# C8051F120/1/2/3/4/5/6/7

## 8K ISP FLASH MCU Family

### Analog Peripherals
- **10 or 12-bit SAR ADC**
  - ± 1 LSB INL
  - Programmable throughput up to 100 ksps
  - Up to 8 external inputs; programmable as single-ended or differential
  - Programmable amplifier gain: 16, 8, 4, 2, 1, 0.5
  - Data-dependent windowed interrupt generator
  - Built-in temperature sensor
- **8-bit SAR ADC ('F12x Only)**
  - Programmable throughput up to 500 ksps
  - 8 external inputs (single-ended or differential)
  - Programmable amplifier gain: 4, 2, 1, 0.5
- **Two 12-bit DACs ('F12x Only)**
  - Can synchronize outputs to timers for jitter-free waveform generation
- **Two Analog Comparators**
- **Voltage Reference**
- **$V_{DD}$ Monitor/Brown-Out Detector**

### On-Chip JTAG Debug & Boundary Scan
- On-chip debug circuitry facilitates full-speed, non-intrusive in-circuit/in-system debugging
- Provides breakpoints, single-stepping, watchpoints, stack monitor; inspect/modify memory and registers
- Superior performance to emulation systems using ICE-chips, target pods, and sockets
- IEEE1149.1 compliant boundary scan
- Complete development kit

### 100-Pin TQFP or 64-Pin TQFP Packaging
- Temperature Range: –40 to +85 °C
- RoHS Available

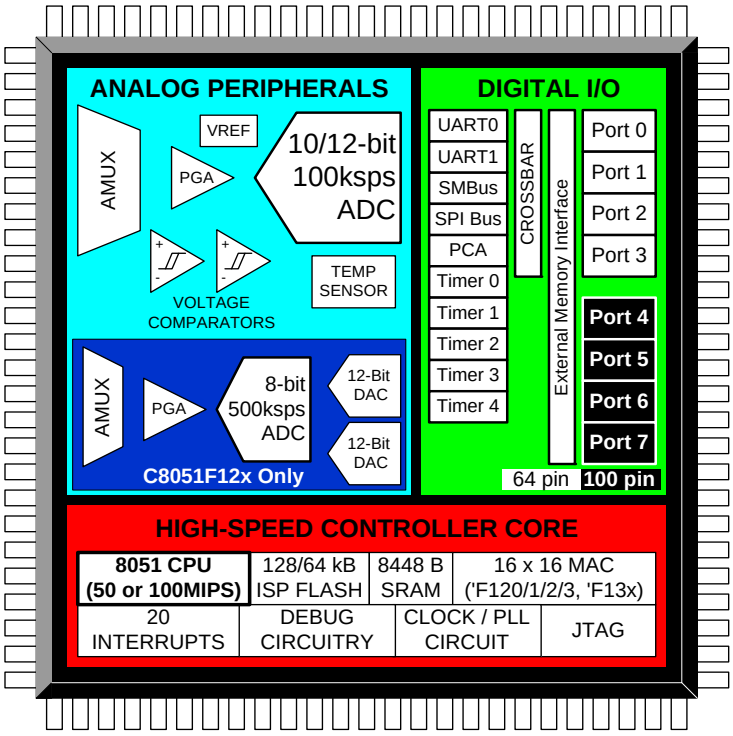
### High Speed 8051 µC Core
- Pipelined instruction architecture; executes 70% of instruction set in 1 or 2 system clocks
- 100 MIPS or 50 MIPS throughput with on-chip PLL
- 2-cycle 16 x 16 MAC engine (C8051F120/1/2/3 and C8051F130/1/2/3 only)

### Memory
- 8448 bytes internal data RAM (8 k + 256)
- 128 or 64 kB Banked Flash; in-system programmable in 1024-byte sectors
- External 64 kB data memory interface (programmable multiplexed or non-multiplexed modes)

### Digital Peripherals
- 8 byte-wide port I/O (100TQFP); 5 V tolerant
- 4 Byte-wide port I/O (64TQFP); 5 V tolerant
- Hardware SMBus™ (I2C™ Compatible), SPI™, and two UART serial ports available concurrently
- Programmable 16-bit counter/timer array with 6 capture/compare modules
- 5 general purpose 16-bit counter/timers
- Dedicated watchdog timer; bi-directional reset pin

### Clock Sources
- Internal precision oscillator: 24.5 MHz
- Flexible PLL technology
- External Oscillator: Crystal, RC, C, or clock

### Voltage Supples
- Range: 2.7–3.6 V (50 MIPS) 3.0–3.6 V (100 MIPS)
- Power saving sleep and shutdown modes




 Copyright © 2003 by Silicon Laboratories 

## NOTES:

## Table of Contents

**1. System Overview ..... 19**
1.1. CIP-51™ Microcontroller Core ..... 27
1.1.1. Fully 8051 Compatible ..... 27
1.1.2. Improved Throughput ..... 27
1.1.3. Additional Features ..... 28
1.2. On-Chip Memory ..... 29
1.3. JTAG Debug and Boundary Scan ..... 30
1.4. 16 x 16 MAC (Multiply and Accumulate) Engine ..... 31
1.5. Programmable Digital I/O and Crossbar ..... 32
1.6. Programmable Counter Array ..... 33
1.7. Serial Ports ..... 33
1.8. 12 or 10-Bit Analog to Digital Converter ..... 34
1.9. 8-Bit Analog to Digital Converter ..... 35
1.10.12-bit Digital to Analog Converters ..... 36
1.11.Analog Comparators ..... 37
**2. Absolute Maximum Ratings ..... 38**
**3. Global DC Electrical Characteristics ..... 39**
**4. Pinout and Package Definitions ..... 41**
**5. ADC0 (12-Bit ADC, C8051F120/1/4/5 Only) ..... 55**
5.1. Analog Multiplexer and PGA ..... 55
5.2. ADC Modes of Operation ..... 57
5.2.1. Starting a Conversion ..... 57
5.2.2. Tracking Modes ..... 58
5.2.3. Settling Time Requirements ..... 59
5.3. ADC0 Programmable Window Detector ..... 66
**6. ADC0 (10-Bit ADC, C8051F122/3/6/7 and C8051F13x Only) ..... 73**
6.1. Analog Multiplexer and PGA ..... 73
6.2. ADC Modes of Operation ..... 75
6.2.1. Starting a Conversion ..... 75
6.2.2. Tracking Modes ..... 76
6.2.3. Settling Time Requirements ..... 77
6.3. ADC0 Programmable Window Detector ..... 84
**7. ADC2 (8-Bit ADC, C8051F12x Only) ..... 91**
7.1. Analog Multiplexer and PGA ..... 91
7.2. ADC2 Modes of Operation ..... 92
7.2.1. Starting a Conversion ..... 92
7.2.2. Tracking Modes ..... 92
7.2.3. Settling Time Requirements ..... 94
7.3. ADC2 Programmable Window Detector ..... 100
7.3.1. Window Detector In Single-Ended Mode ..... 100
7.3.2. Window Detector In Differential Mode ..... 101

**8. DACs, 12-Bit Voltage Mode (C8051F12x Only) ..... 105**
8.1. DAC Output Scheduling ..... 105
8.1.1. Update Output On-Demand ..... 106
8.1.2. Update Output Based on Timer Overflow ..... 106
8.2. DAC Output Scaling/Justification ..... 106
**9. Voltage Reference ..... 113**
9.1. Reference Configuration on the C8051F120/2/4/6 ..... 113
9.2. Reference Configuration on the C8051F121/3/5/7 ..... 115
9.3. Reference Configuration on the C8051F130/1/2/3 ..... 117
**10. Comparators ..... 119**
**11. CIP-51 Microcontroller ..... 127**
11.1. Instruction Set ..... 129
11.1.1. Instruction and CPU Timing ..... 129
11.1.2. MOVX Instruction and Program Memory ..... 129
11.2. Memory Organization ..... 133
11.2.1. Program Memory ..... 133
11.2.2. Data Memory ..... 135
11.2.3. General Purpose Registers ..... 135
11.2.4. Bit Addressable Locations ..... 135
11.2.5. Stack ..... 135
11.2.6. Special Function Registers ..... 136
11.2.7. Register Descriptions ..... 151
11.3. Interrupt Handler ..... 154
11.3.1. MCU Interrupt Sources and Vectors ..... 154
11.3.2. External Interrupts ..... 155
11.3.3. Interrupt Priorities ..... 156
11.3.4. Interrupt Latency ..... 156
11.3.5. Interrupt Register Descriptions ..... 157
11.4. Power Management Modes ..... 163
11.4.1. Idle Mode ..... 163
11.4.2. Stop Mode ..... 164
**12. Multiply And Accumulate (MAC0) ..... 165**
12.1. Special Function Registers ..... 165
12.2. Integer and Fractional Math ..... 166
12.3. Operating in Multiply and Accumulate Mode ..... 167
12.4. Operating in Multiply Only Mode ..... 167
12.5. Accumulator Shift Operations ..... 167
12.6. Rounding and Saturation ..... 168
12.7. Usage Examples ..... 168
12.7.1. Multiply and Accumulate Example ..... 168
12.7.2. Multiply Only Example ..... 169
12.7.3. MAC0 Accumulator Shift Example ..... 169

**13. Reset Sources ..... 177**
13.1. Power-on Reset ..... 178
13.2. Power-fail Reset ..... 178
13.3. External Reset ..... 179
13.4. Missing Clock Detector Reset ..... 179
13.5. Comparator0 Reset ..... 179
13.6. External CNVSTR0 Pin Reset ..... 179
13.7. Watchdog Timer Reset ..... 179
13.7.1. Enable/Reset WDT ..... 180
13.7.2. Disable WDT ..... 180
13.7.3. Disable WDT Lockout ..... 180
13.7.4. Setting WDT Interval ..... 180
**14. Oscillators ..... 185**
14.1. Internal Calibrated Oscillator ..... 185
14.2. External Oscillator Drive Circuit ..... 187
14.3. System Clock Selection ..... 187
14.4. External Crystal Example ..... 190
14.5. External RC Example ..... 190
14.6. External Capacitor Example ..... 190
14.7. Phase-Locked Loop (PLL) ..... 191
14.7.1. PLL Input Clock and Pre-divider ..... 191
14.7.2. PLL Multiplication and Output Clock ..... 191
14.7.3. Powering on and Initializing the PLL ..... 192
**15. Flash Memory ..... 199**
15.1. Programming the Flash Memory ..... 199
15.1.1. Non-volatile Data Storage ..... 200
15.1.2. Erasing Flash Pages From Software ..... 201
15.1.3. Writing Flash Memory From Software ..... 202
15.2. Security Options ..... 203
15.2.1. Summary of Flash Security Options ..... 207
**16. Branch Target Cache ..... 211**
16.1. Cache and Prefetch Operation ..... 211
16.2. Cache and Prefetch Optimization ..... 212
**17. External Data Memory Interface and On-Chip XRAM ..... 219**
17.1. Accessing XRAM ..... 219
17.1.1. 16-Bit MOVX Example ..... 219
17.1.2. 8-Bit MOVX Example ..... 219
17.2. Configuring the External Memory Interface ..... 219
17.3. Port Selection and Configuration ..... 220
17.4. Multiplexed and Non-multiplexed Selection ..... 222
17.4.1. Multiplexed Configuration ..... 222
17.4.2. Non-multiplexed Configuration ..... 223
17.5. Memory Mode Selection ..... 224
17.5.1. Internal XRAM Only ..... 224
17.5.2. Split Mode without Bank Select ..... 224

17.5.3.Split Mode with Bank Select ..... 225
17.5.4.External Only ..... 225
17.6.EMIF Timing ..... 225
17.6.1.Non-multiplexed Mode ..... 227
17.6.2.Multiplexed Mode ..... 230
**18. Port Input/Output ..... 235**
18.1.Ports 0 through 3 and the Priority Crossbar Decoder ..... 238
18.1.1.Crossbar Pin Assignment and Allocation ..... 238
18.1.2.Configuring the Output Modes of the Port Pins ..... 239
18.1.3.Configuring Port Pins as Digital Inputs ..... 240
18.1.4.Weak Pullups ..... 240
18.1.5.Configuring Port 1 Pins as Analog Inputs ..... 240
18.1.6.External Memory Interface Pin Assignments ..... 241
18.1.7.Crossbar Pin Assignment Example ..... 243
18.2.Ports 4 through 7 (100-pin TQFP devices only) ..... 252
18.2.1.Configuring Ports which are not Pinned Out ..... 252
18.2.2.Configuring the Output Modes of the Port Pins ..... 252
18.2.3.Configuring Port Pins as Digital Inputs ..... 253
18.2.4.Weak Pullups ..... 253
18.2.5.External Memory Interface ..... 253
**19. System Management Bus / I2C Bus (SMBus0) ..... 259**
19.1.Supporting Documents ..... 260
19.2.SMBus Protocol ..... 260
19.2.1.Arbitration ..... 261
19.2.2.Clock Low Extension ..... 261
19.2.3.SCL Low Timeout ..... 261
19.2.4.SCL High (SMBus Free) Timeout ..... 261
19.3.SMBus Transfer Modes ..... 262
19.3.1.Master Transmitter Mode ..... 262
19.3.2.Master Receiver Mode ..... 262
19.3.3.Slave Transmitter Mode ..... 263
19.3.4.Slave Receiver Mode ..... 263
19.4.SMBus Special Function Registers ..... 264
19.4.1.Control Register ..... 264
19.4.2.Clock Rate Register ..... 267
19.4.3.Data Register ..... 268
19.4.4.Address Register ..... 268
19.4.5.Status Register ..... 269
**20. Enhanced Serial Peripheral Interface (SPI0) ..... 273**
20.1.Signal Descriptions ..... 274
20.1.1.Master Out, Slave In (MOSI) ..... 274
20.1.2.Master In, Slave Out (MISO) ..... 274
20.1.3.Serial Clock (SCK) ..... 274
20.1.4.Slave Select (NSS) ..... 274

20.2.SPI0 Master Mode Operation ..... 275
20.3.SPI0 Slave Mode Operation ..... 277
20.4.SPI0 Interrupt Sources ..... 277
20.5.Serial Clock Timing ..... 278
20.6.SPI Special Function Registers ..... 280
**21. UART0 ..... 287**
21.1.UART0 Operational Modes ..... 288
21.1.1.Mode 0: Synchronous Mode ..... 288
21.1.2.Mode 1: 8-Bit UART, Variable Baud Rate ..... 289
21.1.3.Mode 2: 9-Bit UART, Fixed Baud Rate ..... 291
21.1.4.Mode 3: 9-Bit UART, Variable Baud Rate ..... 292
21.2.Multiprocessor Communications ..... 293
21.2.1.Configuration of a Masked Address ..... 293
21.2.2.Broadcast Addressing ..... 293
21.3.Frame and Transmission Error Detection ..... 294
**22. UART1 ..... 299**
22.1.Enhanced Baud Rate Generation ..... 300
22.2.Operational Modes ..... 301
22.2.1.8-Bit UART ..... 301
22.2.2.9-Bit UART ..... 302
22.3.Multiprocessor Communications ..... 303
**23. Timers ..... 309**
23.1.Timer 0 and Timer 1 ..... 309
23.1.1.Mode 0: 13-bit Counter/Timer ..... 309
23.1.2.Mode 1: 16-bit Counter/Timer ..... 311
23.1.3.Mode 2: 8-bit Counter/Timer with Auto-Reload ..... 311
23.1.4.Mode 3: Two 8-bit Counter/Timers (Timer 0 Only) ..... 312
23.2.Timer 2, Timer 3, and Timer 4 ..... 317
23.2.1.Configuring Timer 2, 3, and 4 to Count Down ..... 317
23.2.2.Capture Mode ..... 318
23.2.3.Auto-Reload Mode ..... 319
23.2.4.Toggle Output Mode (Timer 2 and Timer 4 Only) ..... 320
**24. Programmable Counter Array ..... 325**
24.1.PCA Counter/Timer ..... 326
24.2.Capture/Compare Modules ..... 328
24.2.1.Edge-triggered Capture Mode ..... 329
24.2.2.Software Timer (Compare) Mode ..... 330
24.2.3.High Speed Output Mode ..... 331
24.2.4.Frequency Output Mode ..... 332
24.2.5.8-Bit Pulse Width Modulator Mode ..... 333
24.2.6.16-Bit Pulse Width Modulator Mode ..... 334
24.3.Register Descriptions for PCA0 ..... 335

**25. JTAG (IEEE 1149.1) ..... 341**
25.1.Boundary Scan ..... 342
25.1.1.EXTEST Instruction ..... 343
25.1.2.SAMPLE Instruction ..... 343
25.1.3.BYPASS Instruction ..... 343
25.1.4.IDCODE Instruction ..... 343
25.2.Flash Programming Commands ..... 344
25.3.Debug Support ..... 347
**Document Change List ..... 349**
**Contact Information ..... 350**

## List of Figures

**1. System Overview**
Figure 1.1. C8051F120/124 Block Diagram ..... 21
Figure 1.2. C8051F121/125 Block Diagram ..... 22
Figure 1.3. C8051F122/126 Block Diagram ..... 23
Figure 1.4. C8051F123/127 Block Diagram ..... 24
Figure 1.5. C8051F130/132 Block Diagram ..... 25
Figure 1.6. C8051F131/133 Block Diagram ..... 26
Figure 1.7. On-Board Clock and Reset ..... 28
Figure 1.8. On-Chip Memory Map ..... 29
Figure 1.9. Development/In-System Debug Diagram ..... 30
Figure 1.10. MAC0 Block Diagram ..... 31
Figure 1.11. Digital Crossbar Diagram ..... 32
Figure 1.12. PCA Block Diagram ..... 33
Figure 1.13. 12-Bit ADC Block Diagram ..... 34
Figure 1.14. 8-Bit ADC Diagram ..... 35
Figure 1.15. DAC System Block Diagram ..... 36
Figure 1.16. Comparator Block Diagram ..... 37
**2. Absolute Maximum Ratings**
**3. Global DC Electrical Characteristics**
**4. Pinout and Package Definitions**
Figure 4.1. C8051F120/2/4/6 Pinout Diagram (TQFP-100) ..... 49
Figure 4.2. C8051F130/2 Pinout Diagram (TQFP-100) ..... 50
Figure 4.3. TQFP-100 Package Drawing ..... 51
Figure 4.4. C8051F121/3/5/7 Pinout Diagram (TQFP-64) ..... 52
Figure 4.5. C8051F131/3 Pinout Diagram (TQFP-64) ..... 53
Figure 4.6. TQFP-64 Package Drawing ..... 54
**5. ADC0 (12-Bit ADC, C8051F120/1/4/5 Only)**
Figure 5.1. 12-Bit ADC0 Functional Block Diagram ..... 55
Figure 5.2. Typical Temperature Sensor Transfer Function ..... 56
Figure 5.3. ADC0 Track and Conversion Example Timing ..... 58
Figure 5.4. ADC0 Equivalent Input Circuits ..... 59
Figure 5.5. ADC0 Data Word Example ..... 65
Figure 5.6. 12-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data ..... 68
Figure 5.7. 12-Bit ADC0 Window Interrupt Example: Right Justified Differential Data ..... 69
Figure 5.8. 12-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data ..... 70
Figure 5.9. 12-Bit ADC0 Window Interrupt Example: Left Justified Differential Data ..... 71

**6. ADC0 (10-Bit ADC, C8051F122/3/6/7 and C8051F13x Only)**
Figure 6.1. 10-Bit ADC0 Functional Block Diagram ..... 73
Figure 6.2. Typical Temperature Sensor Transfer Function..... 74
Figure 6.3. ADC0 Track and Conversion Example Timing..... 76
Figure 6.4. ADC0 Equivalent Input Circuits..... 77
Figure 6.5. ADC0 Data Word Example ..... 83
Figure 6.6. 10-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data ..... 86
Figure 6.7. 10-Bit ADC0 Window Interrupt Example: Right Justified Differential Data ..... 87
Figure 6.8. 10-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data ..... 88
Figure 6.9. 10-Bit ADC0 Window Interrupt Example: Left Justified Differential Data..... 89
**7. ADC2 (8-Bit ADC, C8051F12x Only)**
Figure 7.1. ADC2 Functional Block Diagram..... 91
Figure 7.2. ADC2 Track and Conversion Example Timing..... 93
Figure 7.3. ADC2 Equivalent Input Circuit..... 94
Figure 7.4. ADC2 Data Word Example ..... 99
Figure 7.5. ADC2 Window Compare Examples, Single-Ended Mode..... 100
Figure 7.6. ADC2 Window Compare Examples, Differential Mode ..... 101
**8. DACs, 12-Bit Voltage Mode (C8051F12x Only)**
Figure 8.1. DAC Functional Block Diagram..... 105
**9. Voltage Reference**
Figure 9.1. Voltage Reference Functional Block Diagram (C8051F120/2/4/6) ..... 114
Figure 9.2. Voltage Reference Functional Block Diagram (C8051F121/3/5/7) ..... 115
Figure 9.3. Voltage Reference Functional Block Diagram (C8051F130/1/2/3) ..... 117
**10. Comparators**
Figure 10.1. Comparator Functional Block Diagram ..... 119
Figure 10.2. Comparator Hysteresis Plot ..... 121
**11. CIP-51 Microcontroller**
Figure 11.1. CIP-51 Block Diagram..... 128
Figure 11.2. Memory Map ..... 133
Figure 11.3. Address Memory Map for Instruction Fetches (128 kB Flash Only)... 134
Figure 11.4. SFR Page Stack..... 137
Figure 11.5. SFR Page Stack While Using SFR Page 0x0F To Access Port 5..... 138
Figure 11.6. SFR Page Stack After ADC2 Window Comparator Interrupt Occurs. 139
Figure 11.7. SFR Page Stack Upon PCA Interrupt Occurring During an ADC2 ISR 140
Figure 11.8. SFR Page Stack Upon Return From PCA Interrupt ..... 140
Figure 11.9. SFR Page Stack Upon Return From ADC2 Window Interrupt ..... 141
**12. Multiply And Accumulate (MAC0)**
Figure 12.1. MAC0 Block Diagram..... 165
Figure 12.2. Integer Mode Data Representation ..... 166
Figure 12.3. Fractional Mode Data Representation..... 166
Figure 12.4. MAC0 Pipeline..... 167

**13. Reset Sources**
Figure 13.1. Reset Sources ... 177
Figure 13.2. Reset Timing ... 178
**14. Oscillators**
Figure 14.1. Oscillator Diagram ... 185
Figure 14.2. PLL Block Diagram ... 191
**15. Flash Memory**
Figure 15.1. Flash Memory Map for MOVC Read and MOVX Write Operations ... 201
Figure 15.2. 128 kB Flash Memory Map and Security Bytes ... 204
Figure 15.3. 64 kB Flash Memory Map and Security Bytes ... 205
**16. Branch Target Cache**
Figure 16.1. Branch Target Cache Data Flow ... 211
Figure 16.2. Branch Target Cache Organiztion ... 212
Figure 16.3. Cache Lock Operation ... 214
**17. External Data Memory Interface and On-Chip XRAM**
Figure 17.1. Multiplexed Configuration Example ... 222
Figure 17.2. Non-multiplexed Configuration Example ... 223
Figure 17.3. EMIF Operating Modes ... 224
Figure 17.4. Non-multiplexed 16-bit MOVX Timing ... 227
Figure 17.5. Non-multiplexed 8-bit MOVX without Bank Select Timing ... 228
Figure 17.6. Non-multiplexed 8-bit MOVX with Bank Select Timing ... 229
Figure 17.7. Multiplexed 16-bit MOVX Timing ... 230
Figure 17.8. Multiplexed 8-bit MOVX without Bank Select Timing ... 231
Figure 17.9. Multiplexed 8-bit MOVX with Bank Select Timing ... 232
**18. Port Input/Output**
Figure 18.1. Port I/O Cell Block Diagram ... 235
Figure 18.2. Port I/O Functional Block Diagram ... 237
Figure 18.3. Priority Crossbar Decode Table (EMIFLE = 0; P1MDIN = 0xFF) ... 238
Figure 18.4. Priority Crossbar Decode Table (EMIFLE = 1; EMIF in Multiplexed Mode; P1MDIN = 0xFF) ... 241
Figure 18.5. Priority Crossbar Decode Table (EMIFLE = 1; EMIF in Non-Multiplexed Mode; P1MDIN = 0xFF) ... 242
Figure 18.6. Crossbar Example ... 244
**19. System Management Bus / I2C Bus (SMBus0)**
Figure 19.1. SMBus0 Block Diagram ... 259
Figure 19.2. Typical SMBus Configuration ... 260
Figure 19.3. SMBus Transaction ... 261
Figure 19.4. Typical Master Transmitter Sequence ... 262
Figure 19.5. Typical Master Receiver Sequence ... 262
Figure 19.6. Typical Slave Transmitter Sequence ... 263
Figure 19.7. Typical Slave Receiver Sequence ... 263
**20. Enhanced Serial Peripheral Interface (SPI0)**
Figure 20.1. SPI Block Diagram ... 273
Figure 20.2. Multiple-Master Mode Connection Diagram ... 276
Figure 20.3. 3-Wire Single Master and Slave Mode Connection Diagram ... 276

Figure 20.4. 4-Wire Single Master and Slave Mode Connection Diagram ..... 276
Figure 20.5. Master Mode Data/Clock Timing ..... 278
Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0) ..... 279
Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1) ..... 279
Figure 20.8. SPI Master Timing (CKPHA = 0) ..... 283
Figure 20.9. SPI Master Timing (CKPHA = 1) ..... 283
Figure 20.10. SPI Slave Timing (CKPHA = 0) ..... 284
Figure 20.11. SPI Slave Timing (CKPHA = 1) ..... 284
**21. UART0**
Figure 21.1. UART0 Block Diagram ..... 287
Figure 21.2. UART0 Mode 0 Timing Diagram ..... 288
Figure 21.3. UART0 Mode 0 Interconnect ..... 288
Figure 21.4. UART0 Mode 1 Timing Diagram ..... 289
Figure 21.5. UART0 Modes 2 and 3 Timing Diagram ..... 291
Figure 21.6. UART0 Modes 1, 2, and 3 Interconnect Diagram ..... 292
Figure 21.7. UART Multi-Processor Mode Interconnect Diagram ..... 294
**22. UART1**
Figure 22.1. UART1 Block Diagram ..... 299
Figure 22.2. UART1 Baud Rate Logic ..... 300
Figure 22.3. UART Interconnect Diagram ..... 301
Figure 22.4. 8-Bit UART Timing Diagram ..... 301
Figure 22.5. 9-Bit UART Timing Diagram ..... 302
Figure 22.6. UART Multi-Processor Mode Interconnect Diagram ..... 303
**23. Timers**
Figure 23.1. T0 Mode 0 Block Diagram ..... 310
Figure 23.2. T0 Mode 2 Block Diagram ..... 311
Figure 23.3. T0 Mode 3 Block Diagram ..... 312
Figure 23.4. T2, 3, and 4 Capture Mode Block Diagram ..... 318
Figure 23.5. Tn Auto-reload (T2,3,4) and Toggle Mode (T2,4) Block Diagram ..... 319
**24. Programmable Counter Array**
Figure 24.1. PCA Block Diagram ..... 325
Figure 24.2. PCA Counter/Timer Block Diagram ..... 326
Figure 24.3. PCA Interrupt Block Diagram ..... 328
Figure 24.4. PCA Capture Mode Diagram ..... 329
Figure 24.5. PCA Software Timer Mode Diagram ..... 330
Figure 24.6. PCA High Speed Output Mode Diagram ..... 331
Figure 24.7. PCA Frequency Output Mode ..... 332
Figure 24.8. PCA 8-Bit PWM Mode Diagram ..... 333
Figure 24.9. PCA 16-Bit PWM Mode ..... 334
**25. JTAG (IEEE 1149.1)**

## List Of Tables

**1. System Overview**
Table 1.1. Product Selection Guide ..... 20
**2. Absolute Maximum Ratings**
Table 2.1. Absolute Maximum Ratings ..... 38
**3. Global DC Electrical Characteristics**
Table 3.1. Global DC Electrical Characteristics (C8051F120/1/2/3 and C8051F130/1/2/3) ..... 39
Table 3.2. Global DC Electrical Characteristics (C8051F124/5/6/7) ..... 40
**4. Pinout and Package Definitions**
Table 4.1. Pin Definitions ..... 41
**5. ADC0 (12-Bit ADC, C8051F120/1/4/5 Only)**
Table 5.1. 12-Bit ADC0 Electrical Characteristics (C8051F120/1/4/5) ..... 72
**6. ADC0 (10-Bit ADC, C8051F122/3/6/7 and C8051F13x Only)**
Table 6.1. 10-Bit ADC0 Electrical Characteristics (C8051F122/3/6/7 and C8051F13x) ..... 90
**7. ADC2 (8-Bit ADC, C8051F12x Only)**
Table 7.1. ADC2 Electrical Characteristics ..... 103
**8. DACs, 12-Bit Voltage Mode (C8051F12x Only)**
Table 8.1. DAC Electrical Characteristics ..... 111
**9. Voltage Reference**
Table 9.1. Voltage Reference Electrical Characteristics ..... 118
**10. Comparators**
Table 10.1. Comparator Electrical Characteristics ..... 126
**11. CIP-51 Microcontroller**
Table 11.1. CIP-51 Instruction Set Summary ..... 129
Table 11.2. Special Function Register (SFR) Memory Map ..... 144
Table 11.3. Special Function Registers ..... 146
Table 11.4. Interrupt Summary ..... 155
**12. Multiply And Accumulate (MAC0)**
Table 12.1. MAC0 Rounding (MAC0SAT = 0) ..... 168
**13. Reset Sources**
Table 13.1. Reset Electrical Characteristics ..... 183
**14. Oscillators**
Table 14.1. Oscillator Electrical Characteristics ..... 185
Table 14.2. PLL Frequency Characteristics ..... 195
Table 14.3. PLL Lock Timing Characteristics ..... 196
**15. Flash Memory**
Table 15.1. Flash Electrical Characteristics ..... 200
**16. Branch Target Cache**
**17. External Data Memory Interface and On-Chip XRAM**
Table 17.1. AC Parameters for External Memory Interface ..... 233

**18. Port Input/Output**
Table 18.1. Port I/O DC Electrical Characteristics ..... 236
**19. System Management Bus / I2C Bus (SMBus0)**
Table 19.1. SMB0STA Status Codes and States ..... 270
**20. Enhanced Serial Peripheral Interface (SPI0)**
Table 20.1. SPI Slave Timing Parameters ..... 285
**21. UART0**
Table 21.1. UART0 Modes ..... 288
Table 21.2. Oscillator Frequencies for Standard Baud Rates ..... 295
**22. UART1**
Table 22.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator ..... 305
Table 22.2. Timer Settings for Standard Baud Rates Using an External 25.0 MHz Oscillator ..... 306
Table 22.3. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator ..... 306
Table 22.4. Timer Settings for Standard Baud Rates Using the PLL ..... 307
Table 22.5. Timer Settings for Standard Baud Rates Using the PLL ..... 307
**23. Timers**
**24. Programmable Counter Array**
Table 24.1. PCA Timebase Input Options ..... 326
Table 24.2. PCA0CPM Register Settings for PCA Capture/Compare Modules .... 329
**25. JTAG (IEEE 1149.1)**
Table 25.1. Boundary Data Register Bit Definitions ..... 342

