 28.1. The selected clock source may be shared by other peripherals so long as the timer is left running at all times. For example, Timer 1 overflows may generate the SMBus and UART baud rates simultaneously. Timer configuration is covered in Section “32. Timers” on page 444.

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

Equation 28.1. Minimum SCL High and Low Times

The selected clock source should be configured to establish the minimum SCL High and Low times as per Equation 28.1. When the interface is operating as a master (and SCL is not driven or extended by any other devices on the bus), the typical SMBus bit rate is approximated by Equation 28.1.

$$BitRate = \frac{f_{ClockSourceOverflow}}{3}$$

Equation 28.2. Typical SMBus Bit Rate

Figure 28.4 shows the typical SCL generation described by Equation 28.2. Notice that T_{HIGH} is typically twice as large as T_{LOW}. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by Equation 28.2.

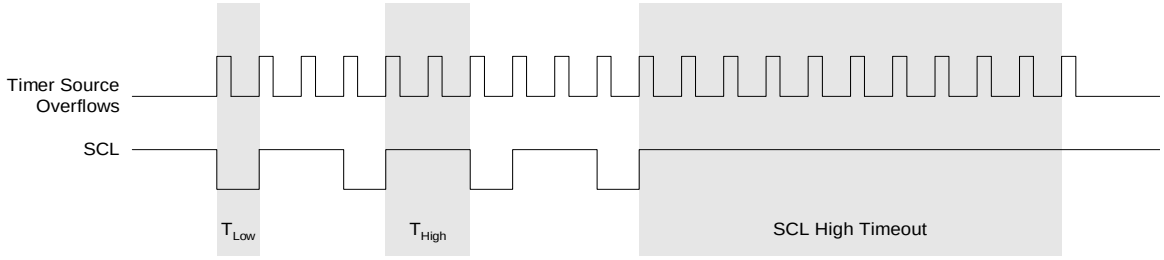


Figure 28.4. Typical SMBus SCL Generation

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 28.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 28.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	T_{low} – 4 system clocks or 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks

***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgement, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “28.3.4. SCL Low Timeout” on page 383). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 28.4).

SFR Definition 28.1. SMB0CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC1

Bit	Name	Function
7	ENSMB	**SMBus Enable.** This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	**SMBus Slave Inhibit.** When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	**SMBus Busy Indicator.** This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	**SMBus Setup and Hold Time Extension Enable.** This bit controls the SDA setup and hold times according to Table 28.2. 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	**SMBus SCL Timeout Detection Enable.** This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	**SMBus Free Timeout Detection Enable.** When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS[1:0]	**SMBus Clock Source Selection.** These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 28.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10:Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

28.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 28.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 28.3 for more details.
Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

28.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

28.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 28.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.
Table 28.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 28.5 for SMBus status decoding using the SMB0CN register.

SFR Definition 28.2. SMB0CN: SMBus Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI
Type	R	R	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xC0; Bit-Addressable

Bit	Name	Description	Read	Write
7	MASTER	**SMBus Master/Slave Indicator.** This read-only bit indicates when the SMBus is operating as a master.	0: SMBus operating in slave mode. 1: SMBus operating in master mode.	N/A
6	TXMODE	**SMBus Transmit Mode Indicator.** This read-only bit indicates when the SMBus is operating as a transmitter.	0: SMBus in Receiver Mode. 1: SMBus in Transmitter Mode.	N/A
5	STA	**SMBus Start Flag.**	0: No Start or repeated Start detected. 1: Start or repeated Start detected.	0: No Start generated. 1: When Configured as a Master, initiates a START or repeated START.
4	STO	**SMBus Stop Flag.**	0: No Stop condition detected. 1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).	0: No STOP condition is transmitted. 1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle. Cleared by Hardware.
3	ACKRQ	**SMBus Acknowledge Request.**	0: No Ack requested 1: ACK requested	N/A
2	ARBLOST	**SMBus Arbitration Lost Indicator.**	0: No arbitration error. 1: Arbitration Lost	N/A
1	ACK	**SMBus Acknowledge.**	0: NACK received. 1: ACK received.	0: Send NACK 1: Send ACK
0	SI	**SMBus Interrupt Flag.** This bit is set by hardware under the conditions listed in Table 15.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.	**0: No interrupt pending** 1: Interrupt Pending	0: Clear interrupt, and initiate next state machine event. 1: Force interrupt.

Table 28.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	• A START is generated.	• A STOP is generated. • Arbitration is lost.
TXMODE	• START is generated. • SMB0DAT is written before the start of an SMBus frame.	• A START is detected. • Arbitration is lost. • SMB0DAT is not written before the start of an SMBus frame.
STA	• A START followed by an address byte is received.	• Must be cleared by software.
STO	• A STOP is detected while addressed as a slave. • Arbitration is lost due to a detected STOP.	• A pending STOP is generated.
ACKRQ	• A byte has been received and an ACK response value is needed (only when hardware ACK is not enabled).	• After each ACK cycle.
ARBLOST	• A repeated START is detected as a MASTER when STA is low (unwanted repeated START). • SCL is sensed low while attempting to generate a STOP or repeated START condition. • SDA is sensed low while transmitting a 1 (excluding ACK bits).	• Each time SI is cleared.
ACK	• The incoming ACK value is low (ACKNOWLEDGE).	• The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	• A START has been generated. • Lost arbitration. • A byte has been transmitted and an ACK/NACK received. • A byte has been received. • A START or repeated START followed by a slave address + R/W has been received. • A STOP has been received.	• Must be cleared by software.

28.4.3. Hardware Slave Address Recognition

The SMBus hardware has the capability to automatically recognize incoming slave addresses and send an ACK without software intervention. Automatic slave address recognition is enabled by setting the EHACK bit in register SMB0ADM to 1. This will enable both automatic slave address recognition and automatic hardware ACK generation for received bytes (as a master or slave). More detail on automatic hardware ACK generation can be found in Section 28.4.2.2.

The registers used to define which address(es) are recognized by the hardware are the SMBus Slave Address register (SFR Definition 28.3) and the SMBus Slave Address Mask register (SFR Definition 28.4). A single address or range of addresses (including the General Call Address 0x00) can be specified using these two registers. The most-significant seven bits of the two registers are used to define which addresses will be ACKed. A 1 in bit positions of the slave address mask SLVM[6:0] enable a comparison between the received slave address and the hardware's slave address SLV[6:0] for those bits. A 0 in a bit of the slave address mask means that bit will be treated as a "don't care" for comparison purposes. In this case, either a 1 or a 0 value are acceptable on the incoming slave address. Additionally, if the GC bit in register SMB0ADR is set to 1, hardware will recognize the General Call Address (0x00). Table 28.4 shows some example parameter settings and the slave addresses that will be recognized by hardware under those conditions.

Table 28.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV[6:0]	Slave Address Mask SLVM[6:0]	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00 (General Call)
0x34	0x7E	0	0x34, 0x35
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

SFR Definition 28.3. SMB0ADR: SMBus Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV[6:0]							GC
Type	R/W							R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xF4

Bit	Name	Function
7:1	SLV[6:0]	**SMBus Hardware Slave Address.** Defines the SMBus Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM[6:0] are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC	**General Call Address Enable.** When hardware address recognition is enabled (EHACK = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

SFR Definition 28.4. SMB0ADM: SMBus Slave Address Mask

Bit	7	6	5	4	3	2	1	0
Name	SLVM[6:0]							EHACK
Type	R/W							R/W
Reset	1	1	1	1	1	1	1	0

SFR Page = 0x0; SFR Address = 0xF5

Bit	Name	Function
7:1	SLVM[6:0]	**SMBus Slave Address Mask.** Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM[6:0] enables comparisons with the corresponding bit in SLV[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK	**Hardware Acknowledge Enable.** Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled.

28.4.4. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 28.5. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xC2

Bit	Name	Function
7:0	SMB0DAT[7:0]	**SMBus Data.** The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

28.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. Note that the position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur **after** the ACK, regardless of whether hardware ACK generation is enabled or not.

28.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt.

Figure 28.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. All “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

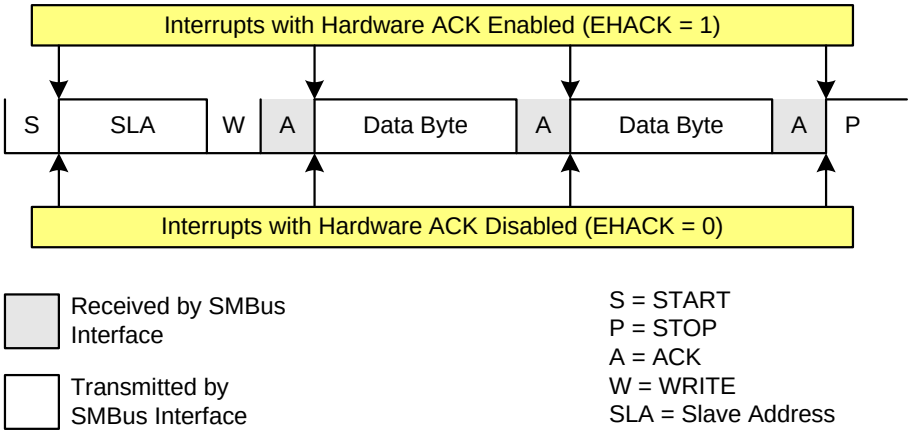


Figure 28.5. Typical Master Write Sequence

28.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 28.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. The “data byte transferred” interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.

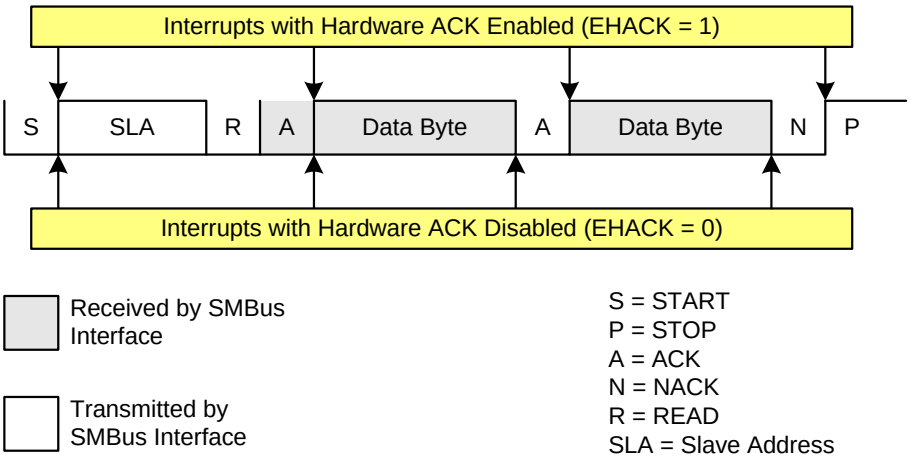


Figure 28.6. Typical Master Read Sequence

28.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **The appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 28.7 shows a typical slave write sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.

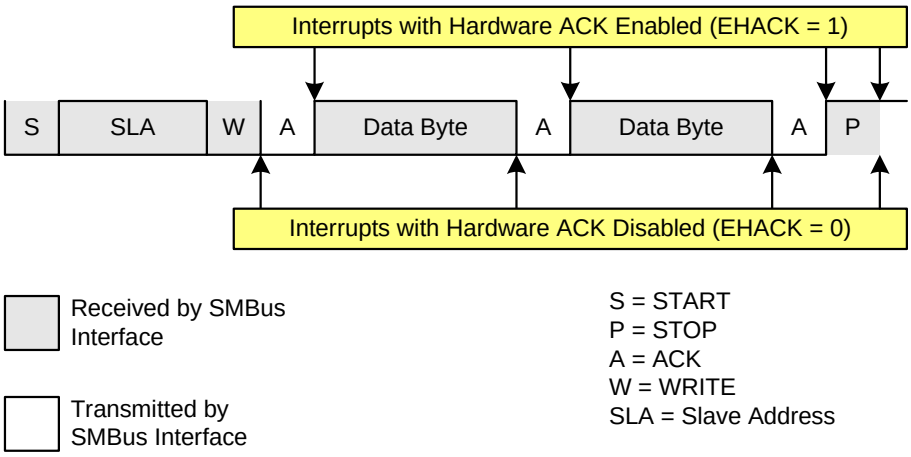


Figure 28.7. Typical Slave Write Sequence

28.5.4. Read Sequence (Slave)

During a read sequence, an SMBus master reads data from a slave device. The slave in this transfer will be a receiver during the address byte, and a transmitter during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode (to receive the slave address) when a START followed by a slave address and direction bit (READ in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are transmitted. If the received slave address is acknowledged, data should be written to SMB0DAT to be transmitted. The interface enters Slave Transmitter Mode, and transmits one or more bytes of data. After each byte is transmitted, the master sends an acknowledge bit; if the acknowledge bit is an ACK, SMB0DAT should be written with the next data byte. If the acknowledge bit is a NACK, SMB0DAT should not be written to before SI is cleared (an error condition may be generated if SMB0DAT is written following a received NACK while in Slave Transmitter Mode). The interface exits Slave Transmitter Mode after receiving a STOP. Note that the interface will switch to Slave Receiver Mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 28.8 shows a typical slave read sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. All of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

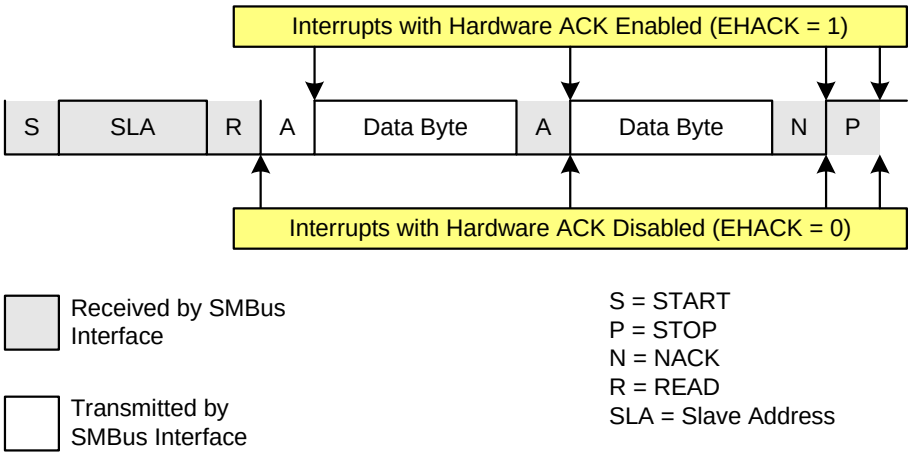


Figure 28.8. Typical Slave Read Sequence

28.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. The appropriate actions to take in response to an SMBus event depend on whether hardware slave address recognition and ACK generation is enabled or disabled. Table 28.5 describes the typical actions when hardware slave address recognition and ACK generation is disabled. Table 28.6 describes the typical actions when hardware slave address recognition and ACK generation is enabled. In the tables, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed by hardware but do not conform to the SMBus specification.

Table 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	-
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	-
						End transfer with STOP and start another transfer.	1	1	X	-
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X	1000
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						Send NACK to indicate last byte, and send STOP.	0	1	0	-
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

Table 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	-
Slave Receiver	0010	1	0	X	A slave address + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	-
		1	1	X	Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	-
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	-
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	-
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	-
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	-
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	-
						Reschedule failed transfer.	1	0	X	1110
	0000	1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0	-
						Reschedule failed transfer.	1	0	0	1110

Table 28.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	-
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	-
						End transfer with STOP and start another transfer.	1	1	X	-
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). Set ACK for initial data byte.	0	0	1	1000
Master Receiver	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
		0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	-
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100

Table 28.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	0	0	X	A slave address + R/W was received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
		0	1	X	Lost arbitration as master; slave address + R/W received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
						Reschedule failed transfer	1	0	X	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		0	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	0	0	X	A slave byte was received.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	0000
						Set NACK for next data byte; Read SMB0DAT.	0	0	0	0000
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	0	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110

29. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “29.1. Enhanced Baud Rate Generation” on page 403). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

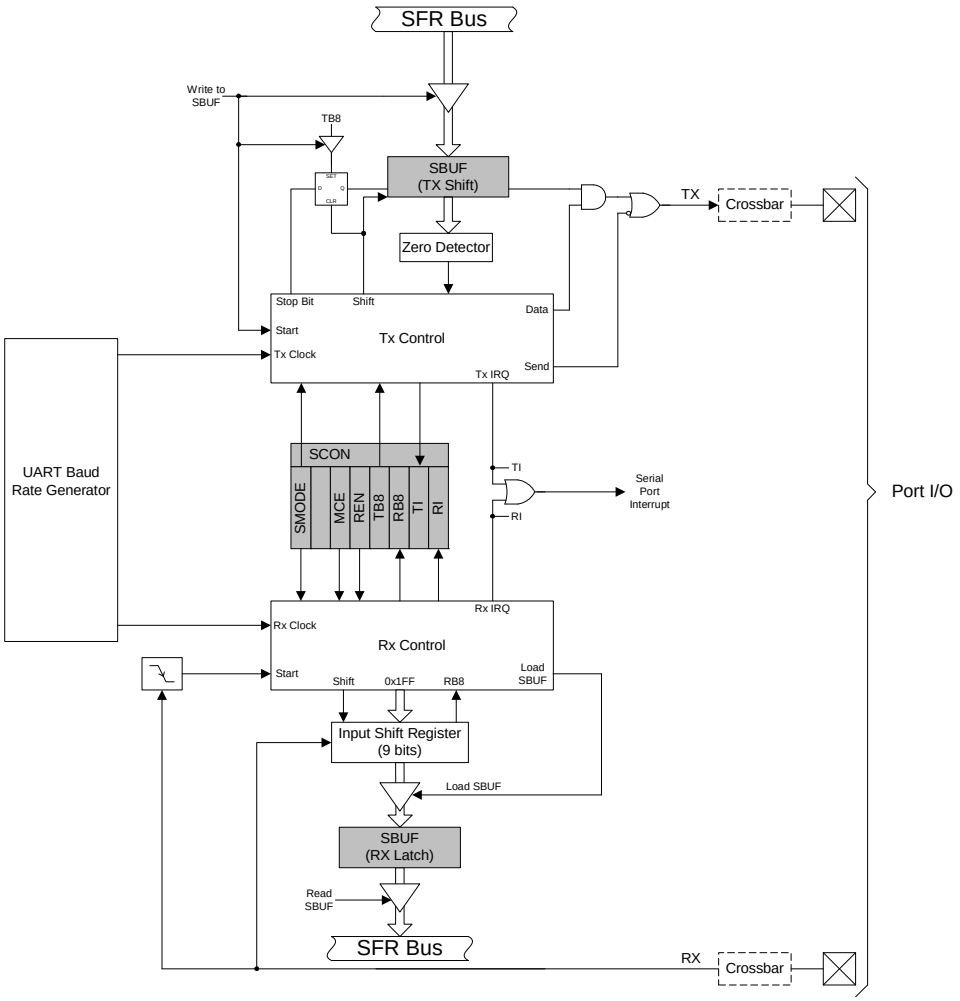


Figure 29.1. UART0 Block Diagram

29.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 29.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

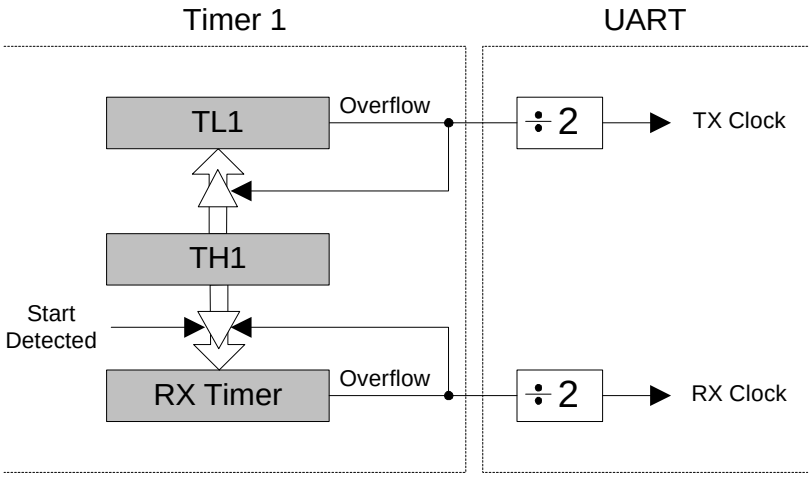


Figure 29.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “32.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 447). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK / 4, SYSCLK / 12, SYSCLK / 48, the external oscillator clock / 8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation -A and Equation -B.