## List of Registers

SFR Definition 5.1. AMX0CF: AMUX0 Configuration . . . . . 60
SFR Definition 5.2. AMX0SL: AMUX0 Channel Select . . . . . 61
SFR Definition 5.3. ADC0CF: ADC0 Configuration . . . . . 62
SFR Definition 5.4. ADC0CN: ADC0 Control . . . . . 63
SFR Definition 5.5. ADC0H: ADC0 Data Word MSB . . . . . 64
SFR Definition 5.6. ADC0L: ADC0 Data Word LSB . . . . . 64
SFR Definition 5.7. ADC0GTH: ADC0 Greater-Than Data High Byte . . . . . 66
SFR Definition 5.8. ADC0GTL: ADC0 Greater-Than Data Low Byte . . . . . 66
SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte . . . . . 67
SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte . . . . . 67
SFR Definition 6.1. AMX0CF: AMUX0 Configuration . . . . . 78
SFR Definition 6.2. AMX0SL: AMUX0 Channel Select . . . . . 79
SFR Definition 6.3. ADC0CF: ADC0 Configuration . . . . . 80
SFR Definition 6.4. ADC0CN: ADC0 Control . . . . . 81
SFR Definition 6.5. ADC0H: ADC0 Data Word MSB . . . . . 82
SFR Definition 6.6. ADC0L: ADC0 Data Word LSB . . . . . 82
SFR Definition 6.7. ADC0GTH: ADC0 Greater-Than Data High Byte . . . . . 84
SFR Definition 6.8. ADC0GTL: ADC0 Greater-Than Data Low Byte . . . . . 84
SFR Definition 6.9. ADC0LTH: ADC0 Less-Than Data High Byte . . . . . 85
SFR Definition 6.10. ADC0LTL: ADC0 Less-Than Data Low Byte . . . . . 85
SFR Definition 7.1. AMX2CF: AMUX2 Configuration . . . . . 95
SFR Definition 7.2. AMX2SL: AMUX2 Channel Select . . . . . 96
SFR Definition 7.3. ADC2CF: ADC2 Configuration . . . . . 97
SFR Definition 7.4. ADC2CN: ADC2 Control . . . . . 98
SFR Definition 7.5. ADC2: ADC2 Data Word . . . . . 99
SFR Definition 7.6. ADC2GT: ADC2 Greater-Than Data Byte . . . . . 102
SFR Definition 7.7. ADC2LT: ADC2 Less-Than Data Byte . . . . . 102
SFR Definition 8.1. DAC0H: DAC0 High Byte . . . . . 107
SFR Definition 8.2. DAC0L: DAC0 Low Byte . . . . . 107
SFR Definition 8.3. DAC0CN: DAC0 Control . . . . . 108
SFR Definition 8.4. DAC1H: DAC1 High Byte . . . . . 109
SFR Definition 8.5. DAC1L: DAC1 Low Byte . . . . . 109
SFR Definition 8.6. DAC1CN: DAC1 Control . . . . . 110
SFR Definition 9.1. REF0CN: Reference Control (C8051F120/2/4/6) . . . . . 114
SFR Definition 9.2. REF0CN: Reference Control (C8051F121/3/5/7) . . . . . 116
SFR Definition 9.3. REF0CN: Reference Control (C8051F130/1/2/3) . . . . . 117
SFR Definition 10.1. CPT0CN: Comparator0 Control . . . . . 122
SFR Definition 10.2. CPT0MD: Comparator0 Mode Selection . . . . . 123
SFR Definition 10.3. CPT1CN: Comparator1 Control . . . . . 124
SFR Definition 10.4. CPT1MD: Comparator1 Mode Selection . . . . . 125
SFR Definition 11.1. PSBANK: Program Space Bank Select . . . . . 134
SFR Definition 11.2. SFRPGCN: SFR Page Control . . . . . 142
SFR Definition 11.3. SFRPAGE: SFR Page . . . . . 142

SFR Definition 11.4. SFRNEXT: SFR Next Register . . . . . 143
SFR Definition 11.5. SFRLAST: SFR Last Register . . . . . 143
SFR Definition 11.6. SP: Stack Pointer . . . . . 151
SFR Definition 11.7. DPL: Data Pointer Low Byte . . . . . 151
SFR Definition 11.8. DPH: Data Pointer High Byte . . . . . 151
SFR Definition 11.9. PSW: Program Status Word . . . . . 152
SFR Definition 11.10. ACC: Accumulator . . . . . 153
SFR Definition 11.11. B: B Register . . . . . 153
SFR Definition 11.12. IE: Interrupt Enable . . . . . 157
SFR Definition 11.13. IP: Interrupt Priority . . . . . 158
SFR Definition 11.14. EIE1: Extended Interrupt Enable 1 . . . . . 159
SFR Definition 11.15. EIE2: Extended Interrupt Enable 2 . . . . . 160
SFR Definition 11.16. EIP1: Extended Interrupt Priority 1 . . . . . 161
SFR Definition 11.17. EIP2: Extended Interrupt Priority 2 . . . . . 162
SFR Definition 11.18. PCON: Power Control . . . . . 164
SFR Definition 12.1. MAC0CF: MAC0 Configuration . . . . . 170
SFR Definition 12.2. MAC0STA: MAC0 Status . . . . . 171
SFR Definition 12.3. MAC0AH: MAC0 A High Byte . . . . . 171
SFR Definition 12.4. MAC0AL: MAC0 A Low Byte . . . . . 172
SFR Definition 12.5. MAC0BH: MAC0 B High Byte . . . . . 172
SFR Definition 12.6. MAC0BL: MAC0 B Low Byte . . . . . 172
SFR Definition 12.7. MAC0ACC3: MAC0 Accumulator Byte 3 . . . . . 173
SFR Definition 12.8. MAC0ACC2: MAC0 Accumulator Byte 2 . . . . . 173
SFR Definition 12.9. MAC0ACC1: MAC0 Accumulator Byte 1 . . . . . 173
SFR Definition 12.10. MAC0ACC0: MAC0 Accumulator Byte 0 . . . . . 174
SFR Definition 12.11. MAC0OVR: MAC0 Accumulator Overflow . . . . . 174
SFR Definition 12.12. MAC0RNDH: MAC0 Rounding Register High Byte . . . . . 174
SFR Definition 12.13. MAC0RNDL: MAC0 Rounding Register Low Byte . . . . . 175
SFR Definition 13.1. WDTCN: Watchdog Timer Control . . . . . 181
SFR Definition 13.2. RSTSRC: Reset Source . . . . . 182
SFR Definition 14.1. OSCICL: Internal Oscillator Calibration. . . . . 186
SFR Definition 14.2. OSCICN: Internal Oscillator Control . . . . . 186
SFR Definition 14.3. CLKSEL: System Clock Selection . . . . . 188
SFR Definition 14.4. OSCXCN: External Oscillator Control . . . . . 189
SFR Definition 14.5. PLL0CN: PLL Control . . . . . 193
SFR Definition 14.6. PLL0DIV: PLL Pre-divider . . . . . 194
SFR Definition 14.7. PLL0MUL: PLL Clock Scaler . . . . . 194
SFR Definition 14.8. PLL0FLT: PLL Filter . . . . . 195
SFR Definition 15.1. FLACL: Flash Access Limit . . . . . 206
SFR Definition 15.2. FLSCL: Flash Memory Control . . . . . 208
SFR Definition 15.3. PSCTL: Program Store Read/Write Control . . . . . 209
SFR Definition 16.1. CCH0CN: Cache Control . . . . . 215
SFR Definition 16.2. CCH0TN: Cache Tuning . . . . . 216
SFR Definition 16.3. CCH0LC: Cache Lock Control . . . . . 216
SFR Definition 16.4. CCH0MA: Cache Miss Accumulator . . . . . 217

SFR Definition 16.5. FLSTAT: Flash Status . . . . . 217
SFR Definition 17.1. EMI0CN: External Memory Interface Control . . . . . 220
SFR Definition 17.2. EMI0CF: External Memory Configuration . . . . . 221
SFR Definition 17.3. EMI0TC: External Memory Timing Control . . . . . 226
SFR Definition 18.1. XBR0: Port I/O Crossbar Register 0 . . . . . 245
SFR Definition 18.2. XBR1: Port I/O Crossbar Register 1 . . . . . 246
SFR Definition 18.3. XBR2: Port I/O Crossbar Register 2 . . . . . 247
SFR Definition 18.4. P0: Port0 Data . . . . . 248
SFR Definition 18.5. P0MDOUT: Port0 Output Mode . . . . . 248
SFR Definition 18.6. P1: Port1 Data . . . . . 249
SFR Definition 18.7. P1MDIN: Port1 Input Mode . . . . . 249
SFR Definition 18.8. P1MDOUT: Port1 Output Mode . . . . . 250
SFR Definition 18.9. P2: Port2 Data . . . . . 250
SFR Definition 18.10. P2MDOUT: Port2 Output Mode . . . . . 251
SFR Definition 18.11. P3: Port3 Data . . . . . 251
SFR Definition 18.12. P3MDOUT: Port3 Output Mode . . . . . 252
SFR Definition 18.13. P4: Port4 Data . . . . . 254
SFR Definition 18.14. P4MDOUT: Port4 Output Mode . . . . . 254
SFR Definition 18.15. P5: Port5 Data . . . . . 255
SFR Definition 18.16. P5MDOUT: Port5 Output Mode . . . . . 255
SFR Definition 18.17. P6: Port6 Data . . . . . 256
SFR Definition 18.18. P6MDOUT: Port6 Output Mode . . . . . 256
SFR Definition 18.19. P7: Port7 Data . . . . . 257
SFR Definition 18.20. P7MDOUT: Port7 Output Mode . . . . . 257
SFR Definition 19.1. SMB0CN: SMBus0 Control . . . . . 266
SFR Definition 19.2. SMB0CR: SMBus0 Clock Rate . . . . . 267
SFR Definition 19.3. SMB0DAT: SMBus0 Data . . . . . 268
SFR Definition 19.4. SMB0ADR: SMBus0 Address . . . . . 269
SFR Definition 19.5. SMB0STA: SMBus0 Status . . . . . 269
SFR Definition 20.1. SPI0CFG: SPI0 Configuration . . . . . 280
SFR Definition 20.2. SPI0CN: SPI0 Control . . . . . 281
SFR Definition 20.3. SPI0CKR: SPI0 Clock Rate . . . . . 282
SFR Definition 20.4. SPI0DAT: SPI0 Data . . . . . 282
SFR Definition 21.1. SCON0: UART0 Control . . . . . 296
SFR Definition 21.2. SSTA0: UART0 Status and Clock Selection . . . . . 297
SFR Definition 21.3. SBUF0: UART0 Data Buffer . . . . . 298
SFR Definition 21.4. SADDR0: UART0 Slave Address . . . . . 298
SFR Definition 21.5. SADEN0: UART0 Slave Address Enable . . . . . 298
SFR Definition 22.1. SCON1: Serial Port 1 Control . . . . . 304
SFR Definition 22.2. SBUF1: Serial (UART1) Port Data Buffer . . . . . 305
SFR Definition 23.1. TCON: Timer Control . . . . . 313
SFR Definition 23.2. TMOD: Timer Mode . . . . . 314
SFR Definition 23.3. CKCON: Clock Control . . . . . 315
SFR Definition 23.4. TL0: Timer 0 Low Byte . . . . . 315
SFR Definition 23.5. TL1: Timer 1 Low Byte . . . . . 316

SFR Definition 23.6. TH0: Timer 0 High Byte . . . . . 316
SFR Definition 23.7. TH1: Timer 1 High Byte . . . . . 316
SFR Definition 23.8. TMRnCN: Timer 2, 3, and 4 Control . . . . . 321
SFR Definition 23.9. TMRnCF: Timer 2, 3, and 4 Configuration . . . . . 322
SFR Definition 23.10. RCAPnL: Timer 2, 3, and 4 Capture Register Low Byte . . . . . 323
SFR Definition 23.11. RCAPnH: Timer 2, 3, and 4 Capture Register High Byte . . . . 323
SFR Definition 23.12. TMRnL: Timer 2, 3, and 4 Low Byte . . . . . 323
SFR Definition 23.13. TMRnH Timer 2, 3, and 4 High Byte . . . . . 324
SFR Definition 24.1. PCA0CN: PCA Control . . . . . 335
SFR Definition 24.2. PCA0MD: PCA0 Mode . . . . . 336
SFR Definition 24.3. PCA0CPMn: PCA0 Capture/Compare Mode . . . . . 337
SFR Definition 24.4. PCA0L: PCA0 Counter/Timer Low Byte . . . . . 338
SFR Definition 24.5. PCA0H: PCA0 Counter/Timer High Byte . . . . . 338
SFR Definition 24.6. PCA0CPLn: PCA0 Capture Module Low Byte . . . . . 338
SFR Definition 24.7. PCA0CPHn: PCA0 Capture Module High Byte . . . . . 339
JTAG Register Definition 25.1. IR: JTAG Instruction Register . . . . . 341
JTAG Register Definition 25.2. DEVICEID: JTAG Device ID . . . . . 343
JTAG Register Definition 25.3. FLASHCON: JTAG Flash Control . . . . . 345
JTAG Register Definition 25.4. FLASHDAT: JTAG Flash Data . . . . . 346
JTAG Register Definition 25.5. FLASHADR: JTAG Flash Address . . . . . 346

# 1. System Overview

The C8051F12x and C8051F13x device families are fully integrated mixed-signal System-on-a-Chip MCUs with 64 digital I/O pins (100-pin TQFP) or 32 digital I/O pins (64-pin TQFP).

Highlighted features are listed below. Refer to Table 1.1 for specific product feature selection.

- High-Speed pipelined 8051-compatible CIP-51 microcontroller core (100 MIPS or 50 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 12 or 10-bit 100 ksps ADC with PGA and 8-channel analog multiplexer
- True 8-bit 500 ksps ADC with PGA and 8-channel analog multiplexer (C8051F12x Family)
- Two 12-bit DACs with programmable update scheduling (C8051F12x Family)
- 2-cycle 16 by 16 Multiply and Accumulate Engine (C8051F120/1/2/3 and C8051F130/1/2/3)
- 128 or 64 kB of in-system programmable Flash memory
- 8448 (8 k + 256) bytes of on-chip RAM
- External Data Memory Interface with 64 kB address space
- SPI, SMBus/I2C, and (2) UART serial interfaces implemented in hardware
- Five general purpose 16-bit Timers
- Programmable Counter/Timer Array with 6 capture/compare modules
- On-chip Watchdog Timer, $V_{DD}$ Monitor, and Temperature Sensor

With on-chip $V_{DD}$ monitor, Watchdog Timer, and clock oscillator, the C8051F12x and C8051F13x devices are truly stand-alone System-on-a-Chip solutions. All analog and digital peripherals are enabled/disabled and configured by user firmware. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware.

On-board JTAG debug circuitry allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug system supports inspection and modification of memory and registers, setting breakpoints, watchpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using JTAG.

Each MCU is specified for operation over the industrial temperature range (–45 to +85 °C). The Port I/O, $\overline{RST}$, and JTAG pins are tolerant for input signals up to 5 V. The devices are available in 100-pin TQFP or 64-pin TQFP packaging. Table 1.1 lists the specific device features and package offerings for each part number. Figure 1.1 through Figure 1.6 show functional block diagrams for each device.

**Table 1.1. Product Selection Guide**

| Ordering Part Number | MIPS (Peak) | Flash Memory | RAM | 2-cycle 16 by 16 MAC | External Memory Interface | SMBus/I2C | SPI | UARTS | Timers (16-bit) | Programmable Counter Array | Digital Port I/O's | 12-bit 100ksps ADC Inputs | 10-bit 100ksps ADC Inputs | 8-bit 500ksps ADC Inputs | Voltage Reference | Temperature Sensor | DAC Resolution (bits) | DAC Outputs | Analog Comparators | Lead-Free (RoHS Compliant) | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C8051F120 | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 100TQFP |
| C8051F120-GQ | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 100TQFP |
| C8051F121 | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 64TQFP |
| C8051F121-GQ | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 64TQFP |
| C8051F122 | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 100TQFP |
| C8051F122-GQ | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 100TQFP |
| C8051F123 | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 64TQFP |
| C8051F123-GQ | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 64TQFP |
| C8051F124 | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 100TQFP |
| C8051F124-GQ | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 100TQFP |
| C8051F125 | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 64TQFP |
| C8051F125-GQ | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | 8 | - | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 64TQFP |
| C8051F126 | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 100TQFP |
| C8051F126-GQ | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 100TQFP |
| C8051F127 | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | - | 64TQFP |
| C8051F127-GQ | 50 | 128 k | 8448 | - | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | 8 | ✓ | ✓ | 12 | 2 | 2 | ✓ | 64TQFP |
| C8051F130 | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | - | ✓ | ✓ | - | - | 2 | - | 100TQFP |
| C8051F130-GQ | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | - | ✓ | ✓ | - | - | 2 | ✓ | 100TQFP |
| C8051F131 | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | - | ✓ | ✓ | - | - | 2 | - | 64TQFP |
| C8051F131-GQ | 100 | 128 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | - | ✓ | ✓ | - | - | 2 | ✓ | 64TQFP |
| C8051F132 | 100 | 64 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | - | ✓ | ✓ | - | - | 2 | - | 100TQFP |
| C8051F132-GQ | 100 | 64 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 64 | - | 8 | - | ✓ | ✓ | - | - | 2 | ✓ | 100TQFP |
| C8051F133 | 100 | 64 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | - | ✓ | ✓ | - | - | 2 | - | 64TQFP |
| C8051F133-GQ | 100 | 64 k | 8448 | ✓ | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 32 | - | 8 | - | ✓ | ✓ | - | - | 2 | ✓ | 64TQFP |

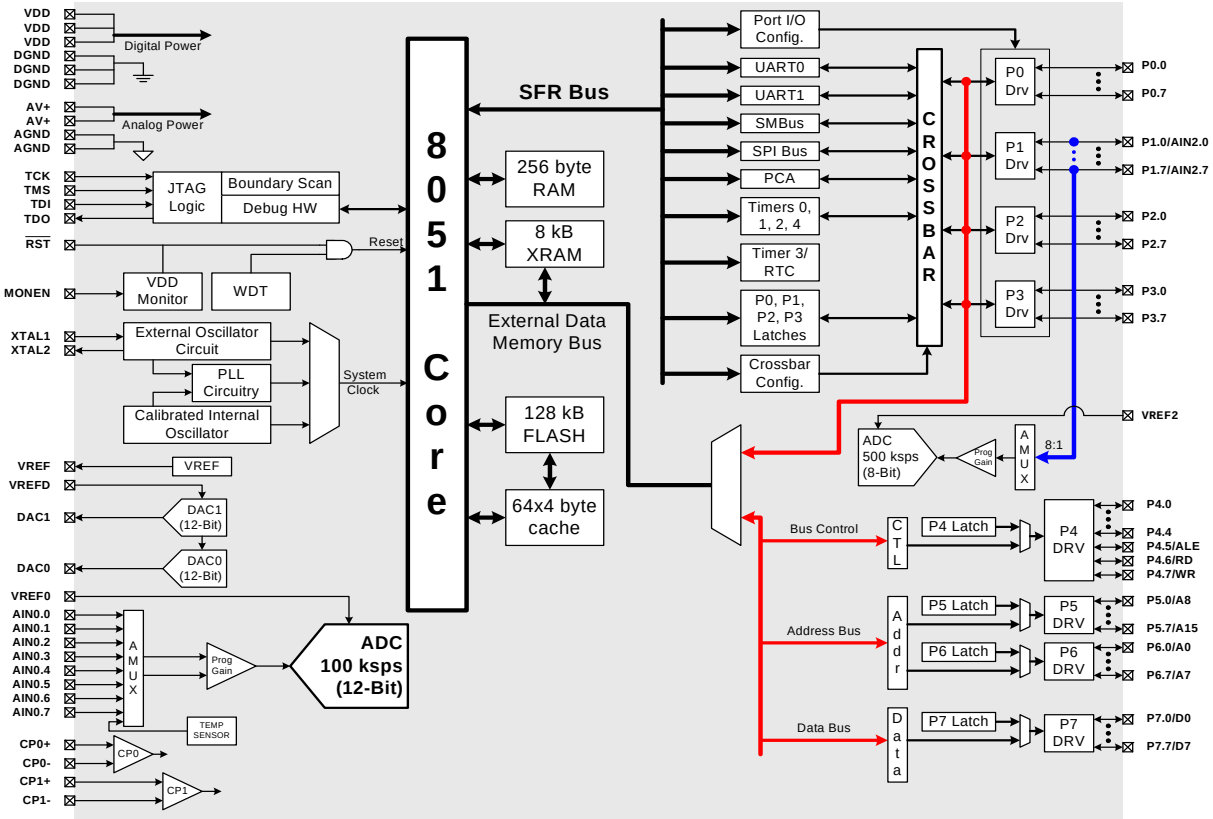



**Figure 1.1. C8051F120/124 Block Diagram**

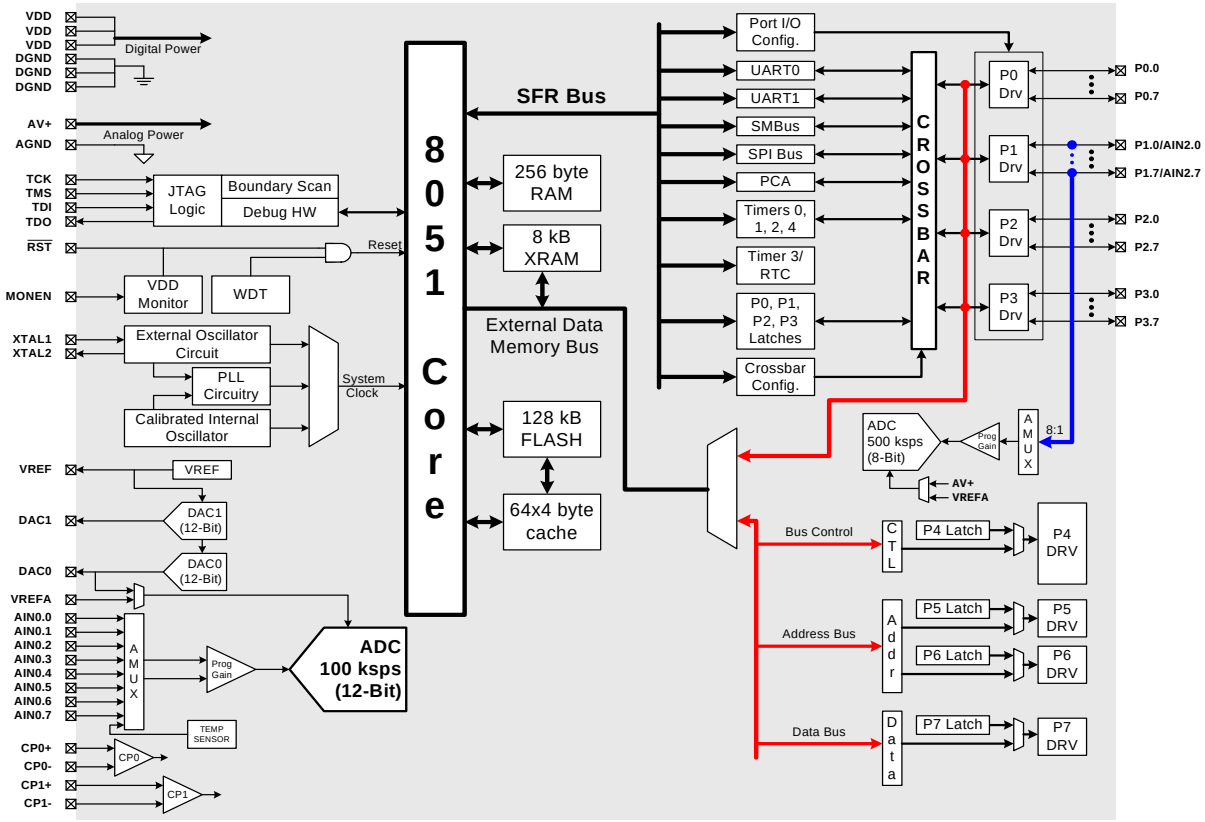



**Figure 1.2. C8051F121/125 Block Diagram**

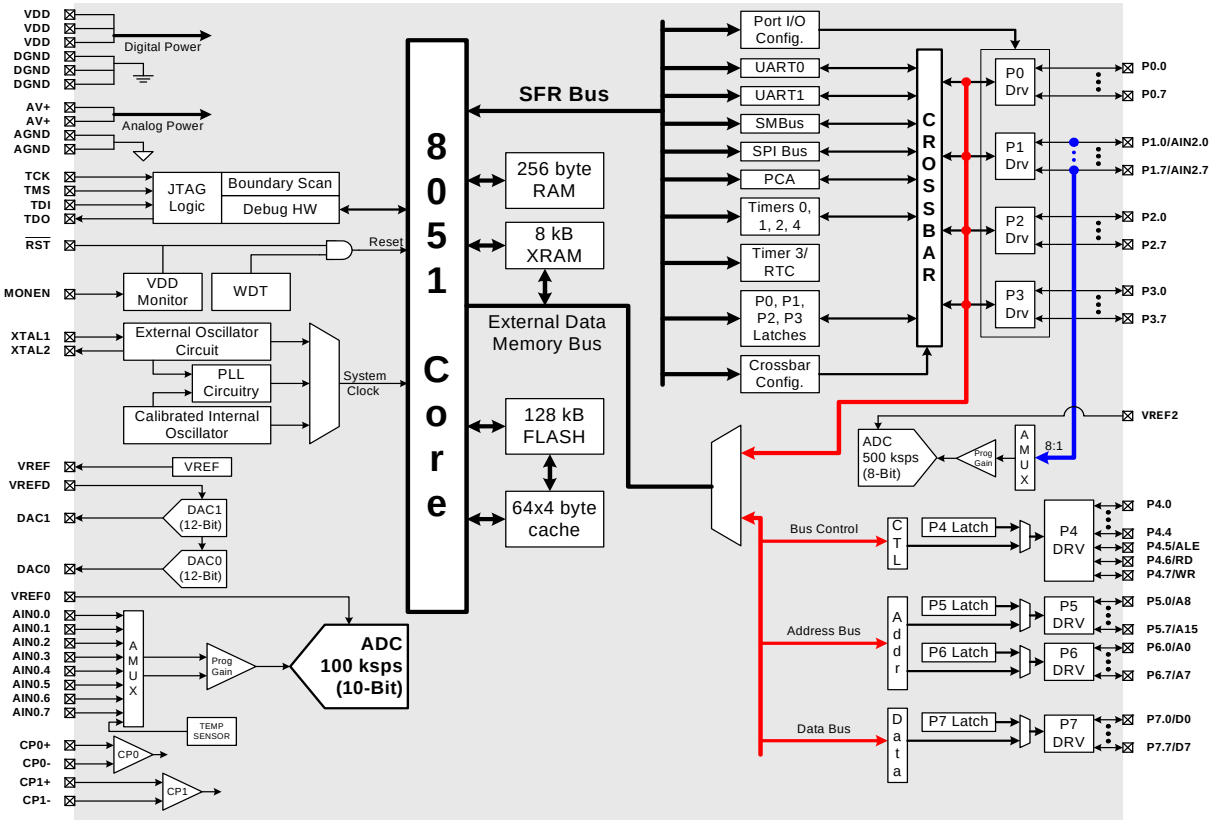



**Figure 1.3. C8051F122/126 Block Diagram**

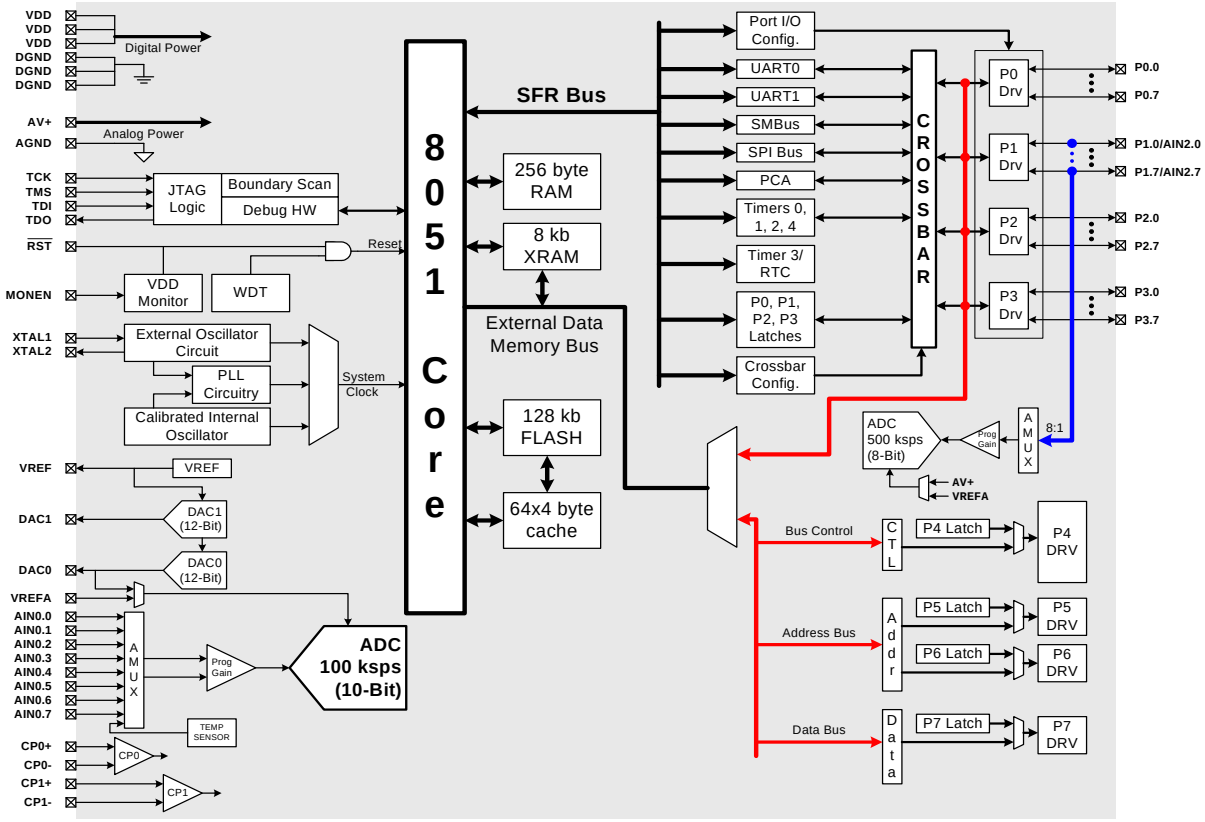



**Figure 1.4. C8051F123/127 Block Diagram**

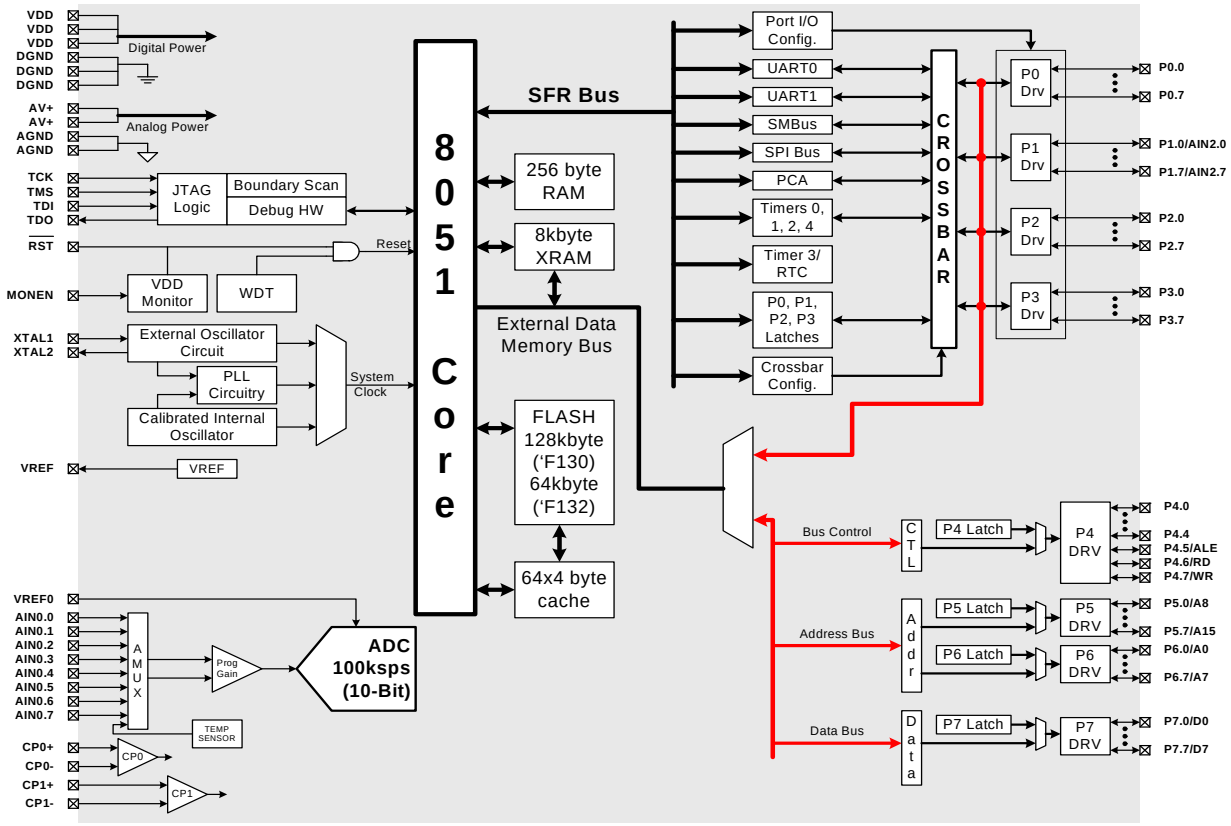



**Figure 1.5. C8051F130/132 Block Diagram**

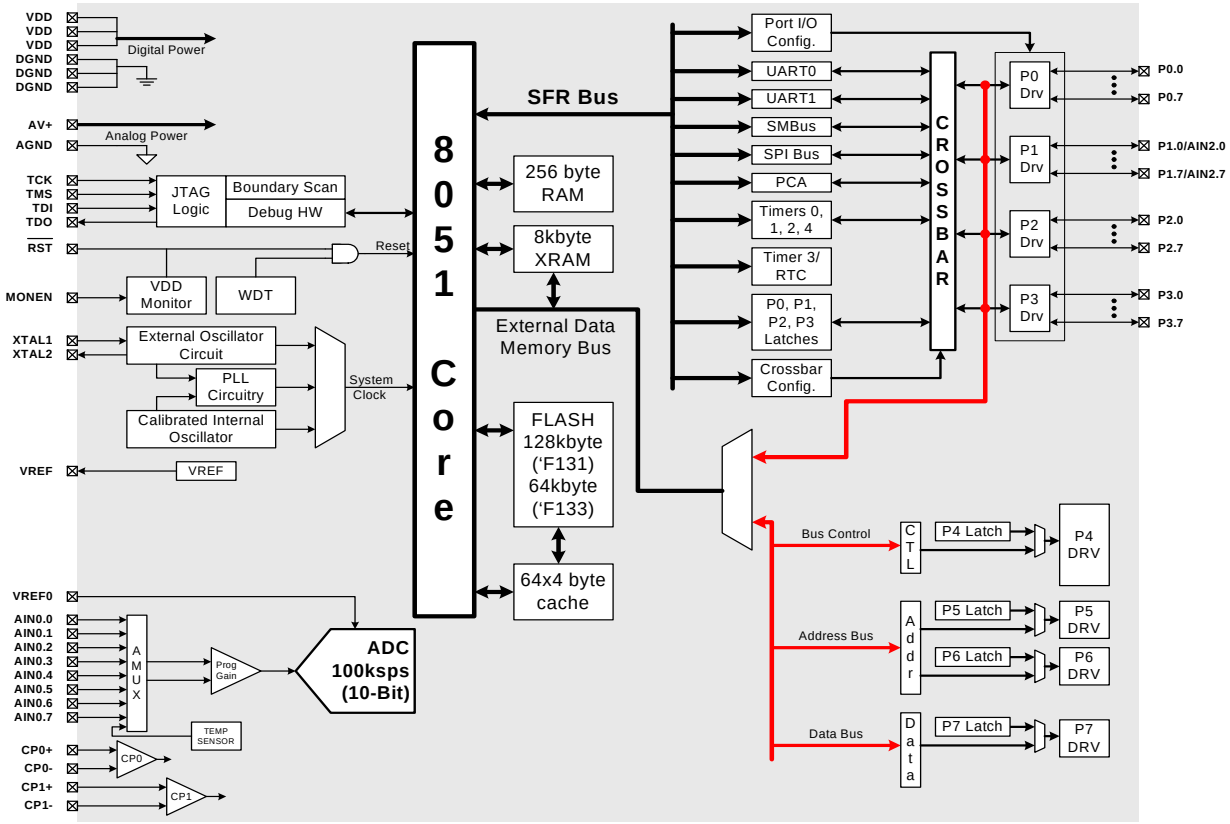



**Figure 1.6. C8051F131/133 Block Diagram**

## 1.1. CIP-51™ Microcontroller Core

### 1.1.1. Fully 8051 Compatible

The C8051F12x and C8051F13x utilize 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 core has all the peripherals included with a standard 8052, including five 16-bit counter/timers, two full-duplex UARTs, 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space, and 8/4 byte-wide I/O Ports.

### 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-to-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 100 MHz, the C8051F120/1/2/3 and C8051F130/1/2/3 have a peak throughput of 100 MIPS (the C8051F124/5/6/7 have a peak throughput of 50 MIPS).

### 1.1.3. Additional Features

Several key enhancements are implemented in the CIP-51 core and peripherals to improve overall performance and ease of use in end applications.

The extended interrupt handler provides 20 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing the 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.

There are up to seven reset sources for the MCU: an on-board $V_{DD}$ monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator0, a forced software reset, the CNVSTR0 input pin, and the $\overline{RST}$ pin. The $\overline{RST}$ pin is bi-directional, accommodating an external reset, or allowing the internally generated POR to be output on the $\overline{RST}$ pin. Each reset source except for the $V_{DD}$ monitor and Reset Input pin may be disabled by the user in software; the $V_{DD}$ monitor is enabled/disabled via the MONEN pin. The Watchdog Timer may be permanently enabled in software after a power-on reset during MCU initialization.

The MCU has an internal, stand alone clock generator which is used by default as the system clock after any reset. If desired, the clock source may be switched on the fly to the external oscillator, which can use a crystal, ceramic resonator, capacitor, RC, or external clock source to generate the system clock. This can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the 24.5 MHz internal oscillator as needed. Additionally, an on-chip PLL is provided to achieve higher system clock speeds for increased throughput.

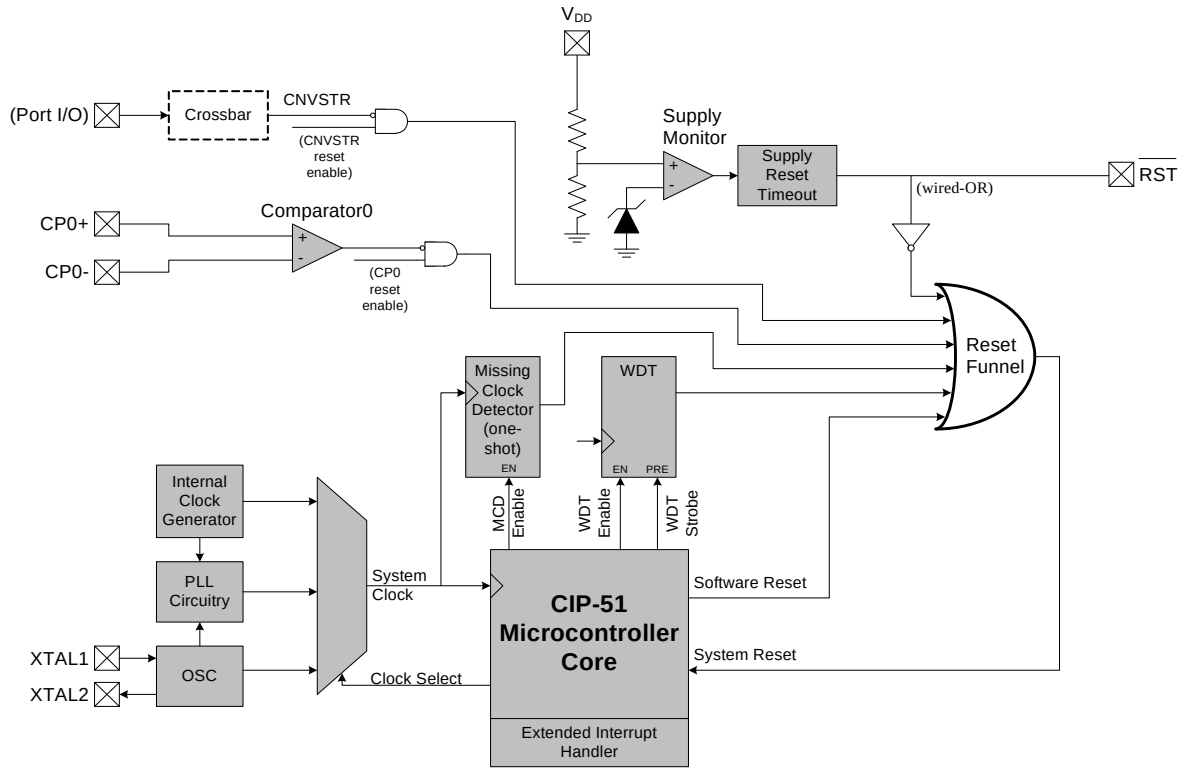



**Figure 1.7. On-Board Clock and Reset**

## 1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

The devices include an on-chip 8k byte RAM block and an external memory interface (EMIF) for accessing off-chip data memory. The on-chip 8k byte block can be addressed over the entire 64k external data memory address range (overlapping 8k boundaries). External data memory address space can be mapped to on-chip memory only, off-chip memory only, or a combination of the two (addresses up to 8k directed to on-chip, above 8k directed to EMIF). The EMIF is also configurable for multiplexed or non-multiplexed address/data lines.

On the C8051F12x and C8051F130/1, the MCU's program memory consists of 128 k bytes of banked Flash memory. The 1024 bytes from addresses 0x1FC00 to 0x1FFFF are reserved. On the C8051F132/3, the MCU's program memory consists of 64 k bytes of Flash memory. This memory may be reprogrammed in-system in 1024 byte sectors, and requires no special off-chip programming voltage.

On all devices, there are also two 128 byte sectors at addresses 0x20000 to 0x200FF, which may be used by software for data storage. See Figure 1.8 for the MCU system memory map.

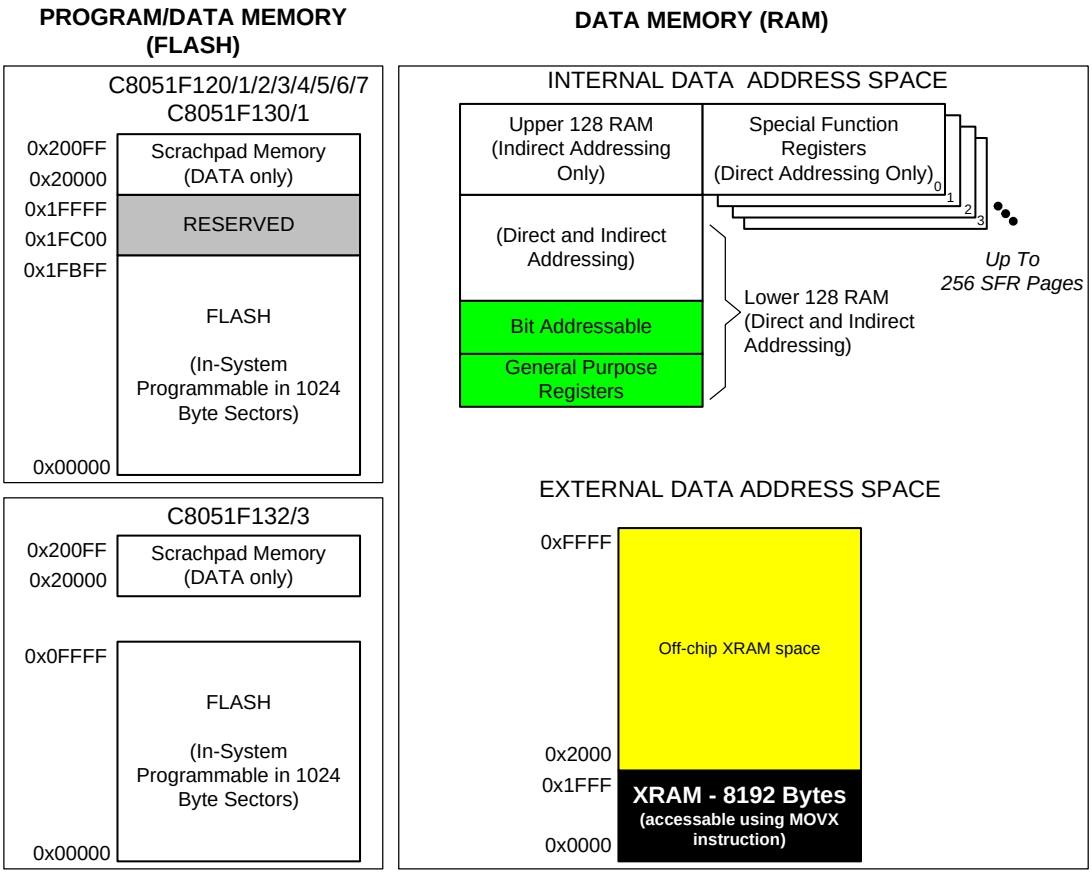



**Figure 1.8. On-Chip Memory Map**

## 1.3. JTAG Debug and Boundary Scan

JTAG boundary scan and debug circuitry is included which provides *non-intrusive, full speed, in-circuit debugging using the production part installed in the end application*, via the four-pin JTAG interface. The JTAG port is fully compliant to IEEE 1149.1, providing full boundary scan for test and manufacturing purposes.

Silicon Labs' debugging system supports inspection and modification of memory and registers, breakpoints, watchpoints, a stack monitor, and single stepping. No additional target RAM, program memory, timers, or communications channels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the ADC and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized.

The C8051F120DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F12x or C8051F13x MCUs.

The kit includes a Windows (95 or later) development environment, a serial adapter for connecting to the JTAG port, and a target application board with a C8051F120 MCU installed. All of the necessary communication cables and a wall-mount power supply are also supplied with the development kit. Silicon Labs' debug environment is a vastly superior configuration for developing and debugging embedded applications compared to standard MCU emulators, which use on-board "ICE Chips" and target cables and require the MCU in the application board to be socketed. Silicon Labs' debug environment both increases ease of use and preserves the performance of the precision, on-chip analog peripherals.

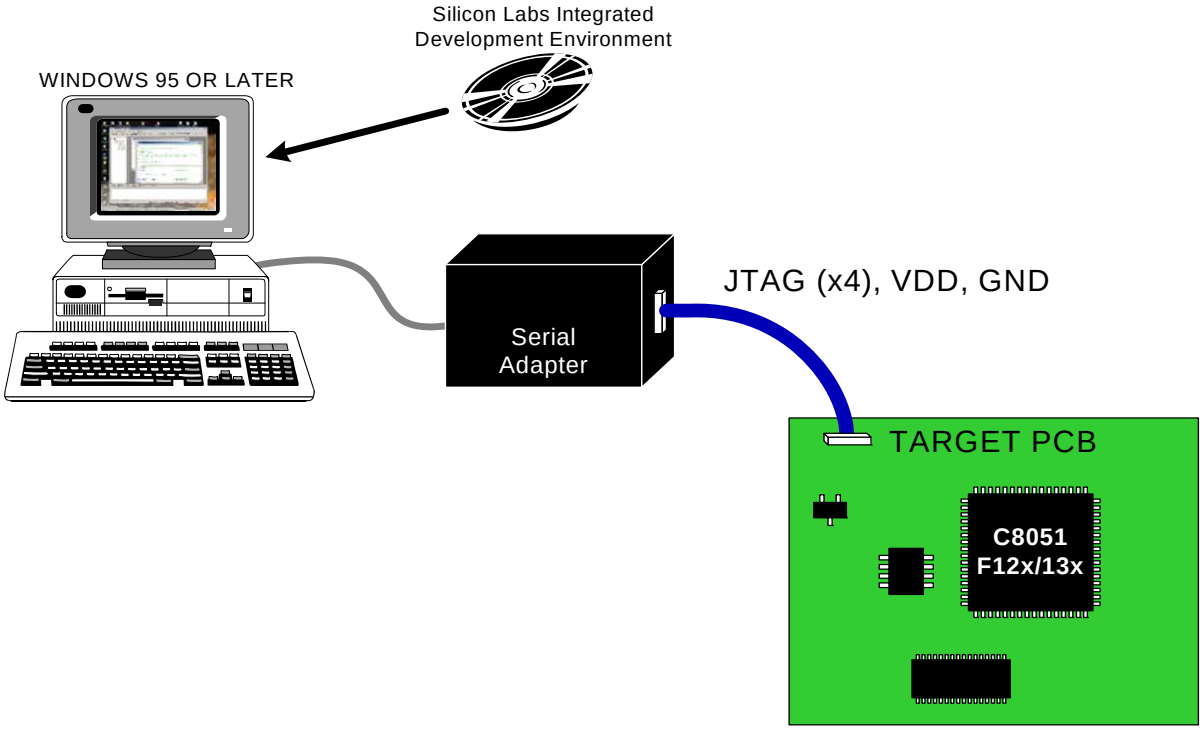



**Figure 1.9. Development/In-System Debug Diagram**

## 1.4. 16 x 16 MAC (Multiply and Accumulate) Engine

The C8051F120/1/2/3 and C8051F130/1/2/3 devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. MAC0 contains a 16-by-16 bit multiplier and a 40-bit adder, which can perform integer or fractional multiply-accumulate and multiply operations on signed input values in two SYSCLK cycles. A rounding engine provides a rounded 16-bit fractional result after an additional (third) SYSCLK cycle. MAC0 also contains a 1-bit arithmetic shifter that will left or right-shift the contents of the 40-bit accumulator in a single SYSCLK cycle.

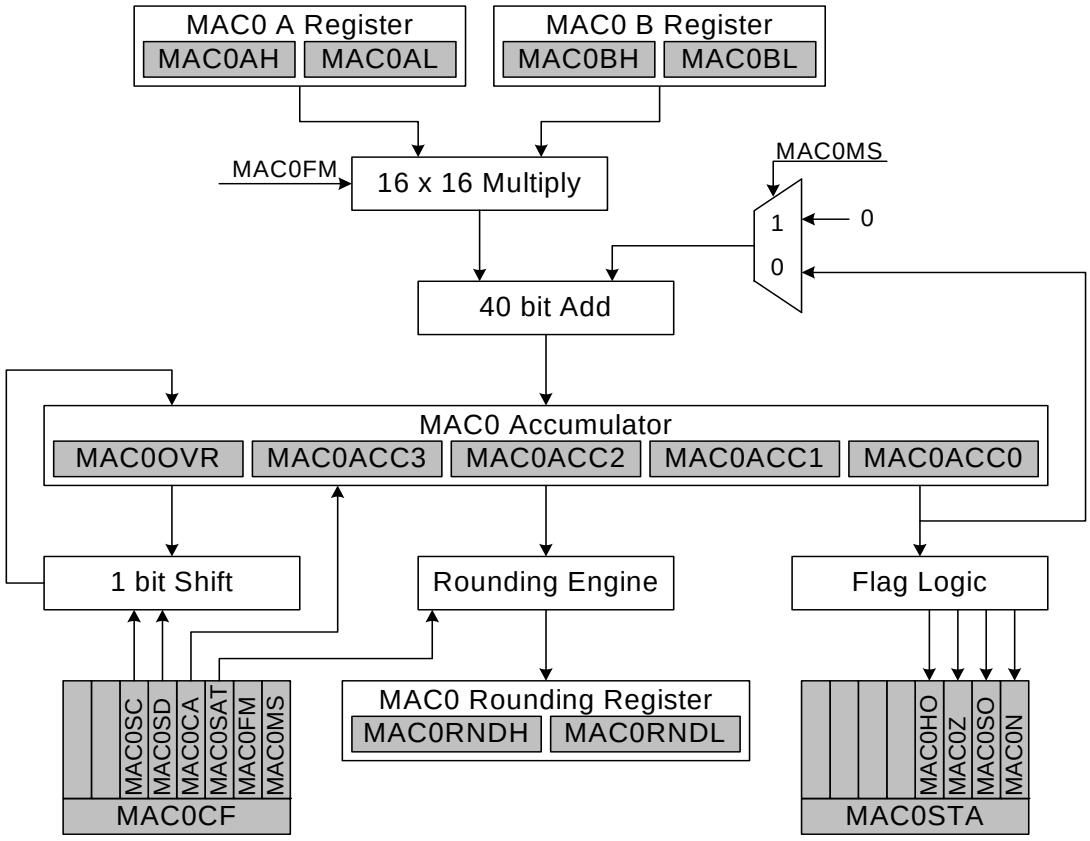



**Figure 1.10. MAC0 Block Diagram**

## 1.5. Programmable Digital I/O and Crossbar

The standard 8051 8-bit Ports (0, 1, 2, and 3) are available on the MCUs. The devices in the larger (100-pin TQFP) packaging have 4 additional ports (4, 5, 6, and 7) for a total of 64 general-purpose port I/O. The Port I/O behave like the standard 8051 with a few enhancements.

Each Port I/O pin can be configured as either a push-pull or open-drain output. Also, the "weak pullups" which are normally fixed on an 8051 can be globally disabled, providing additional power saving capabilities for low-power applications.

Perhaps the most unique enhancement is the Digital Crossbar. This is a large digital switching network that allows mapping of internal digital system resources to Port I/O pins on P0, P1, P2, and P3. (See Figure 1.11) Unlike microcontrollers with standard multiplexed digital I/O, all combinations of functions are supported.

The on-chip counter/timers, serial buses, HW interrupts, ADC Start of Conversion inputs, comparator outputs, and other digital signals in the controller can be configured to appear on the Port I/O pins specified in the Crossbar Control registers. This allows the user to select the exact mix of general purpose Port I/O and digital resources needed for the particular application.

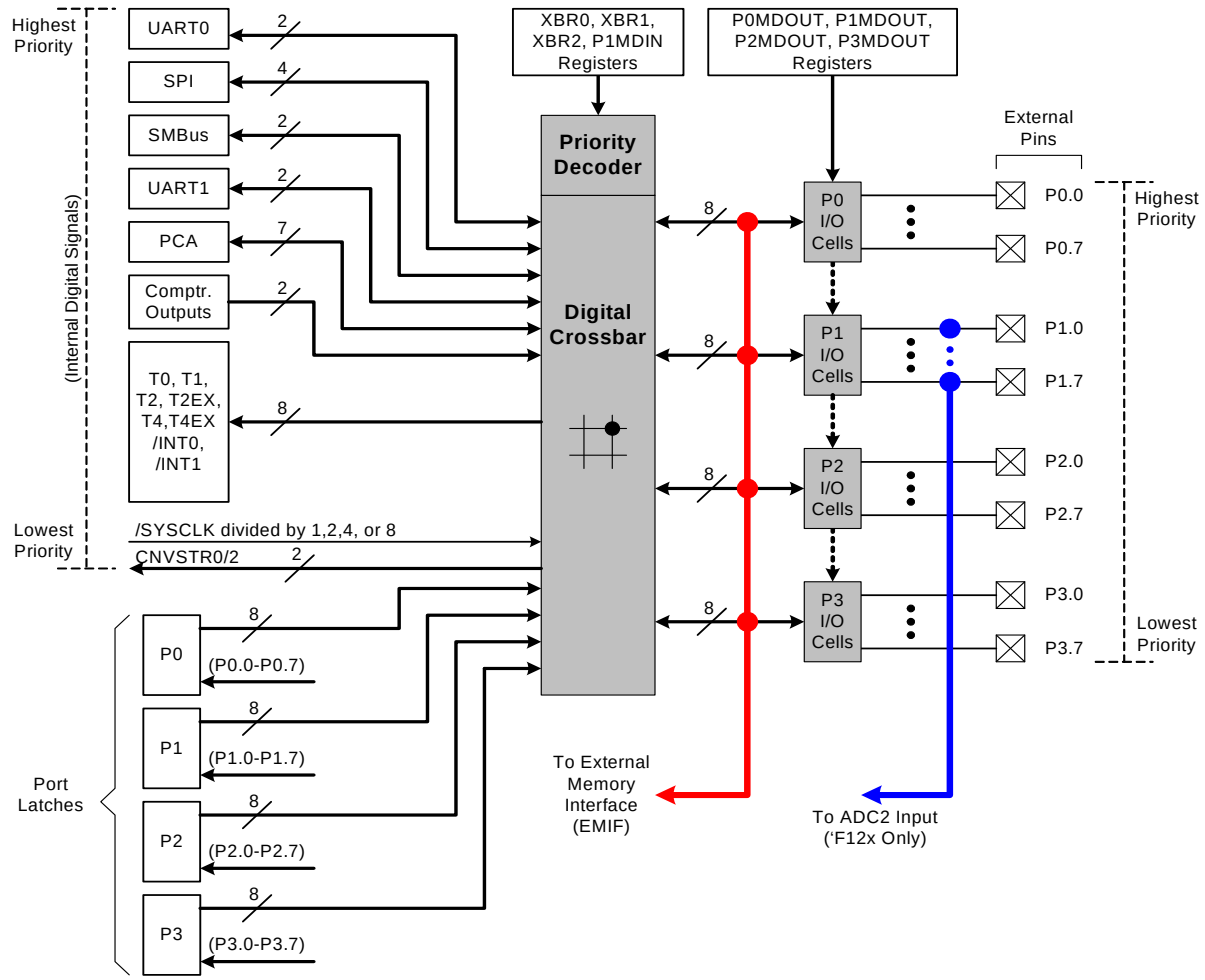



**Figure 1.11. Digital Crossbar Diagram**

## 1.6. Programmable Counter Array

An on-board Programmable Counter/Timer Array (PCA) is included in addition to the five 16-bit general purpose counter/timers. The PCA consists of a dedicated 16-bit counter/timer time base with 6 programmable capture/compare modules. The timebase is clocked from one of six sources: the system clock divided by 12, the system clock divided by 4, Timer 0 overflow, an External Clock Input (ECI pin), the system clock, or the external oscillator source divided by 8.

Each capture/compare module can be configured to operate in one of six modes: Edge-Triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. The PCA Capture/Compare Module I/O and External Clock Input are routed to the MCU Port I/O via the Digital Crossbar.

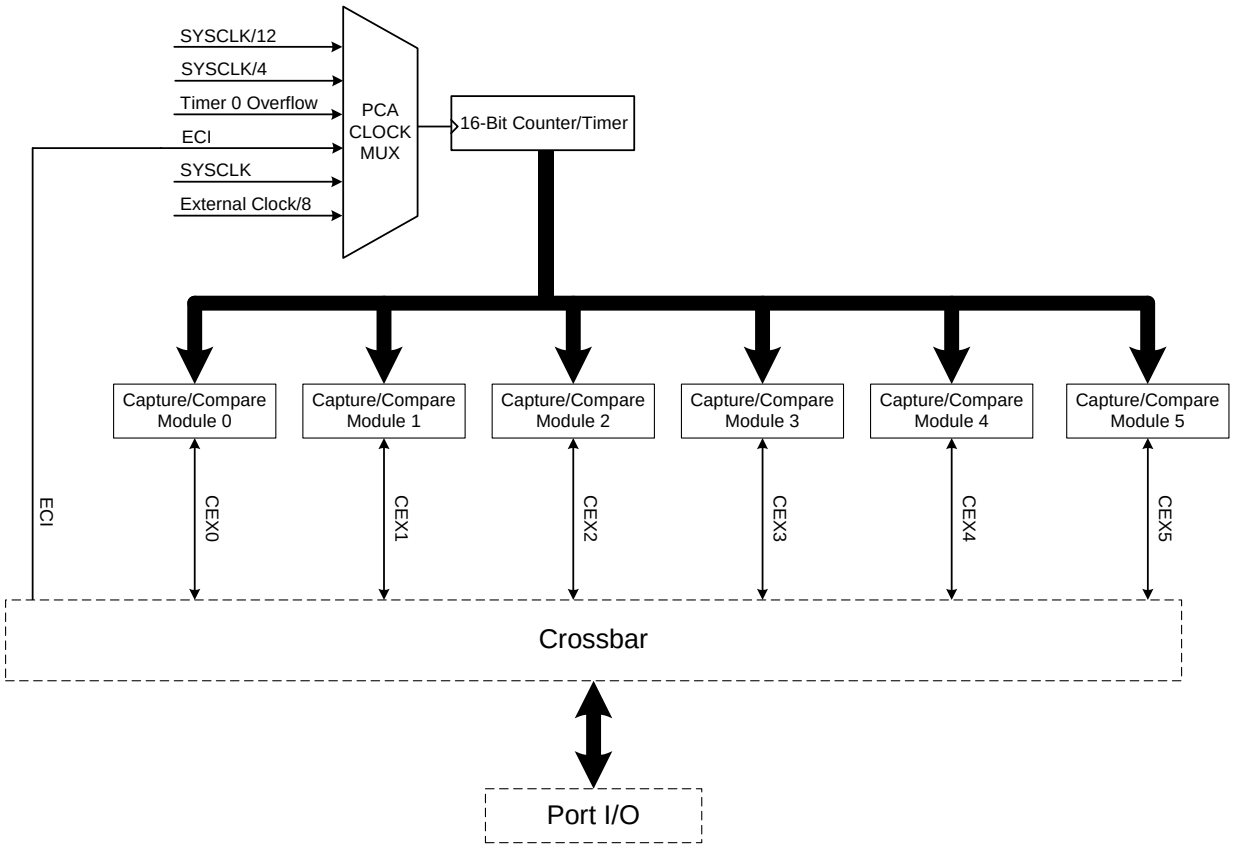



**Figure 1.12. PCA Block Diagram**

## 1.7. Serial Ports

Serial peripherals included on the devices are two Enhanced Full-Duplex UARTs, SPI Bus, and SMBus/I2C. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little intervention by the CPU. The serial buses do not "share" resources such as timers, interrupts, or Port I/O, so any or all of the serial buses may be used together with any other.

## 1.8. 12 or 10-Bit Analog to Digital Converter

All devices include either a 12 or 10-bit SAR ADC (ADC0) with a 9-channel input multiplexer and programmable gain amplifier. With a maximum throughput of 100 ksps, the 12 and 10-bit ADCs offer true 12-bit linearity with an INL of ±1LSB. The ADC0 voltage reference can be selected from an external VREF pin, or (on the C8051F12x devices) the DAC0 output. On the 100-pin TQFP devices, ADC0 has its own dedicated Voltage Reference input pin; on the 64-pin TQFP devices, the ADC0 shares a Voltage Reference input pin with the 8-bit ADC2. The on-chip voltage reference may generate the voltage reference for other system components or the on-chip ADCs via the VREF output pin.

The ADC is under full control of the CIP-51 microcontroller via its associated Special Function Registers. One input channel is tied to an internal temperature sensor, while the other eight channels are available externally. Each pair of the eight external input channels can be configured as either two single-ended inputs or a single differential input. The system controller can also put the ADC into shutdown mode to save power.

A programmable gain amplifier follows the analog multiplexer. The gain can be set in software from 0.5 to 16 in powers of 2. The gain stage can be especially useful when different ADC input channels have widely varied input voltage signals, or when it is necessary to "zoom in" on a signal with a large DC offset (in differential mode, a DAC could be used to provide the DC offset).

Conversions can be started in four ways; a software command, an overflow of Timer 2, an overflow of Timer 3, or an external signal input. This flexibility allows the start of conversion to be triggered by software events, external HW signals, or a periodic timer overflow signal. Conversion completions are indicated by a status bit and an interrupt (if enabled). The resulting 10 or 12-bit data word is latched into two SFRs upon completion of a conversion. The data can be right or left justified in these registers under software control.

Window Compare registers for the ADC data can be configured to interrupt the controller when ADC data is within or outside of a specified range. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within the specified window.

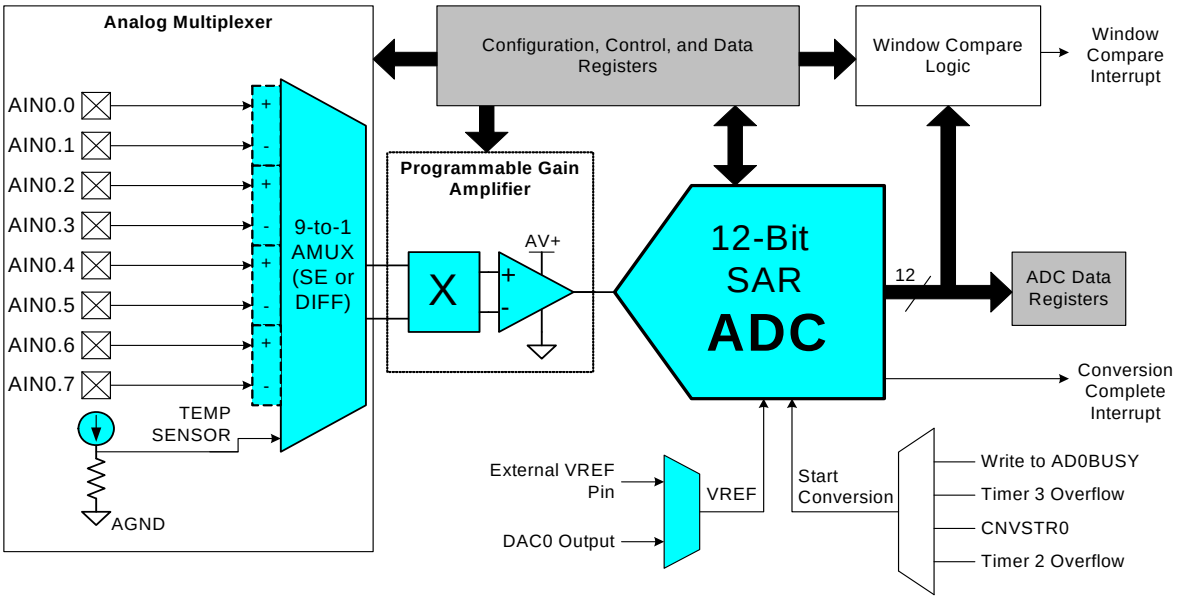



**Figure 1.13. 12-Bit ADC Block Diagram**

## 1.9. 8-Bit Analog to Digital Converter

The C8051F12x devices have an on-board 8-bit SAR ADC (ADC2) with an 8-channel input multiplexer and programmable gain amplifier. This ADC features a 500 ksps maximum throughput and true 8-bit linearity with an INL of ±1LSB. Eight input pins are available for measurement. The ADC is under full control of the CIP-51 microcontroller via the Special Function Registers. The ADC2 voltage reference is selected between the analog power supply (AV+) and an external VREF pin. On the 100-pin TQFP devices, ADC2 has its own dedicated Voltage Reference input pin; on the 64-pin TQFP devices, ADC2 shares a Voltage Reference input pin with ADC0. User software may put ADC2 into shutdown mode to save power.