A) $$\text{UartBaudRate} = \frac{1}{2} \times \text{T1_Overflow_Rate}$$

B) $$\text{T1_Overflow_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$

UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section “32.1. Timer 0 and Timer 1” on page 446. A quick reference for typical baud rates and system clock frequencies is given in Table 29.1 through Table 29.2. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

29.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown below.

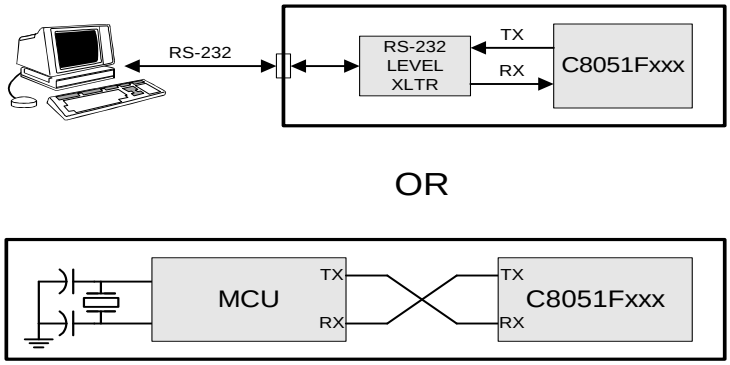


Figure 29.3. UART Interconnect Diagram

29.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

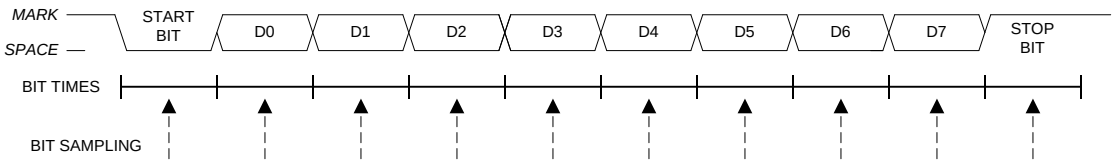


Figure 29.4. 8-Bit UART Timing Diagram

29.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

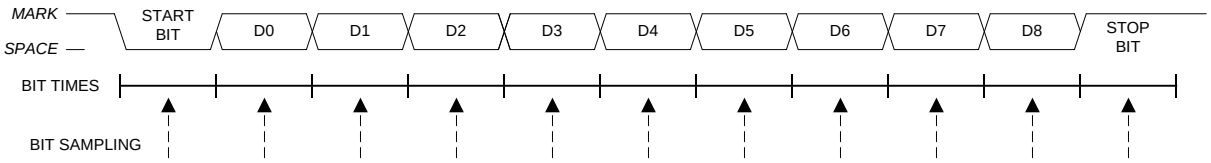


Figure 29.5. 9-Bit UART Timing Diagram

29.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

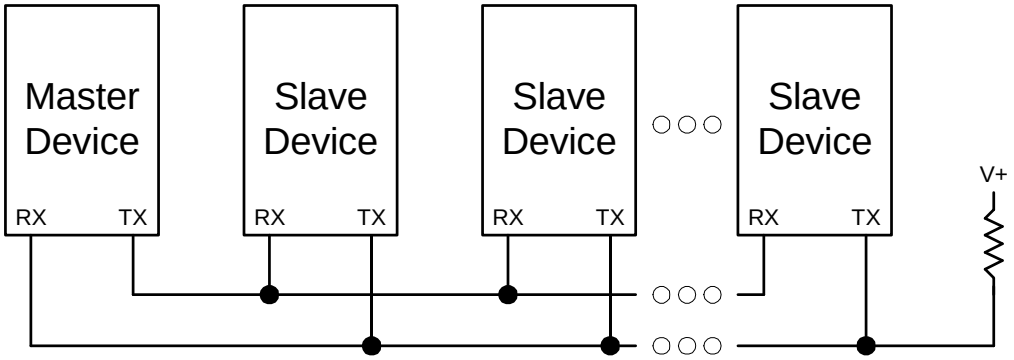


Figure 29.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 29.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	S0MODE		MCE0	REN0	TB80	RB80	TI0	RI0
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x98; Bit-Addressable

Bit	Name	Function
7	S0MODE	**Serial Port 0 Operation Mode.** Selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate. 1: 9-bit UART with Variable Baud Rate.
6	Unused	Read = 1b. Write = Don't Care.
5	MCE0	**Multiprocessor Communication Enable.** **For Mode 0 (8-bit UART): Checks for valid stop bit.** 0: Logic level of stop bit is ignored. 1: RI0 will only be activated if stop bit is logic level 1. **For Mode 1 (9-bit UART): Multiprocessor Communications Enable.** 0: Logic level of ninth bit is ignored. 1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.
4	REN0	**Receive Enable.** 0: UART0 reception disabled. 1: UART0 reception enabled.
3	TB80	**Ninth Transmission Bit.** The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB80	**Ninth Receive Bit.** RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI0	**Transmit Interrupt Flag.** Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	**Receive Interrupt Flag.** Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

SFR Definition 29.2. SBUF0: Serial (UART0) Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF0[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x99

Bit	Name	Function
7:0	SBUF0	**Serial Data Buffer Bits 7:0 (MSB–LSB).** This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Table 29.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX[2]	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 32.1.
2. X = Don't care.

Table 29.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	96	SYSCLK	XX[2]	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK / 12	00	0	0xE0
	14400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	9600	0.00%	2304	SYSCLK / 12	00	0	0xA0
	2400	0.00%	9216	SYSCLK / 48	10	0	0xA0
	1200	0.00%	18432	SYSCLK / 48	10	0	0x40

Table 29.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:
1. SCA1−SCA0 and T1M bit definitions can be found in Section 32.1.
2. X = Don't care.

30. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

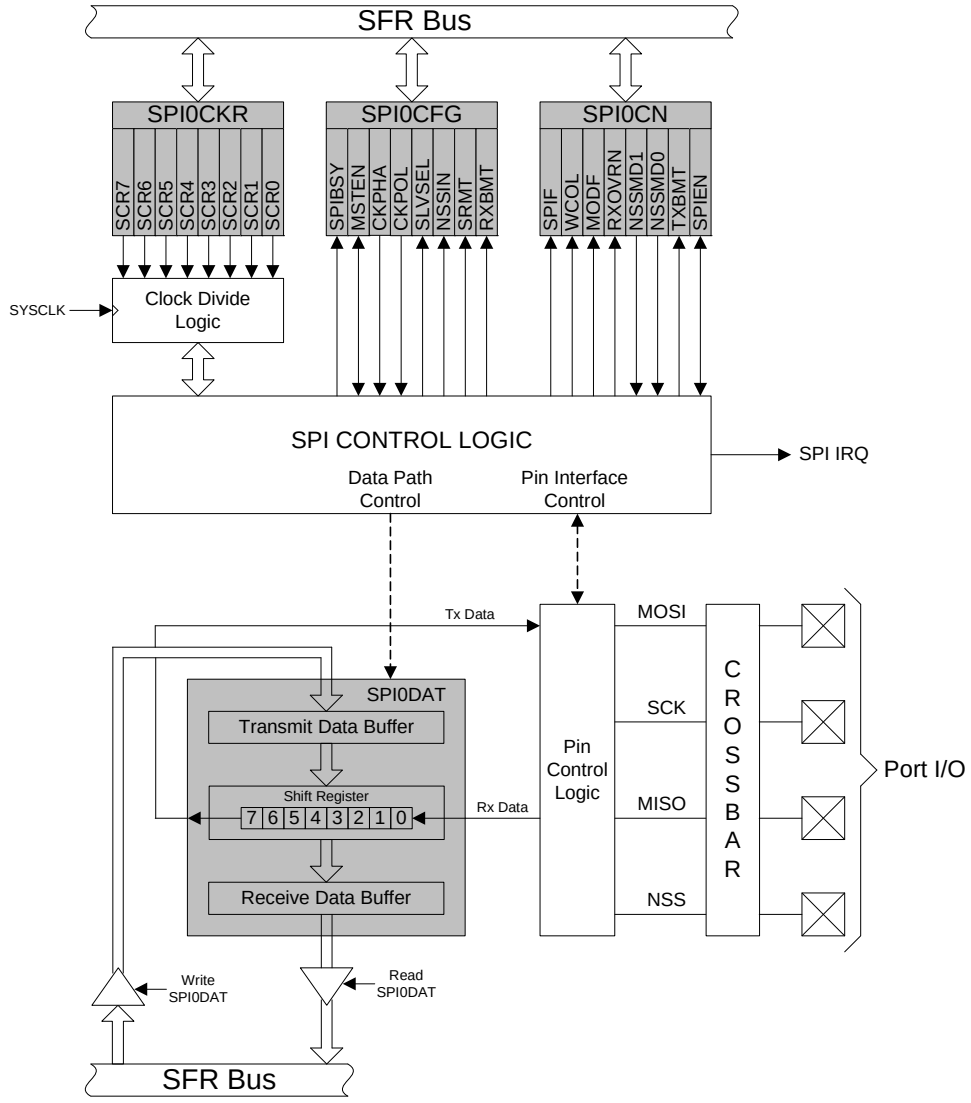


Figure 30.1. SPI Block Diagram

30.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

30.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

30.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

30.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

30.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 30.2, Figure 30.3, and Figure 30.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “27. Port Input/Output” on page 351 for general purpose port I/O and crossbar information.

30.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag

is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 30.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 30.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 30.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

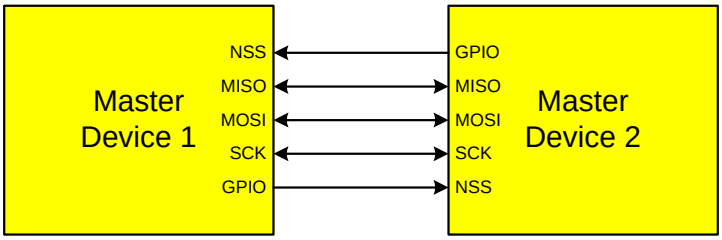


Figure 30.2. Multiple-Master Mode Connection Diagram

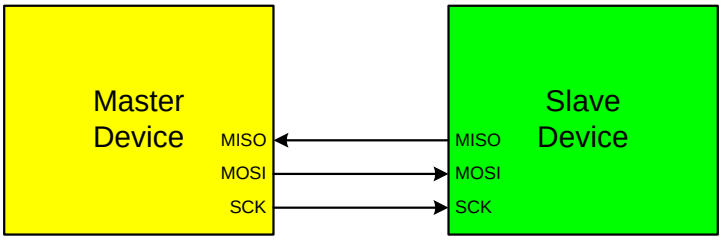


Figure 30.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

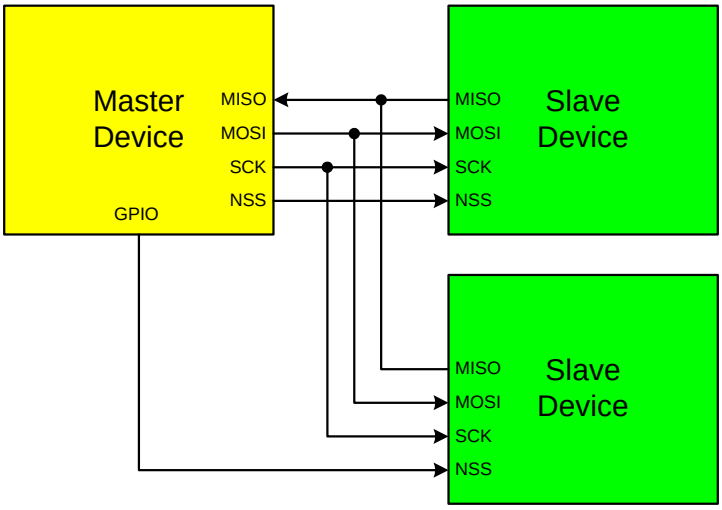


Figure 30.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

30.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data,

the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 30.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 30.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

30.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

- The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
- The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
- The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
- The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

30.5. Serial Clock Phase and Polarity

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 30.5. For slave mode, the clock and data relationships are shown in Figure 30.6 and Figure 30.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 30.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-

wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

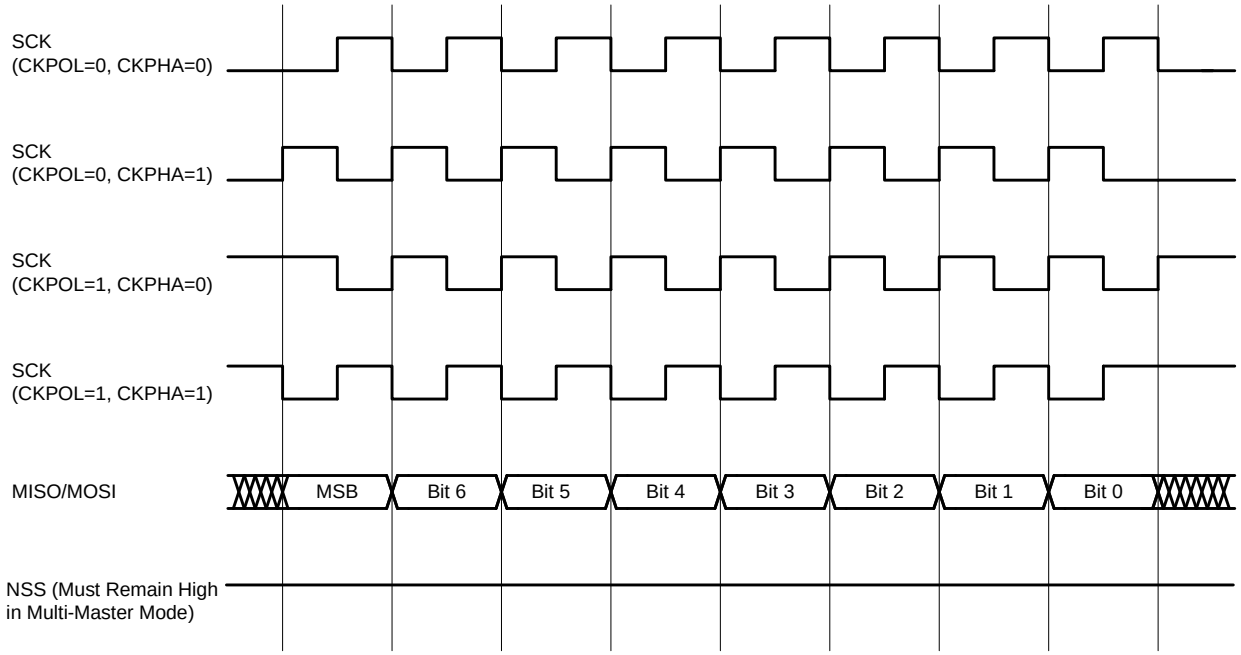


Figure 30.5. Master Mode Data/Clock Timing

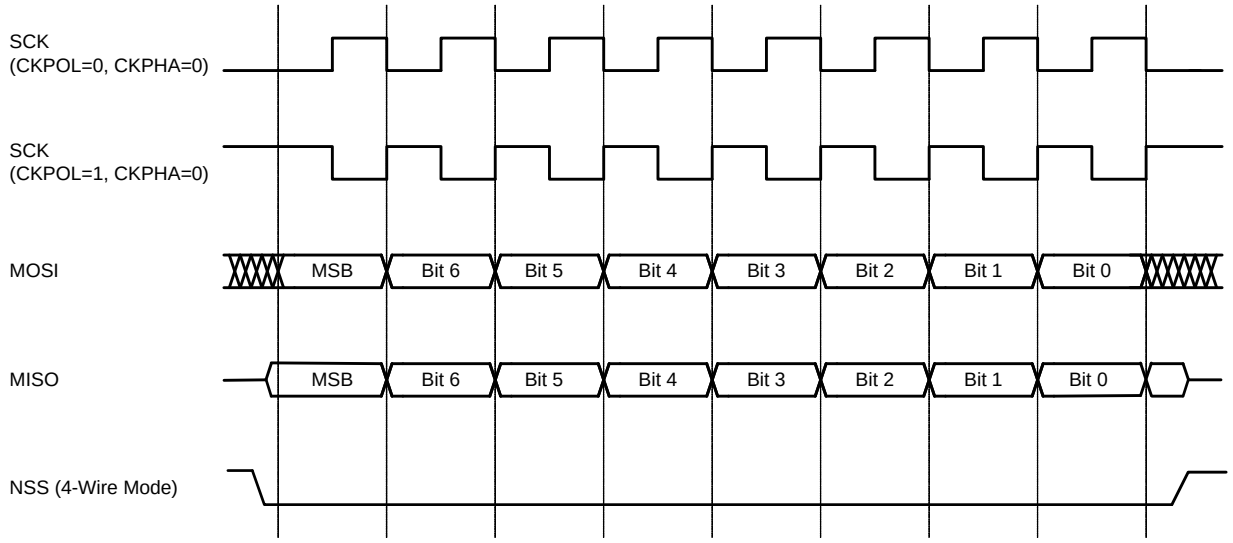


Figure 30.6. Slave Mode Data/Clock Timing (CKPHA = 0)

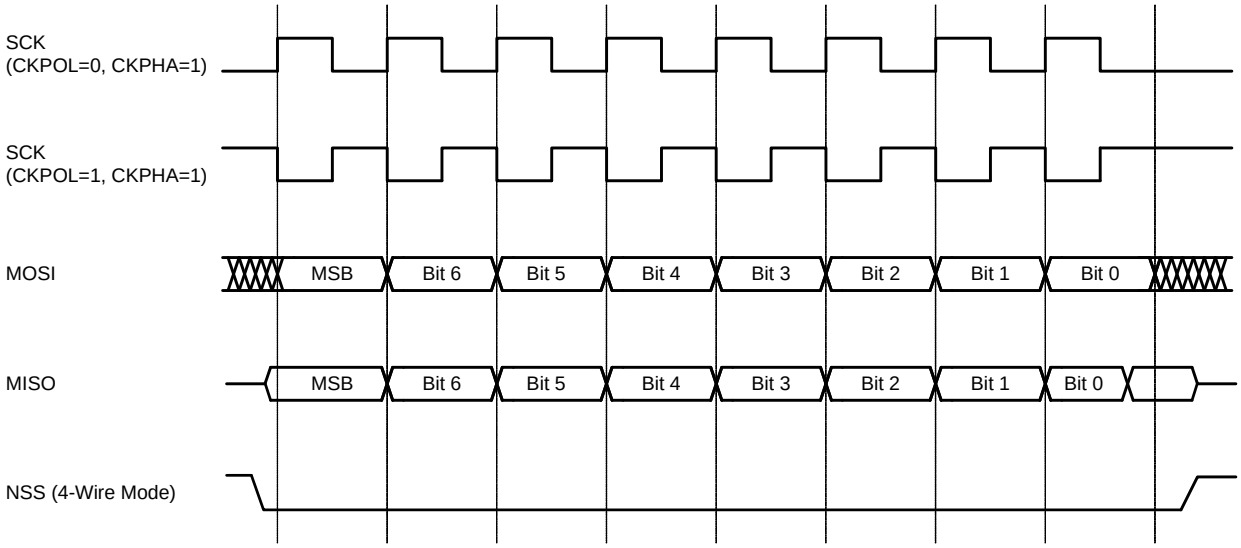


Figure 30.7. Slave Mode Data/Clock Timing (CKPHA = 1)

30.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 30.1. SPI0CFG: SPI0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Page = 0x0; SFR Address = 0xA1

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI0 Clock Phase.** 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	**SPI0 Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	**NSS Instantaneous Pin Input.** This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	**Shift Register Empty (valid in slave mode only).** This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
0	RXBMT	**Receive Buffer Empty (valid in slave mode only).** This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.

Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 30.1 for timing parameters.

SFR Definition 30.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0]		TXBMT	SPIEN
Type	R/W	R/W	R/W	R/W	R/W		R	R/W
Reset	0	0	0	0	0	1	1	0

SFR Page = 0x0; SFR Address = 0xF8; Bit-Addressable

Bit	Name	Function
7	SPIF	**SPI0 Interrupt Flag.** This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	**Write Collision Flag.** This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	**Mode Fault Flag.** This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	**Receive Overrun Flag (valid in slave mode only).** This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD[1:0]	**Slave Select Mode.** Selects between the following NSS operation modes: (See Section 30.2 and Section 30.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMT	**Transmit Buffer Empty.** This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPIEN	**SPI0 Enable.** 0: SPI disabled. 1: SPI enabled.