A programmable gain amplifier follows the analog multiplexer. The gain stage can be especially useful when different ADC input channels have widely varied input voltage signals, or when it is necessary to "zoom in" on a signal with a large DC offset (in differential mode, a DAC could be used to provide the DC offset). The PGA gain can be set in software to 0.5, 1, 2, or 4.

A flexible conversion scheduling system allows ADC2 conversions to be initiated by software commands, timer overflows, or an external input signal. ADC2 conversions may also be synchronized with ADC0 software-commanded conversions. Conversion completions are indicated by a status bit and an interrupt (if enabled), and the resulting 8-bit data word is latched into an SFR upon completion.

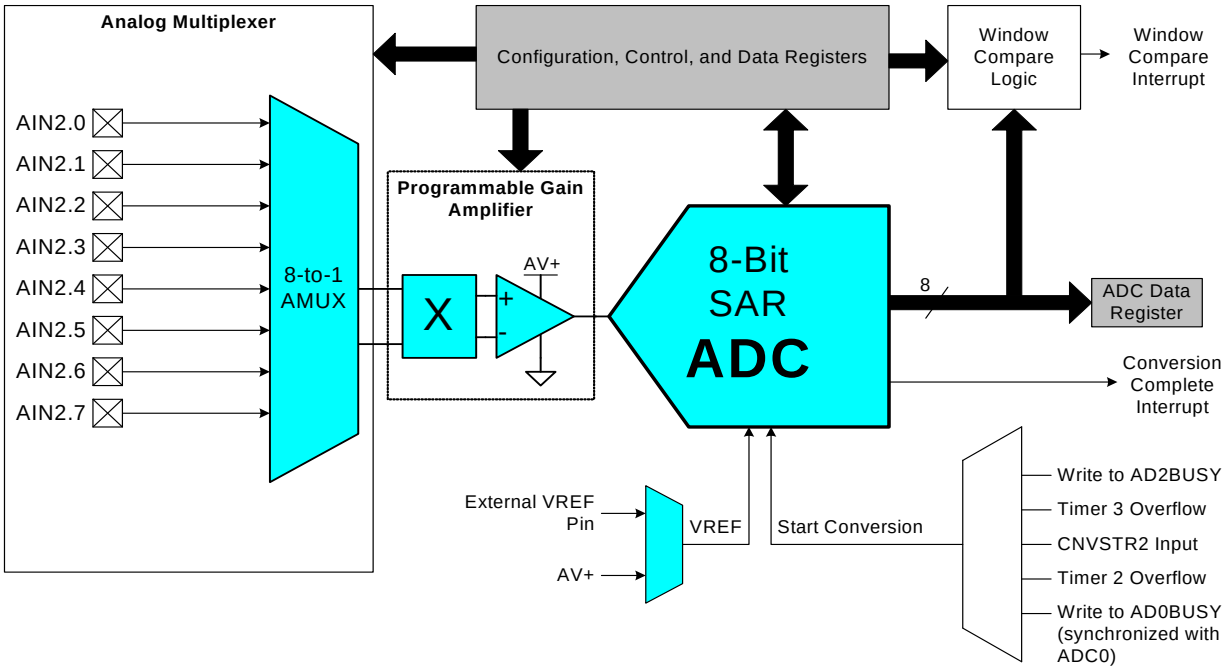



**Figure 1.14. 8-Bit ADC Diagram**

## 1.10. 12-bit Digital to Analog Converters

The C8051F12x devices have two integrated 12-bit Digital to Analog Converters (DACs). The MCU data and control interface to each DAC is via the Special Function Registers. The MCU can place either or both of the DACs in a low power shutdown mode.

The DACs are voltage output mode and include a flexible output scheduling mechanism. This scheduling mechanism allows DAC output updates to be forced by a software write or scheduled on a Timer 2, 3, or 4 overflow. The DAC voltage reference is supplied from the dedicated VREFD input pin on the 100-pin TQFP devices or via the internal Voltage reference on the 64-pin TQFP devices. The DACs are especially useful as references for the comparators or offsets for the differential inputs of the ADCs.

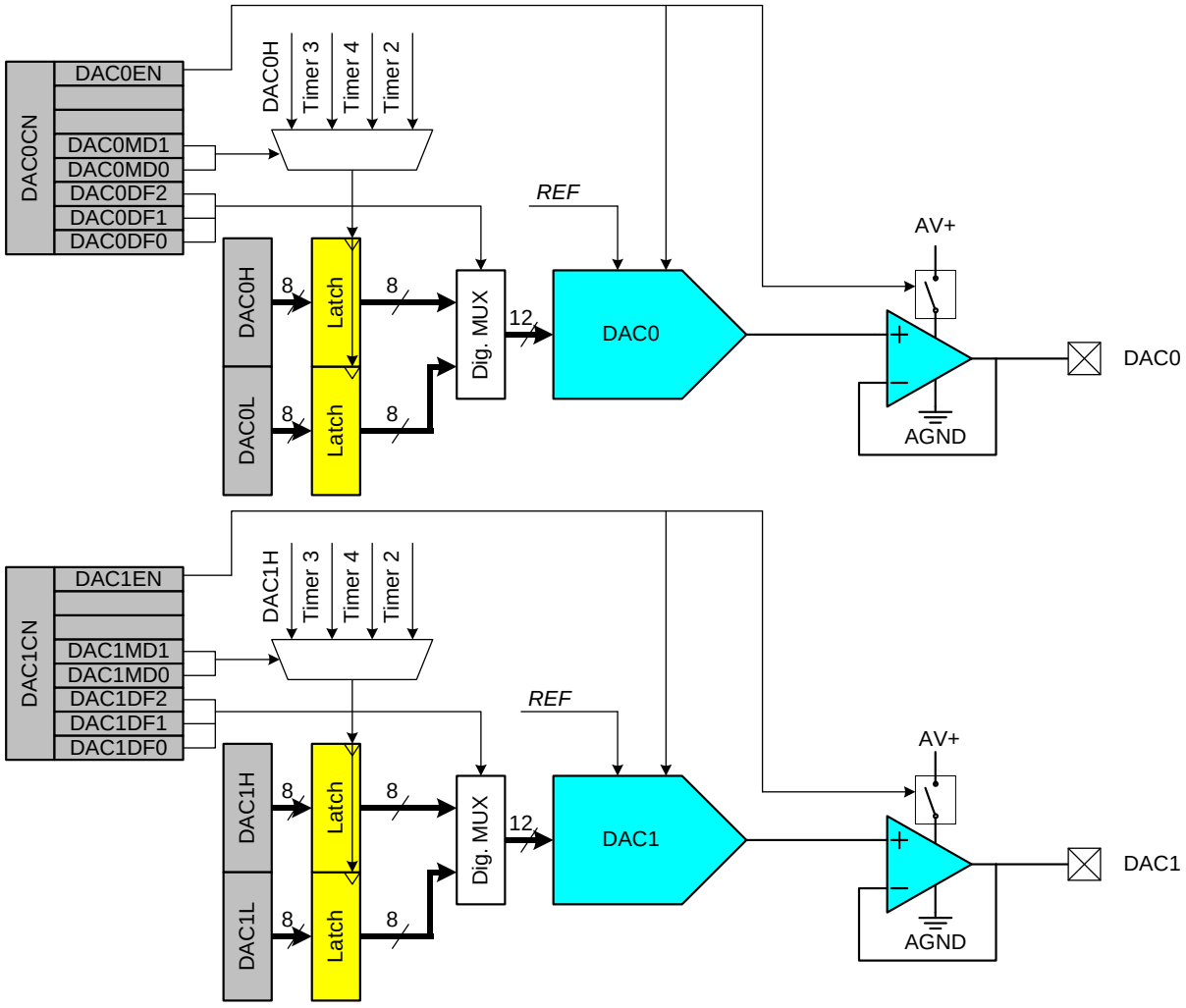



**Figure 1.15. DAC System Block Diagram**

## 1.11. Analog Comparators

Two analog comparators with dedicated input pins are included on-chip. The comparators have software programmable hysteresis and response time. Each comparator can generate an interrupt on a rising edge, falling edge, or both. The interrupts are capable of waking up the MCU from sleep mode, and Comparator 0 can be used as a reset source. The output state of the comparators can be polled in software or routed to Port I/O pins via the Crossbar. The comparators can be programmed to a low power shutdown mode when not in use.

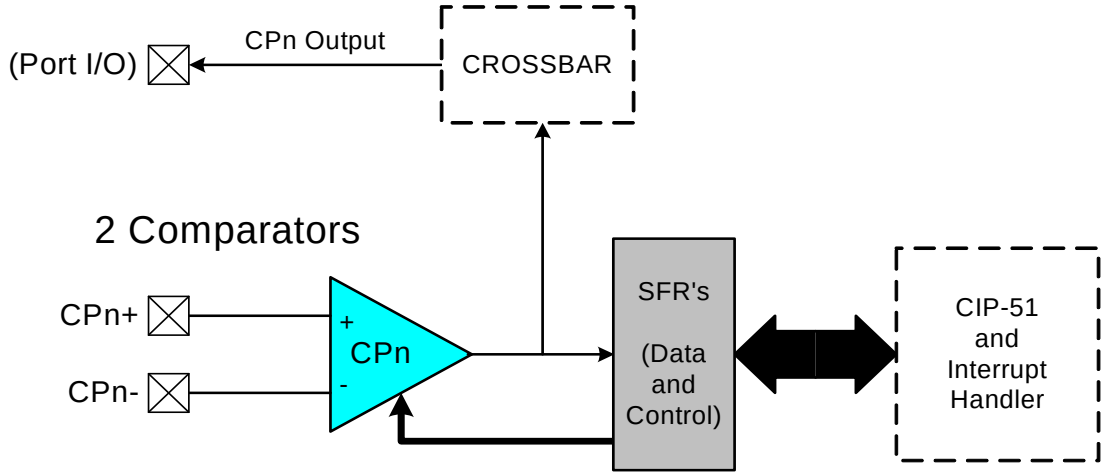



**Figure 1.16. Comparator Block Diagram**

## 2. Absolute Maximum Ratings

**Table 2.1. Absolute Maximum Ratings***

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Ambient temperature under bias | | –55 | — | 125 | °C |
| Storage Temperature | | –65 | — | 150 | °C |
| Voltage on any Pin (except $V_{DD}$ and Port I/O) with Respect to DGND | | –0.3 | — | $V_{DD}$ + 0.3 | V |
| Voltage on any Port I/O Pin or $\overline{RST}$ with Respect to DGND | | –0.3 | — | 5.8 | V |
| Voltage on $V_{DD}$ with Respect to DGND | | –0.3 | — | 4.2 | V |
| Maximum Total Current through $V_{DD}$, AV+, DGND, and AGND | | — | — | 800 | mA |
| Maximum Output Current Sunk by any Port pin | | — | — | 100 | mA |
| Maximum Output Current Sunk by any other I/O pin | | — | — | 50 | mA |
| Maximum Output Current Sourced by any Port pin | | — | — | 100 | mA |
| Maximum Output Current Sourced by any other I/O Pin | | — | — | 50 | 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.

# 3. Global DC Electrical Characteristics

### Table 3.1. Global DC Electrical Characteristics (C8051F120/1/2/3 and C8051F130/1/2/3)

–40 to +85 °C, 100 MHz System Clock unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Analog Supply Voltage[1] | SYSCLK = 0 to 50 MHz<br>SYSCLK > 50 MHz | 2.7<br>3.0 | 3.0<br>3.3 | 3.6<br>3.6 | V<br>V |
| Analog Supply Current | Internal REF, ADCs, DACs, Comparators all active | — | 1.7 | — | mA |
| Analog Supply Current with analog sub-systems inactive | Internal REF, ADCs, DACs, Comparators all disabled, oscillator disabled | — | 0.2 | — | µA |
| Analog-to-Digital Supply Delta ($\lvert V_{DD} - AV+\rvert$) | | — | — | 0.5 | V |
| Digital Supply Voltage | SYSCLK = 0 to 50 MHz<br>SYSCLK > 50 MHz | 2.7<br>3.0 | 3.0<br>3.3 | 3.6<br>3.6 | V<br>V |
| Digital Supply Current with CPU active | $V_{DD}$ = 3.0 V, Clock = 100 MHz<br>$V_{DD}$ = 3.0 V, Clock = 50 MHz<br>$V_{DD}$ = 3.0 V, Clock = 1 MHz<br>$V_{DD}$ = 3.0 V, Clock = 32 kHz | — | 65<br>35<br>1<br>33 | — | mA<br>mA<br>mA<br>µA |
| Digital Supply Current with CPU inactive (not accessing Flash) | $V_{DD}$ = 3.0 V, Clock = 100 MHz<br>$V_{DD}$ = 3.0 V, Clock = 50 MHz<br>$V_{DD}$ = 3.0 V, Clock = 1 MHz<br>$V_{DD}$ = 3.0 V, Clock = 32 kHz | — | 40<br>20<br>0.4<br>15 | — | mA<br>mA<br>mA<br>µA |
| Digital Supply Current (shutdown) | Oscillator not running | — | 0.4 | — | µA |
| Digital Supply RAM Data Retention Voltage | | — | 1.5 | — | V |
| SYSCLK (System Clock)[2,3] | $V_{DD}$, AV+ = 2.7 to 3.6 V<br>$V_{DD}$, AV+ = 3.0 to 3.6 V | 0<br>0 | — | 50<br>100 | MHz<br>MHz |
| Specified Operating Temperature Range | | –40 | — | +85 | °C |

**Notes:**

1. Analog Supply AV+ must be greater than 1 V for $V_{DD}$ monitor to operate.
2. SYSCLK is the internal device clock. For operational speeds in excess of 30 MHz, SYSCLK must be derived from the Phase-Locked Loop (PLL).
3. SYSCLK must be at least 32 kHz to enable debugging.

### Table 3.2. Global DC Electrical Characteristics (C8051F124/5/6/7)

–40 to +85 °C, 50 MHz System Clock unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Analog Supply Voltage[1] | | 2.7 | 3.0 | 3.6 | V |
| Analog Supply Current | Internal REF, ADC, DAC, Comparators all active | — | 1.7 | — | mA |
| Analog Supply Current with analog sub-systems inactive | Internal REF, ADC, DAC, Comparators all disabled, oscillator disabled | — | 0.2 | — | µA |
| Analog-to-Digital Supply Delta ($\|V_{DD}$ – AV+$\|$) | | — | — | 0.5 | V |
| Digital Supply Voltage | | 2.7 | 3.0 | 3.6 | V |
| Digital Supply Current with CPU active | $V_{DD}$ = 3.0 V, Clock = 50 MHz<br>$V_{DD}$ = 3.0 V, Clock = 1 MHz<br>$V_{DD}$ = 3.0 V, Clock = 32 kHz | — | 35<br>1<br>33 | — | mA<br>mA<br>µA |
| Digital Supply Current with CPU inactive (not accessing Flash) | $V_{DD}$ = 3.0 V, Clock = 50 MHz<br>$V_{DD}$ = 3.0 V, Clock = 1 MHz<br>$V_{DD}$ = 3.0 V, Clock = 32 kHz | — | 27<br>0.4<br>15 | — | mA<br>mA<br>µA |
| Digital Supply Current (shutdown) | Oscillator not running | — | 0.4 | — | µA |
| Digital Supply RAM Data Retention Voltage | | — | 1.5 | — | V |
| SYSCLK (System Clock)[2,3] | | 0 | — | 50 | MHz |
| Specified Operating Temperature Range | | –40 | — | +85 | °C |

**Notes:**

1. Analog Supply AV+ must be greater than 1 V for $V_{DD}$ monitor to operate.
2. SYSCLK is the internal device clock. For operational speeds in excess of 30 MHz, SYSCLK must be derived from the phase-locked loop (PLL).
3. SYSCLK must be at least 32 kHz to enable debugging.

# 4. Pinout and Package Definitions

**Table 4.1. Pin Definitions**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | **'F120 'F122 'F124 'F126** | **'F121 'F123 'F125 'F127** | **'F130 'F132** | **'F131 'F133** | | |
| $V_{DD}$ | 37, 64, 90 | 24, 41, 57 | 37, 64, 90 | 24, 41, 57 | | Digital Supply Voltage. Must be tied to +2.7 to +3.6 V. |
| DGND | 38, 63, 89 | 25, 40, 56 | 38, 63, 89 | 25, 40, 56 | | Digital Ground. Must be tied to Ground. |
| AV+ | 11, 14 | 6 | 11, 14 | 6 | | Analog Supply Voltage. Must be tied to +2.7 to +3.6 V. |
| AGND | 10, 13 | 5 | 10, 13 | 5 | | Analog Ground. Must be tied to Ground. |
| TMS | 1 | 58 | 1 | 58 | D In | JTAG Test Mode Select with internal pullup. |
| TCK | 2 | 59 | 2 | 59 | D In | JTAG Test Clock with internal pullup. |
| TDI | 3 | 60 | 3 | 60 | D In | JTAG Test Data Input with internal pullup. TDI is latched on the rising edge of TCK. |
| TDO | 4 | 61 | 4 | 61 | D Out | JTAG Test Data Output with internal pullup. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver. |
| $\overline{RST}$ | 5 | 62 | 5 | 62 | D I/O | Device Reset. Open-drain output of internal $V_{DD}$ monitor. Is driven low when $V_{DD}$ is < $V_{RST}$ and MONEN is high. An external source can initiate a system reset by driving this pin low. |
| XTAL1 | 26 | 17 | 26 | 17 | A In | Crystal Input. This pin is the return for the internal oscillator circuit for a crystal or ceramic resonator. For a precision internal clock, connect a crystal or ceramic resonator from XTAL1 to XTAL2. If overdriven by an external CMOS clock, this becomes the system clock. |
| XTAL2 | 27 | 18 | 27 | 18 | A Out | Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator. |
| MONEN | 28 | 19 | 28 | 19 | D In | $V_{DD}$ Monitor Enable. When tied high, this pin enables the internal $V_{DD}$ monitor, which forces a system reset when $V_{DD}$ is < $V_{RST}$. When tied low, the internal $V_{DD}$ monitor is disabled. **This pin must be tied high or low.** |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | ‘F120 ‘F122 ‘F124 ‘F126 | ‘F121 ‘F123 ‘F125 ‘F127 | ‘F130 ‘F132 | ‘F131 ‘F133 | | |
| VREF | 12 | 7 | 12 | 7 | A I/O | Bandgap Voltage Reference Output (all devices).<br>DAC Voltage Reference Input (C8051F121/3/5/7 only). |
| VREFA | | 8 | | | A In | ADC0 and ADC2 Voltage Reference Input. |
| VREF0 | 16 | | 16 | 8 | A In | ADC0 Voltage Reference Input. |
| VREF2 | 17 | | 17 | | A In | ADC2 Voltage Reference Input. |
| VREFD | 15 | | 15 | | A In | DAC Voltage Reference Input. |
| AIN0.0 | 18 | 9 | 18 | 9 | A In | ADC0 Input Channel 0 (See ADC0 Specification for complete description). |
| AIN0.1 | 19 | 10 | 19 | 10 | A In | ADC0 Input Channel 1 (See ADC0 Specification for complete description). |
| AIN0.2 | 20 | 11 | 20 | 11 | A In | ADC0 Input Channel 2 (See ADC0 Specification for complete description). |
| AIN0.3 | 21 | 12 | 21 | 12 | A In | ADC0 Input Channel 3 (See ADC0 Specification for complete description). |
| AIN0.4 | 22 | 13 | 22 | 13 | A In | ADC0 Input Channel 4 (See ADC0 Specification for complete description). |
| AIN0.5 | 23 | 14 | 23 | 14 | A In | ADC0 Input Channel 5 (See ADC0 Specification for complete description). |
| AIN0.6 | 24 | 15 | 24 | 15 | A In | ADC0 Input Channel 6 (See ADC0 Specification for complete description). |
| AIN0.7 | 25 | 16 | 25 | 16 | A In | ADC0 Input Channel 7 (See ADC0 Specification for complete description). |
| CP0+ | 9 | 4 | 9 | 4 | A In | Comparator 0 Non-Inverting Input. |
| CP0- | 8 | 3 | 8 | 3 | A In | Comparator 0 Inverting Input. |
| CP1+ | 7 | 2 | 7 | 2 | A In | Comparator 1 Non-Inverting Input. |
| CP1– | 6 | 1 | 6 | 1 | A In | Comparator 1 Inverting Input. |
| DAC0 | 100 | 64 | | | A Out | Digital to Analog Converter 0 Voltage Output. (See DAC Specification for complete description). |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | ‘F120 ‘F122 ‘F124 ‘F126 | ‘F121 ‘F123 ‘F125 ‘F127 | ‘F130 ‘F132 | ‘F131 ‘F133 | | |
| DAC1 | 99 | 63 | | | A Out | Digital to Analog Converter 1 Voltage Output. (See DAC Specification for complete description). |
| P0.0 | 62 | 55 | 62 | 55 | D I/O | Port 0.0. See Port Input/Output section for complete description. |
| P0.1 | 61 | 54 | 61 | 54 | D I/O | Port 0.1. See Port Input/Output section for complete description. |
| P0.2 | 60 | 53 | 60 | 53 | D I/O | Port 0.2. See Port Input/Output section for complete description. |
| P0.3 | 59 | 52 | 59 | 52 | D I/O | Port 0.3. See Port Input/Output section for complete description. |
| P0.4 | 58 | 51 | 58 | 51 | D I/O | Port 0.4. See Port Input/Output section for complete description. |
| ALE/P0.5 | 57 | 50 | 57 | 50 | D I/O | ALE Strobe for External Memory Address bus (multiplexed mode)<br>Port 0.5<br>See Port Input/Output section for complete description. |
| $\overline{RD}$/P0.6 | 56 | 49 | 56 | 49 | D I/O | /RD Strobe for External Memory Address bus<br>Port 0.6<br>See Port Input/Output section for complete description. |
| $\overline{WR}$/P0.7 | 55 | 48 | 55 | 48 | D I/O | /WR Strobe for External Memory Address bus<br>Port 0.7<br>See Port Input/Output section for complete description. |
| AIN2.0/A8/P1.0 | 36 | 29 | 36 | 29 | A In<br>D I/O | ADC2 Input Channel 0 (See ADC2 Specification for complete description).<br>Bit 8 External Memory Address bus (Non-multiplexed mode)<br>Port 1.0<br>See Port Input/Output section for complete description. |
| AIN2.1/A9/P1.1 | 35 | 28 | 35 | 28 | A In<br>D I/O | Port 1.1. See Port Input/Output section for complete description. |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | ‘F120 ‘F122 ‘F124 ‘F126 | ‘F121 ‘F123 ‘F125 ‘F127 | ‘F130 ‘F132 | ‘F131 ‘F133 | | |
| AIN2.2/A10/P1.2 | 34 | 27 | 34 | 27 | A In D I/O | Port 1.2. See Port Input/Output section for complete description. |
| AIN2.3/A11/P1.3 | 33 | 26 | 33 | 26 | A In D I/O | Port 1.3. See Port Input/Output section for complete description. |
| AIN2.4/A12/P1.4 | 32 | 23 | 32 | 23 | A In D I/O | Port 1.4. See Port Input/Output section for complete description. |
| AIN2.5/A13/P1.5 | 31 | 22 | 31 | 22 | A In D I/O | Port 1.5. See Port Input/Output section for complete description. |
| AIN2.6/A14/P1.6 | 30 | 21 | 30 | 21 | A In D I/O | Port 1.6. See Port Input/Output section for complete description. |
| AIN2.7/A15/P1.7 | 29 | 20 | 29 | 20 | A In D I/O | Port 1.7. See Port Input/Output section for complete description. |
| A8m/A0/P2.0 | 46 | 37 | 46 | 37 | D I/O | Bit 8 External Memory Address bus (Multiplexed mode)<br>Bit 0 External Memory Address bus (Non-multiplexed mode)<br>Port 2.0<br>See Port Input/Output section for complete description. |
| A9m/A1/P2.1 | 45 | 36 | 45 | 36 | D I/O | Port 2.1. See Port Input/Output section for complete description. |
| A10m/A2/P2.2 | 44 | 35 | 44 | 35 | D I/O | Port 2.2. See Port Input/Output section for complete description. |
| A11m/A3/P2.3 | 43 | 34 | 43 | 34 | D I/O | Port 2.3. See Port Input/Output section for complete description. |
| A12m/A4/P2.4 | 42 | 33 | 42 | 33 | D I/O | Port 2.4. See Port Input/Output section for complete description. |
| A13m/A5/P2.5 | 41 | 32 | 41 | 32 | D I/O | Port 2.5. See Port Input/Output section for complete description. |
| A14m/A6/P2.6 | 40 | 31 | 40 | 31 | D I/O | Port 2.6. See Port Input/Output section for complete description. |
| A15m/A7/P2.7 | 39 | 30 | 39 | 30 | D I/O | Port 2.7. See Port Input/Output section for complete description. |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | ‘F120 ‘F122 ‘F124 ‘F126 | ‘F121 ‘F123 ‘F125 ‘F127 | ‘F130 ‘F132 | ‘F131 ‘F133 | | |
| AD0/D0/P3.0 | 54 | 47 | 54 | 47 | D I/O | Bit 0 External Memory Address/Data bus (Multiplexed mode)<br>Bit 0 External Memory Data bus (Non-multiplexed mode)<br>Port 3.0<br>See Port Input/Output section for complete description. |
| AD1/D1/P3.1 | 53 | 46 | 53 | 46 | D I/O | Port 3.1. See Port Input/Output section for complete description. |
| AD2/D2/P3.2 | 52 | 45 | 52 | 45 | D I/O | Port 3.2. See Port Input/Output section for complete description. |
| AD3/D3/P3.3 | 51 | 44 | 51 | 44 | D I/O | Port 3.3. See Port Input/Output section for complete description. |
| AD4/D4/P3.4 | 50 | 43 | 50 | 43 | D I/O | Port 3.4. See Port Input/Output section for complete description. |
| AD5/D5/P3.5 | 49 | 42 | 49 | 42 | D I/O | Port 3.5. See Port Input/Output section for complete description. |
| AD6/D6/P3.6 | 48 | 39 | 48 | 39 | D I/O | Port 3.6. See Port Input/Output section for complete description. |
| AD7/D7/P3.7 | 47 | 38 | 47 | 38 | D I/O | Port 3.7. See Port Input/Output section for complete description. |
| P4.0 | 98 | | 98 | | D I/O | Port 4.0. See Port Input/Output section for complete description. |
| P4.1 | 97 | | 97 | | D I/O | Port 4.1. See Port Input/Output section for complete description. |
| P4.2 | 96 | | 96 | | D I/O | Port 4.2. See Port Input/Output section for complete description. |
| P4.3 | 95 | | 95 | | D I/O | Port 4.3. See Port Input/Output section for complete description. |
| P4.4 | 94 | | 94 | | D I/O | Port 4.4. See Port Input/Output section for complete description. |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | ‘F120 ‘F122 ‘F124 ‘F126 | ‘F121 ‘F123 ‘F125 ‘F127 | ‘F130 ‘F132 | ‘F131 ‘F133 | | |
| ALE/P4.5 | 93 | | 93 | | D I/O | ALE Strobe for External Memory Address bus (multiplexed mode)<br>Port 4.5<br>See Port Input/Output section for complete description. |
| $\overline{RD}$/P4.6 | 92 | | 92 | | D I/O | /RD Strobe for External Memory Address bus<br>Port 4.6<br>See Port Input/Output section for complete description. |
| $\overline{WR}$/P4.7 | 91 | | 91 | | D I/O | /WR Strobe for External Memory Address bus<br>Port 4.7<br>See Port Input/Output section for complete description. |
| A8/P5.0 | 88 | | 88 | | D I/O | Bit 8 External Memory Address bus (Non-multiplexed mode)<br>Port 5.0<br>See Port Input/Output section for complete description. |
| A9/P5.1 | 87 | | 87 | | D I/O | Port 5.1. See Port Input/Output section for complete description. |
| A10/P5.2 | 86 | | 86 | | D I/O | Port 5.2. See Port Input/Output section for complete description. |
| A11/P5.3 | 85 | | 85 | | D I/O | Port 5.3. See Port Input/Output section for complete description. |
| A12/P5.4 | 84 | | 84 | | D I/O | Port 5.4. See Port Input/Output section for complete description. |
| A13/P5.5 | 83 | | 83 | | D I/O | Port 5.5. See Port Input/Output section for complete description. |
| A14/P5.6 | 82 | | 82 | | D I/O | Port 5.6. See Port Input/Output section for complete description. |
| A15/P5.7 | 81 | | 81 | | D I/O | Port 5.7. See Port Input/Output section for complete description. |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | ‘F120 ‘F122 ‘F124 ‘F126 | ‘F121 ‘F123 ‘F125 ‘F127 | ‘F130 ‘F132 | ‘F131 ‘F133 | | |
| A8m/A0/P6.0 | 80 | | 80 | | D I/O | Bit 8 External Memory Address bus (Multiplexed mode)<br>Bit 0 External Memory Address bus (Non-multiplexed mode)<br>Port 6.0<br>See Port Input/Output section for complete description. |
| A9m/A1/P6.1 | 79 | | 79 | | D I/O | Port 6.1. See Port Input/Output section for complete description. |
| A10m/A2/P6.2 | 78 | | 78 | | D I/O | Port 6.2. See Port Input/Output section for complete description. |
| A11m/A3/P6.3 | 77 | | 77 | | D I/O | Port 6.3. See Port Input/Output section for complete description. |
| A12m/A4/P6.4 | 76 | | 76 | | D I/O | Port 6.4. See Port Input/Output section for complete description. |
| A13m/A5/P6.5 | 75 | | 75 | | D I/O | Port 6.5. See Port Input/Output section for complete description. |
| A14m/A6/P6.6 | 74 | | 74 | | D I/O | Port 6.6. See Port Input/Output section for complete description. |
| A15m/A7/P6.7 | 73 | | 73 | | D I/O | Port 6.7. See Port Input/Output section for complete description. |
| AD0/D0/P7.0 | 72 | | 72 | | D I/O | Bit 0 External Memory Address/Data bus (Multiplexed mode)<br>Bit 0 External Memory Data bus (Non-multiplexed mode)<br>Port 7.0<br>See Port Input/Output section for complete description. |
| AD1/D1/P7.1 | 71 | | 71 | | D I/O | Port 7.1. See Port Input/Output section for complete description. |
| AD2/D2/P7.2 | 70 | | 70 | | D I/O | Port 7.2. See Port Input/Output section for complete description. |
| AD3/D3/P7.3 | 69 | | 69 | | D I/O | Port 7.3. See Port Input/Output section for complete description. |
| AD4/D4/P7.4 | 68 | | 68 | | D I/O | Port 7.4. See Port Input/Output section for complete description. |

**Table 4.1. Pin Definitions (Continued)**

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | **'F120 'F122 'F124 'F126** | **'F121 'F123 'F125 'F127** | **'F130 'F132** | **'F131 'F133** | | |
| AD5/D5/P7.5 | 67 | | 67 | | D I/O | Port 7.5. See Port Input/Output section for complete description. |
| AD6/D6/P7.6 | 66 | | 66 | | D I/O | Port 7.6. See Port Input/Output section for complete description. |
| AD7/D7/P7.7 | 65 | | 65 | | D I/O | Port 7.7. See Port Input/Output section for complete description. |
| NC | | | 15, 17, 99, 100 | 63, 64 | | No Connection. |

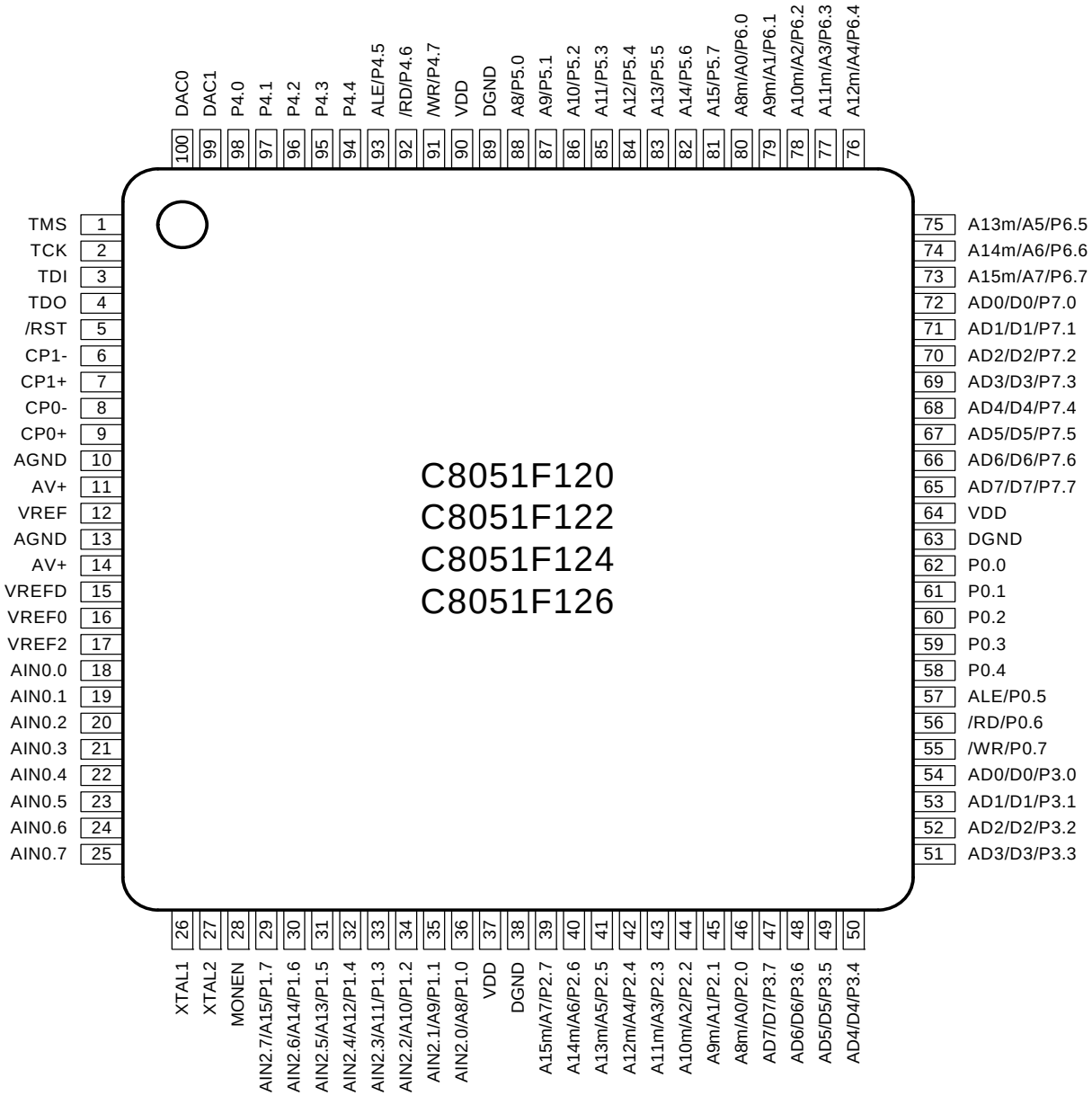



**Figure 4.1. C8051F120/2/4/6 Pinout Diagram (TQFP-100)**

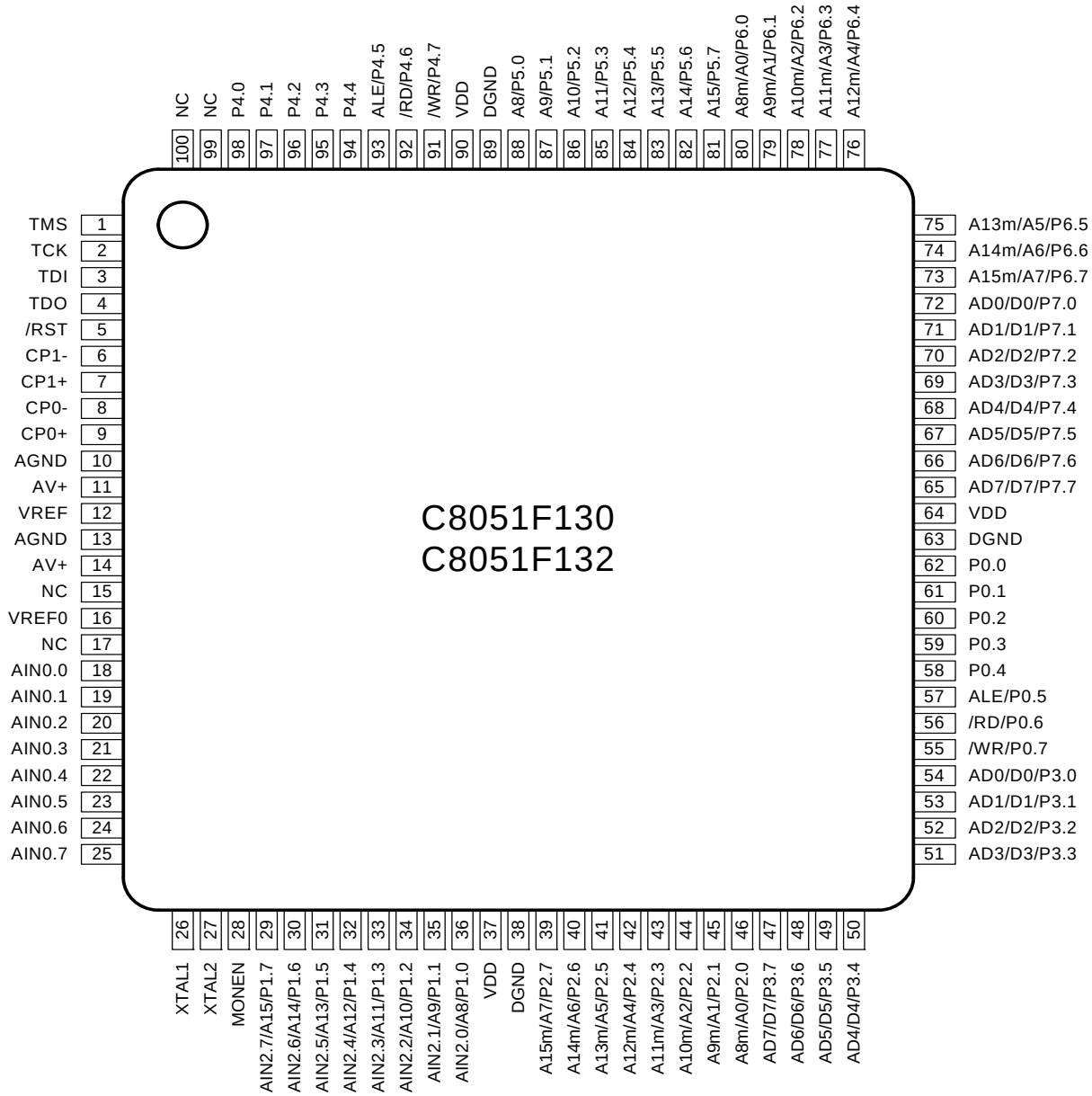
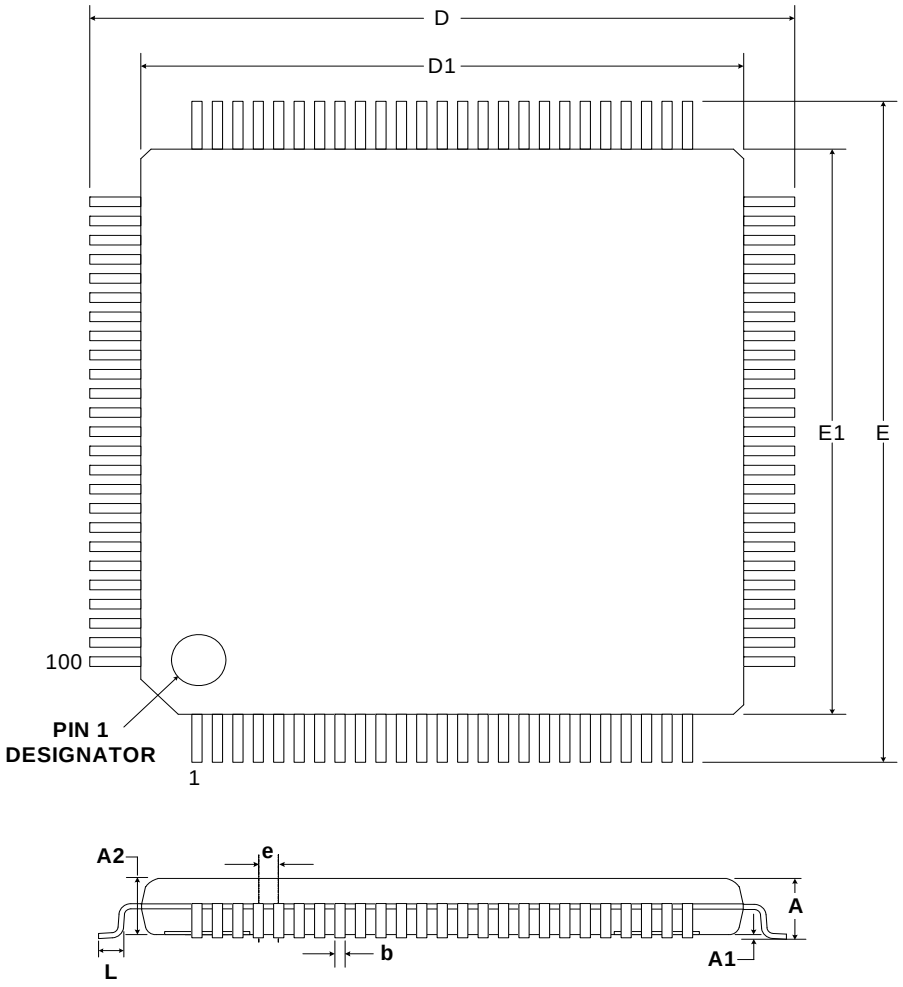



**Figure 4.2. C8051F130/2 Pinout Diagram (TQFP-100)**




| | MIN (mm) | NOM (mm) | MAX (mm) |
|---|---|---|---|
| A | - | - | 1.20 |
| A1 | 0.05 | - | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| b | 0.17 | 0.22 | 0.27 |
| D | - | 16.00 | - |
| D1 | - | 14.00 | - |
| e | - | 0.50 | - |
| E | - | 16.00 | - |
| E1 | - | 14.00 | - |
| L | 0.45 | 0.60 | 0.75 |

**Figure 4.3. TQFP-100 Package Drawing**

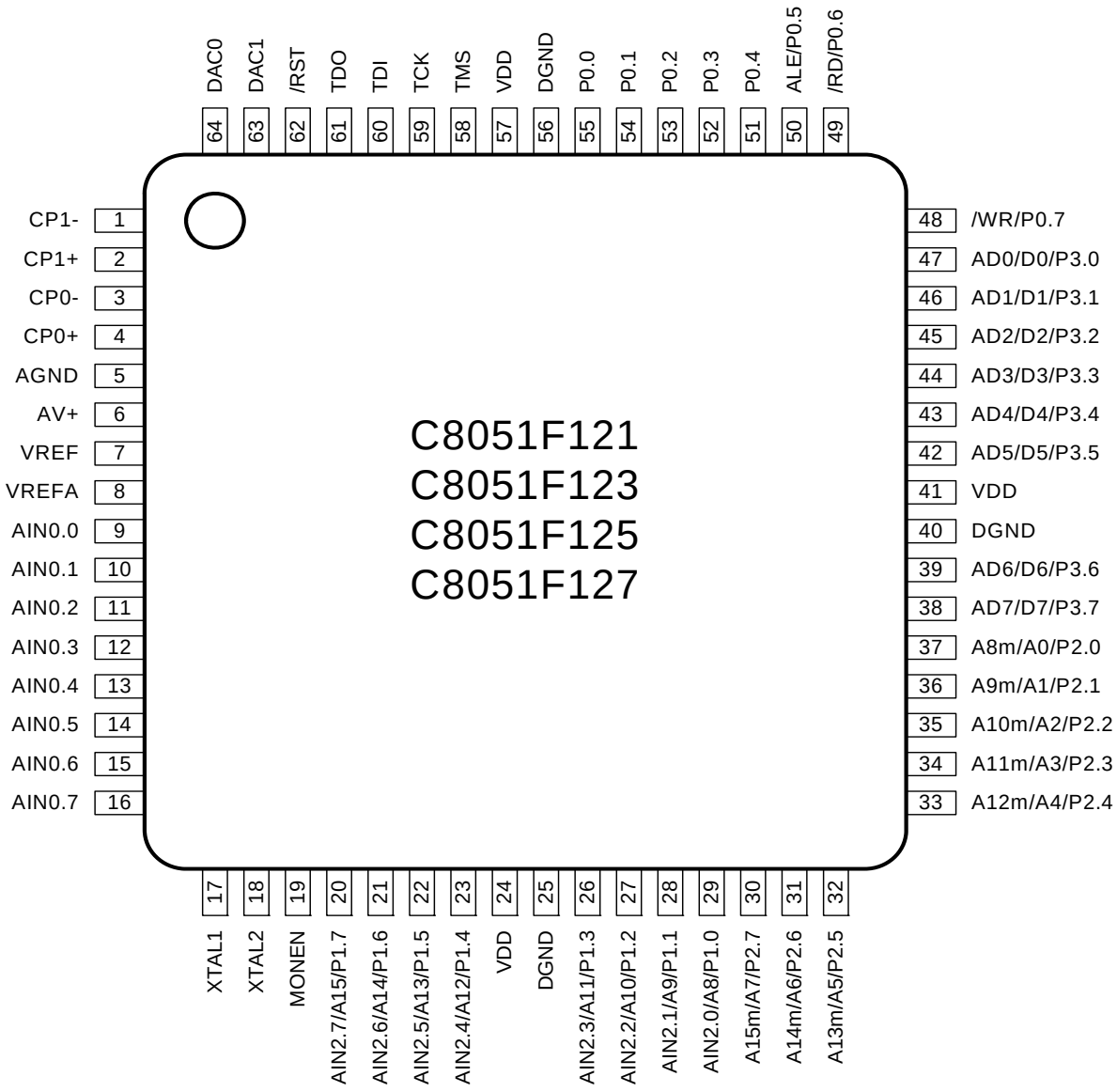



**Figure 4.4. C8051F121/3/5/7 Pinout Diagram (TQFP-64)**

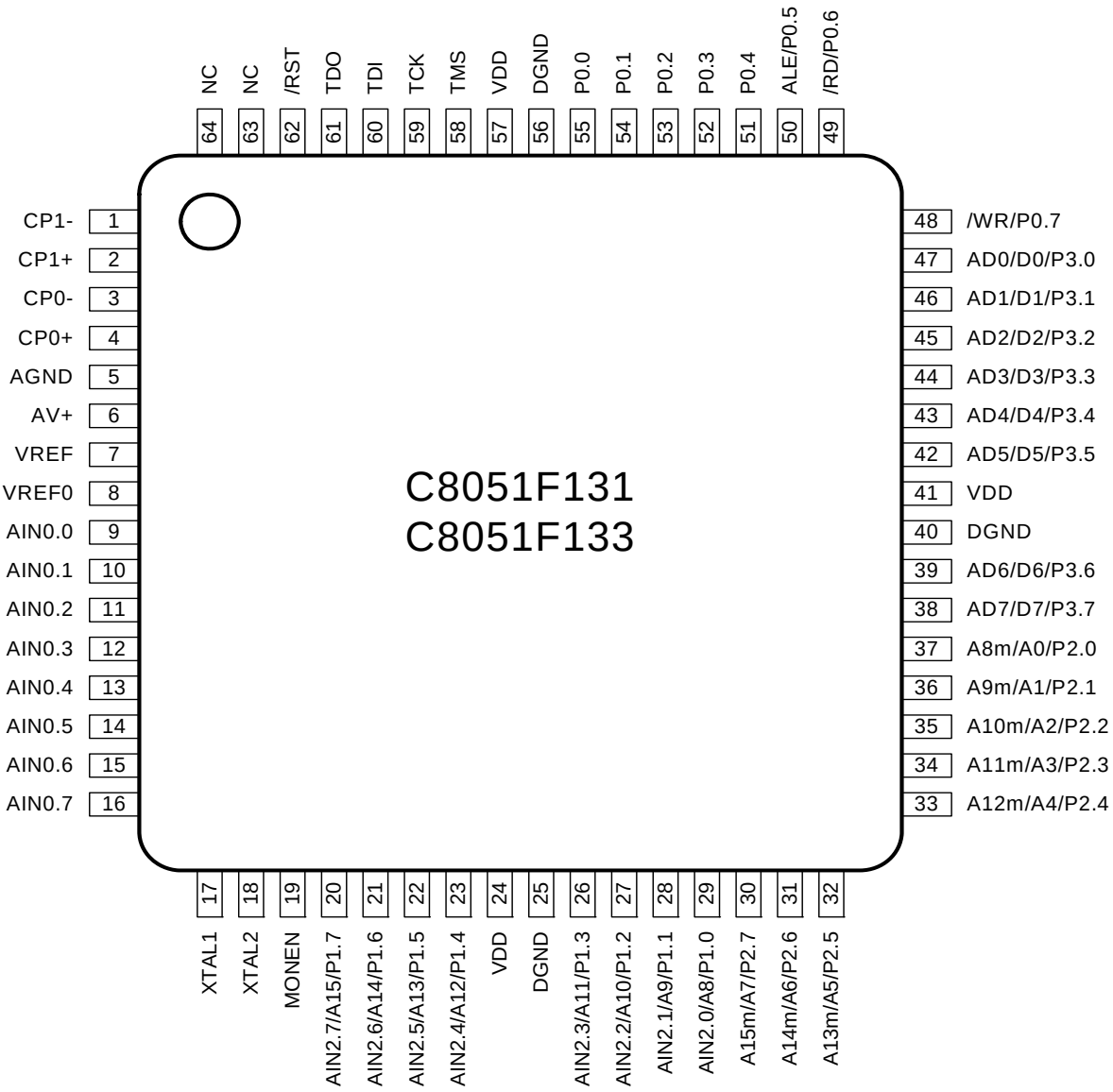
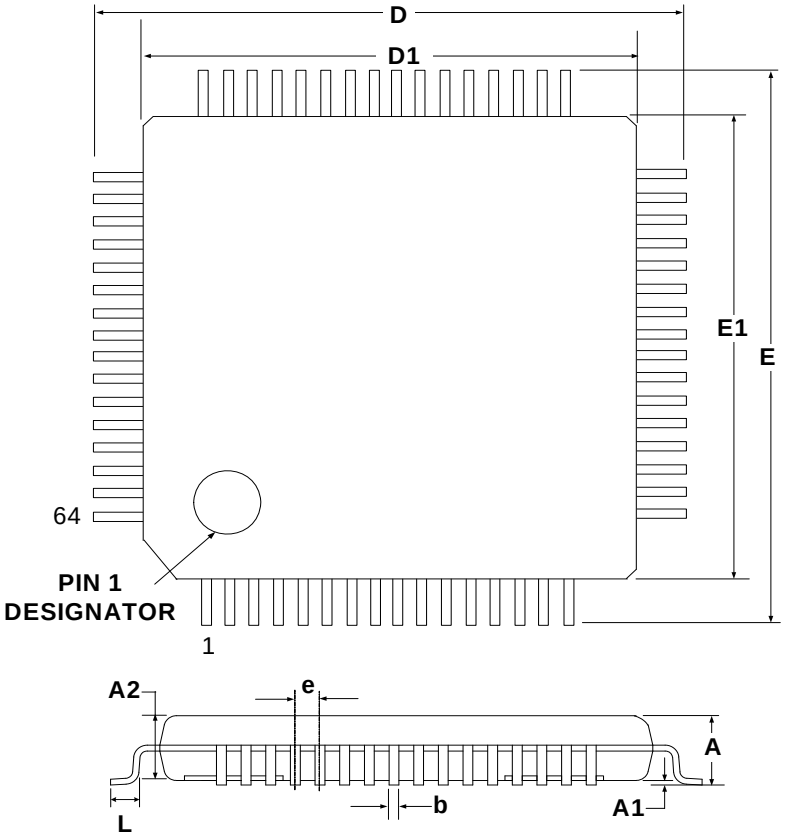



**Figure 4.5. C8051F131/3 Pinout Diagram (TQFP-64)**




| | MIN (mm) | NOM (mm) | MAX (mm) |
|---|---|---|---|
| A | - | - | 1.20 |
| A1 | 0.05 | - | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| b | 0.17 | 0.22 | 0.27 |
| D | - | 12.00 | - |
| D1 | - | 10.00 | - |
| e | - | 0.50 | - |
| E | - | 12.00 | - |
| E1 | - | 10.00 | - |
| L | 0.45 | 0.60 | 0.75 |

**Figure 4.6. TQFP-64 Package Drawing**

# 5. ADC0 (12-Bit ADC, C8051F120/1/4/5 Only)

The ADC0 subsystem for the C8051F120/1/4/5 consists of a 9-channel, configurable analog multiplexer (AMUX0), a programmable gain amplifier (PGA0), and a 100 ksps, 12-bit successive-approximation-register ADC with integrated track-and-hold and Programmable Window Detector (see block diagram in Figure 5.1). The AMUX0, PGA0, Data Conversion Modes, and Window Detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. The voltage reference used by ADC0 is selected as described in **Section "9. Voltage Reference" on page 113**. The ADC0 subsystem (ADC0, track-and-hold and PGA0) is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

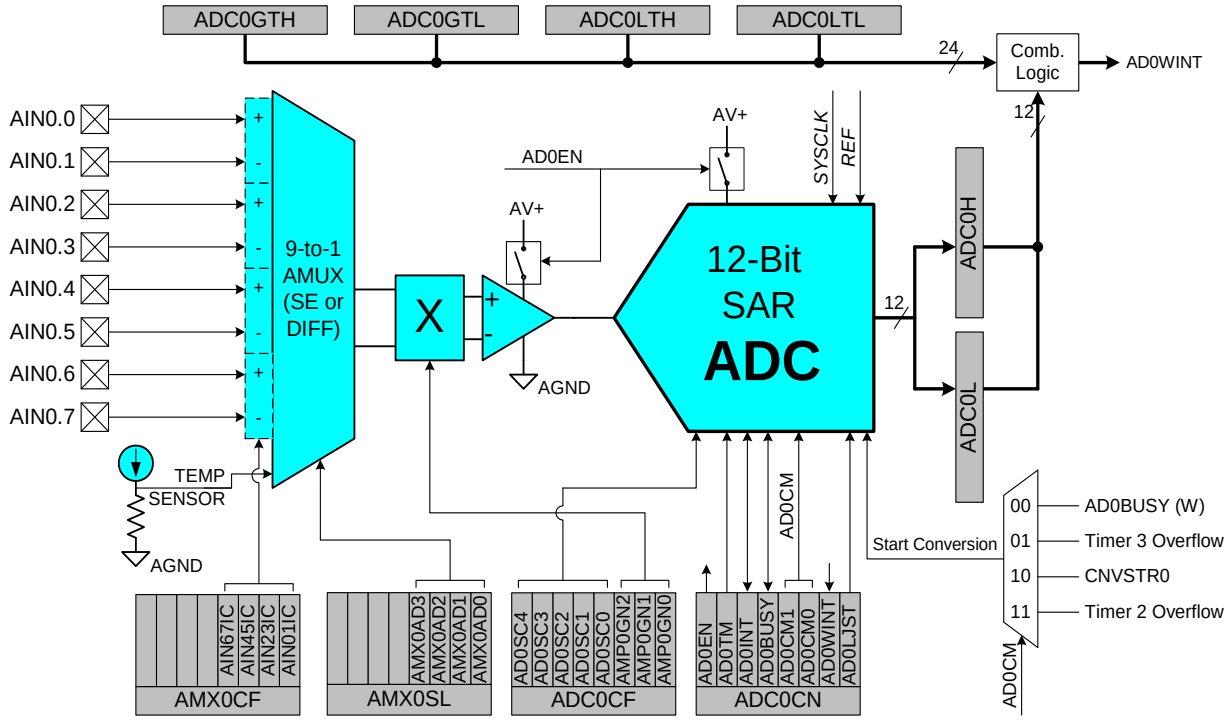



**Figure 5.1. 12-Bit ADC0 Functional Block Diagram**

## 5.1. Analog Multiplexer and PGA

Eight of the AMUX channels are available for external measurements while the ninth channel is internally connected to an on-chip temperature sensor (temperature transfer function is shown in Figure 5.2). AMUX input pairs can be programmed to operate in either differential or single-ended mode. This allows the user to select the best measurement technique for each input channel, and even accommodates mode changes "on-the-fly". The AMUX defaults to all single-ended inputs upon reset. There are two registers associated with the AMUX: the Channel Selection register AMX0SL (SFR Definition 5.2), and the Configuration register AMX0CF (SFR Definition 5.1). The table in SFR Definition 5.2 shows AMUX functionality by channel, for each possible configuration. The PGA amplifies the AMUX output signal by an amount determined by the states of the AMP0GN2-0 bits in the ADC0 Configuration register, ADC0CF (SFR Definition 5.3). The PGA can be software-programmed for gains of 0.5, 2, 4, 8 or 16. Gain defaults to unity on reset.

The Temperature Sensor transfer function is shown in Figure 5.2. The output voltage ($V_{TEMP}$) is the PGA input when the Temperature Sensor is selected by bits AMX0AD3-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings. Typical values for the Slope and Offset parameters can be found in Table 5.1.

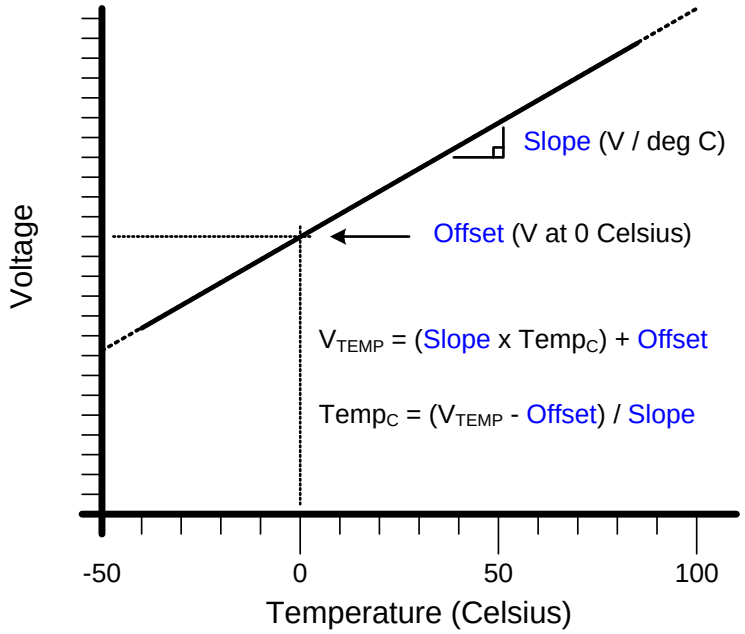



**Figure 5.2. Typical Temperature Sensor Transfer Function**

## 5.2. ADC Modes of Operation

ADC0 has a maximum conversion speed of 100 ksps. The ADC0 conversion clock is derived from the system clock divided by the value held in the ADCSC bits of register ADC0CF.

### 5.2.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1, AD0CM0) in ADC0CN. Conversions may be initiated by:

1. Writing a '1' to the AD0BUSY bit of ADC0CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR0;
4. A Timer 2 overflow (i.e. timed continuous conversions).

The AD0BUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the AD0INT interrupt flag (ADC0CN.5). Converted data is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 5.5) depending on the programmed state of the AD0LJST bit in the ADC0CN register.

When initiating conversions by writing a '1' to AD0BUSY, the AD0INT bit should be polled to determine when a conversion has completed (ADC0 interrupts may also be used). The recommended polling procedure is shown below.

Step 1. Write a '0' to AD0INT;
Step 2. Write a '1' to AD0BUSY;
Step 3. Poll AD0INT for '1';
Step 4. Process ADC0 data.

When CNVSTR0 is used as a conversion start source, it must be enabled in the crossbar, and the corresponding pin must be set to open-drain, high-impedance mode (see **Section "18. Port Input/Output" on page 235** for more details on Port I/O configuration).

### 5.2.2. Tracking Modes

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked when a conversion is not 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 CNVSTR0 signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR0 is low; conversion begins on the rising edge of CNVSTR0 (see Figure 5.3). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX or PGA settings are frequently changed, to ensure that settling time requirements are met (see **Section "5.2.3. Settling Time Requirements" on page 59**).

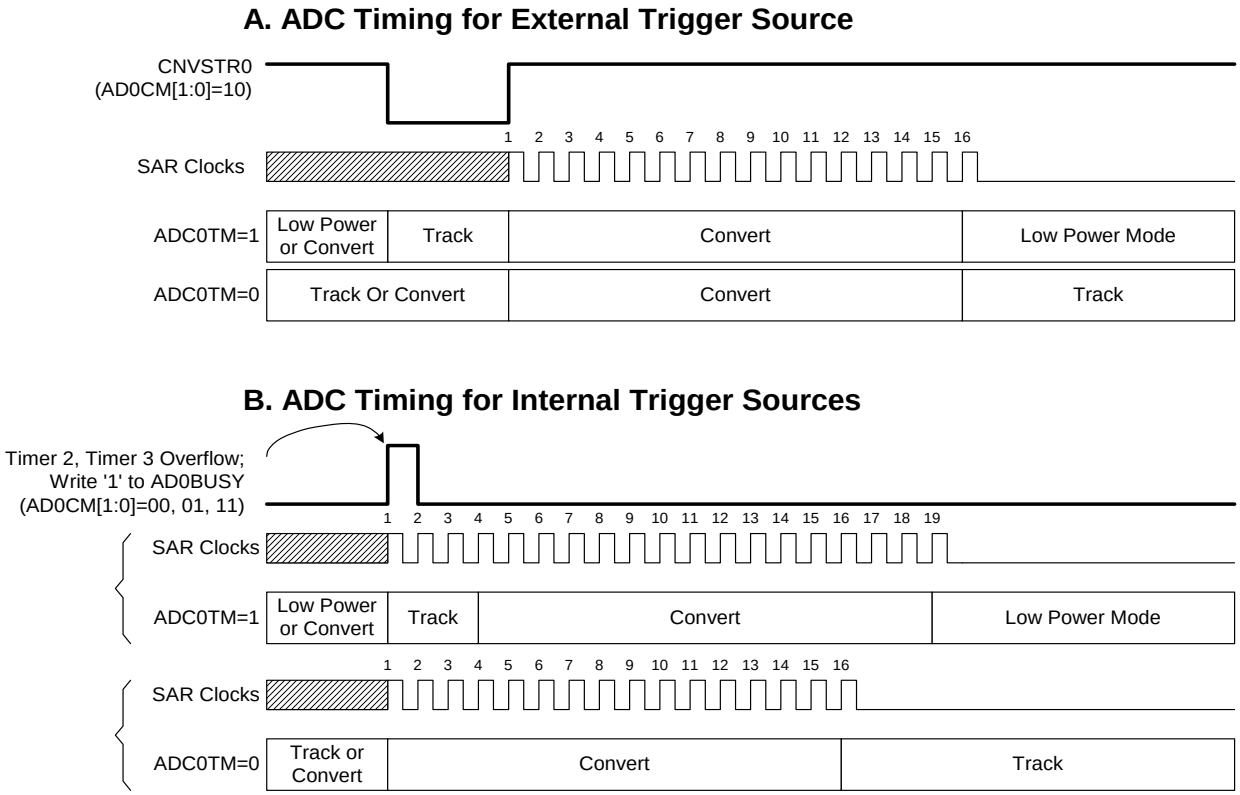



**Figure 5.3. ADC0 Track and Conversion Example Timing**

### 5.2.3. Settling Time Requirements

A minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the ADC0 MUX resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 5.4 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required settling time for a given settling accuracy (*SA*) may be approximated by Equation 5.1. When measuring the Temperature Sensor output, $R_{TOTAL}$ reduces to $R_{MUX}$. An absolute minimum settling time of 1.5 µs is required after any MUX or PGA selection. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the tracking requirements.

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

**Equation 5.1. 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 ADC0 MUX resistance and any external source resistance.
*n* is the ADC resolution in bits (12).