SFR Definition 30.3. SPI0CKR: SPI0 Clock Rate

Bit	7	6	5	4	3	2	1	0
Name	SCR[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xA2

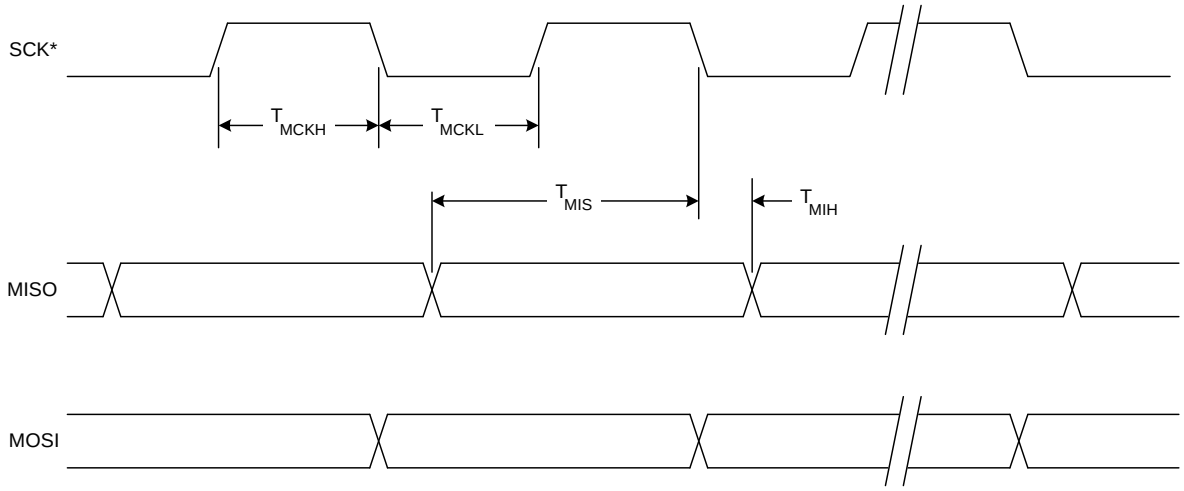
Bit	Name	Function
7:0	SCR[7:0]	**SPI0 Clock Rate.** These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR[7:0] + 1)}$ for 0 <= SPI0CKR <= 255 Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 30.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

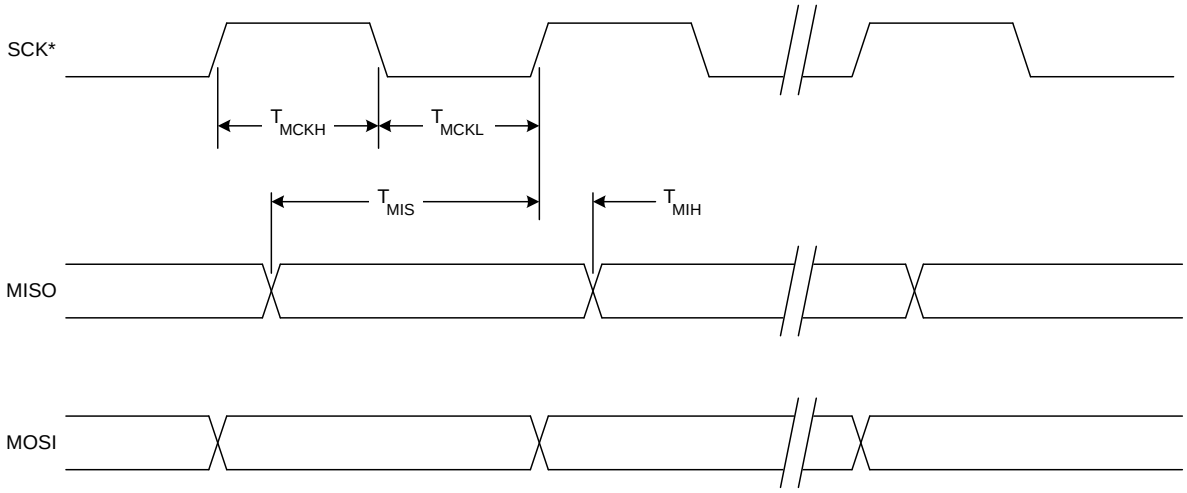
SFR Page = 0x0; SFR Address = 0xA3

Bit	Name	Function
7:0	SPI0DAT[7:0]	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.



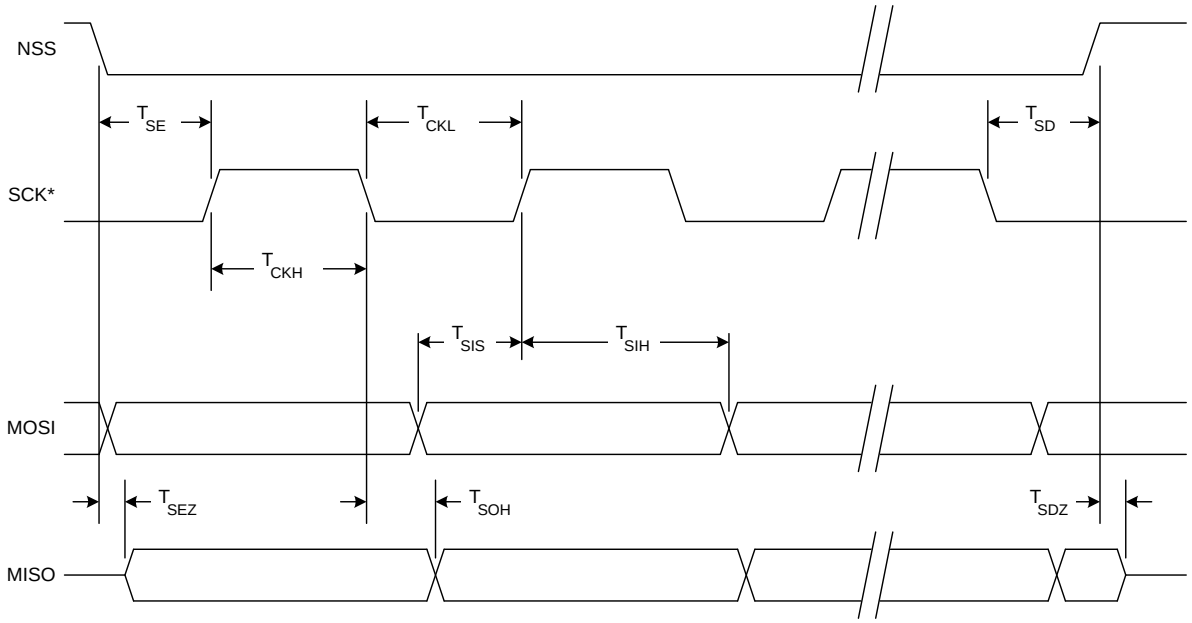
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.8. SPI Master Timing (CKPHA = 0)



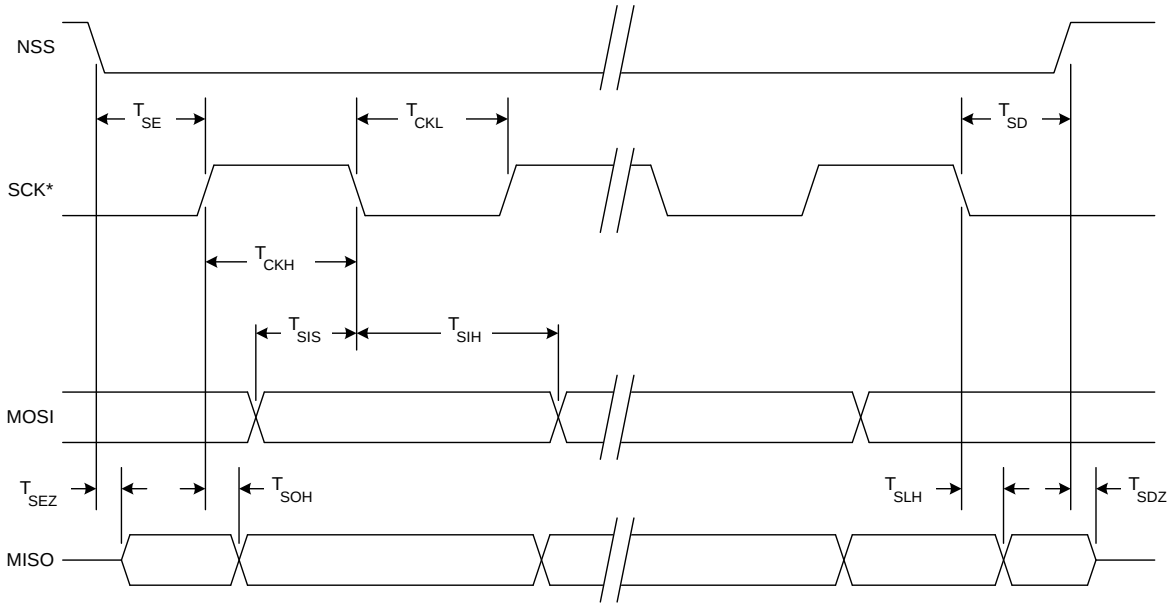
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.9. SPI Master Timing (CKPHA = 1)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 30.11. SPI Slave Timing (CKPHA = 1)

Table 30.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing (See Figure 30.8 and Figure 30.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing (See Figure 30.10 and Figure 30.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns
Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

31. Enhanced Serial Peripheral Interface with DMA Support (SPI1)

The Enhanced Serial Peripheral Interface (SPI1) provides access to a flexible, full-duplex synchronous serial bus. SPI1 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI1 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

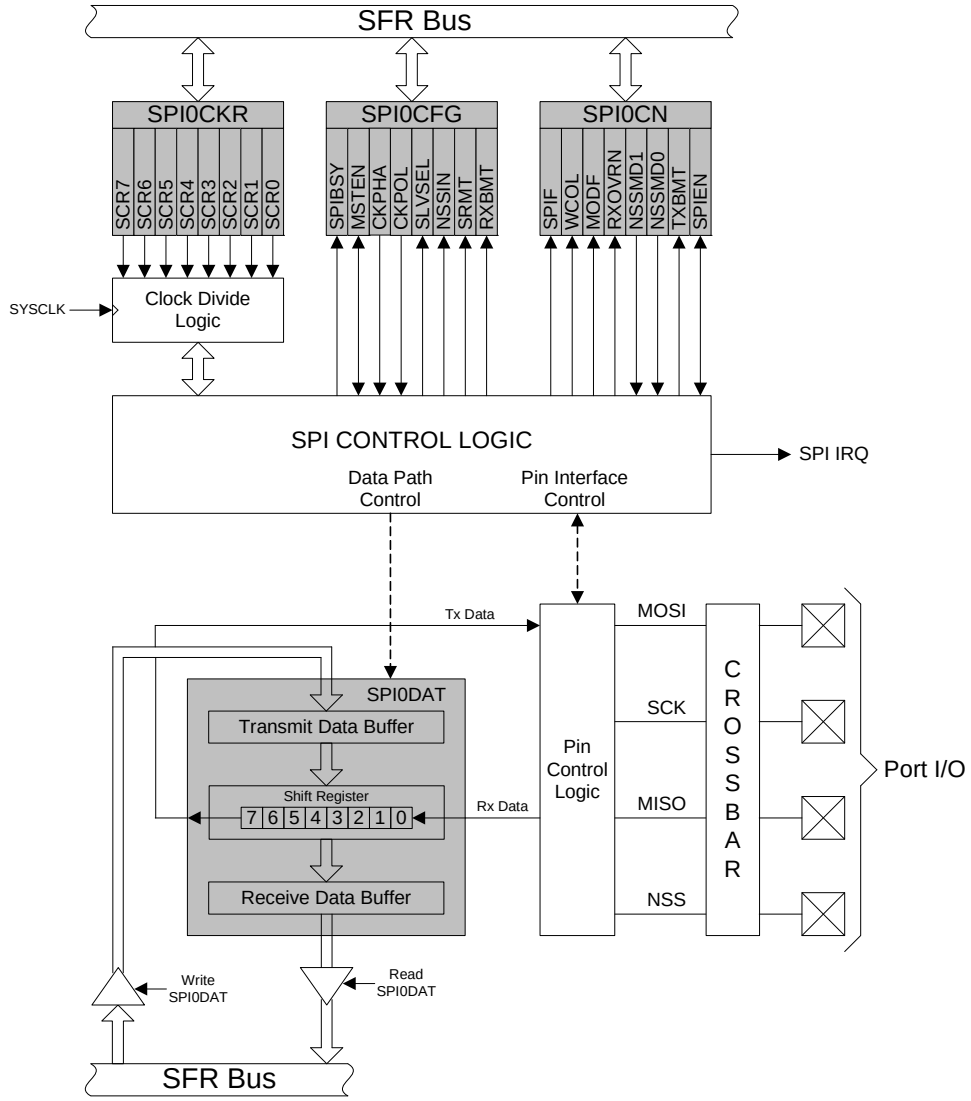


Figure 31.1. SPI Block Diagram

31.1. Signal Descriptions

The four signals used by SPI1 (MOSI, MISO, SCK, NSS) are described below.

31.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI1 is operating as a master and an input when SPI1 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

31.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI1 is operating as a master and an output when SPI1 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

31.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI1 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

31.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI1CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI1 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI1 is always selected in 3-wire mode. Since no select signal is present, SPI1 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI1 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI1 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI1 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI1 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI1 as a master device.

See Figure 31.2, Figure 31.3, and Figure 31.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “27. Port Input/Output” on page 351 for general purpose port I/O and crossbar information.

31.2. SPI1 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI1 is placed in master mode by setting the Master Enable flag (MSTEN, SPI1CN.6). Writing a byte of data to the SPI1 data register (SPI1DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI1 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI1CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI1 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI1DAT.

When configured as a master, SPI1 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI1CN.3) = 0 and NSSMD0 (SPI1CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI1 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI1CN.6) and SPIEN (SPI1CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI1CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI1 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 31.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI1CN.3) = 0 and NSSMD0 (SPI1CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 31.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI1CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI1CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 31.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

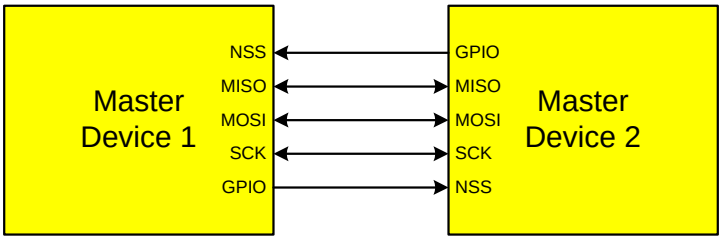


Figure 31.2. Multiple-Master Mode Connection Diagram

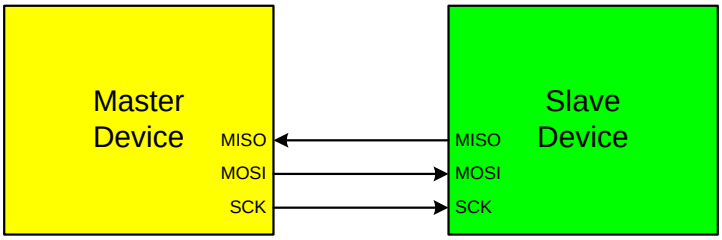


Figure 31.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

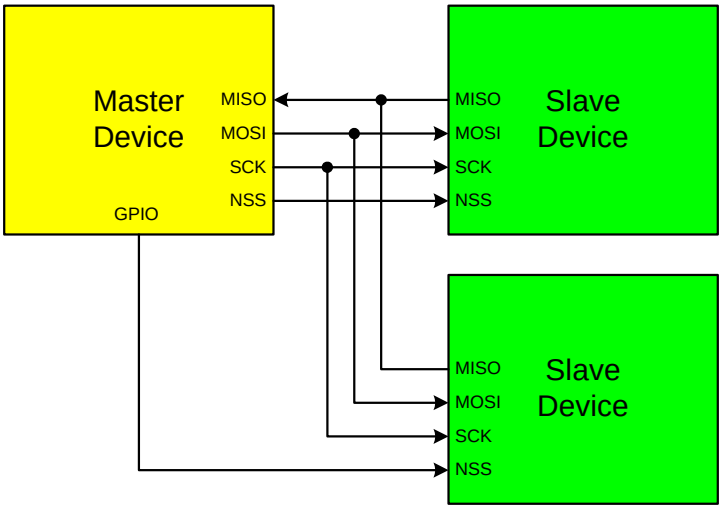


Figure 31.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

31.3. SPI1 Slave Mode Operation

When SPI1 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI1 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI1DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI1DAT. Writes to SPI1DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI1 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI1CN.3) = 0 and NSSMD0 (SPI1CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI1 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 31.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI1CN.3) = 0 and NSSMD0 (SPI1CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI1 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI1 with the SPIEN bit. Figure 31.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

31.4. SPI1 Interrupt Sources

When SPI1 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

- The SPI Interrupt Flag, SPIF (SPI1CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI1 modes.
- The Write Collision Flag, WCOL (SPI1CN.6) is set to logic 1 if a write to SPI1DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI1DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI1 modes.
- The Mode Fault Flag MODF (SPI1CN.5) is set to logic 1 when SPI1 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI1CN are set to logic 0 to disable SPI1 and allow another master device to access the bus.
- The Receive Overrun Flag RXOVRN (SPI1CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

31.5. Serial Clock Phase and Polarity

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI1 Configuration Register (SPI1CFG). The CKPHA bit (SPI1CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI1CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI1 should be disabled (by clearing the SPIEN bit, SPI1CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 31.5. For slave mode, the clock and data relationships are shown in Figure 31.6 and Figure 31.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

The SPI1 Clock Rate Register (SPI1CKR) as shown in SFR Definition 31.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