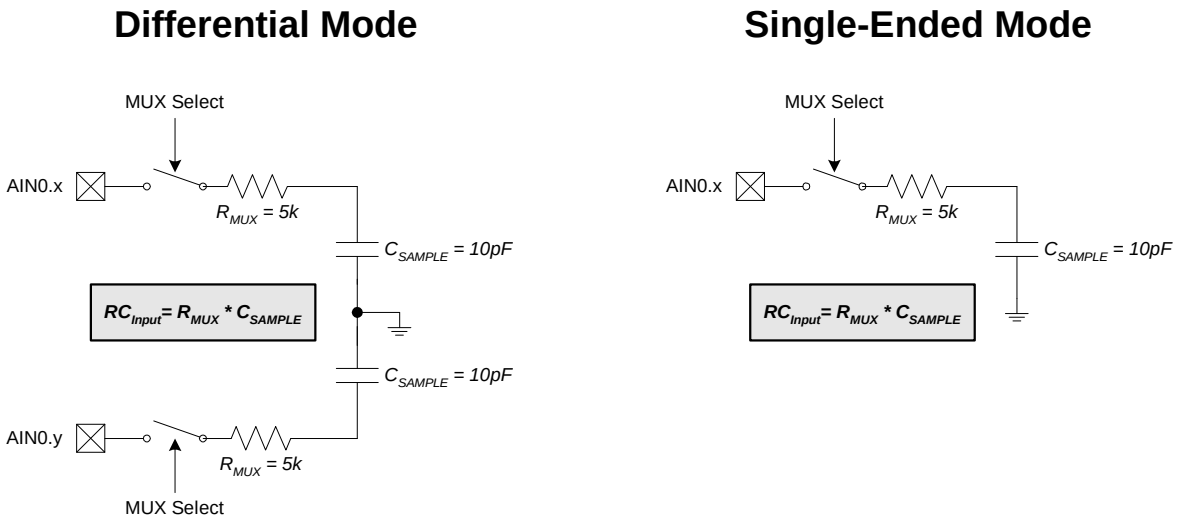



**Figure 5.4. ADC0 Equivalent Input Circuits**

## SFR Definition 5.1. AMX0CF: AMUX0 Configuration

SFR Page: 0
SFR Address: 0xBA

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | AIN67IC | AIN45IC | AIN23IC | AIN01IC | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bit3: AIN67IC: AIN0.6, AIN0.7 Input Pair Configuration Bit.
0: AIN0.6 and AIN0.7 are independent single-ended inputs.
1: AIN0.6, AIN0.7 are (respectively) +, – differential input pair.
Bit2: AIN45IC: AIN0.4, AIN0.5 Input Pair Configuration Bit.
0: AIN0.4 and AIN0.5 are independent single-ended inputs.
1: AIN0.4, AIN0.5 are (respectively) +, – differential input pair.
Bit1: AIN23IC: AIN0.2, AIN0.3 Input Pair Configuration Bit.
0: AIN0.2 and AIN0.3 are independent single-ended inputs.
1: AIN0.2, AIN0.3 are (respectively) +, – differential input pair.
Bit0: AIN01IC: AIN0.0, AIN0.1 Input Pair Configuration Bit.
0: AIN0.0 and AIN0.1 are independent single-ended inputs.
1: AIN0.0, AIN0.1 are (respectively) +, – differential input pair.

**Note:** The ADC0 Data Word is in 2's complement format for channels configured as differential.

## SFR Definition 5.2. AMX0SL: AMUX0 Channel Select

SFR Page: 0
SFR Address: 0xBB

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | AMX0AD3 | AMX0AD2 | AMX0AD1 | AMX0AD0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bits3–0: AMX0AD3–0: AMX0 Address Bits.
0000-1111b: ADC Inputs selected per chart below.

| AMX0CF Bits 3–0 | AMX0AD3–0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1xxx** |
| **0000** | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0001** | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0010** | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0011** | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0100** | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0101** | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0110** | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **0111** | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| **1000** | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1001** | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1010** | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1011** | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1100** | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1101** | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1110** | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| **1111** | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |

## SFR Definition 5.3. ADC0CF: ADC0 Configuration

SFR Page: 0
SFR Address: 0xBC

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0SC4 | AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | AMP0GN2 | AMP0GN1 | AMP0GN0 | 11111000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
The SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in AD0SC4-0, and $CLK_{SAR0}$ refers to the desired ADC0 SAR clock (Note: the ADC0 SAR Conversion Clock should be less than or equal to 2.5 MHz).

$$AD0SC = \frac{SYSCLK}{2 \times CLK_{SAR0}} - 1 \qquad (AD0SC > 00000\text{b})$$

When the AD0SC bits are equal to 00000b, the SAR Conversion clock is equal to SYSCLK to facilitate faster ADC conversions at slower SYSCLK speeds.

Bits2–0: AMP0GN2–0: ADC0 Internal Amplifier Gain (PGA).
000: Gain = 1
001: Gain = 2
010: Gain = 4
011: Gain = 8
10x: Gain = 16
11x: Gain = 0.5

## SFR Definition 5.4. ADC0CN: ADC0 Control

SFR Page: 0
SFR Address: 0xE8 (bit addressable)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0EN | AD0TM | AD0INT | AD0BUSY | AD0CM1 | AD0CM0 | AD0WINT | AD0LJST | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit6: AD0TM: ADC Track Mode Bit.
0: When the ADC is enabled, tracking is continuous unless a conversion is in process.
1: Tracking Defined by ADCM1-0 bits.

Bit5: AD0INT: ADC0 Conversion Complete Interrupt Flag.
This flag must be cleared by software.
0: ADC0 has not completed a data conversion since the last time this flag was cleared.
1: ADC0 has completed a data conversion.

Bit4: AD0BUSY: ADC0 Busy Bit.
Read:
0: ADC0 Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.
1: ADC0 Conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0CM1-0 = 00b.

Bits3–2: AD0CM1–0: ADC0 Start of Conversion Mode Select.
If AD0TM = 0:
00: ADC0 conversion initiated on every write of ‘1’ to AD0BUSY.
01: ADC0 conversion initiated on overflow of Timer 3.
10: ADC0 conversion initiated on rising edge of external CNVSTR0.
11: ADC0 conversion initiated on overflow of Timer 2.
If AD0TM = 1:
00: Tracking starts with the write of ‘1’ to AD0BUSY and lasts for 3 SAR clocks, followed by conversion.
01: Tracking started by the overflow of Timer 3 and lasts for 3 SAR clocks, followed by conversion.
10: ADC0 tracks only when CNVSTR0 input is logic low; conversion starts on rising CNVSTR0 edge.
11: Tracking started by the overflow of Timer 2 and lasts for 3 SAR clocks, followed by conversion.

Bit1: AD0WINT: ADC0 Window Compare Interrupt Flag.
This bit must be cleared by software.
0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC0 Window Comparison Data match has occurred.

Bit0: AD0LJST: ADC0 Left Justify Select.
0: Data in ADC0H:ADC0L registers are right-justified.
1: Data in ADC0H:ADC0L registers are left-justified.

### SFR Definition 5.5. ADC0H: ADC0 Data Word MSB

SFR Page: 0
SFR Address: 0xBF

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0: Bits 7–4 are the sign extension of Bit3. Bits 3–0 are the upper 4 bits of the 12-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 12-bit ADC0 Data Word.

### SFR Definition 5.6. ADC0L: ADC0 Data Word LSB

SFR Page: 0
SFR Address: 0xBE

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 12-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7–4 are the lower 4 bits of the 12-bit ADC0 Data Word. Bits 3–0 will always read ‘0’.

**12-bit ADC0 Data Word appears in the ADC0 Data Word Registers as follows:**
ADC0H[3:0]:ADC0L[7:0], if AD0LJST = 0
(ADC0H[7:4] will be sign-extension of ADC0H.3 for a differential reading, otherwise =
0000b).

ADC0H[7:0]:ADC0L[7:4], if AD0LJST = 1
(ADC0L[3:0] = 0000b).

Example: ADC0 Data Word Conversion Map, AIN0.0 Input in Single-Ended Mode
(AMX0CF = 0x00, AMX0SL = 0x00)

| AIN0.0–AGND (Volts) | **ADC0H:ADC0L** (AD0LJST = 0) | **ADC0H:ADC0L** (AD0LJST = 1) |
|---|---|---|
| VREF x (4095/4096) | 0x0FFF | 0xFFF0 |
| VREF / 2 | 0x0800 | 0x8000 |
| VREF x (2047/4096) | 0x07FF | 0x7FF0 |
| 0 | 0x0000 | 0x0000 |

Example: ADC0 Data Word Conversion Map, AIN0.0-AIN0.1 Differential Input Pair
(AMX0CF = 0x01, AMX0SL = 0x00)

| AIN0.0–AIN0.1 (Volts) | **ADC0H:ADC0L** (AD0LJST = 0) | **ADC0H:ADC0L** (AD0LJST = 1) |
|---|---|---|
| VREF x (2047/2048) | 0x07FF | 0x7FF0 |
| VREF / 2 | 0x0400 | 0x4000 |
| VREF x (1/2048) | 0x0001 | 0x0010 |
| 0 | 0x0000 | 0x0000 |
| –VREF x (1/2048) | 0xFFFF (–1d) | 0xFFF0 |
| –VREF / 2 | 0xFC00 (–1024d) | 0xC000 |
| –VREF | 0xF800 (–2048d) | 0x8000 |

For AD0LJST = 0:

$$Code = Vin \times \frac{Gain}{VREF} \times 2^n$$ ; 'n' = 12 for Single-Ended; 'n'=11 for Differential.

**Figure 5.5. ADC0 Data Word Example**

## 5.3. ADC0 Programmable Window Detector

The ADC0 Programmable Window Detector continuously compares the ADC0 output to user-programmed limits, and notifies the system when an out-of-bound 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 ADC0CN) can also be used in polled mode. The high and low bytes of the reference words are loaded into the ADC0 Greater-Than and ADC0 Less-Than registers (ADC0GTH, ADC0GTL, ADC0LTH, and ADC0LTL). Reference comparisons are shown starting on page 68. Notice that the window detector flag can be asserted when the measured data is inside or outside the user-programmed limits, depending on the programming of the ADC0GTx and ADC0LTx registers.

### SFR Definition 5.7. ADC0GTH: ADC0 Greater-Than Data High Byte

SFR Page: 0
SFR Address: 0xC5

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: High byte of ADC0 Greater-Than Data Word.

### SFR Definition 5.8. ADC0GTL: ADC0 Greater-Than Data Low Byte

SFR Page: 0
SFR Address: 0xC4

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: Low byte of ADC0 Greater-Than Data Word.

### SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte

SFR Page: 0
SFR Address: 0xC7

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

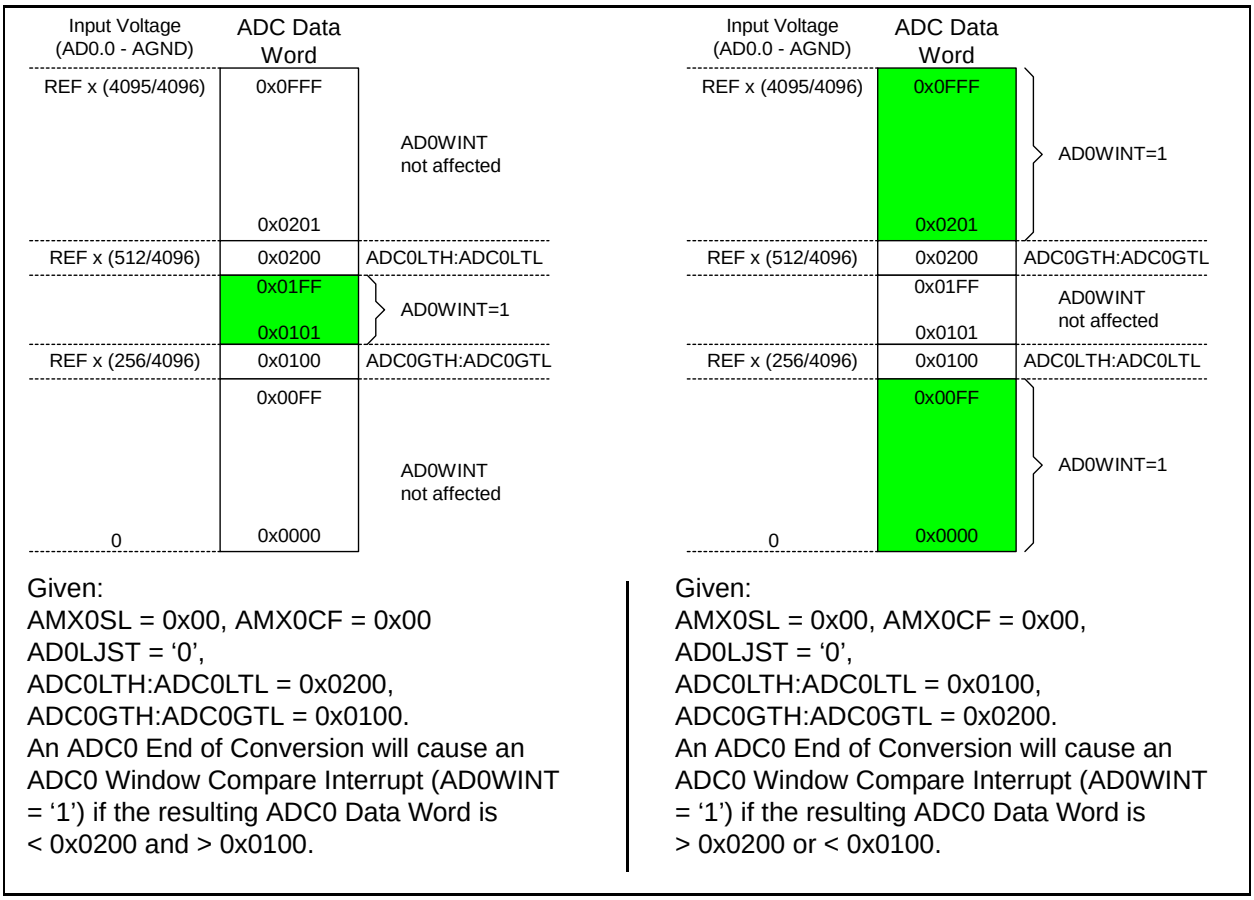
Bits7–0: High byte of ADC0 Less-Than Data Word.

### SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte

SFR Page: 0
SFR Address: 0xC6

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

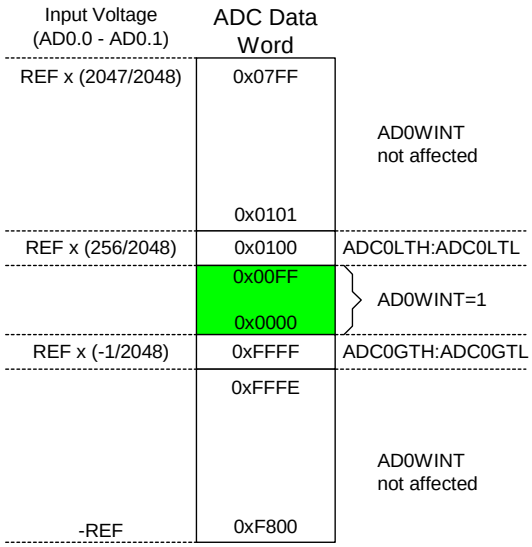
Bits7–0: Low byte of ADC0 Less-Than Data Word.



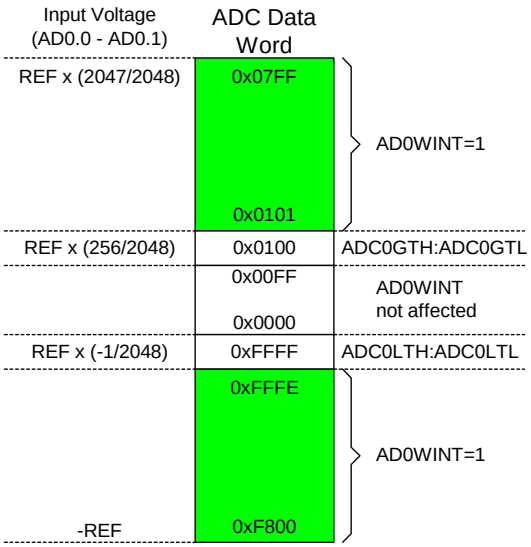

Given:
AMX0SL = 0x00, AMX0CF = 0x00
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0200,
ADC0GTH:ADC0GTL = 0x0100.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x0200 and > 0x0100.

Given:
AMX0SL = 0x00, AMX0CF = 0x00,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0100,
ADC0GTH:ADC0GTL = 0x0200.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is > 0x0200 or < 0x0100.

**Figure 5.6. 12-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data**

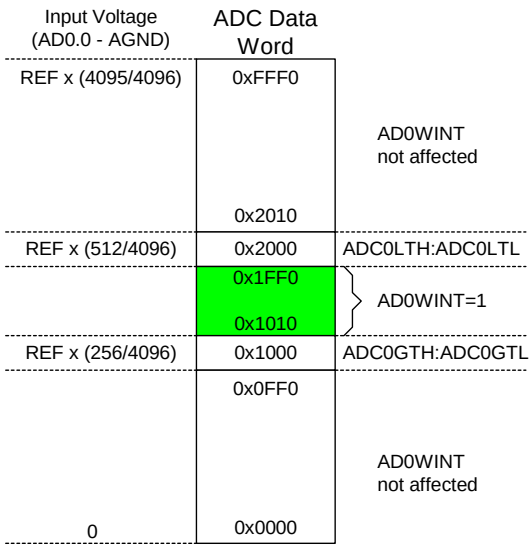



Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0100,
ADC0GTH:ADC0GTL = 0xFFFF.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x0100 and > 0xFFFF. (In 2s-complement math, 0xFFFF = -1.)

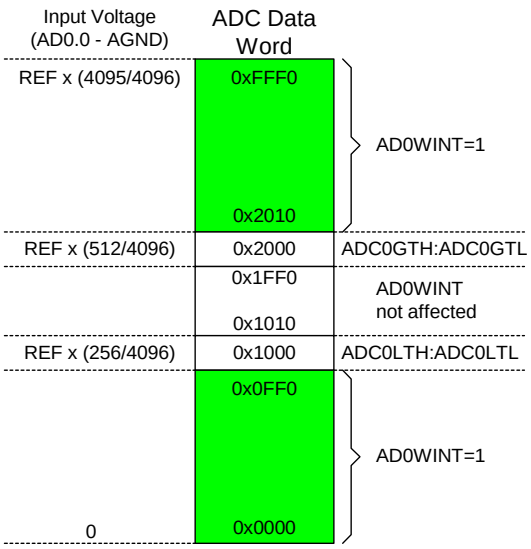



Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0xFFFF,
ADC0GTH:ADC0GTL = 0x0100.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0xFFFF or > 0x0100. (In 2s-complement math, 0xFFFF = -1.)

**Figure 5.7. 12-Bit ADC0 Window Interrupt Example: Right Justified Differential Data**

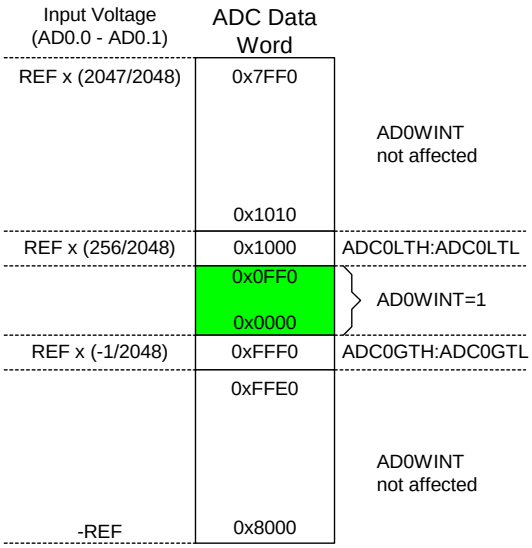



Given:
AMX0SL = 0x00, AMX0CF = 0x00,
AD0LJST = '1',
ADC0LTH:ADC0LTL = 0x2000,
ADC0GTH:ADC0GTL = 0x1000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x2000 and > 0x1000.

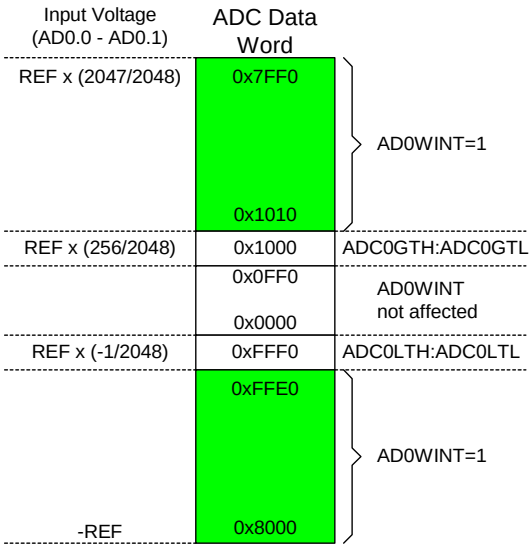



Given:
AMX0SL = 0x00, AMX0CF = 0x00,
AD0LJST = '1'
ADC0LTH:ADC0LTL = 0x1000,
ADC0GTH:ADC0GTL = 0x2000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x1000 or > 0x2000.

**Figure 5.8. 12-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data**




Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '1',
ADC0LTH:ADC0LTL = 0x1000,
ADC0GTH:ADC0GTL = 0xFFF0.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x1000 and > 0xFFF0. (2s-complement math.)




Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '1',
ADC0LTH:ADC0LTL = 0xFFF0,
ADC0GTH:ADC0GTL = 0x1000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0xFFF0 or > 0x1000. (2s-complement math.)

**Figure 5.9. 12-Bit ADC0 Window Interrupt Example: Left Justified Differential Data**

### Table 5.1. 12-Bit ADC0 Electrical Characteristics (C8051F120/1/4/5)

$V_{DD}$ = 3.0 V, AV+ = 3.0 V, VREF = 2.40 V (REFBE = 0), PGA Gain = 1, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 12 | | bits |
| Integral Nonlinearity | | — | — | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | — | ±1 | LSB |
| Offset Error | | — | –3±1 | — | LSB |
| Full Scale Error | Differential mode | — | –7±3 | — | LSB |
| Offset Temperature Coefficient | | — | ±0.25 | — | ppm/°C |
| **Dynamic Performance (10 kHz sine-wave input, 0 to 1 dB below Full Scale, 100 ksps** | | | | | |
| Signal-to-Noise Plus Distortion | | 66 | — | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | –75 | — | dB |
| Spurious-Free Dynamic Range | | — | 80 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Clock Frequency | | — | — | 2.5 | MHz |
| Conversion Time in SAR Clocks | | 16 | — | — | clocks |
| Track/Hold Acquisition Time | | 1.5 | — | — | µs |
| Throughput Rate | | — | — | 100 | ksps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | Single-ended operation | 0 | — | VREF | V |
| ***Common-mode Voltage Range** | Differential operation | AGND | — | AV+ | V |
| Input Capacitance | | — | 10 | — | pF |
| **Temperature Sensor** | | | | | |
| Linearity[1] | | — | ±0.2 | — | °C |
| Offset | (Temp = 0 °C) | — | 776 | — | mV |
| Offset Error[1, 2] | (Temp = 0 °C) | — | ±8.5 | — | mV |
| Slope | | — | 2.86 | — | mV / °C |
| Slope Error[2] | | — | ±0.034 | — | mV / °C |
| **Power Specifications** | | | | | |
| Power Supply Current (AV+ supplied to ADC) | Operating Mode, 100 ksps | — | 450 | 900 | µA |
| Power Supply Rejection | | — | ±0.3 | — | mV/V |

**Notes:**
1. Includes ADC offset, gain, and linearity variations.
2. Represents one standard deviation from the mean.

# 6. ADC0 (10-Bit ADC, C8051F122/3/6/7 and C8051F13x Only)

The ADC0 subsystem for the C8051F122/3/6/7 and C8051F13x consists of a 9-channel, configurable analog multiplexer (AMUX0), a programmable gain amplifier (PGA0), and a 100 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and Programmable Window Detector (see block diagram in Figure 6.1). The AMUX0, PGA0, Data Conversion Modes, and Window Detector are all configurable under software control via the Special Function Registers shown in Figure 6.1. The voltage reference used by ADC0 is selected as described in **Section "9. Voltage Reference" on page 113**. The ADC0 subsystem (ADC0, track-and-hold and PGA0) is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

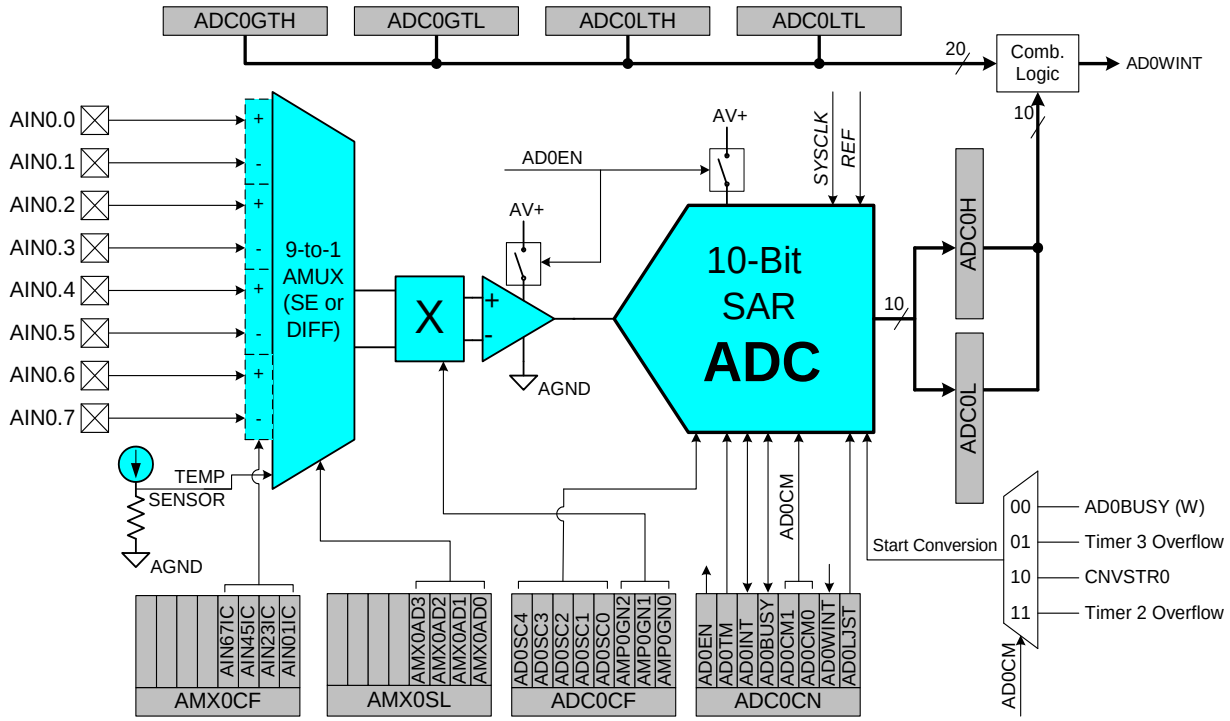



**Figure 6.1. 10-Bit ADC0 Functional Block Diagram**

## 6.1. Analog Multiplexer and PGA

Eight of the AMUX channels are available for external measurements while the ninth channel is internally connected to an on-chip temperature sensor (temperature transfer function is shown in Figure 6.2). AMUX input pairs can be programmed to operate in either differential or single-ended mode. This allows the user to select the best measurement technique for each input channel, and even accommodates mode changes "on-the-fly". The AMUX defaults to all single-ended inputs upon reset. There are two registers associated with the AMUX: the Channel Selection register AMX0SL (SFR Definition 6.2), and the Configuration register AMX0CF (SFR Definition 6.1). The table in SFR Definition 6.2 shows AMUX functionality by channel, for each possible configuration. The PGA amplifies the AMUX output signal by an amount determined by the states of the AMP0GN2-0 bits in the ADC0 Configuration register, ADC0CF (SFR Definition 6.3). The PGA can be software-programmed for gains of 0.5, 2, 4, 8 or 16. Gain defaults to unity on reset.

The Temperature Sensor transfer function is shown in Figure 6.2. The output voltage ($V_{TEMP}$) is the PGA input when the Temperature Sensor is selected by bits AMX0AD3-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings. Typical values for the Slope and Offset parameters can be found in Table 6.1.

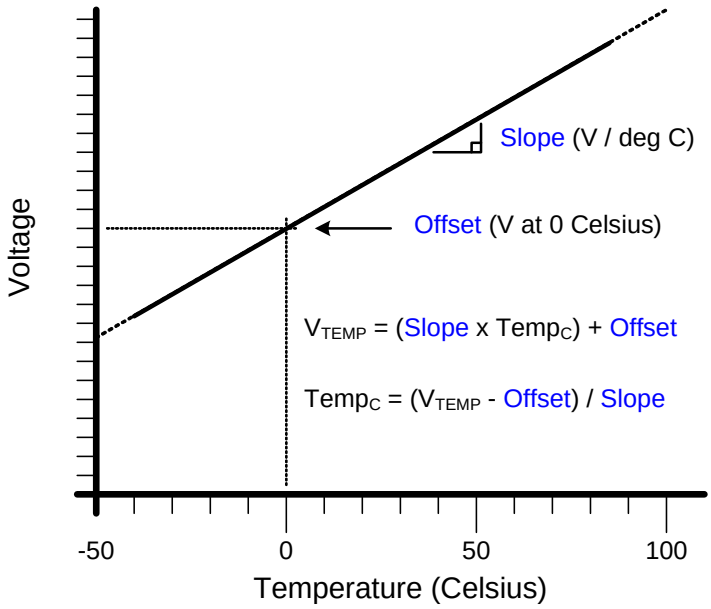



**Figure 6.2. Typical Temperature Sensor Transfer Function**

## 6.2. ADC Modes of Operation

ADC0 has a maximum conversion speed of 100 ksps. The ADC0 conversion clock is derived from the system clock divided by the value held in the ADCSC bits of register ADC0CF.

### 6.2.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1, AD0CM0) in ADC0CN. Conversions may be initiated by:

1. Writing a '1' to the AD0BUSY bit of ADC0CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR0;
4. A Timer 2 overflow (i.e. timed continuous conversions).

The AD0BUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the AD0INT interrupt flag (ADC0CN.5). Converted data is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 6.5) depending on the programmed state of the AD0LJST bit in the ADC0CN register.

When initiating conversions by writing a '1' to AD0BUSY, the AD0INT bit should be polled to determine when a conversion has completed (ADC0 interrupts may also be used). The recommended polling procedure is shown below.

Step 1. Write a '0' to AD0INT;
Step 2. Write a '1' to AD0BUSY;
Step 3. Poll AD0INT for '1';
Step 4. Process ADC0 data.

When CNVSTR0 is used as a conversion start source, it must be enabled in the crossbar, and the corresponding pin must be set to open-drain, high-impedance mode (see **Section "18. Port Input/Output" on page 235** for more details on Port I/O configuration).

### 6.2.2. Tracking Modes

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked when a conversion is not 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 CNVSTR0 signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR0 is low; conversion begins on the rising edge of CNVSTR0 (see Figure 6.3). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX or PGA settings are frequently changed, to ensure that settling time requirements are met (see **Section "6.2.3. Settling Time Requirements" on page 77**).

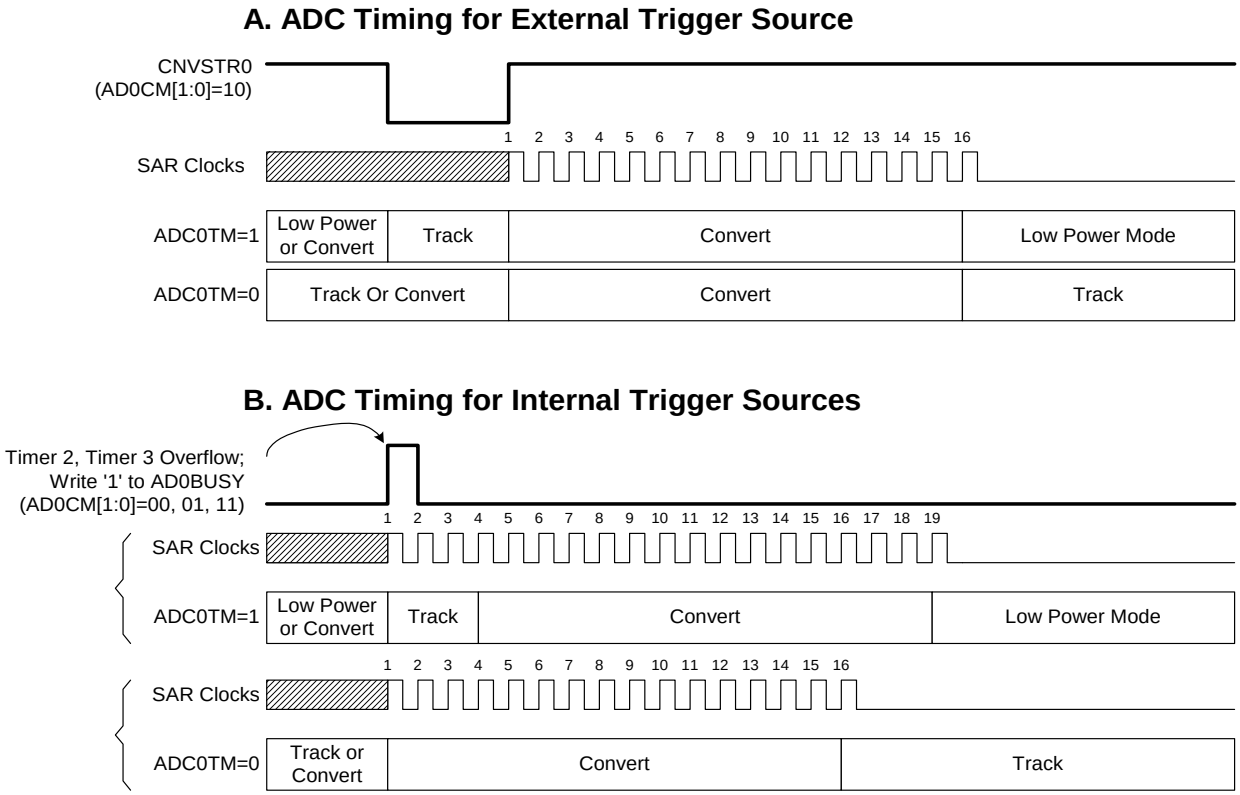



**Figure 6.3. ADC0 Track and Conversion Example Timing**

### 6.2.3. Settling Time Requirements

A minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the ADC0 MUX resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 6.4 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required settling time for a given settling accuracy (*SA*) may be approximated by Equation 6.1. When measuring the Temperature Sensor output, $R_{TOTAL}$ reduces to $R_{MUX}$. An absolute minimum settling time of 1.5 µs is required after any MUX or PGA selection. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the tracking requirements.