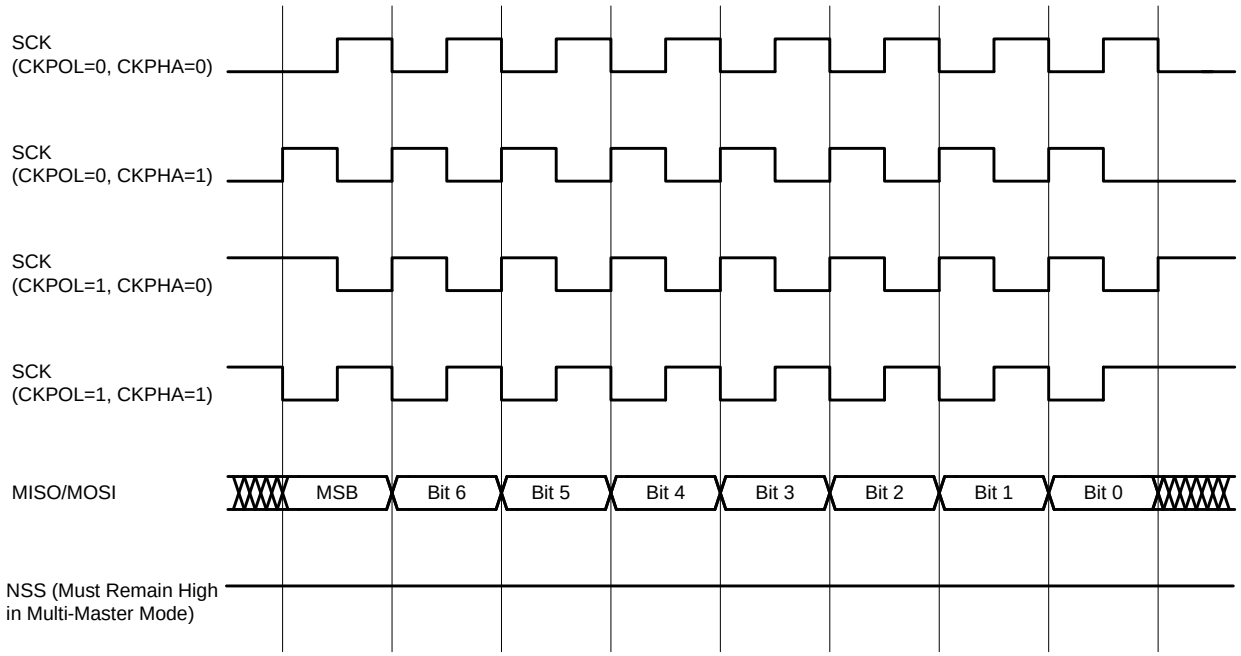


Figure 31.5. Master Mode Data/Clock Timing

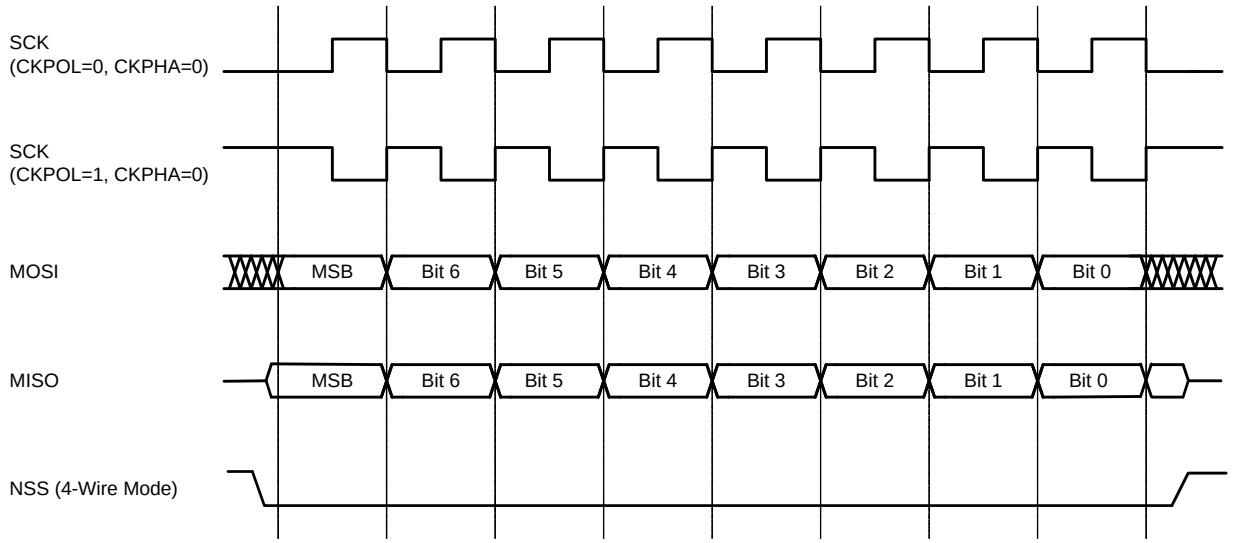


Figure 31.6. Slave Mode Data/Clock Timing (CKPHA = 0)

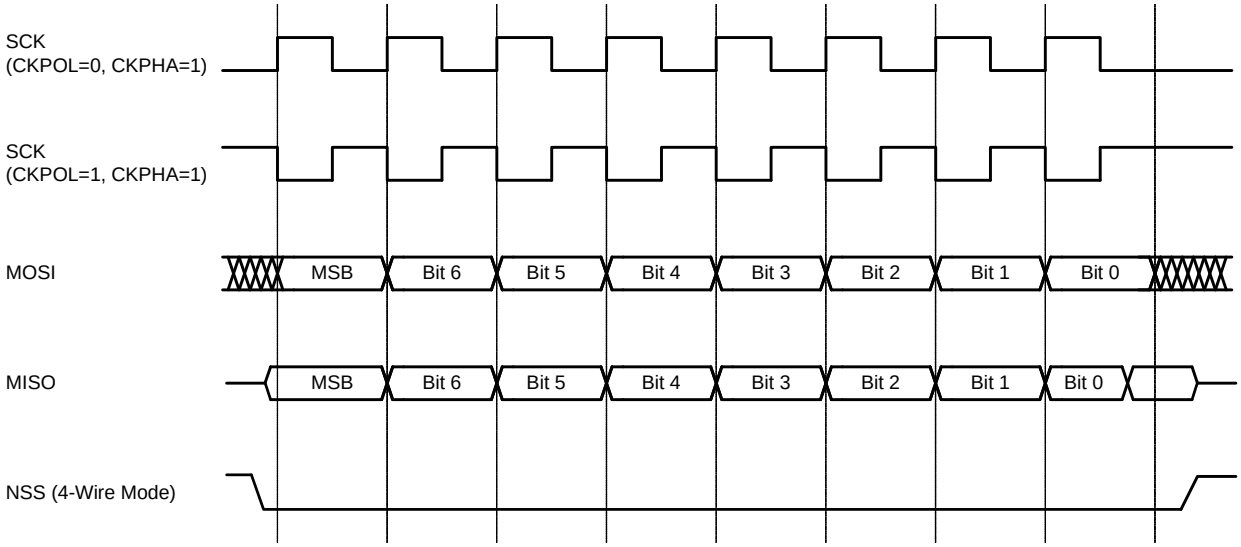


Figure 31.7. Slave Mode Data/Clock Timing (CKPHA = 1)

31.6. Using SPI1 with the DMA

SPI1 is a DMA-enabled peripheral that can provide autonomous data transfers when used with the DMA. The DMA-enabled SPI1 supports both master and slave mode. The SPI requires two DMA channels for a bidirectional data transfer and also supports unidirectional data transfers using a single DMA channel.

There are no additional control bits in the SPI1 control and configuration SFRs. The configuration is the same in DMA and non-DMA mode. While the SPIF flag and/or SPI interrupts are normally used for non-DMA SPI transfers, a DMA transfer is managed using the DMA enable and DMA full transfer complete flags.

More information on using the SPI1 peripheral can be found in the detailed example code for SPI1 Master and Slave modes.

31.7. Master Mode SPI1 DMA Transfers

The SPI interface does not normally have any handshaking or flow control. Therefore, the Master will transmit all of the output data without waiting on the slave peripheral. The system designer must ensure that the slave peripheral can accept all of the data at the transfer rate.

31.8. Master Mode Bidirectional Data Transfer

A bidirectional SPI Master Mode DMA transfer will transmit a specified number of bytes out on the MOSI pin and receive the same number of bytes on the MISO pin. The MOSI data must be stored in XRAM before initiating the DMA transfers. The DMA will also transfer all the MISO data to XRAM, overwriting any data at the target location.

A bidirectional transfer requires two DMA channels. The first DMA channel transfers data from XRAM to the SPI1DAT SFR and the second DMA channel transfers data from the SPI1DAT SFR to XRAM. The second channel DMA interrupt indicates SPI transfer completion.

In master mode, the NSS pin is an output and the hardware does not manage the NSS pin automatically. Normally, firmware should assert the NSS pin before the SPI transfer and deassert it upon completion of the transfer. When using 4-wire Master mode, bit 2 of SPI1CN controls the state of the NSS pin. When using 3-wire master mode, firmware may use any GPIO pin as NSS.

To initiate a Master mode Bidirectional data transfer:

1. Configure the SPI1 SFRs normally for Master mode.
   a. Enable Master mode by setting bit 6 in SPI1CFG.
   b. Configure the clock polarity CKPOL and clock phase CKPHA as desired in SPI1CFG.
   c. Configure SPI1CKR for the desired SPI clock rate.
   d. Configure the desired 4-wire master or 3-wire master mode in SPI1CN.
   e. Enable the SPI by setting bit 0 of SPI1CN.
2. Configure the first DMA channel for the XRAM-to-SPI1DATA transfer:
   a. Disable the first DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the first DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the XRAM-to-SPI1DAT peripheral request by writing 0x03 to DMA0NCF.
   d. Write 0 to DMA0NMD to disable wrapping.
   e. Write the address of the first byte of master output (MOSI) data to DMA0NBAH:L.
   f. Write the size of the SPI transfer in bytes to DMA0NSZH:L.
   g. Clear the address offset SFRs CMA0A0H:L.
3. Configure the second DMA channel for the SPI1DAT-to-XRAM transfer:
   a. Disable the second DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the second DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the SPI1DAT-to-XRAM peripheral request by writing 0x04 to DMA0NCF.
   d. Enable DMA interrupts for the second channel by setting bit 7 of DMA0NCF.
   e. Write 0 to DMA0NMD to disable wrapping.
   f. Write the address for the first byte of master input (MISO) data to DMA0NBAH:L.
   g. Write the size of the SPI transfer in bytes to DMA0NSZH:L.
   h. Clear the address offset SFRs CMA0A0H:L.
   i. Enable the interrupt on the second channel by setting the corresponding bit in DMA0INT.
   j. Enable DMA interrupts by setting bit 5 of EIE2.
4. Clear the interrupt bits in DMA0INT for both channels.
5. Enable both channels by setting the corresponding bits in the DMA0EN SFR to initiate the SPI transfer operation.
6. Wait on the DMA interrupt.
7. Clear the DMA enables in the DMA0EN SFR.
8. Clear the DMA interrupts in the DMA0INT SFR.

31.9. Master Mode Unidirectional Data Transfer

A unidirectional SPI master mode DMA transfer will transfer a specified number of bytes out on the MOSI pin. The MOSI data must be stored in XRAM before initiating the DMA transfers. The SPI1DAT-to-XRAM peripheral request is not used. Since the DMA does not read the SPI1DAT SFR, the SPI will discard the MISO data.

A unidirectional transfer only requires one DMA channel to transfer XRAM data to the SPI1DAT SFR. The DMA interrupt will indicate the completion of the data transfer to the SPI1DAT SFR. When the interrupt occurs, the DMA has written all of the data to the SPI1DAT SFR, but the SPI has not transmitted the last byte. Firmware may poll on the SPIBSY bit to determine when the SPI has transmitted the last byte. Firmware should not deassert the NSS pin until after the SPI has transmitted the last byte.

To initiate a master mode unidirectional data transfer:

1. Configure the SPI1 SFRs normally for Master mode.
   a. Enable Master mode by setting bit 6 in SPI1CFG.
   b. Configure the clock polarity CKPOL and clock phase CKPHA as desired in SPI1CFG.
   c. Configure SPI1CKR for the desired SPI clock rate.
   d. Configure the desired 4-wire master or 3-wire master mode in SPI1CN.
   e. Enable the SPI by setting bit 0 of SPI1CN.
2. Configure the desired DMA channel for the XRAM-to-SPI1DAT transfer.
   a. Disable the desired DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the desired DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the XRAM-to-SPI1DAT XRAM peripheral request by writing 0x03 to DMA0NCF.
   d. Enable DMA interrupts for the desired channel by setting bit 7 of DMA0NCF.
   e. Write 0 to DMA0NMD to disable wrapping.
   f. Write the address for the first byte of master output (MOSI) data to DMA0NBAH:L.
   g. Write the size of the SPI transfer in bytes to DMA0NSZH:L.
   h. Clear the address offset SFRs CMA0A0H:L.
   i. Enable the interrupt on the desired channel by setting the corresponding bit in DMA0INT.
   j. Enable DMA interrupts by setting bit 5 of EIE2.
3. Clear the interrupt bit in DMA0INT for the desired channel.
4. Enable the desired channel by setting the corresponding bit in the DMA0EN SFR to initiate the SPI transfer operation.
5. Wait on the DMA interrupt.
6. Clear the DMA enables in the DMA0EN SFR.
7. Clear the DMA interrupts in the DMA0INT SFR.
8. If desired, wait on the SPIBSY bit in SPI1CFG for the last byte transfer to complete.

31.10. Slave Mode DMA Transfers

SPI1 also supports using the DMA with Slave mode. The maximum SPI bit rate for a bidirectional Slave mode transfer is SYSCLK/10.

In master mode, the master is responsible for initiating the transfer, clocking the data, managing the NSS pin, and has control over the number of bytes transferred. In slave mode, the slave depends on the master for the clock and NSS signal. The slave also depends on the master to set the time between bytes and the number of bytes per transfer.

Firmware implementations of a SPI slave often have some restrictions on the time between bytes. When using SPI0 in slave mode, an interrupt service routine commonly processes each byte received. This imposes a limitation on the time between bytes. When using the SPI in Slave mode with the DMA, the time between bytes must be long enough to accommodate the DMA latency.

The time between bytes in master mode and the minimum time required between bytes in slave mode will depend on the DMA latency. The DMA latency will depend on a number of factors - the CPU state, the number of active DMA channels, and the DMA channel priority. Using only the two required DMA channels and putting the CPU in Idle mode will provide the lowest latency. If the CPU is actively executing instructions, the DMA may have to wait for the current instruction to execute before it can complete a transfer. If other DMA channels are active, the SPI DMA channels may have to wait for other DMA transfers to complete. This could be a very long time for long DMA transfers. Assigning the SPI to the first two DMA channels will ensure they have the highest DMA priority.

Note that in master mode, the time between bytes may prolong the DMA transfer, but does not usually result in data loss. In slave mode, the slave may drop data if the DMA cannot keep up with the master data coming in. Since the SPI slave data rate is limited to SYSCLK/10 and the longest instruction is 8 clock cycles, a delay between bytes of one SPI clock will prevent data loss. Using a SPI DMA slave with additional active DMA channels may result in data loss and is not recommended.

31.11. Bidirectional SPI Slave Mode DMA Transfer

A bidirectional SPI Slave mode DMA transfer will transfer a specified number of bytes out on the MISO pin and also receive the same number of bytes on the MOSI pin. The MISO data must be stored in XRAM before initiating the DMA transfers. After the complete transfer, the MOSI data will be stored in XRAM.

Since the MISO data must be stored in XRAM before the transfer, the MISO data is fixed and should not depend on the MOSI data received in the same transfer. The protocol designer should carefully consider this behavior when designing a SPI slave protocol. Firmware can easily modify the MISO data after each message. For example, one message can request data and a second message can read the data previously requested. This approach is much simpler and more efficient than attempting to modify the MISO data buffer on-the-fly.

If the slave transfer is a fixed constant length, the DMA interrupt will indicate one complete transfer. Firmware may implement a variable length slave transfer using an external interrupt connected to the NSS signal. In this case, firmware may use the DMA interrupt for a buffer overflow condition.

To to initiate a fixed-length SPI Slave mode bidirectional data transfer:

1. Configure the SPI1 SFRs normally for Slave mode.
   a. Enable Slave mode by clearing bit 6 in SPI1CFG.
   b. Configure the clock polarity CKPOL and clock phase CKPHA as desired in SPI1CFG.
   c. Configure SPI1CKR for the desired SPI clock rate.
   d. Configure SPI1CN for 4-wire slave mode.
   e. Enable the SPI by setting bit 0 of SPI1CN.
2. Configure the first DMA channel for the XRAM-to-SPI1DATA transfer:
   a. Disable the first DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the first DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the XRAM-to-SPI1DAT peripheral request by writing 0x03 to DMA0NCF.
   d. Write 0 to DMA0NMD to disable wrapping.
   e. Write the address of the first byte of the slave output (MISO) data to DMA0NBAH:L.
   f. Write the size of the SPI transfer in bytes to DMA0NSZH:L.
   g. Clear the address offset SFRs DMA0A0H:L.
3. Configure the second DMA channel for the SPI1DAT-to-XRAM transfer:
   a. Disable the second DMA channel by clearing the corresponding bit in DMA0EN.
   b. Select the second DMA channel by writing to DMA0SEL.
   c. Configure the selected DMA channel to use the SPI1DAT-to-XRAM peripheral request by writing 0x04 to DMA0NCF.
   d. Enable DMA interrupts for the second channel by setting bit 7 of DMA0NCF.
   e. Write 0 to DMA0NMD to disable wrapping.
   f. Write the address for the first byte of the slave input (MOSI) data to DMA0NBAH:L.
   g. Write the size of the SPI transfer in bytes to DMA0NSZH:L.
   h. Clear the address offset SFRs DMA0A0H:L.
   i. Enable the interrupt on the second channel by setting the corresponding bit in DMA0INT.
   j. Enable DMA interrupts by setting bit 5 of EIE2.
4. Clear the interrupt bits in DMA0INT for both channels.
5. Enable both channels by setting the corresponding bits in the DMA0EN SFR to initiate the SPI transfer operation.
6. Wait on the DMA interrupt.
7. Clear the DMA enables in the DMA0EN SFR.
8. Clear the DMA interrupts in the DMA0INT SFR.

31.12. SPI Special Function Registers

SPI1 is accessed and controlled through four special function registers in the system controller: SPI1CN Control Register, SPI1DAT Data Register, SPI1CFG Configuration Register, and SPI1CKR Clock Rate Register. The four special function registers related to the operation of the SPI1 Bus are described in the following SFR definitions.

SFR Definition 31.1. SPI1CFG: SPI1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Page = 0x2; SFR Address = 0xA1

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI1 Clock Phase.** 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	**SPI1 Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** This bit is set to logic 1 whenever the NSS pin is low indicating SPI1 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	**NSS Instantaneous Pin Input.** This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	**Shift Register Empty (valid in slave mode only).** This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
0	RXBMT	**Receive Buffer Empty (valid in slave mode only).** This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.

Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 31.1 for timing parameters.

SFR Definition 31.2. SPI1CN: SPI1 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0]		TXBMT	SPIEN
Type	R/W	R/W	R/W	R/W	R/W		R	R/W
Reset	0	0	0	0	0	1	1	0

SFR Page = 0x2; SFR Address = 0xF8; Bit-Addressable

Bit	Name	Function
7	SPIF	**SPI1 Interrupt Flag.** This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	**Write Collision Flag.** This bit is set to logic 1 if a write to SPI1DAT is attempted when TXBMT is 0. When this occurs, the write to SPI1DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	**Mode Fault Flag.** This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	**Receive Overrun Flag (valid in slave mode only).** This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI1 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD[1:0]	**Slave Select Mode.** Selects between the following NSS operation modes: (See Section 31.2 and Section 31.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMT	**Transmit Buffer Empty.** This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPIEN	**SPI1 Enable.** 0: SPI disabled. 1: SPI enabled.

SFR Definition 31.3. SPI1CKR: SPI1 Clock Rate

Bit	7	6	5	4	3	2	1	0
Name	SCR[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xA2

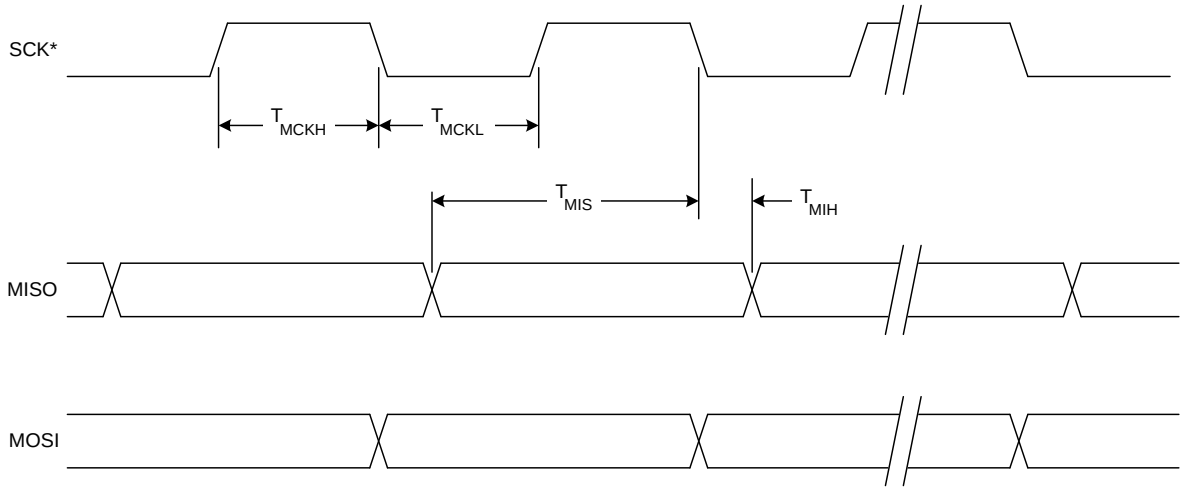
Bit	Name	Function
7:0	SCR[7:0]	**SPI1 Clock Rate.** These bits determine the frequency of the SCK output when the SPI1 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI1CKR* is the 8-bit value held in the SPI1CKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPI1CKR[7:0] + 1)}$ for 0 <= SPI1CKR <= 255 Example: If SYSCLK = 2 MHz and SPI1CKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 31.4. SPI1DAT: SPI1 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI1DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

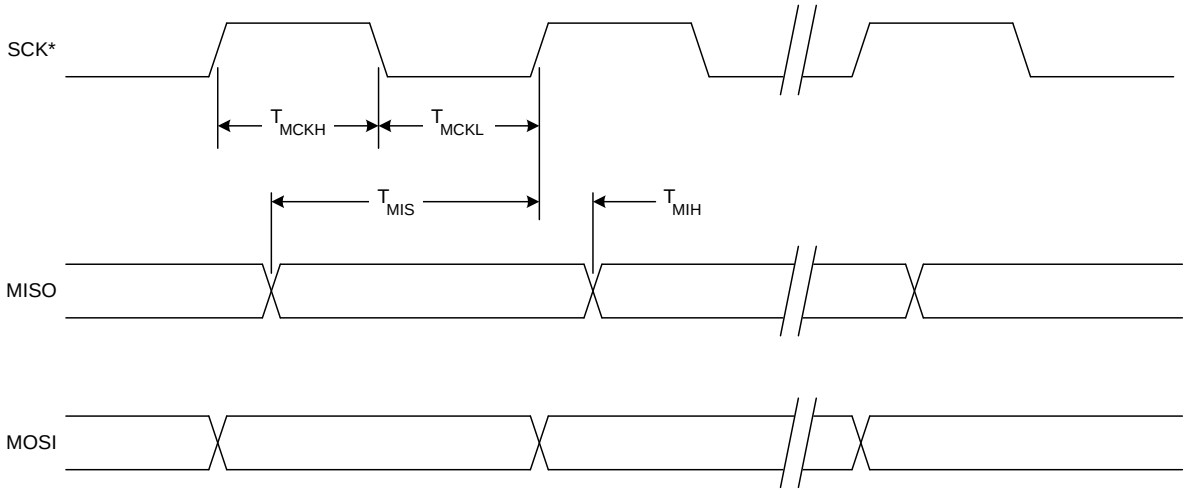
SFR Page = 0x2; SFR Address = 0xA3

Bit	Name	Function
7:0	SPI1DAT[7:0]	**SPI1 Transmit and Receive Data.** The SPI1DAT register is used to transmit and receive SPI1 data. Writing data to SPI1DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI1DAT returns the contents of the receive buffer.



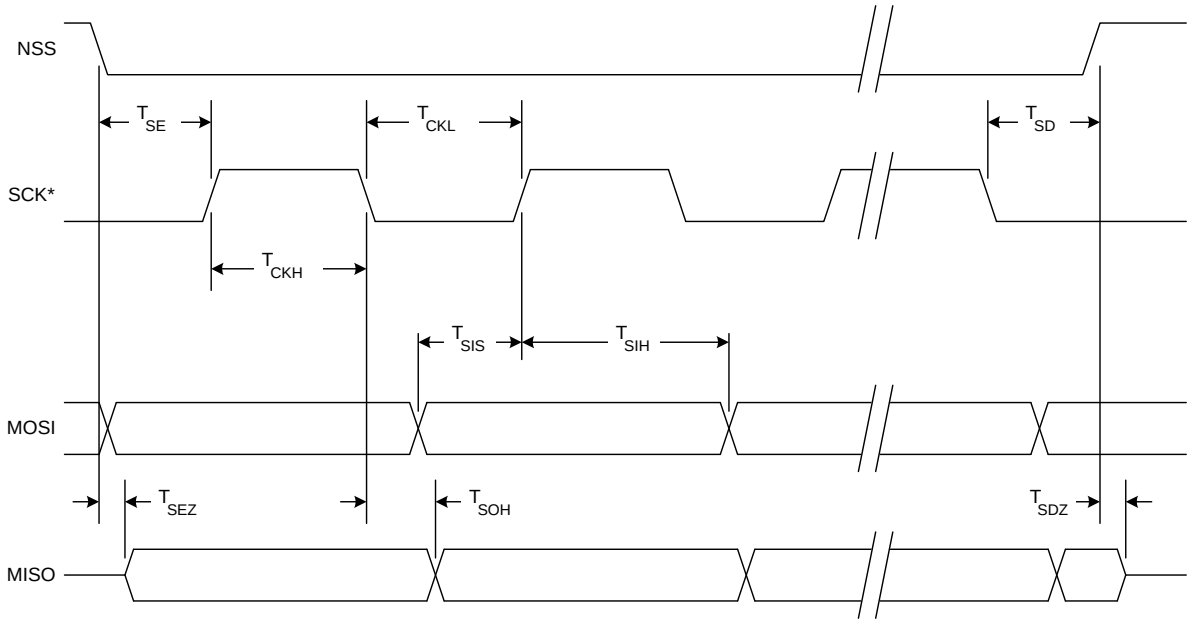
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.8. SPI Master Timing (CKPHA = 0)



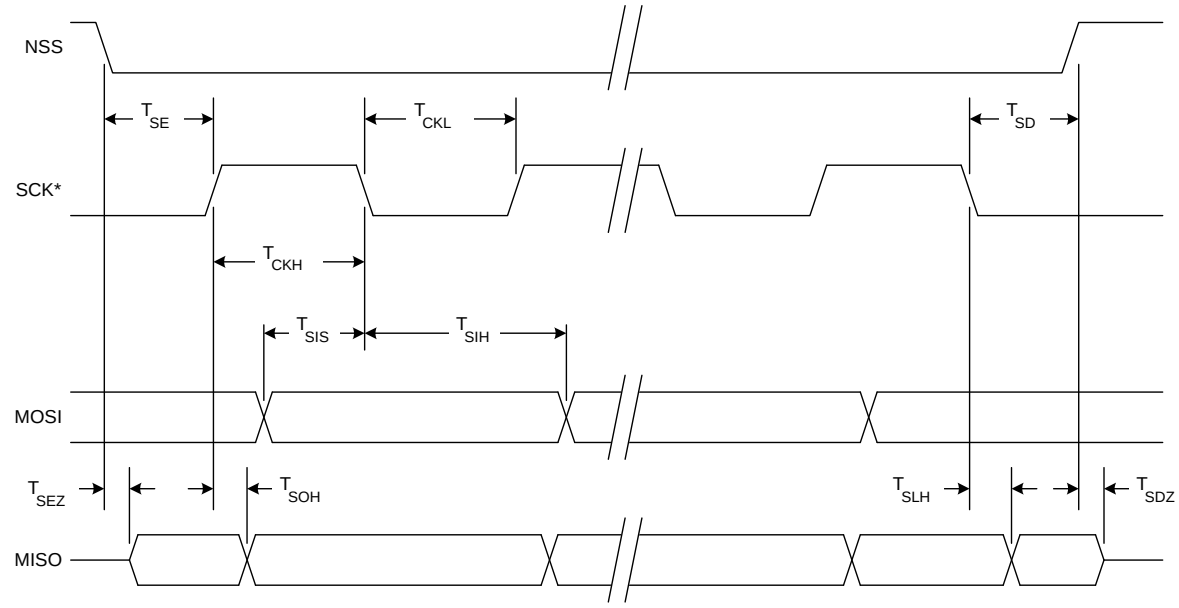
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.9. SPI Master Timing (CKPHA = 1)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.11. SPI Slave Timing (CKPHA = 1)

Table 31.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing (See Figure 31.8 and Figure 31.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing (See Figure 31.10 and Figure 31.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns
Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

32. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload. Additionally, Timer 2 and Timer 3 have a Capture Mode that can be used to measure the SmaRTClock, Comparator, or external clock period with respect to another oscillator. The ability to measure the Comparator period with respect to another oscillator is particularly useful when interfacing to capacitive sensors.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 32.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12. Timer 2 may additionally be clocked by the SmaRTClock divided by 8 or the Comparator0 output. Timer 3 may additionally be clocked by the external oscillator clock source divided by 8 or the Comparator1 output.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

SFR Definition 32.1. CKCON: Clock Control

Bit	7	6	5	4	3	2	1	0
Name	T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8E

Bit	Name	Function
7	T3MH	**Timer 3 High Byte Clock Select.** Selects the clock supplied to the Timer 3 high byte (split 8-bit timer mode only). 0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 high byte uses the system clock.
6	T3ML	**Timer 3 Low Byte Clock Select.** Selects the clock supplied to Timer 3. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 low byte uses the system clock.
5	T2MH	**Timer 2 High Byte Clock Select.** Selects the clock supplied to the Timer 2 high byte (split 8-bit timer mode only). 0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 high byte uses the system clock.
4	T2ML	**Timer 2 Low Byte Clock Select.** Selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 low byte uses the system clock.
3	T1M	**Timer 1 Clock Select.** Selects the clock source supplied to Timer 1. Ignored when C/T1 is set to 1. 0: Timer 1 uses the clock defined by the prescale bits SCA[1:0]. 1: Timer 1 uses the system clock.
2	T0M	**Timer 0 Clock Select.** Selects the clock source supplied to Timer 0. Ignored when C/T0 is set to 1. 0: Counter/Timer 0 uses the clock defined by the prescale bits SCA[1:0]. 1: Counter/Timer 0 uses the system clock.
1:0	SCA[1:0]	**Timer 0/1 Prescale Bits.** These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12 01: System clock divided by 4 10: System clock divided by 48 11: External clock divided by 8 (synchronized with the system clock)

32.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section “17.5. Interrupt Register Descriptions” on page 235); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section “17.5. Interrupt Register Descriptions” on page 235). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

32.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section “27.3. Priority Crossbar Decoder” on page 355 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 32.1).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 17.7). Setting GATE0 to 1 allows the timer to be controlled by the external input signal $\overline{\text{INT0}}$ (see Section “17.5. Interrupt Register Descriptions” on page 235), facilitating pulse width measurements

Table 32.1. Timer 0 Running Modes

TR0	GATE0	$\overline{\text{INT0}}$	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal $\overline{\text{INT1}}$ is used with Timer 1; the $\overline{\text{INT1}}$ polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 17.7).

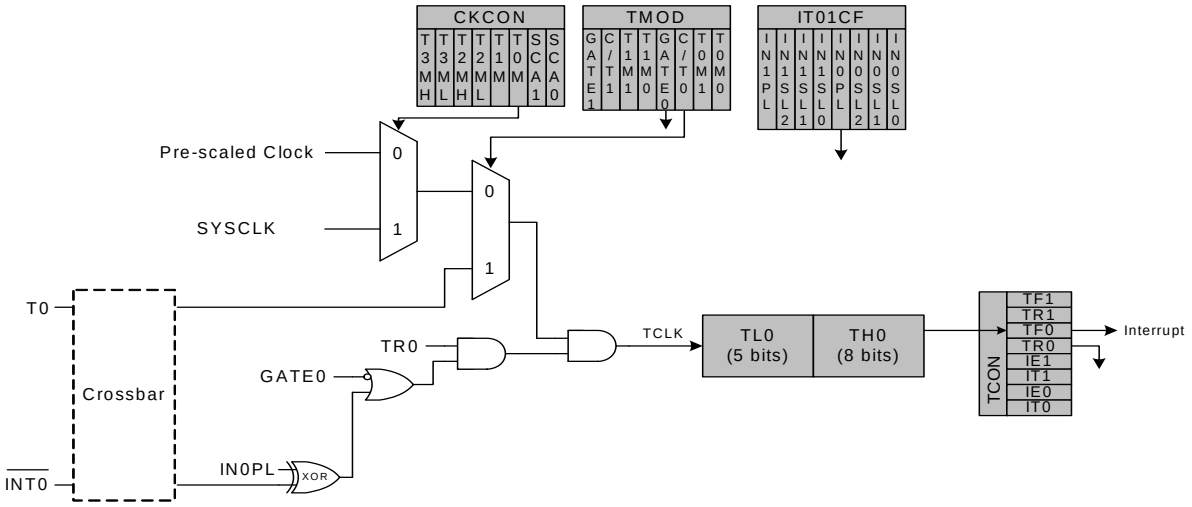


Figure 32.1. T0 Mode 0 Block Diagram

32.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

32.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register IT01CF (see Section “17.6. External Interrupts INT0 and INT1” on page 242 for details on the external input signals INT0 and INT1).

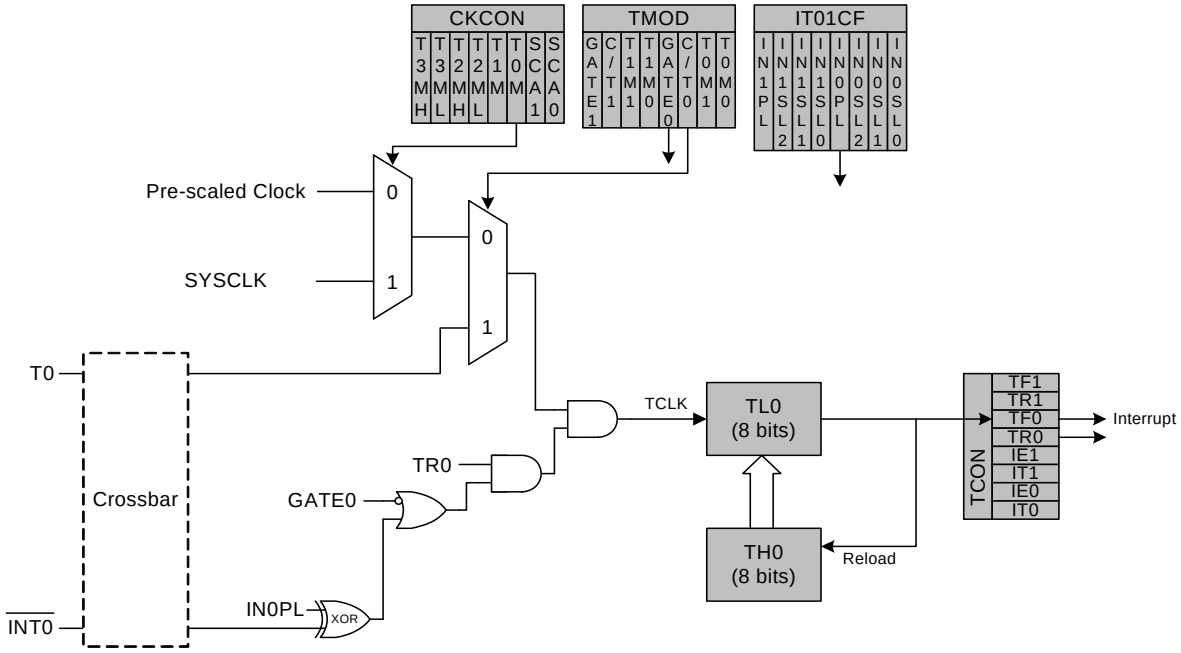


Figure 32.2. T0 Mode 2 Block Diagram

32.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

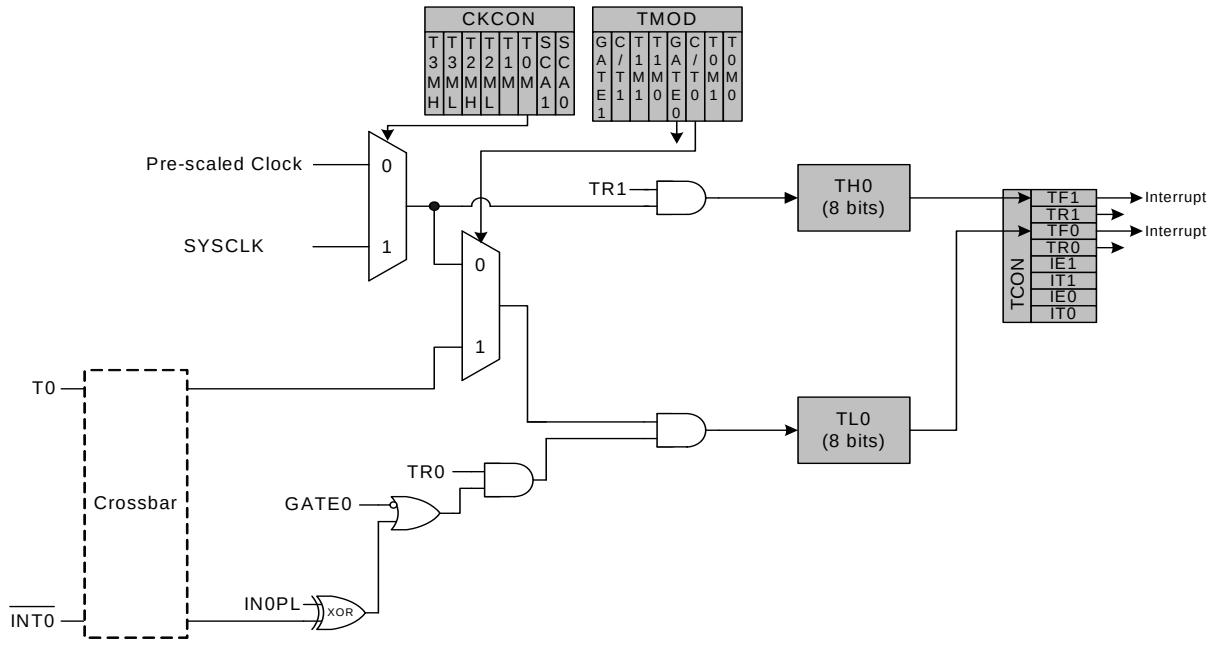


Figure 32.3. T0 Mode 3 Block Diagram

SFR Definition 32.2. TCON: Timer Control

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0x88; Bit-Addressable

Bit	Name	Function
7	TF1	**Timer 1 Overflow Flag.** Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
6	TR1	**Timer 1 Run Control.** Timer 1 is enabled by setting this bit to 1.
5	TF0	**Timer 0 Overflow Flag.** Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
4	TR0	**Timer 0 Run Control.** Timer 0 is enabled by setting this bit to 1.
3	IE1	**External Interrupt 1.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine in edge-triggered mode.
2	IT1	**Interrupt 1 Type Select.** This bit selects whether the configured $\overline{\text{INT1}}$ interrupt will be edge or level sensitive. $\overline{\text{INT1}}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 17.7). 0: $\overline{\text{INT1}}$ is level triggered. 1: $\overline{\text{INT1}}$ is edge triggered.
1	IE0	**External Interrupt 0.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine in edge-triggered mode.
0	IT0	**Interrupt 0 Type Select.** This bit selects whether the configured $\overline{\text{INT0}}$ interrupt will be edge or level sensitive. $\overline{\text{INT0}}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 17.7). 0: $\overline{\text{INT0}}$ is level triggered. 1: $\overline{\text{INT0}}$ is edge triggered.

SFR Definition 32.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x89

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 17.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 17.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

SFR Definition 32.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8A

Bit	Name	Function
7:0	TL0[7:0]	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 32.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8B

Bit	Name	Function
7:0	TL1[7:0]	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

SFR Definition 32.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8C

Bit	Name	Function
7:0	TH0[7:0]	**Timer 0 High Byte.** The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 32.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8D

Bit	Name	Function
7:0	TH1[7:0]	**Timer 1 High Byte.** The TH1 register is the high byte of the 16-bit Timer 1.

32.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode. Timer 2 can also be used in Capture Mode to measure the SmaRTClock or the Comparator 0 period with respect to another oscillator. The ability to measure the Comparator 0 period with respect to the system clock is makes using Touch Sense Switches very easy.

Timer 2 may be clocked by the system clock, the system clock divided by 12, SmaRTClock divided by 8, or Comparator 0 output. Note that the SmaRTClock divided by 8 and Comparator 0 output is synchronized with the system clock.