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

**Equation 6.1. 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 ADC0 MUX resistance and any external source resistance.
*n* is the ADC resolution in bits (10).

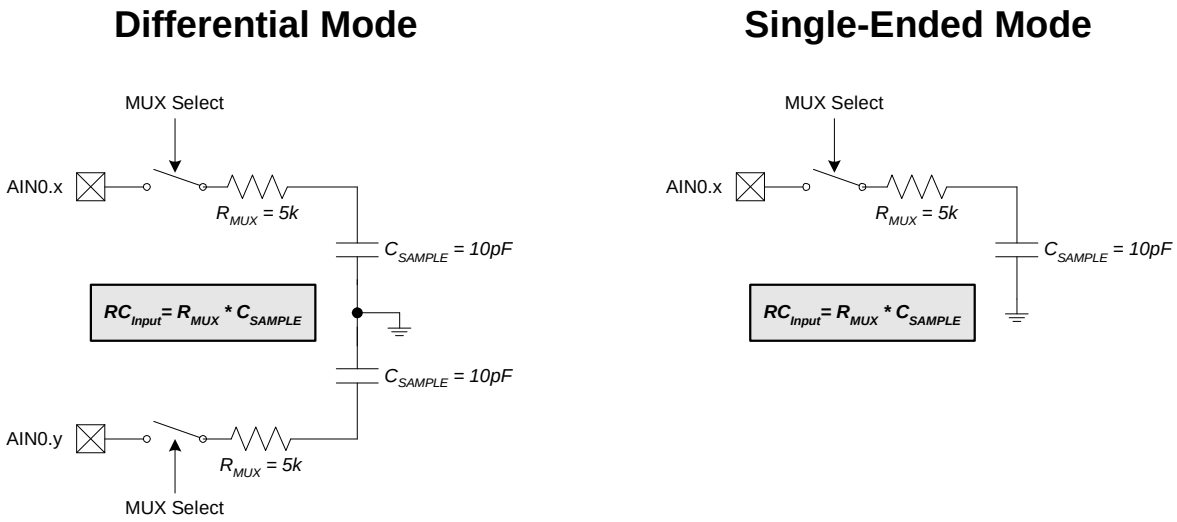



**Figure 6.4. ADC0 Equivalent Input Circuits**

## SFR Definition 6.1. AMX0CF: AMUX0 Configuration

SFR Page: 0
SFR Address: 0xBA

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | AIN67IC | AIN45IC | AIN23IC | AIN01IC | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bit3: AIN67IC: AIN0.6, AIN0.7 Input Pair Configuration Bit.
0: AIN0.6 and AIN0.7 are independent single-ended inputs.
1: AIN0.6, AIN0.7 are (respectively) +, - differential input pair.
Bit2: AIN45IC: AIN0.4, AIN0.5 Input Pair Configuration Bit.
0: AIN0.4 and AIN0.5 are independent single-ended inputs.
1: AIN0.4, AIN0.5 are (respectively) +, - differential input pair.
Bit1: AIN23IC: AIN0.2, AIN0.3 Input Pair Configuration Bit.
0: AIN0.2 and AIN0.3 are independent single-ended inputs.
1: AIN0.2, AIN0.3 are (respectively) +, - differential input pair.
Bit0: AIN01IC: AIN0.0, AIN0.1 Input Pair Configuration Bit.
0: AIN0.0 and AIN0.1 are independent single-ended inputs.
1: AIN0.0, AIN0.1 are (respectively) +, - differential input pair.

**Note:** The ADC0 Data Word is in 2's complement format for channels configured as differential.

## SFR Definition 6.2. AMX0SL: AMUX0 Channel Select

SFR Page: 0
SFR Address: 0xBB

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | AMX0AD3 | AMX0AD2 | AMX0AD1 | AMX0AD0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bits3–0: AMX0AD3–0: AMX0 Address Bits.
0000-1111b: ADC Inputs selected per chart below.

| | | AMX0AD3-0 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1xxx |
| AMX0CF Bits 3-0 | 0000 | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0001 | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0010 | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0011 | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0100 | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0101 | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0110 | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 0111 | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | AIN0.6 | AIN0.7 | TEMP SENSOR |
| | 1000 | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1001 | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1010 | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1011 | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | AIN0.4 | AIN0.5 | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1100 | AIN0.0 | AIN0.1 | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1101 | +(AIN0.0) –(AIN0.1) | | AIN0.2 | AIN0.3 | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1110 | AIN0.0 | AIN0.1 | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |
| | 1111 | +(AIN0.0) –(AIN0.1) | | +(AIN0.2) –(AIN0.3) | | +(AIN0.4) –(AIN0.5) | | +(AIN0.6) –(AIN0.7) | | TEMP SENSOR |

## SFR Definition 6.3. ADC0CF: ADC0 Configuration

SFR Page: 0
SFR Address: 0xBC

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0SC4 | AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | AMP0GN2 | AMP0GN1 | AMP0GN0 | 11111000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in AD0SC4-0, and $CLK_{SAR0}$ refers to the desired ADC0 SAR clock (Note: the ADC0 SAR Conversion Clock should be less than or equal to 2.5 MHz).

$$AD0SC = \frac{SYSCLK}{2 \times CLK_{SAR0}} - 1 \qquad (AD0SC > 00000\text{b})$$

When the AD0SC bits are equal to 00000b, the SAR Conversion clock is equal to SYSCLK to facilitate faster ADC conversions at slower SYSCLK speeds.

Bits2–0: AMP0GN2–0: ADC0 Internal Amplifier Gain (PGA).
000: Gain = 1
001: Gain = 2
010: Gain = 4
011: Gain = 8
10x: Gain = 16
11x: Gain = 0.5

## SFR Definition 6.4. ADC0CN: ADC0 Control

SFR Page: 0
SFR Address: 0xE8 (bit addressable)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0EN | AD0TM | AD0INT | AD0BUSY | AD0CM1 | AD0CM0 | AD0WINT | AD0LJST | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit6: AD0TM: ADC Track Mode Bit.
0: When the ADC is enabled, tracking is continuous unless a conversion is in process.
1: Tracking Defined by ADCM1-0 bits.

Bit5: AD0INT: ADC0 Conversion Complete Interrupt Flag.
This flag must be cleared by software.
0: ADC0 has not completed a data conversion since the last time this flag was cleared.
1: ADC0 has completed a data conversion.

Bit4: AD0BUSY: ADC0 Busy Bit.
Read:
0: ADC0 Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.
1: ADC0 Conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0CM1-0 = 00b.

Bits3–2: AD0CM1–0: ADC0 Start of Conversion Mode Select.
If AD0TM = 0:
00: ADC0 conversion initiated on every write of ‘1’ to AD0BUSY.
01: ADC0 conversion initiated on overflow of Timer 3.
10: ADC0 conversion initiated on rising edge of external CNVSTR0.
11: ADC0 conversion initiated on overflow of Timer 2.
If AD0TM = 1:
00: Tracking starts with the write of ‘1’ to AD0BUSY and lasts for 3 SAR clocks, followed by conversion.
01: Tracking started by the overflow of Timer 3 and lasts for 3 SAR clocks, followed by conversion.
10: ADC0 tracks only when CNVSTR0 input is logic low; conversion starts on rising CNVSTR0 edge.
11: Tracking started by the overflow of Timer 2 and lasts for 3 SAR clocks, followed by conversion.

Bit1: AD0WINT: ADC0 Window Compare Interrupt Flag.
This bit must be cleared by software.
0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC0 Window Comparison Data match has occurred.

Bit0: AD0LJST: ADC0 Left Justify Select.
0: Data in ADC0H:ADC0L registers are right-justified.
1: Data in ADC0H:ADC0L registers are left-justified.

### SFR Definition 6.5. ADC0H: ADC0 Data Word MSB

SFR Page: 0
SFR Address: 0xBF

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0: Bits 7–4 are the sign extension of Bit3. Bits 3–0 are the upper 4 bits of the 10-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 10-bit ADC0 Data Word.

### SFR Definition 6.6. ADC0L: ADC0 Data Word LSB

SFR Page: 0
SFR Address: 0xBE

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 10-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7–4 are the lower 4 bits of the 10-bit ADC0 Data Word. Bits 3–0 will always read ‘0’.

**10-bit ADC0 Data Word appears in the ADC0 Data Word Registers as follows:**
ADC0H[1:0]:ADC0L[7:0], if AD0LJST = 0
(ADC0H[7:2] will be sign-extension of ADC0H.1 for a differential reading, otherwise =
000000b).

ADC0H[7:0]:ADC0L[7:6], if AD0LJST = 1
(ADC0L[5:0] = 00b).

Example: ADC0 Data Word Conversion Map, AIN0.0 Input in Single-Ended Mode
(AMX0CF = 0x00, AMX0SL = 0x00)

| AIN0.0–AGND (Volts) | **ADC0H:ADC0L** (AD0LJST = 0) | **ADC0H:ADC0L** (AD0LJST = 1) |
|---|---|---|
| VREF x (1023/1024) | 0x03FF | 0xFFC0 |
| VREF / 2 | 0x0200 | 0x8000 |
| VREF x (511/1024) | 0x01FF | 0x7FC0 |
| 0 | 0x0000 | 0x0000 |

Example: ADC0 Data Word Conversion Map, AIN0.0-AIN0.1 Differential Input Pair
(AMX0CF = 0x01, AMX0SL = 0x00)

| AIN0.0–AIN0.1 (Volts) | **ADC0H:ADC0L** (AD0LJST = 0) | **ADC0H:ADC0L** (AD0LJST = 1) |
|---|---|---|
| VREF x (511/512) | 0x01FF | 0x7FC0 |
| VREF / 2 | 0x0100 | 0x4000 |
| VREF x (1/512) | 0x0001 | 0x0040 |
| 0 | 0x0000 | 0x0000 |
| –VREF x (1/512) | 0xFFFF (–1d) | 0xFFC0 |
| –VREF / 2 | 0xFF00 (–256d) | 0xC000 |
| –VREF | 0xFE00 (–512d) | 0x8000 |

For AD0LJST = 0:

$$Code = Vin \times \frac{Gain}{VREF} \times 2^n$$ ; 'n' = 10 for Single-Ended; 'n'= 9 for Differential.

**Figure 6.5. ADC0 Data Word Example**

## 6.3. ADC0 Programmable Window Detector

The ADC0 Programmable Window Detector continuously compares the ADC0 output to user-programmed limits, and notifies the system when an out-of-bound 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 ADC0CN) can also be used in polled mode. The high and low bytes of the reference words are loaded into the ADC0 Greater-Than and ADC0 Less-Than registers (ADC0GTH, ADC0GTL, ADC0LTH, and ADC0LTL). Reference comparisons are shown starting on page 87. Notice that the window detector flag can be asserted when the measured data is inside or outside the user-programmed limits, depending on the programming of the ADC0GTx and ADC0LTx registers.

### SFR Definition 6.7. ADC0GTH: ADC0 Greater-Than Data High Byte

SFR Page: 0
SFR Address: 0xC5

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: High byte of ADC0 Greater-Than Data Word.

### SFR Definition 6.8. ADC0GTL: ADC0 Greater-Than Data Low Byte

SFR Page: 0
SFR Address: 0xC4

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: Low byte of ADC0 Greater-Than Data Word.

### SFR Definition 6.9. ADC0LTH: ADC0 Less-Than Data High Byte

SFR Page: 0
SFR Address: 0xC7

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

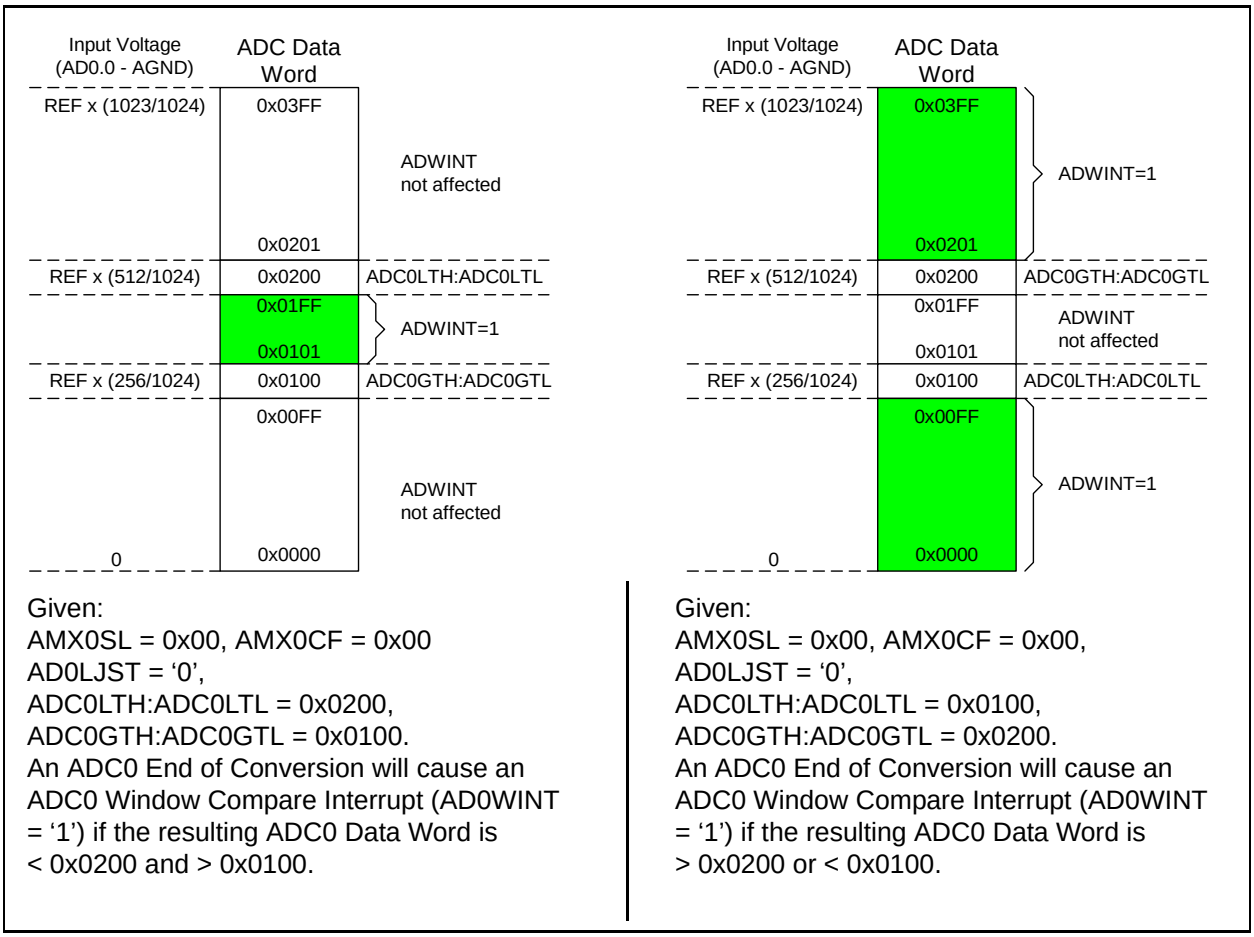
Bits7–0: High byte of ADC0 Less-Than Data Word.

### SFR Definition 6.10. ADC0LTL: ADC0 Less-Than Data Low Byte

SFR Page: 0
SFR Address: 0xC6

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

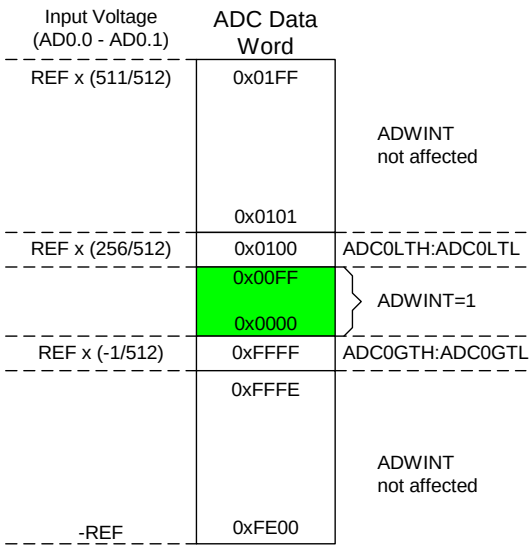
Bits7–0: Low byte of ADC0 Less-Than Data Word.



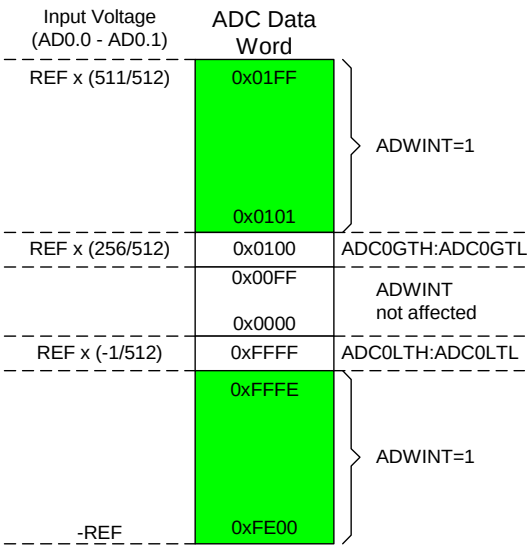

Given:
AMX0SL = 0x00, AMX0CF = 0x00
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0200,
ADC0GTH:ADC0GTL = 0x0100.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x0200 and > 0x0100.

Given:
AMX0SL = 0x00, AMX0CF = 0x00,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0100,
ADC0GTH:ADC0GTL = 0x0200.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is > 0x0200 or < 0x0100.

**Figure 6.6. 10-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data**




Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0x0100,
ADC0GTH:ADC0GTL = 0xFFFF.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x0100 and > 0xFFFF. (In 2s-complement math, 0xFFFF = -1.)




Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '0',
ADC0LTH:ADC0LTL = 0xFFFF,
ADC0GTH:ADC0GTL = 0x0100.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0xFFFF or > 0x0100. (In 2s-complement math, 0xFFFF = -1.)

**Figure 6.7. 10-Bit ADC0 Window Interrupt Example: Right Justified Differential Data**

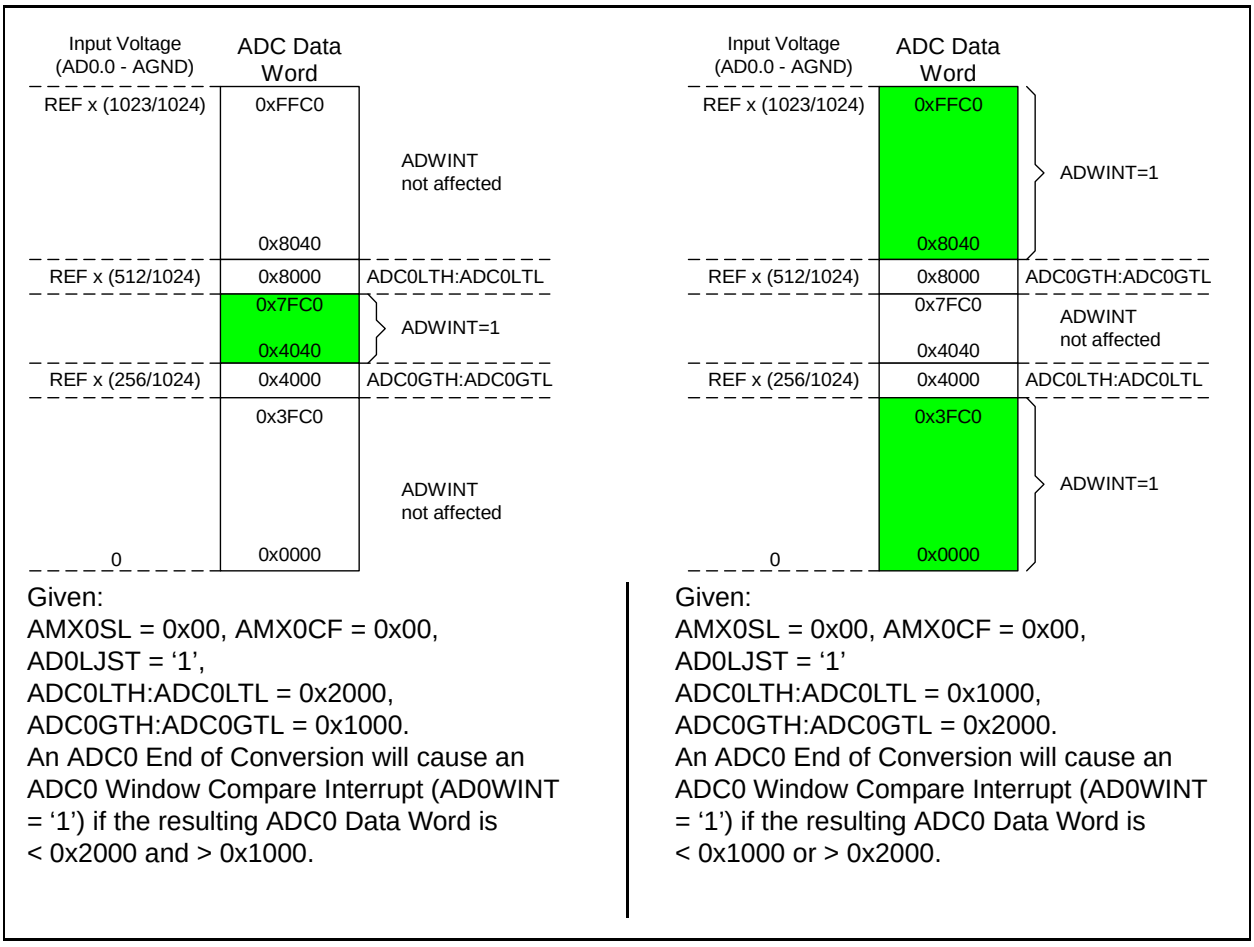
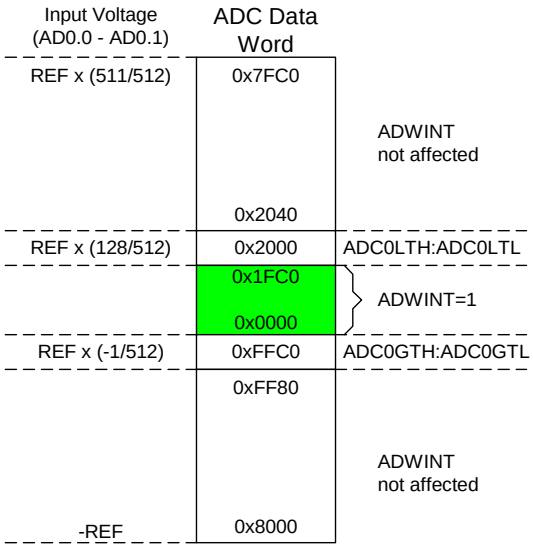
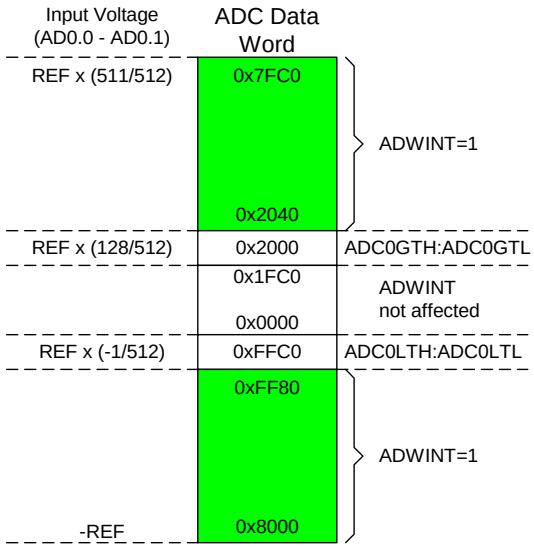



**Figure 6.8. 10-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data**




Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '1',
ADC0LTH:ADC0LTL = 0x2000,
ADC0GTH:ADC0GTL = 0xFFC0.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0x2000 and > 0xFFC0. (2s-complement math.)




Given:
AMX0SL = 0x00, AMX0CF = 0x01,
AD0LJST = '1',
ADC0LTH:ADC0LTL = 0xFFC0,
ADC0GTH:ADC0GTL = 0x2000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = '1') if the resulting ADC0 Data Word is < 0xFFC0 or > 0x2000. (2s-complement math.)

**Figure 6.9. 10-Bit ADC0 Window Interrupt Example: Left Justified Differential Data**

**Table 6.1. 10-Bit ADC0 Electrical Characteristics (C8051F122/3/6/7 and C8051F13x)**

$V_{DD}$ = 3.0 V, AV+ = 3.0 V, VREF = 2.40 V (REFBE = 0), PGA Gain = 1, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 10 | | bits |
| Integral Nonlinearity | | — | — | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | — | ±1 | LSB |
| Offset Error | | — | ±0.5 | — | LSB |
| Full Scale Error | Differential mode | — | –1.5±0.5 | — | LSB |
| Offset Temperature Coefficient | | — | ±0.25 | — | ppm/°C |
| **Dynamic Performance (10 kHz sine-wave input, 0 to 1 dB below Full Scale, 100 ksps** | | | | | |
| Signal-to-Noise Plus Distortion | | 59 | — | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | –70 | — | dB |
| Spurious-Free Dynamic Range | | — | 80 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Clock Frequency | | — | — | 2.5 | MHz |
| Conversion Time in SAR Clocks | | 16 | — | — | clocks |
| Track/Hold Acquisition Time | | 1.5 | — | — | µs |
| Throughput Rate | | — | — | 100 | ksps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | Single-ended operation | 0 | — | VREF | V |
| ***Common-mode Voltage Range** | Differential operation | AGND | — | AV+ | V |
| Input Capacitance | | — | 10 | — | pF |
| **Temperature Sensor** | | | | | |
| Linearity[1] | | — | ±0.2 | — | °C |
| Offset | (Temp = 0 °C) | — | 776 | — | mV |
| Offset Error[1,2] | (Temp = 0 °C) | — | ±8.5 | — | mV |
| Slope | | — | 2.86 | — | mV/°C |
| Slope Error[2] | | — | ±0.034 | — | mV/°C |
| **Power Specifications** | | | | | |
| Power Supply Current (AV+ supplied to ADC) | Operating Mode, 100 ksps | — | 450 | 900 | µA |
| Power Supply Rejection | | — | ±0.3 | — | mV/V |

**Notes:**
1. Includes ADC offset, gain, and linearity variations.
2. Represents one standard deviation from the mean.

# 7. ADC2 (8-Bit ADC, C8051F12x Only)

The C8051F12x devices include a second ADC peripheral (ADC2), which consists of an 8-channel, configurable analog multiplexer, a programmable gain amplifier, and a 500 ksps, 8-bit successive-approximation-register ADC with integrated track-and-hold (see block diagram in Figure 7.1). ADC2 is fully configurable under software control via the Special Function Registers shown in Figure 7.1. The ADC2 subsystem (8-bit ADC, track-and-hold and PGA) is enabled only when the AD2EN bit in the ADC2 Control register (ADC2CN) is set to logic 1. The ADC2 subsystem is in low power shutdown when this bit is logic 0. The voltage reference used by ADC2 is selected as described in **Section "9. Voltage Reference" on page 113**.

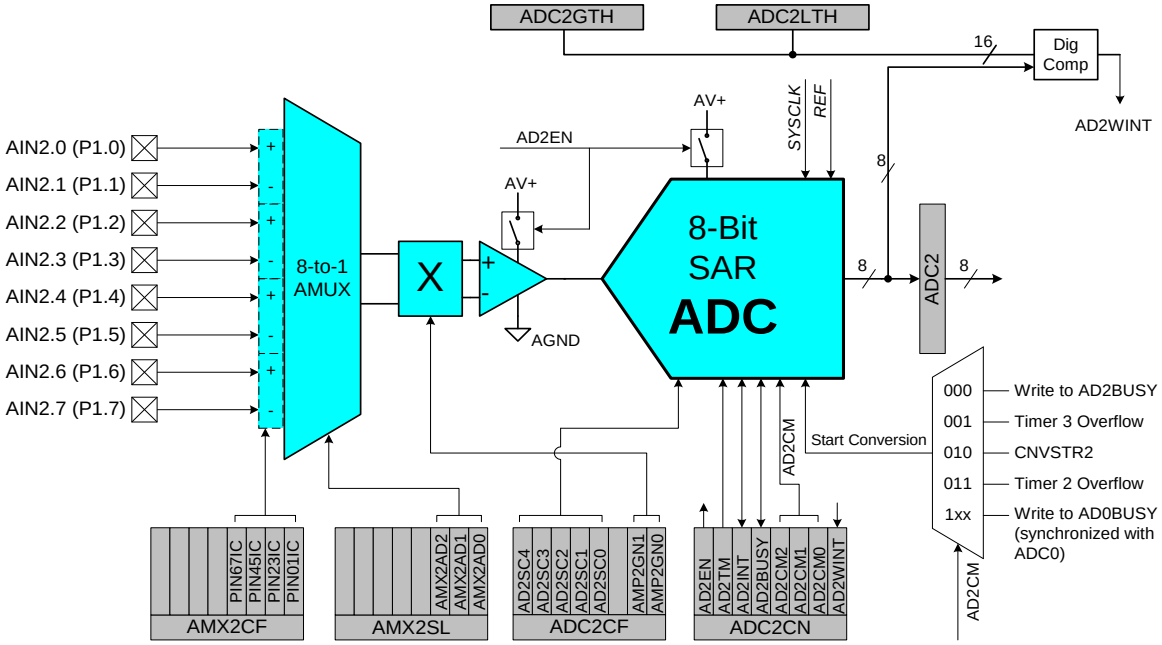



**Figure 7.1. ADC2 Functional Block Diagram**

## 7.1. Analog Multiplexer and PGA

Eight ADC2 channels are available for measurement, as selected by the AMX2SL register (see SFR Definition 7.2). The PGA amplifies the ADC2 output signal by an amount determined by the states of the AMP2GN2-0 bits in the ADC2 Configuration register, ADC2CF (SFR Definition 7.3). The PGA can be software-programmed for gains of 0.5, 1, 2, or 4. Gain defaults to 0.5 on reset.

**Important Note**: AIN2 pins also function as Port 1 I/O pins, and must be configured as analog inputs when used as ADC2 inputs. To configure an AIN2 pin for analog input, set to '0' the corresponding bit in register P1MDIN. Port 1 pins selected as analog inputs are skipped by the Digital I/O Crossbar. See **Section "18.1.5. Configuring Port 1 Pins as Analog Inputs" on page 240** for more information on configuring the AIN2 pins.

## 7.2. ADC2 Modes of Operation

ADC2 has a maximum conversion speed of 500 ksps. The ADC2 conversion clock (SAR2 clock) is a divided version of the system clock, determined by the AD2SC bits in the ADC2CF register. The maximum ADC2 conversion clock is 6 MHz.

### 7.2.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC2 Start of Conversion Mode bits (AD2CM2-0) in ADC2CN. Conversions may be initiated by:

1. Writing a '1' to the AD2BUSY bit of ADC2CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR2;
4. A Timer 2 overflow (i.e. timed continuous conversions);
5. Writing a '1' to the AD0BUSY of register ADC0CN (initiate conversion of ADC2 and ADC0 with a single software command).

During conversion, the AD2BUSY bit is set to logic 1 and restored to 0 when conversion is complete. The falling edge of AD2BUSY triggers an interrupt (when enabled) and sets the interrupt flag in ADC2CN. Converted data is available in the ADC2 data word, ADC2.

When a conversion is initiated by writing a '1' to AD2BUSY, it is recommended to poll AD2INT to determine when the conversion is complete. The recommended procedure is:

Step 1. Write a '0' to AD2INT;
Step 2. Write a '1' to AD2BUSY;
Step 3. Poll AD2INT for '1';
Step 4. Process ADC2 data.