32.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, SmaRTClock divided by 8, or Comparator 0 output. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 32.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

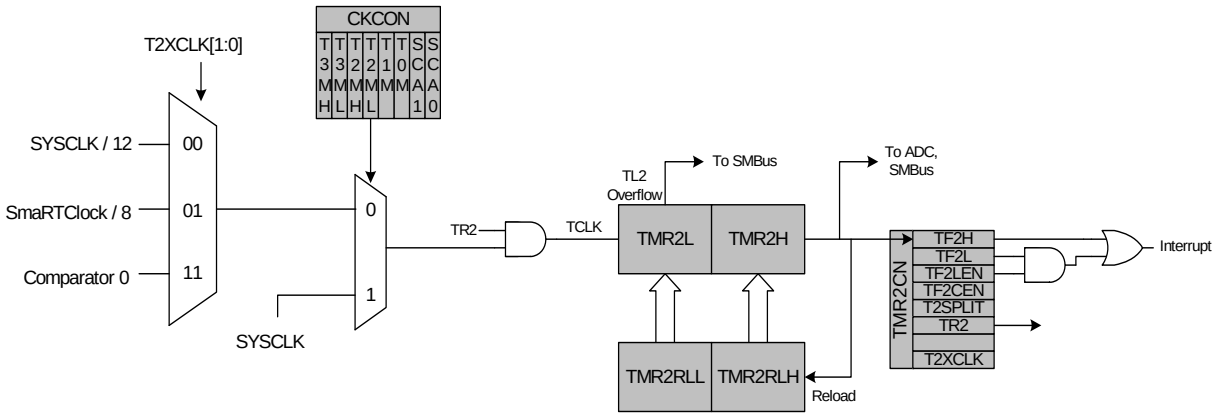


Figure 32.4. Timer 2 16-Bit Mode Block Diagram

32.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 32.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, SmaRTClock divided by 8 or Comparator 0 output. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bits (T2XCLK[1:0] in TMR2CN), as follows:

T2MH	T2XCLK[1:0]	TMR2H Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock / 8
0	10	Reserved
0	11	Comparator 0
1	X	SYSCLK

T2ML	T2XCLK[1:0]	TMR2L Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock / 8
0	10	Reserved
0	11	Comparator 0
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

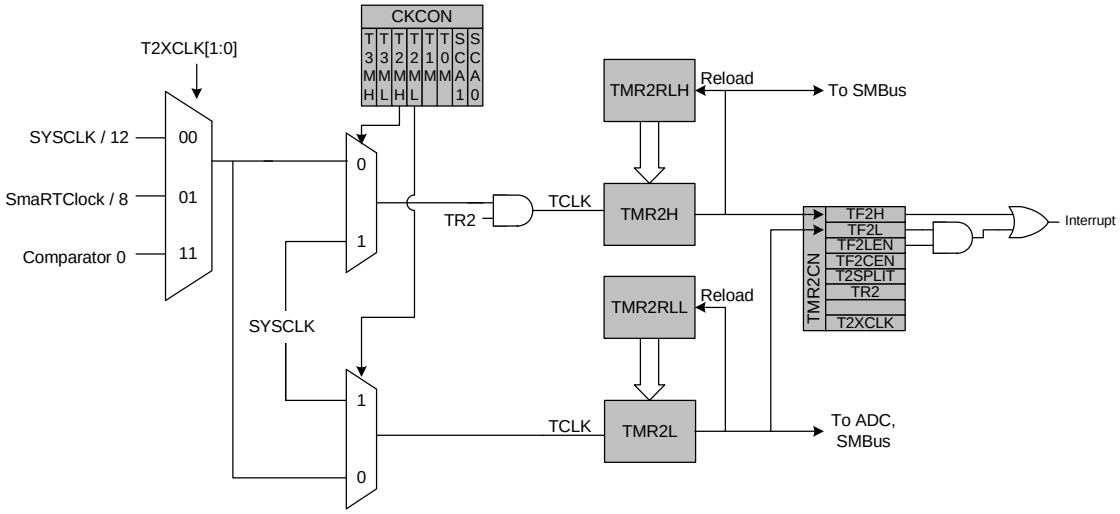


Figure 32.5. Timer 2 8-Bit Mode Block Diagram

32.2.3. Comparator 0/SmaRTClock Capture Mode

The Capture Mode in Timer 2 allows either Comparator 0 or the SmaRTClock period to be measured against the system clock or the system clock divided by 12. Comparator 0 and the SmaRTClock period can also be compared against each other. Timer 2 Capture Mode is enabled by setting TF2CEN to 1. Timer 2 should be in 16-bit auto-reload mode when using Capture Mode.

When Capture Mode is enabled, a capture event will be generated either every Comparator 0 rising edge or every 8 SmaRTClock clock cycles, depending on the T2XCLK1 setting. When the capture event occurs, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set (triggering an interrupt if Timer 2 interrupts are enabled). By recording the difference between two successive timer capture values, the Comparator 0 or SmaRTClock period can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading.

For example, if T2ML = 1b, T2XCLK1 = 0b, and TF2CEN = 1b, Timer 2 will clock every SYSCLK and capture every SmaRTClock clock divided by 8. If the SYSCLK is 24.5 MHz and the difference between two successive captures is 5984, then the SmaRTClock clock is as follows:

24.5 MHz/(5984/8) = 0.032754 MHz or 32.754 kHz.

This mode allows software to determine the exact SmaRTClock frequency in self-oscillate mode and the time between consecutive Comparator 0 rising edges, which is useful for detecting changes in the capacitance of a Touch Sense Switch.

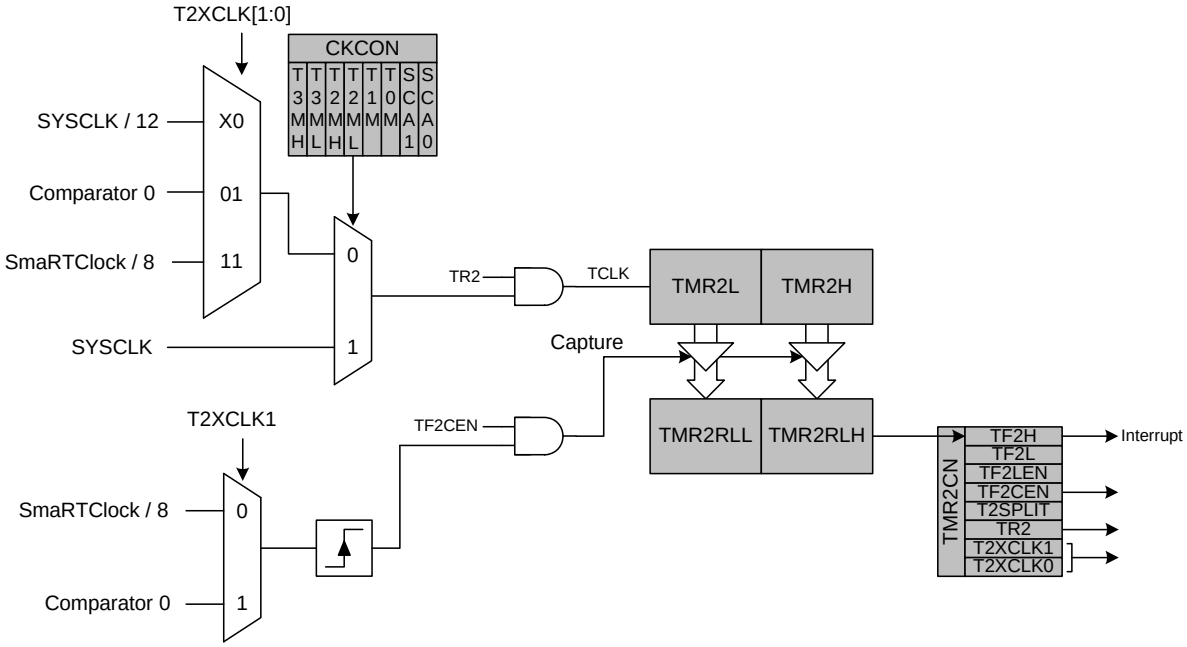


Figure 32.6. Timer 2 Capture Mode Block Diagram

SFR Definition 32.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	T2XCLK[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xC8; Bit-Addressable

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Capture Enable.** When set to 1, this bit enables Timer 2 Capture Mode.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When set to 1, Timer 2 operates as two 8-bit timers with auto-reload. Otherwise, Timer 2 operates in 16-bit auto-reload mode.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1:0	T2XCLK[1:0]	**Timer 2 External Clock Select.** This bit selects the “external” and “capture trigger” clock sources for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the “external” clock source for both timer bytes. Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the “external” clock and the system clock for either timer. Note: External clock sources are synchronized with the system clock. 00: External Clock is SYSCLK/12. Capture trigger is SmaRTClock/8. 01: External Clock is Comparator 0. Capture trigger is SmaRTClock/8. 10: External Clock is SYSCLK/12. Capture trigger is Comparator 0. 11: External Clock is SmaRTClock/8. Capture trigger is Comparator 0.

SFR Definition 32.9. TMR2RLL: Timer 2 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCA

Bit	Name	Function
7:0	TMR2RLL[7:0]	**Timer 2 Reload Register Low Byte.** TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 32.10. TMR2RLH: Timer 2 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCB

Bit	Name	Function
7:0	TMR2RLH[7:0]	**Timer 2 Reload Register High Byte.** TMR2RLH holds the high byte of the reload value for Timer 2.

SFR Definition 32.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCC

Bit	Name	Function
7:0	TMR2L[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

SFR Definition 32.12. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xCD

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

32.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR2CN.3) defines the Timer 3 operation mode. Timer 3 can also be used in Capture Mode to measure the external oscillator source or the SmaRTClock oscillator period with respect to another oscillator.

Timer 3 may be clocked by the system clock, the system clock divided by 12, external oscillator source divided by 8, or the SmaRTClock oscillator. The external oscillator source divided by 8 and SmaRTClock oscillator is synchronized with the system clock.

32.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, external oscillator clock source divided by 8, or SmaRTClock oscillator. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 32.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

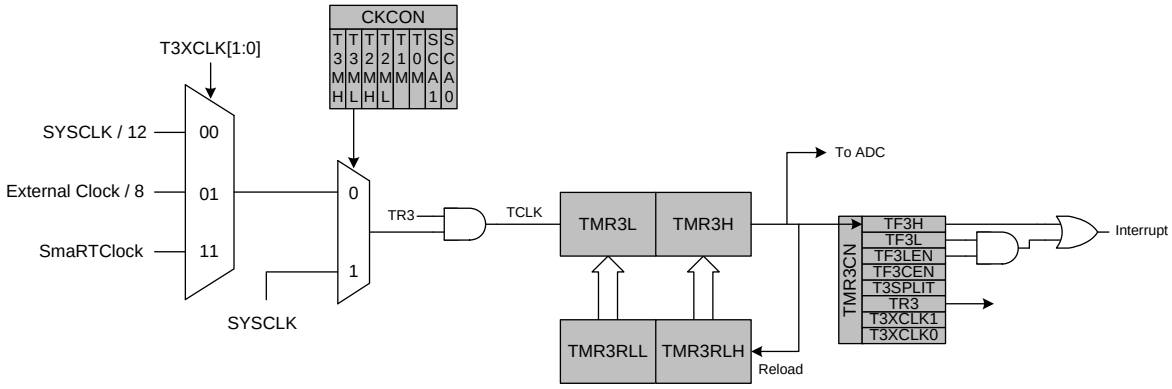


Figure 32.7. Timer 3 16-Bit Mode Block Diagram

32.3.2. 8-Bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 32.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, the external oscillator clock source divided by 8, or the SmaRTClock. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bits (T3XCLK[1:0] in TMR3CN), as follows:

T3MH	T3XCLK[1:0]	TMR3H Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK[1:0]	TMR3L Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

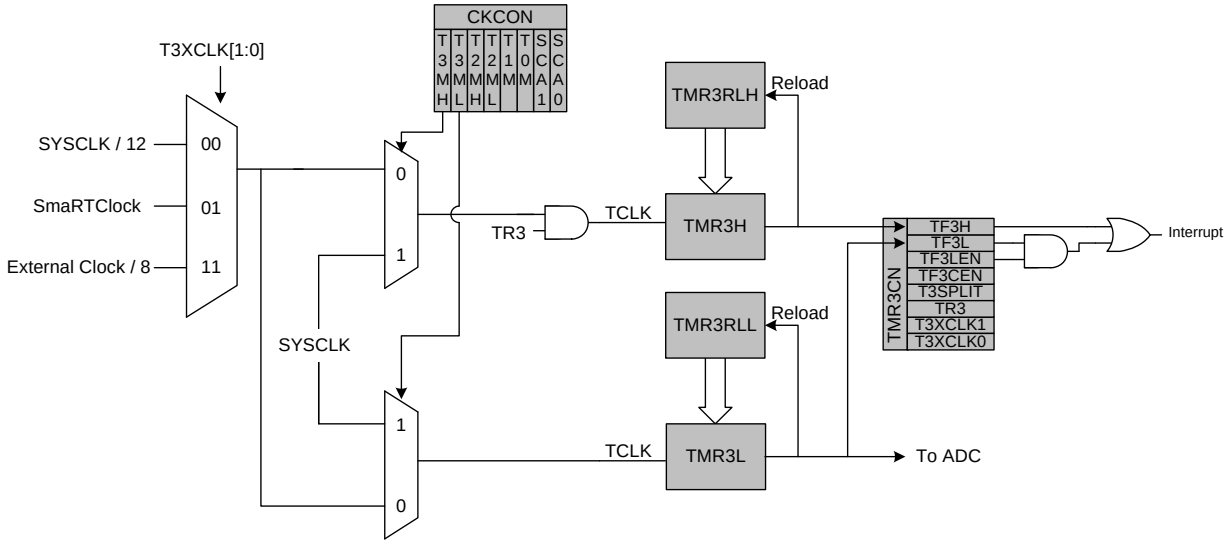


Figure 32.8. Timer 3 8-Bit Mode Block Diagram

32.3.3. SmaRTClock/External Oscillator Capture Mode

The Capture Mode in Timer 3 allows either SmaRTClock or the external oscillator period to be measured against the system clock or the system clock divided by 12. SmaRTClock and the external oscillator period can also be compared against each other.

Setting TF3CEN to 1 enables the SmaRTClock/External Oscillator Capture Mode for Timer 3. In this mode, T3SPLIT should be set to 0, as the full 16-bit timer is used.

When Capture Mode is enabled, a capture event will be generated either every SmaRTClock rising edge or every 8 external clock cycles, depending on the T3XCLK1 setting. When the capture event occurs, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set (triggering an interrupt if Timer 3 interrupts are enabled). By recording the difference between two successive timer capture values, the SmaRTClock or external clock period can be determined with respect to the Timer 3 clock. The Timer 3 clock should be much faster than the capture clock to achieve an accurate reading.

For example, if T3ML = 1b, T3XCLK1 = 0b, and TF3CEN = 1b, Timer 3 will clock every SYSCLK and capture every SmaRTClock rising edge. If SYSCLK is 24.5 MHz and the difference between two successive captures is 350 counts, then the SmaRTClock period is as follows:

350 x (1 / 24.5 MHz) = 14.2 µs.

This mode allows software to determine the exact frequency of the external oscillator in C and RC mode or the time between consecutive SmaRTClock rising edges, which is useful for determining the SmaRTClock frequency.

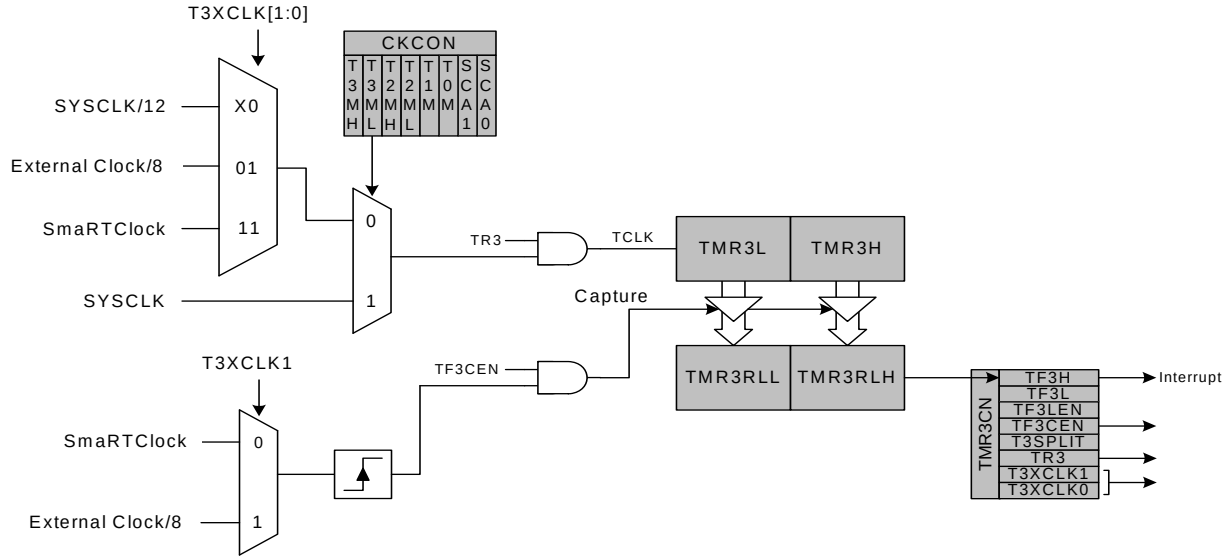


Figure 32.9. Timer 3 Capture Mode Block Diagram

SFR Definition 32.13. TMR3CN: Timer 3 Control

Bit	7	6	5	4	3	2	1	0
Name	TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3	T3XCLK[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x91

Bit	Name	Function
7	TF3H	**Timer 3 High Byte Overflow Flag.** Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF3L	**Timer 3 Low Byte Overflow Flag.** Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. TF3L will be set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.
5	TF3LEN	**Timer 3 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 3 Low Byte interrupts. If Timer 3 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
4	TF3CEN	**Timer 3 SmaRTClock/External Oscillator Capture Enable.** When set to 1, this bit enables Timer 3 Capture Mode.
3	T3SPLIT	**Timer 3 Split Mode Enable.** When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload. 0: Timer 3 operates in 16-bit auto-reload mode. 1: Timer 3 operates as two 8-bit auto-reload timers.
2	TR3	**Timer 3 Run Control.** Timer 3 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in split mode.
1:0	T3XCLK[1:0]	**Timer 3 External Clock Select.** This bit selects the “external” and “capture trigger” clock sources for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the “external” clock source for both timer bytes. Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the “external” clock and the system clock for either timer. Note: External clock sources are synchronized with the system clock. 00: External Clock is SYSCLK /12. Capture trigger is SmaRTClock. 01: External Clock is External Oscillator/8. Capture trigger is SmaRTClock. 10: External Clock is SYSCLK/12. Capture trigger is External Oscillator/8. 11: External Clock is SmaRTClock. Capture trigger is External Oscillator/8.

SFR Definition 32.14. TMR3RLL: Timer 3 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x92

Bit	Name	Function
7:0	TMR3RLL[7:0]	**Timer 3 Reload Register Low Byte.** TMR3RLL holds the low byte of the reload value for Timer 3.

SFR Definition 32.15. TMR3RLH: Timer 3 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x93

Bit	Name	Function
7:0	TMR3RLH[7:0]	**Timer 3 Reload Register High Byte.** TMR3RLH holds the high byte of the reload value for Timer 3.

SFR Definition 32.16. TMR3L: Timer 3 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x94

Bit	Name	Function
7:0	TMR3L[7:0]	**Timer 3 Low Byte.** In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value.

SFR Definition 32.17. TMR3H Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x95

Bit	Name	Function
7:0	TMR3H[7:0]	**Timer 3 High Byte.** In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

33. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and six 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled. The counter/timer is driven by a programmable timebase that can select between seven sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, SmaRTClock divided by 8, Timer 0 overflows, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8 to 11-Bit PWM, or 16-Bit PWM (each mode is described in Section “33.3. Capture/Compare Modules” on page 469). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 33.1

Important Note: The PCA Module 5 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. **Access to certain PCA registers is restricted while WDT mode is enabled**. See Section 33.4 for details.

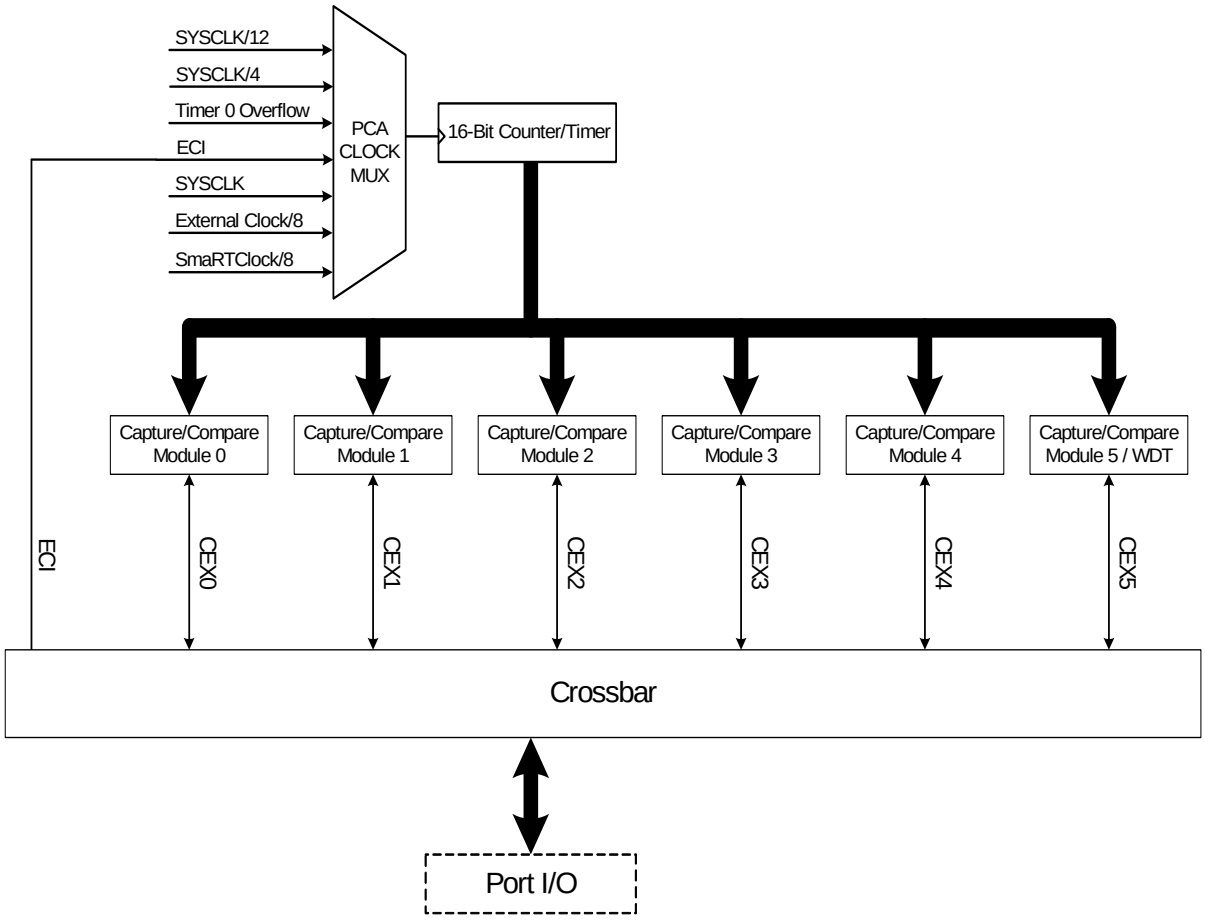


Figure 33.1. PCA Block Diagram

33.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. **Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter.** Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2–CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 33.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 33.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8[1]
1	1	0	SmaRTClock oscillator source divided by 8[2]
1	1	1	Reserved

Notes:
1. External oscillator source divided by 8 is synchronized with the system clock.
2. SmaRTClock oscillator source divided by 8 is synchronized with the system clock.

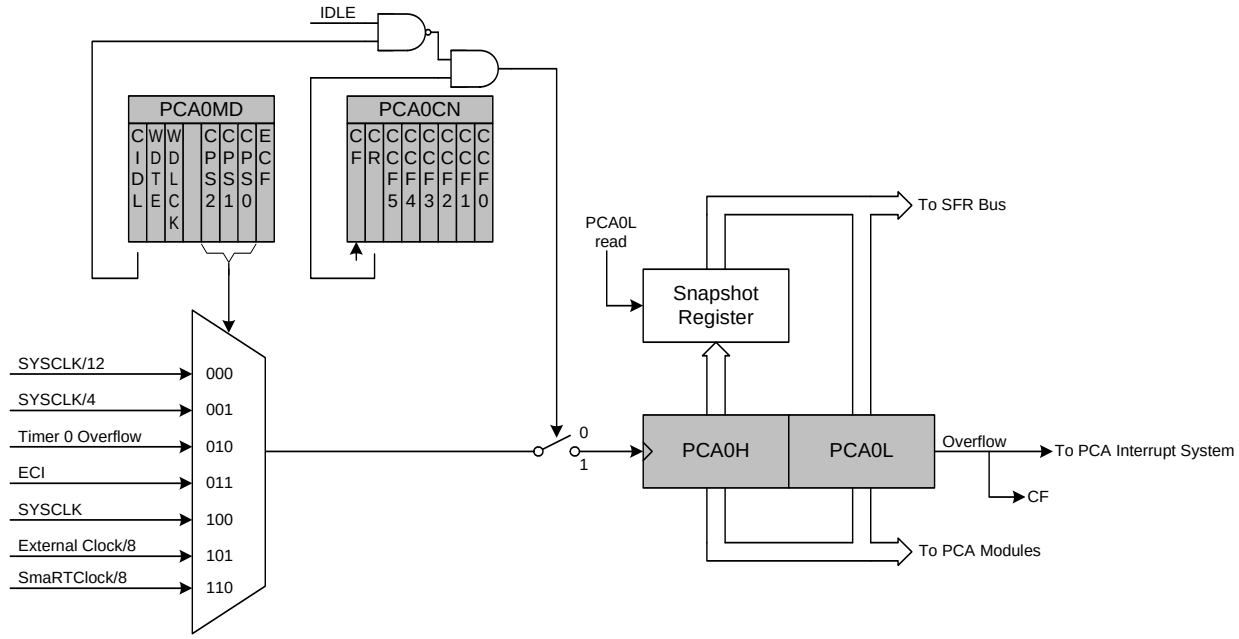


Figure 33.2. PCA Counter/Timer Block Diagram

33.2. PCA0 Interrupt Sources

Figure 33.3 shows a diagram of the PCA interrupt tree. There are eight independent event flags that can be used to generate a PCA0 interrupt. They are: the main PCA counter overflow flag (CF), which is set upon a 16-bit overflow of the PCA0 counter, an intermediate overflow flag (COVF), which can be set on an overflow from the 8th, 9th, 10th, or 11th bit of the PCA0 counter, and the individual flags for each PCA channel (CCF0, CCF1, CCF2, CCF3, CCF4, and CCF5), which are set according to the operation mode of that module. These event flags are always set when the trigger condition occurs. Each of these flags can be individually selected to generate a PCA0 interrupt, using the corresponding interrupt enable flag (ECF for CF, ECOV for COVF, and ECCFn for each CCFn). PCA0 interrupts must be globally enabled before any individual interrupt sources are recognized by the processor. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1.

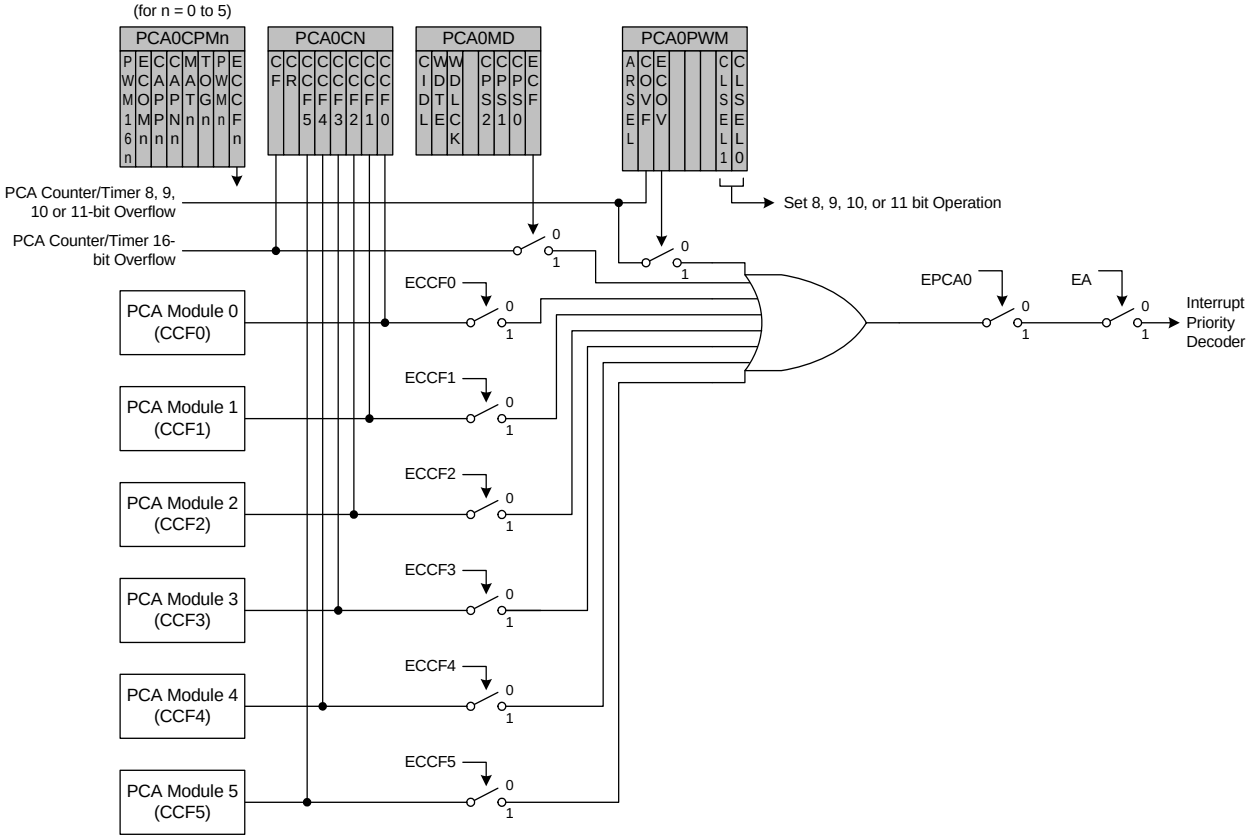


Figure 33.3. PCA Interrupt Block Diagram

33.3. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: edge-triggered capture, software timer, high speed output, frequency output, 8 to 11-bit pulse width modulator, or 16-bit pulse width modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation. Table 33.2 summarizes the bit settings in the PCA0CPMn and PCA0PWM registers used to select the PCA capture/compare module's operating mode. Note that all modules set to use 8, 9, 10, or 11-bit PWM mode must use the same cycle length (8-11 bits). Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt.

Table 33.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules

Operational Mode	PCA0CPMn								PCA0PWM				
Bit Number	7	6	5	4	3	2	1	0	7	6	5	4–2	1–0
Capture triggered by positive edge on CEXn	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
Capture triggered by negative edge on CEXn	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
Capture triggered by any transition on CEXn	X	X	1	1	0	0	0	A	0	X	B	XXX	XX

Table 33.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules

Operational Mode	PCA0CPMn								PCA0PWM				
Software Timer	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
High Speed Output	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
Frequency Output	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16-Bit Pulse Width Modulator	1	C	0	0	E	0	1	A	0	X	B	XXX	XX

Notes:
1. X = Don't Care (no functional difference for individual module if 1 or 0).
2. A = Enable interrupts for this module (PCA interrupt triggered on CCFn set to 1).
3. B = Enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL[1:0]).
4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0).
5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA0CPHn and PCA0CPLn.
6. E = When set, a match event will cause the CCFn flag for the associated channel to be set.
7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting.

33.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

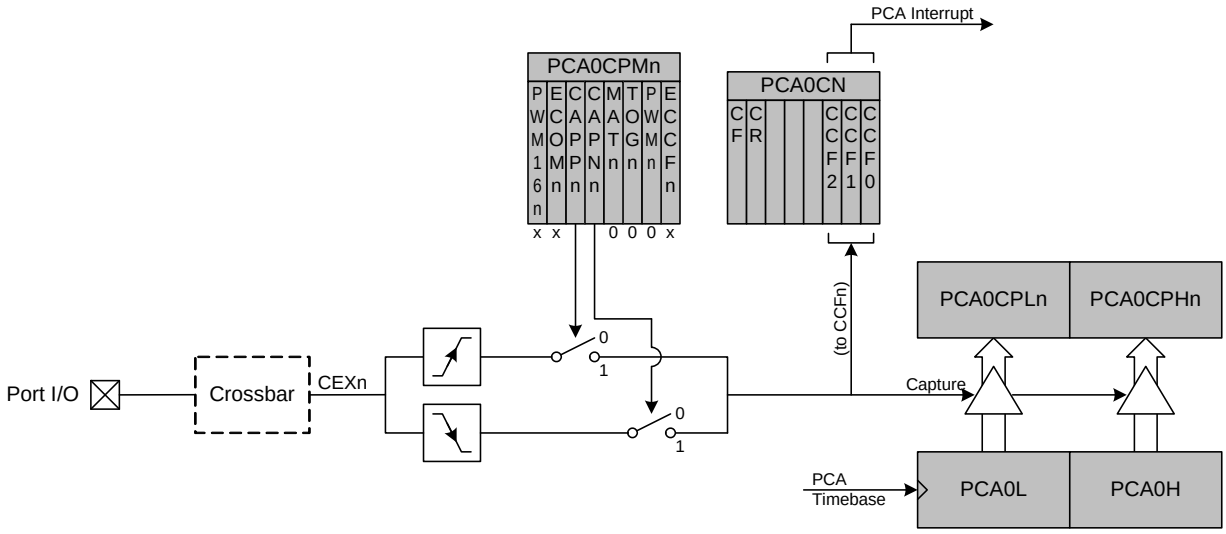


Figure 33.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

33.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

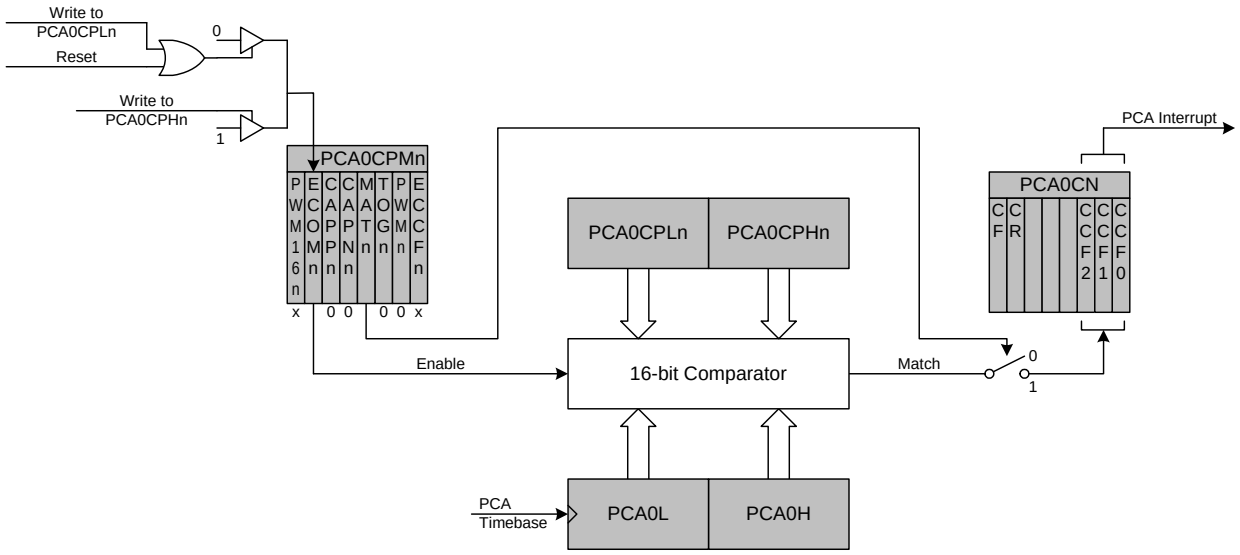


Figure 33.5. PCA Software Timer Mode Diagram

33.3.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode. If ECOMn is cleared, the associated pin will retain its state, and not toggle on the next match event.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

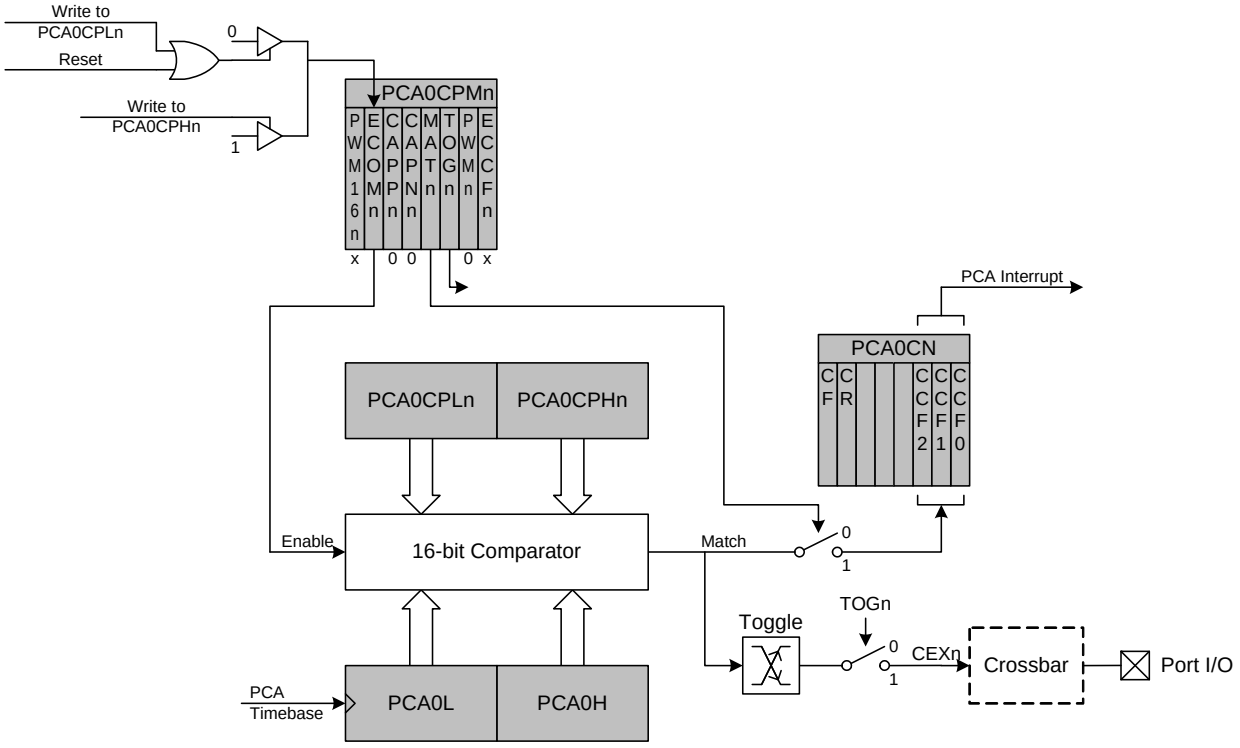


Figure 33.6. PCA High-Speed Output Mode Diagram

33.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 33.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 33.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. The MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

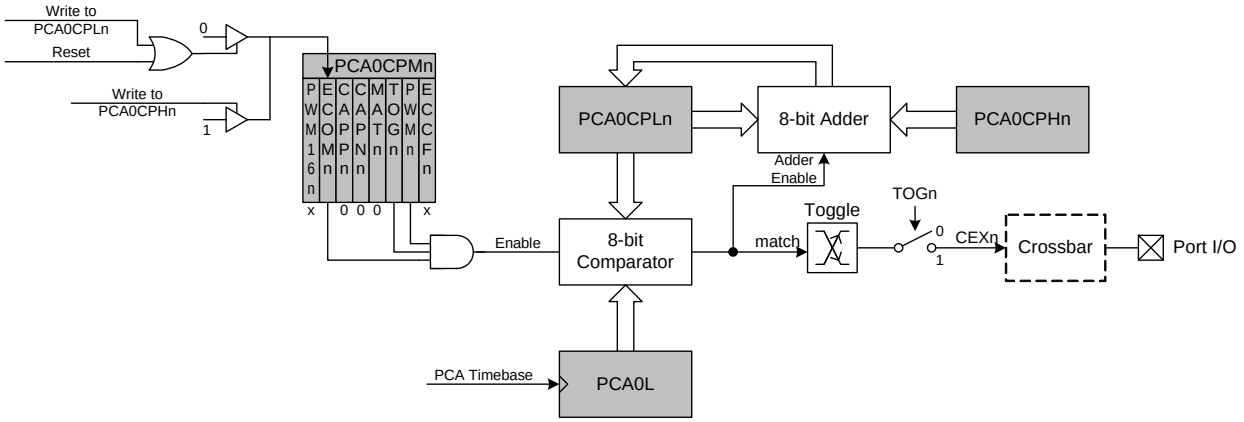


Figure 33.7. PCA Frequency Output Mode

33.3.5. 8-Bit, 9-Bit, 10-Bit and 11-Bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

33.3.5.1. 8-Bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 33.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module's capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 33.2.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(256 - PCA0CPHn)}{256}$$

Equation 33.2. 8-Bit PWM Duty Cycle

Using Equation 33.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

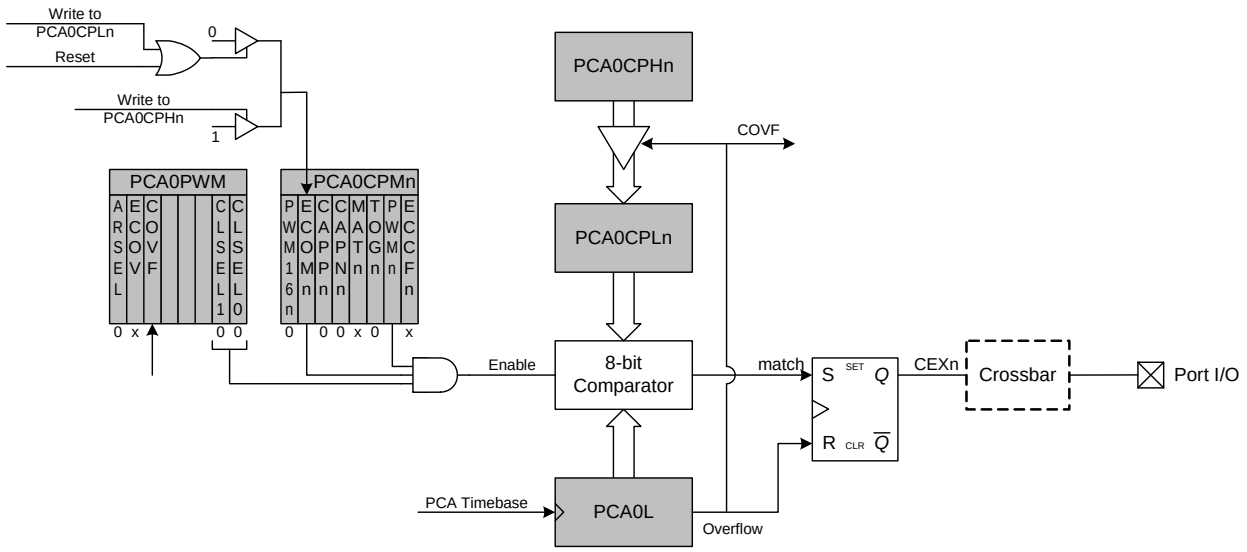


Figure 33.8. PCA 8-Bit PWM Mode Diagram

33.3.5.2. 9/10/11-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 9/10/11-bit PWM mode should be varied by writing to an "Auto-Reload" Register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0.

When the least-significant N bits of the PCA0 counter match the value in the associated module's capture/compare register (PCA0CPn), the output on CEXn is asserted high. When the counter overflows from the Nth bit, CEXn is asserted low (see Figure 33.9). Upon an overflow from the Nth bit, the COVF flag is set, and the value stored in the module's auto-reload register is loaded into the capture/compare register. The value of N is determined by the CLSEL bits in register PCA0PWM.

The 9, 10 or 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 512 (9-bit), 1024 (10-bit) or 2048 (11-bit) PCA clock cycles. The duty cycle for 9/10/11-Bit PWM Mode is given in Equation 33.3, where N is the number of bits in the PWM cycle.

Important Note About PCA0CPHn and PCA0CPLn Registers: When writing a 16-bit value to the PCA0CPn registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(2^N - PCA0CPn)}{2^N}$$

Equation 33.3. 9, 10, and 11-Bit PWM Duty Cycle

A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

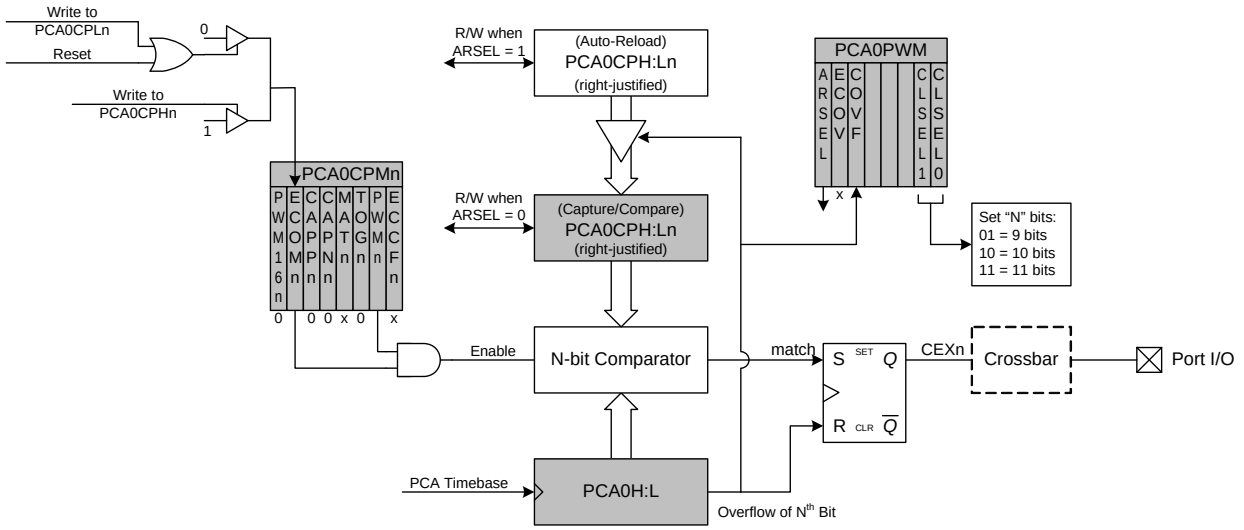


Figure 33.9. PCA 9, 10 and 11-Bit PWM Mode Diagram

33.3.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 33.4.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 33.4. 16-Bit PWM Duty Cycle

Using Equation 33.4, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

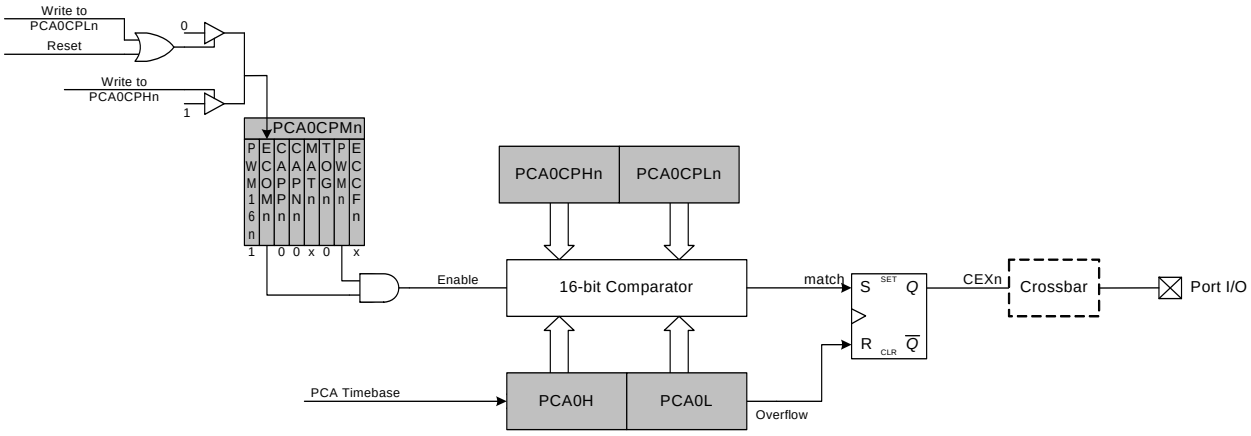


Figure 33.10. PCA 16-Bit PWM Mode

33.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 5. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 5 operates as a watchdog timer (WDT). The Module 5 high byte is compared to the PCA counter high byte; the Module 5 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

33.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2–CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 5 is forced into software timer mode.
- Writes to the Module 5 mode register (PCA0CPM5) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH5 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH5. Upon a PCA0CPH5 write, PCA0H plus the offset held in PCA0CPL5 is loaded into PCA0CPH5. (See Figure 33.11.)

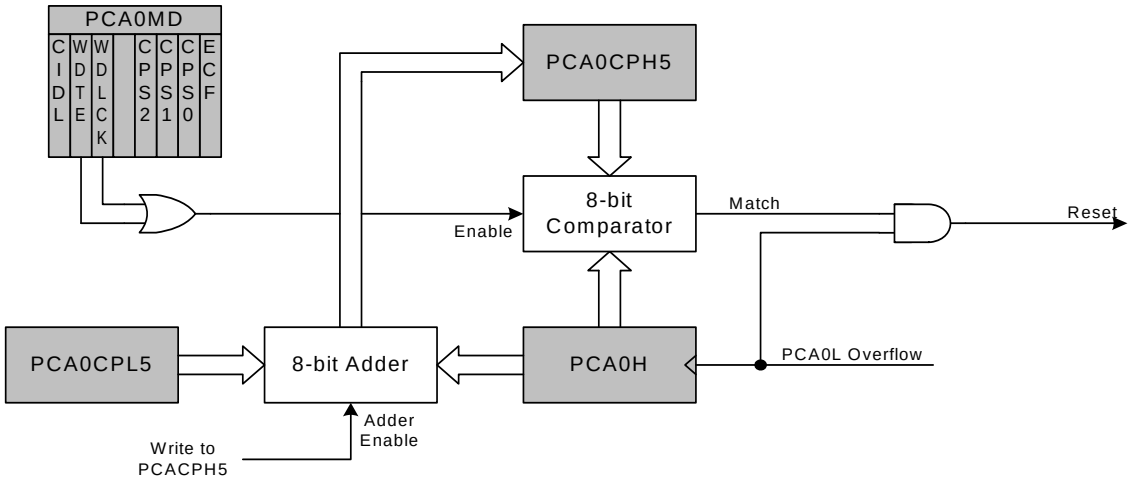


Figure 33.11. PCA Module 5 with Watchdog Timer Enabled

Note that the 8-bit offset held in PCA0CPH5 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 33.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL5) + (256 - PCA0L)$$

Equation 33.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH5 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF5 flag (PCA0CN.5) while the WDT is enabled.

33.4.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a 0 to the WDTE bit.
- Select the desired PCA clock source (with the CPS2–CPS0 bits).
- Load PCA0CPL5 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to 1.
- Reset the WDT timer by writing to PCA0CPH5.

The PCA clock source and idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL5 defaults to 0x00. Using Equation 33.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 33.3 lists some example time-out intervals for typical system clocks.

Table 33.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500[2]	255	257
3,062,500[2]	128	129.5
3,062,500[2]	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:
1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 8.

33.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 33.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xD8; Bit-Addressable

Bit	Name	Function
7	CF	**PCA Counter/Timer Overflow Flag.** Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	**PCA Counter/Timer Run Control.** This bit enables/disables the PCA Counter/Timer. 0: PCA Counter/Timer disabled. 1: PCA Counter/Timer enabled.
5:0	CCF[5:0]	**PCA Module n Capture/Compare Flag.** These bits are set by hardware when a match or capture occurs in the associated PCA Module n. When the CCFn interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

SFR Definition 33.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK		CPS2	CPS1	CPS0	ECF
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD9

Bit	Name	Function
7	CIDL	**PCA Counter/Timer Idle Control.** Specifies PCA behavior when CPU is in Idle Mode. 0: PCA continues to function normally while the system controller is in Idle Mode. 1: PCA operation is suspended while the system controller is in Idle Mode.
6	WDTE	**Watchdog Timer Enable.** If this bit is set, PCA Module 5 is used as the watchdog timer. 0: Watchdog Timer disabled. 1: PCA Module 5 enabled as Watchdog Timer.
5	WDLCK	**Watchdog Timer Lock.** This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Unused	Read = 0b, Write = don't care.
3:1	CPS[2:0]	**PCA Counter/Timer Pulse Select.** These bits select the timebase source for the PCA counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 110: SmaRTClock divided by 8 (synchronized with the system clock) 111: Reserved
0	ECF	**PCA Counter/Timer Overflow Interrupt Enable.** This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt. 0: Disable the CF interrupt. 1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.
Note: When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.		

SFR Definition 33.3. PCA0PWM: PCA PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL	ECOV	COVF				CLSEL[1:0]	
Type	R/W	R/W	R/W	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xDF

Bit	Name	Function
7	ARSEL	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9, 10, and 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function. 0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn. 1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn.
6	ECOV	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt. 0: COVF will not generate PCA interrupts. 1: A PCA interrupt will be generated when COVF is set.
5	COVF	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or software, but must be cleared by software. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:2	Unused	Read = 000b; Write = don’t care.
1:0	CLSEL[1:0]	**Cycle Length Select.** When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle, between 8, 9, 10, or 11 bits. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to16-bit PWM mode. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.

SFR Definition 33.4. PCA0CPMn: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address, Page: PCA0CPM0 = 0xDA, 0x0; PCA0CPM1 = 0xDB, 0x0; PCA0CPM2 = 0xDC, 0x0
PCA0CPM3 = 0xDD, 0x0; PCA0CPM4 = 0xDE, 0x0; PCA0CPM5 = 0xCE, 0x0

Bit	Name	Function
7	PWM16n	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOMn	**Comparator Function Enable.** This bit enables the comparator function for PCA module n when set to 1.
5	CAPPn	**Capture Positive Function Enable.** This bit enables the positive edge capture for PCA module n when set to 1.
4	CAPNn	**Capture Negative Function Enable.** This bit enables the negative edge capture for PCA module n when set to 1.
3	MATn	**Match Function Enable.** This bit enables the match function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	**Toggle Function Enable.** This bit enables the toggle function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function for PCA module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	ECCFn	**Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.

Note: When the WDTE bit is set to 1, the PCA0CPM5 register cannot be modified, and module 5 acts as the watchdog timer. To change the contents of the PCA0CPM5 register or the function of module 5, the Watchdog Timer must be disabled.

SFR Definition 33.5. PCA0L: PCA Counter/Timer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xF9

Bit	Name	Function
7:0	PCA0[7:0]	**PCA Counter/Timer Low Byte.** The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

Note: When the WDTE bit is set to 1, the PCA0L register cannot be modified by software. To change the contents of the PCA0L register, the Watchdog Timer must first be disabled.

SFR Definition 33.6. PCA0H: PCA Counter/Timer High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xFA

Bit	Name	Function
7:0	PCA0[15:8]	**PCA Counter/Timer High Byte.** The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Reads of this register will read the contents of a “snapshot” register, whose contents are updated only when the contents of PCA0L are read (see Section 33.1).

Note: When the WDTE bit is set to 1, the PCA0H register cannot be modified by software. To change the contents of the PCA0H register, the Watchdog Timer must first be disabled.

SFR Definition 33.7. PCA0CPLn: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPL0 = 0xFB, PCA0CPL1 = 0xE9, PCA0CPL2 = 0xEB,
PCA0CPL3 = 0xED, PCA0CPL4 = 0xFD, PCA0CPL5 = 0xD2

SFR Pages: PCA0CPL0 = 0x0, PCA0CPL1 = 0x0, PCA0CPL2 = 0x0,
PCA0CPL3 = 0x0, PCA0CPL4 = 0x0, PCA0CPL5 = 0x0

Bit	Name	Function
7:0	PCA0CPn[7:0]	**PCA Capture Module Low Byte.** The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOMn bit to a 0.		

SFR Definition 33.8. PCA0CPHn: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPH0 = 0xFC, PCA0CPH1 = 0xEA, PCA0CPH2 = 0xEC,
PCA0CPH3 = 0xEE, PCA0CPH4 = 0xFE, PCA0CPH5 = 0xD3

SFR Pages: PCA0CPH0 = 0x0, PCA0CPH1 = 0x0, PCA0CPH2 = 0x0,
PCA0CPH3 = 0x0, PCA0CPH4 = 0x0, PCA0CPH5 = 0x0

Bit	Name	Function
7:0	PCA0CPn[15:8]	**PCA Capture Module High Byte.** The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOMn bit to a 1.		

34. C2 Interface

C8051F96x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

34.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 34.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function	
7:0	C2ADD[7:0]	**C2 Address.** The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.	
		Address	**Description**
		0x00	Selects the Device ID register for Data Read instructions
		0x01	Selects the Revision ID register for Data Read instructions
		0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
		0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions

C2 Register Definition 34.2. DEVICEID: C2 Device ID

Bit	7	6	5	4	3	2	1	0
Name	DEVICEID[7:0]							
Type	R/W							
Reset	0	0	0	1	0	1	0	0

C2 Address: 0x00

Bit	Name	Function
7:0	DEVICEID[7:0]	**Device ID.** This read-only register returns the 8-bit device ID: 0x2A (C8051F96x).

C2 Register Definition 34.3. REVID: C2 Revision ID

Bit	7	6	5	4	3	2	1	0
Name	REVID[7:0]							
Type	R/W							
Reset	Varies	Varies	Varies	Varies	Varies	Varies	Varies	Varies

C2 Address: 0x01

Bit	Name	Function
7:0	REVID[7:0]	**Revision ID.** This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.

C2 Register Definition 34.4. FPCTL: C2 Flash Programming Control

Bit	7	6	5	4	3	2	1	0
Name	FPCTL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 34.5. FPDAT: C2 Flash Programming Data

Bit	7	6	5	4	3	2	1	0
Name	FPDAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xB4

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase

34.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (RST) and C2D pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 34.1.

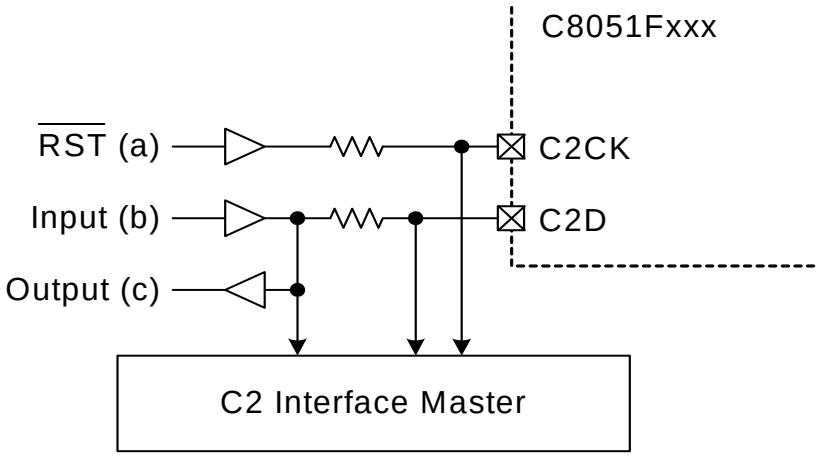


Figure 34.1. Typical C2 Pin Sharing

The configuration in Figure 34.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Added new content to DC0 chapter.
- Reordered chapters.
- Corrections to SFR tables.
- Updated Electrical Specifications.

Revision 0.2 to Revision 0.3

- Added new content to DMA0, CRC1, ENC0, SPI1, and Pulse Counter chapters.
- Added TQFP-80 package variant.
- Added package drawings and landing diagram for TQFP-80 package.
- Added via placement recommendations for DQFN-76 package.
- Updated electrical specifications.
- Corrections to SFR tables.
- Fixed inconsistencies in SFR names.
- Fixed inconsistencies in acronyms and terminology.

Revision 0.3 to Revision 0.5

- Updated maximum IBAT current using precision oscillator in Table 4.4.
- Updated sleep currents in Table 4.4.
- Added Note 1 to Table 4.6.
- Deleted SFR Page Stack Example in Special Function Registers chapter.
- Change description of SFRPGEN bit in SFRPGCN SFR definition.
- Added paragraph to Flash chapter to explain lock byte behavior on 128 kB devices.
- Corrected SFRPAGE in SPI1 SFR definitions 32.1/2/3.

NOTES:

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701

Please visit the Silicon Labs Technical Support web page:
https://www.silabs.com/support/pages/contacttechnicalsupport.aspx
and register to submit a technical support request.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.