When CNVSTR2 is used as a conversion start source, it must be enabled in the crossbar, and the corresponding pin must be set to open-drain, high-impedance mode (see **Section "18. Port Input/Output" on page 235** for more details on Port I/O configuration).

### 7.2.2. Tracking Modes

The AD2TM bit in register ADC2CN controls the ADC2 track-and-hold mode. In its default state, the ADC2 input is continuously tracked, except when a conversion is in progress. When the AD2TM bit is logic 1, ADC2 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 CNVSTR2 signal is used to initiate conversions in low-power tracking mode, ADC2 tracks only when CNVSTR2 is low; conversion begins on the rising edge of CNVSTR2 (see Figure 7.2). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power Track-and-Hold mode is also useful when AMUX or PGA settings are frequently changed, due to the settling time requirements described in **Section "7.2.3. Settling Time Requirements" on page 94**.

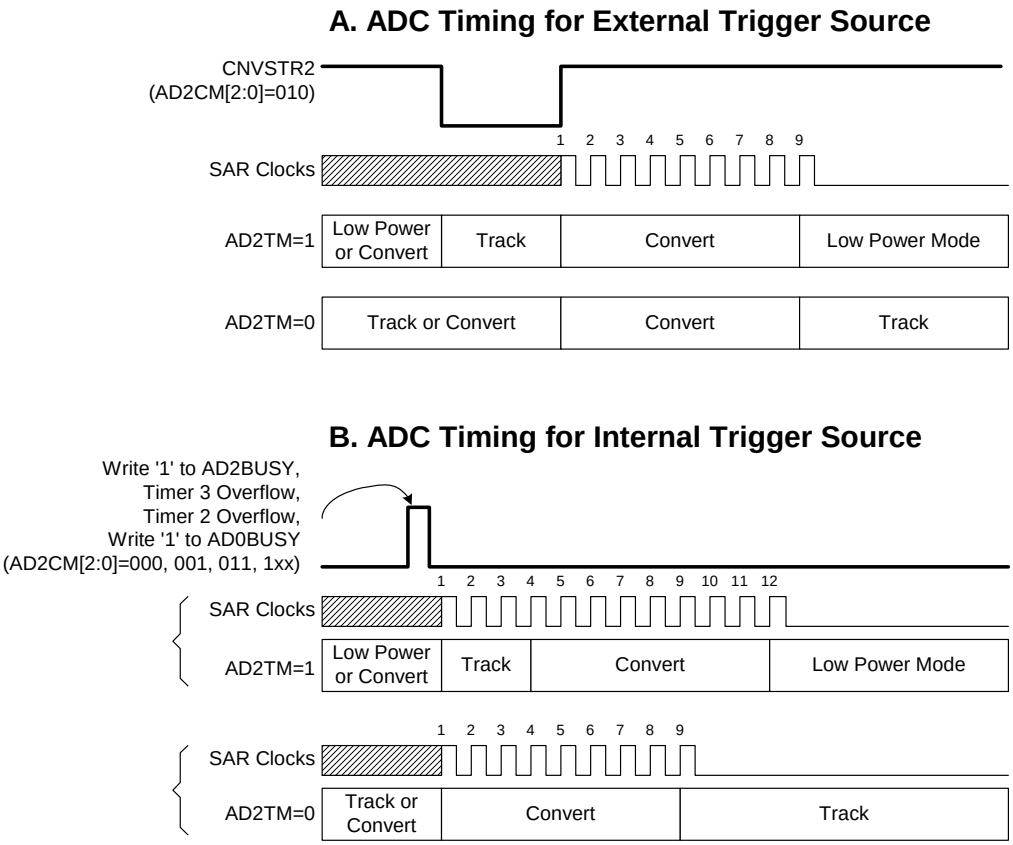



**Figure 7.2. ADC2 Track and Conversion Example Timing**

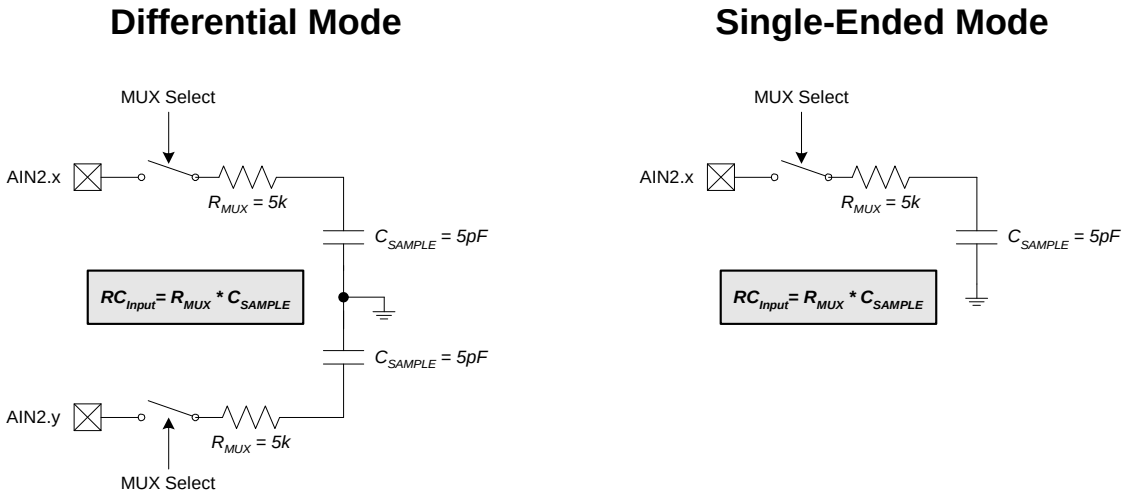
### 7.2.3. Settling Time Requirements

A minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the ADC2 MUX resistance, the ADC2 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 7.3 shows the equivalent ADC2 input circuit. The required ADC2 settling time for a given settling accuracy (SA) may be approximated by Equation 7.1. Note: An absolute minimum settling time of 800 ns required after any MUX selection. In low-power tracking mode, three SAR2 clocks are used for tracking at the start of every conversion. For most applications, these three SAR2 clocks will meet the tracking requirements.

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

**Equation 7.1. ADC2 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 ADC2 MUX resistance and any external source resistance.
*n* is the ADC resolution in bits (8).




Note: When the PGA gain is set to 0.5, $C_{SAMPLE}$ = 3pF

**Figure 7.3. ADC2 Equivalent Input Circuit**

## SFR Definition 7.1. AMX2CF: AMUX2 Configuration

SFR Page: 2
SFR Address: 0xBA

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | PIN67IC | PIN45IC | PIN23IC | PIN01IC | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bit3: PIN67IC: AIN2.6, AIN2.7 Input Pair Configuration Bit.
0: AIN2.6 and AIN2.7 are independent single-ended inputs.
1: AIN2.6 and AIN2.7 are (respectively) +, – differential input pair.
Bit2: PIN45IC: AIN2.4, AIN2.5 Input Pair Configuration Bit.
0: AIN2.4 and AIN2.5 are independent single-ended inputs.
1: AIN2.4 and AIN2.5 are (respectively) +, – differential input pair.
Bit1: PIN23IC: AIN2.2, AIN2.3 Input Pair Configuration Bit.
0: AIN2.2 and AIN2.3 are independent single-ended inputs.
1: AIN2.2 and AIN2.3 are (respectively) +, – differential input pair.
Bit0: PIN01IC: AIN2.0, AIN2.1 Input Pair Configuration Bit.
0: AIN2.0 and AIN2.1 are independent single-ended inputs.
1: AIN2.0 and AIN2.1 are (respectively) +, – differential input pair.

**Note:** The ADC2 Data Word is in 2's complement format for channels configured as differential.

## SFR Definition 7.2. AMX2SL: AMUX2 Channel Select

SFR Page: 2
SFR Address: 0xBB

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | | AMX2AD2 | AMX2AD1 | AMX2AD0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–3: UNUSED. Read = 00000b; Write = don't care.
Bits2–0: AMX2AD2–0: AMX2 Address Bits.
000-111b: ADC Inputs selected per chart below.

| | | AMX2AD2–0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| AMX2CF Bits 3–0 | 0000 | AIN2.0 | AIN2.1 | AIN2.2 | AIN2.3 | AIN2.4 | AIN2.5 | AIN2.6 | AIN2.7 |
| | 0001 | +(AIN2.0) –(AIN2.1) | | AIN2.2 | AIN2.3 | AIN2.4 | AIN2.5 | AIN2.6 | AIN2.7 |
| | 0010 | AIN2.0 | AIN2.1 | +(AIN2.2) –(AIN2.3) | | AIN2.4 | AIN2.5 | AIN2.6 | AIN2.7 |
| | 0011 | +(AIN2.0) –(AIN2.1) | | +(AIN2.2) –(AIN2.3) | | AIN2.4 | AIN2.5 | AIN2.6 | AIN2.7 |
| | 0100 | AIN2.0 | AIN2.1 | AIN2.2 | AIN2.3 | +(AIN2.4) –(AIN2.5) | | AIN2.6 | AIN2.7 |
| | 0101 | +(AIN2.0) –(AIN2.1) | | AIN2.2 | AIN2.3 | +(AIN2.4) –(AIN2.5) | | AIN2.6 | AIN2.7 |
| | 0110 | AIN2.0 | AIN2.1 | +(AIN2.2) –(AIN2.3) | | +(AIN2.4) –(AIN2.5) | | AIN2.6 | AIN2.7 |
| | 0111 | +(AIN2.0) –(AIN2.1) | | +(AIN2.2) –(AIN2.3) | | +(AIN2.4) –(AIN2.5) | | AIN2.6 | AIN2.7 |
| | 1000 | AIN2.0 | AIN2.1 | AIN2.2 | AIN2.3 | AIN2.4 | AIN2.5 | +(AIN2.6) –(AIN2.7) | |
| | 1001 | +(AIN2.0) –(AIN2.1) | | AIN2.2 | AIN2.3 | AIN2.4 | AIN2.5 | +(AIN2.6) –(AIN2.7) | |
| | 1010 | AIN2.0 | AIN2.1 | +(AIN2.2) –(AIN2.3) | | AIN2.4 | AIN2.5 | +(AIN2.6) –(AIN2.7) | |
| | 1011 | +(AIN2.0) –(AIN2.1) | | +(AIN2.2) –(AIN2.3) | | AIN2.4 | AIN2.5 | +(AIN2.6) –(AIN2.7) | |
| | 1100 | AIN2.0 | AIN2.1 | AIN2.2 | AIN2.3 | +(AIN2.4) –(AIN2.5) | | +(AIN2.6) –(AIN2.7) | |
| | 1101 | +(AIN2.0) –(AIN2.1) | | AIN2.2 | AIN2.3 | +(AIN2.4) –(AIN2.5) | | +(AIN2.6) –(AIN2.7) | |
| | 1110 | AIN2.0 | AIN2.1 | +(AIN2.2) –(AIN2.3) | | +(AIN2.4) –(AIN2.5) | | +(AIN2.6) –(AIN2.7) | |
| | 1111 | +(AIN2.0) –(AIN2.1) | | +(AIN2.2) –(AIN2.3) | | +(AIN2.4) –(AIN2.5) | | +(AIN2.6) –(AIN2.7) | |

## SFR Definition 7.3. ADC2CF: ADC2 Configuration

SFR Page: 2
SFR Address: 0xBC

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD2SC4 | AD2SC3 | AD2SC2 | AD2SC1 | AD2SC0 | - | AMP2GN1 | AMP2GN0 | 11111000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–3: AD2SC4–0: ADC2 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD2SC* refers to the 5-bit value held in AD2SC4–0, and $CLK_{SAR2}$ refers to the desired ADC2 SAR clock (Note: the ADC2 SAR Conversion Clock should be less than or equal to 6 MHz).

$$AD2SC = \frac{SYSCLK}{CLK_{SAR2}} - 1$$

Bit2: UNUSED. Read = 0b; Write = don't care.
Bits1–0: AMP2GN1–0: ADC2 Internal Amplifier Gain (PGA).
00: Gain = 0.5
01: Gain = 1
10: Gain = 2
11: Gain = 4

## SFR Definition 7.4. ADC2CN: ADC2 Control

SFR Page: 2
SFR Address: 0xE8 (bit addressable)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD2EN | AD2TM | AD2INT | AD2BUSY | AD2CM2 | AD2CM1 | AD2CM0 | AD2WINT | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit7: AD2EN: ADC2 Enable Bit.
0: ADC2 Disabled. ADC2 is in low-power shutdown.
1: ADC2 Enabled. ADC2 is active and ready for data conversions.

Bit6: AD2TM: ADC2 Track Mode Bit.
0: Normal Track Mode: When ADC2 is enabled, tracking is continuous unless a conversion is in process.
1: Low-power Track Mode: Tracking Defined by AD2CM2-0 bits (see below).

Bit5: AD2INT: ADC2 Conversion Complete Interrupt Flag.
This flag must be cleared by software.
0: ADC2 has not completed a data conversion since the last time this flag was cleared.
1: ADC2 has completed a data conversion.

Bit4: AD2BUSY: ADC2 Busy Bit.
Read:
0: ADC2 Conversion is complete or a conversion is not currently in progress. AD2INT is set to logic 1 on the falling edge of AD2BUSY.
1: ADC2 Conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC2 Conversion if AD2CM2-0 = 000b

Bits3–1: AD2CM2–0: ADC2 Start of Conversion Mode Select.
AD2TM = 0:
000: ADC2 conversion initiated on every write of ‘1’ to AD2BUSY.
001: ADC2 conversion initiated on overflow of Timer 3.
010: ADC2 conversion initiated on rising edge of external CNVSTR2.
011: ADC2 conversion initiated on overflow of Timer 2.
1xx: ADC2 conversion initiated on write of ‘1’ to AD0BUSY (synchronized with ADC0 software-commanded conversions).
AD2TM = 1:
000: Tracking initiated on write of ‘1’ to AD2BUSY for 3 SAR2 clocks, followed by conversion.
001: Tracking initiated on overflow of Timer 3 for 3 SAR2 clocks, followed by conversion.
010: ADC2 tracks only when CNVSTR2 input is logic low; conversion starts on rising CNVSTR2 edge.
011: Tracking initiated on overflow of Timer 2 for 3 SAR2 clocks, followed by conversion.
1xx: Tracking initiated on write of ‘1’ to AD0BUSY and lasts 3 SAR2 clocks, followed by conversion.

Bit0: AD2WINT: ADC2 Window Compare Interrupt Flag.
This bit must be cleared by software.
0: ADC2 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC2 Window Comparison Data match has occurred.

**SFR Definition 7.5. ADC2: ADC2 Data Word**

SFR Page: 2
SFR Address: 0xBE

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC2 Data Word.

**Single-Ended Example:**
**8-bit ADC Data Word appears in the ADC2 Data Word Register as follows:**
Example: ADC2 Data Word Conversion Map, Single-Ended AIN2.0 Input
(AMX2CF = 0x00; AMX2SL = 0x00)

| AIN2.0–AGND (Volts) | ADC2 |
|---|---|
| VREF * (255/256) | 0xFF |
| VREF * (128/256) | 0x80 |
| VREF * (64/256) | 0x40 |
| 0 | 0x00 |

$$Code = Vin \times \frac{Gain}{VREF} \times 256$$

**Differential Example:**
**8-bit ADC Data Word appears in the ADC2 Data Word Register as follows:**
Example: ADC2 Data Word Conversion Map, Differential AIN2.0-AIN2.1 Input
(AMX2CF = 0x01; AMX2SL = 0x00)

| AIN2.0–AIN2.1 (Volts) | ADC2 |
|---|---|
| VREF * (127/128) | 0x7F |
| VREF * (64/128) | 0x40 |
| 0 | 0x00 |
| –VREF * (64/128) | 0xC0 (-64d) |
| –VREF * (128/128) | 0x80 (-128d) |

$$Code = Vin \times \frac{Gain}{2 \times VREF} \times 256$$

**Figure 7.4. ADC2 Data Word Example**

## 7.3. ADC2 Programmable Window Detector

The ADC2 Programmable Window Detector continuously compares the ADC2 output 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 (AD2WINT in register ADC2CN) can also be used in polled mode. The ADC2 Greater-Than (ADC2GT) and Less-Than (ADC2LT) registers hold the comparison values. Example comparisons for Differential and Single-ended modes are shown in Figure 7.6 and Figure 7.5, respectively. Notice that 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 ADC2LT and ADC2GT registers.

### 7.3.1. Window Detector In Single-Ended Mode

Figure 7.5 shows two example window comparisons for Single-ended mode, with ADC2LT = 0x20 and ADC2GT = 0x10. Notice that in Single-ended mode, the codes vary from 0 to VREF*(255/256) and are represented as 8-bit unsigned integers. In the left example, an AD2WINT interrupt will be generated if the ADC2 conversion word (ADC2) is within the range defined by ADC2GT and ADC2LT (if 0x10 < ADC2 < 0x20). In the right example, and AD2WINT interrupt will be generated if ADC2 is outside of the range defined by ADC2GT and ADC2LT (if ADC2 < 0x10 or ADC2 > 0x20).

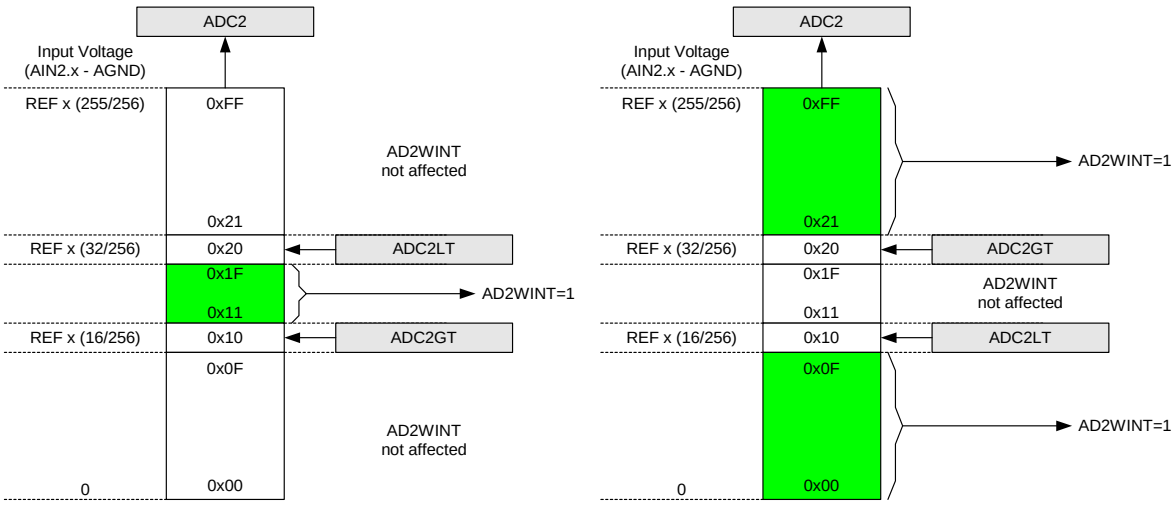



**Figure 7.5. ADC2 Window Compare Examples, Single-Ended Mode**

### 7.3.2. Window Detector In Differential Mode

Figure 7.6 shows two example window comparisons for differential mode, with ADC2LT = 0x10 (+16d) and ADC2GT = 0xFF (-1d). Notice that in Differential mode, the codes vary from -VREF to VREF*(127/128) and are represented as 8-bit 2's complement signed integers. In the left example, an AD2WINT interrupt will be generated if the ADC2 conversion word (ADC2L) is within the range defined by ADC2GT and ADC2LT (if 0xFF (-1d) < ADC2 < 0x0F (16d)). In the right example, an AD2WINT interrupt will be generated if ADC2 is outside of the range defined by ADC2GT and ADC2LT (if ADC2 < 0xFF (-1d) or ADC2 > 0x10 (+16d)).

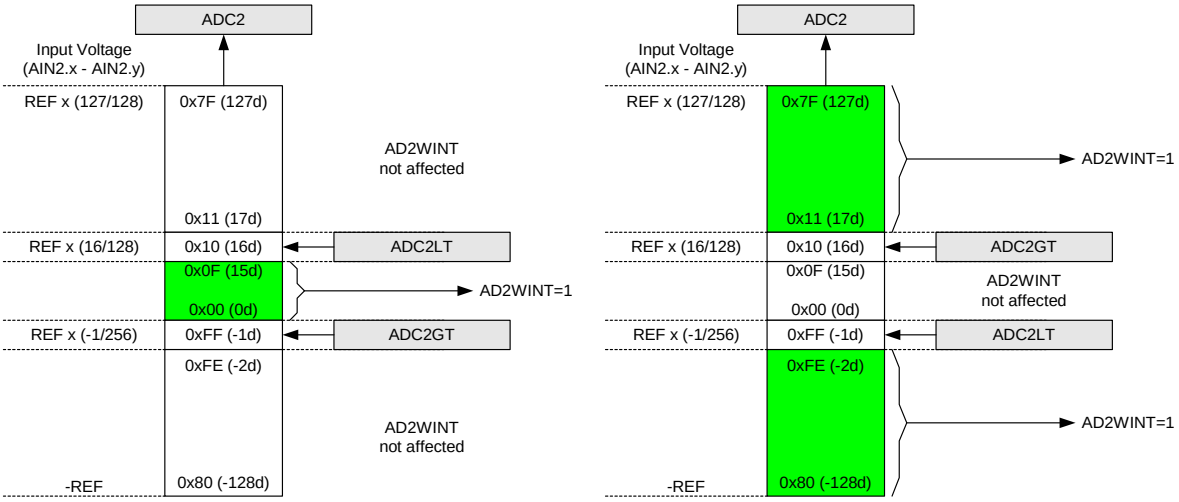



**Figure 7.6. ADC2 Window Compare Examples, Differential Mode**

### SFR Definition 7.6. ADC2GT: ADC2 Greater-Than Data Byte

SFR Page: 2
SFR Address: 0xC4

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC2 Greater-Than Data Word.

### SFR Definition 7.7. ADC2LT: ADC2 Less-Than Data Byte

SFR Page: 2
SFR Address: 0xC6

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–0: ADC2 Less-Than Data Word.

## Table 7.1. ADC2 Electrical Characteristics

$V_{DD}$ = 3.0 V, AV+ = 3.0 V, VREF2 = 2.40 V (REFBE = 0), PGA gain = 1, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 8 | | bits |
| Integral Nonlinearity | | — | — | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | — | — | ±1 | LSB |
| Offset Error | | — | 0.5±0.3 | — | LSB |
| Full Scale Error | Differential mode | — | –1±0.2 | — | LSB |
| Offset Temperature Coefficient | | — | 10 | — | ppm/°C |
| **Dynamic Performance (10 kHz sine-wave input, 1 dB below Full Scale, 500 ksps** | | | | | |
| Signal-to-Noise Plus Distortion | | 45 | 47 | — | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | — | -51 | — | dB |
| Spurious-Free Dynamic Range | | — | 52 | — | dB |
| **Conversion Rate** | | | | | |
| SAR Clock Frequency | | — | — | 6 | MHz |
| Conversion Time in SAR Clocks | | 8 | — | — | clocks |
| Track/Hold Acquisition Time | | 300 | — | — | ns |
| Throughput Rate | | — | — | 500 | ksps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | | 0 | — | VREF | V |
| Input Capacitance | | — | 5 | — | pF |
| **Power Specifications** | | | | | |
| Power Supply Current (AV+ supplied to ADC2) | Operating Mode, 500 ksps | — | 420 | 900 | µA |
| Power Supply Rejection | | — | ±0.3 | — | mV/V |

## NOTES:

# 8. DACs, 12-Bit Voltage Mode (C8051F12x Only)

The C8051F12x devices include two on-chip 12-bit voltage-mode Digital-to-Analog Converters (DACs). Each DAC has an output swing of 0 V to (VREF-1LSB) for a corresponding input code range of 0x000 to 0xFFF. The DACs may be enabled/disabled via their corresponding control registers, DAC0CN and DAC1CN. While disabled, the DAC output is maintained in a high-impedance state, and the DAC supply current falls to 1 µA or less. The voltage reference for each DAC is supplied at the VREFD pin (C8051F120/2/4/6 devices) or the VREF pin (C8051F121/3/5/7 devices). Note that the VREF pin on C8051F121/3/5/7 devices may be driven by the internal voltage reference or an external source. If the internal voltage reference is used it must be enabled in order for the DAC outputs to be valid. See **Section "9. Voltage Reference" on page 113** for more information on configuring the voltage reference for the DACs.

## 8.1. DAC Output Scheduling

Each DAC features a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. The following examples are written in terms of DAC0, but DAC1 operation is identical.

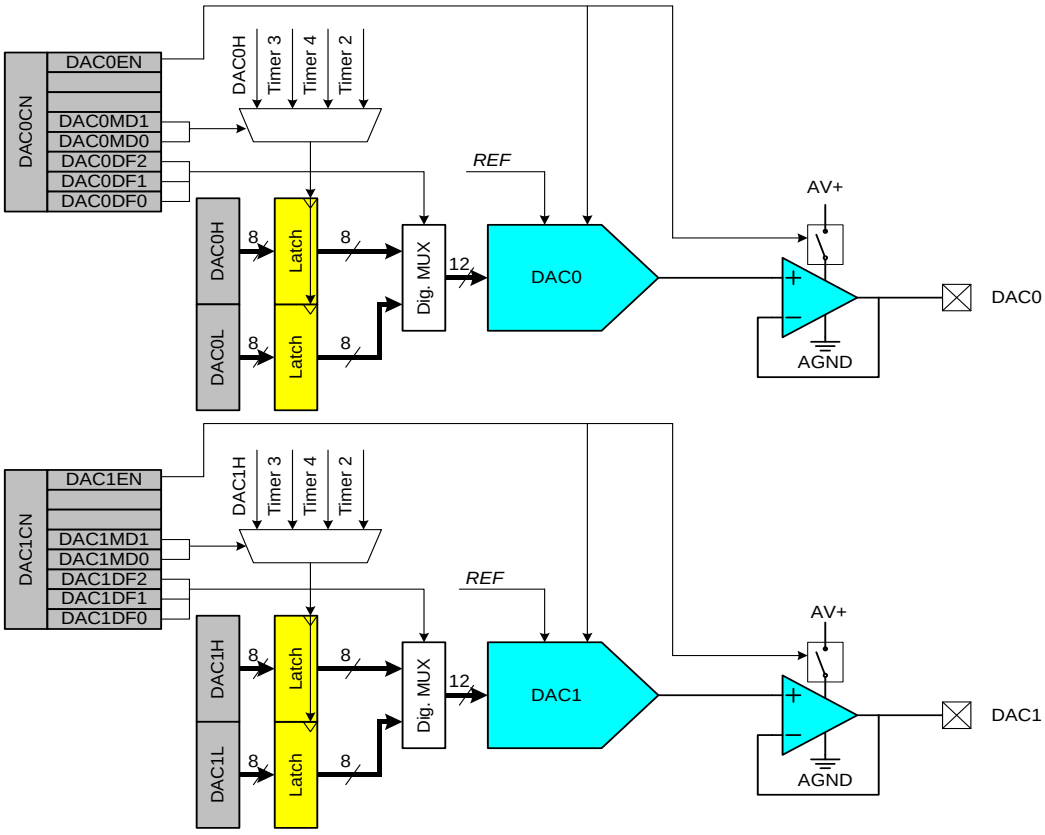



**Figure 8.1. DAC Functional Block Diagram**

### 8.1.1. Update Output On-Demand

In its default mode (DAC0CN.[4:3] = ‘00’) the DAC0 output is updated “on-demand” on a write to the high-byte of the DAC0 data register (DAC0H). It is important to note that writes to DAC0L are held, and have no effect on the DAC0 output until a write to DAC0H takes place. If writing a full 12-bit word to the DAC data registers, the 12-bit data word is written to the low byte (DAC0L) and high byte (DAC0H) data registers. Data is latched into DAC0 after a write to the corresponding DAC0H register, **so the write sequence should be DAC0L followed by DAC0H** if the full 12-bit resolution is required. The DAC can be used in 8-bit mode by initializing DAC0L to the desired value (typically 0x00), and writing data to only DAC0H (also see **Section 8.2** for information on formatting the 12-bit DAC data word within the 16-bit SFR space).

### 8.1.2. Update Output Based on Timer Overflow

Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the DAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the DAC is used to generate a waveform of a defined sampling rate by eliminating the effects of variable interrupt latency and instruction execution on the timing of the DAC output. When the DAC0MD bits (DAC0CN.[4:3]) are set to ‘01’, ‘10’, or ‘11’, writes to both DAC data registers (DAC0L and DAC0H) are held until an associated Timer overflow event (Timer 3, Timer 4, or Timer 2, respectively) occurs, at which time the DAC0H:DAC0L contents are copied to the DAC input latches allowing the DAC output to change to the new value.

## 8.2. DAC Output Scaling/Justification

In some instances, input data should be shifted prior to a DAC0 write operation to properly justify data within the DAC input registers. This action would typically require one or more load and shift operations, adding software overhead and slowing DAC throughput. To alleviate this problem, the data-formatting feature provides a means for the user to program the orientation of the DAC0 data word within data registers DAC0H and DAC0L. The three DAC0DF bits (DAC0CN.[2:0]) allow the user to specify one of five data word orientations as shown in the DAC0CN register definition.

DAC1 is functionally the same as DAC0 described above. The electrical specifications for both DAC0 and DAC1 are given in Table 8.1.

### SFR Definition 8.1. DAC0H: DAC0 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD3
SFR Page: 0

Bits7–0: DAC0 Data Word Most Significant Byte.

### SFR Definition 8.2. DAC0L: DAC0 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD2
SFR Page: 0

Bits7–0: DAC0 Data Word Least Significant Byte.

## SFR Definition 8.3. DAC0CN: DAC0 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| DAC0EN | - | - | DAC0MD1 | DAC0MD0 | DAC0DF2 | DAC0DF1 | DAC0DF0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD4
SFR Page: 0

Bit7: DAC0EN: DAC0 Enable Bit.
0: DAC0 Disabled. DAC0 Output pin is disabled; DAC0 is in low-power shutdown mode.
1: DAC0 Enabled. DAC0 Output pin is active; DAC0 is operational.

Bits6–5: UNUSED. Read = 00b; Write = don't care.

Bits4–3: DAC0MD1–0: DAC0 Mode Bits.
00: DAC output updates occur on a write to DAC0H.
01: DAC output updates occur on Timer 3 overflow.
10: DAC output updates occur on Timer 4 overflow.
11: DAC output updates occur on Timer 2 overflow.

Bits2–0: DAC0DF2–0: DAC0 Data Format Bits:

000: The most significant nibble of the DAC0 Data Word is in DAC0H[3:0], while the least significant byte is in DAC0L.

| DAC0H | | | | | | | | DAC0L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MSB | | | | | | | | | | | LSB |

001: The most significant 5-bits of the DAC0 Data Word is in DAC0H[4:0], while the least significant 7-bits are in DAC0L[7:1].

| DAC0H | | | | | | | | DAC0L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MSB | | | | | | | | | | | LSB | |

010: The most significant 6-bits of the DAC0 Data Word is in DAC0H[5:0], while the least significant 6-bits are in DAC0L[7:2].

| DAC0H | | | | | | | | DAC0L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSB | | | | | | | | | | | LSB | | |

011: The most significant 7-bits of the DAC0 Data Word is in DAC0H[6:0], while the least significant 5-bits are in DAC0L[7:3].

| DAC0H | | | | | | | | DAC0L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSB | | | | | | | | | | | LSB | | | |

1xx: The most significant 8-bits of the DAC0 Data Word is in DAC0H[7:0], while the least significant 4-bits are in DAC0L[7:4].

| DAC0H | | | | | | | | DAC0L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSB | | | | | | | | | | | LSB | | | | |

### SFR Definition 8.4. DAC1H: DAC1 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD3
SFR Page: 1

Bits7–0: DAC1 Data Word Most Significant Byte.

### SFR Definition 8.5. DAC1L: DAC1 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD2
SFR Page: 1

Bits7–0: DAC1 Data Word Least Significant Byte.

## SFR Definition 8.6. DAC1CN: DAC1 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| DAC1EN | - | - | DAC1MD1 | DAC1MD0 | DAC1DF2 | DAC1DF1 | DAC1DF0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD4
SFR Page: 1

Bit7: DAC1EN: DAC1 Enable Bit.
0: DAC1 Disabled. DAC1 Output pin is disabled; DAC1 is in low-power shutdown mode.
1: DAC1 Enabled. DAC1 Output pin is active; DAC1 is operational.

Bits6–5: UNUSED. Read = 00b; Write = don't care.

Bits4–3: DAC1MD1–0: DAC1 Mode Bits:
00: DAC output updates occur on a write to DAC1H.
01: DAC output updates occur on Timer 3 overflow.
10: DAC output updates occur on Timer 4 overflow.
11: DAC output updates occur on Timer 2 overflow.

Bits2–0: DAC1DF2: DAC1 Data Format Bits:

000: The most significant nibble of the DAC1 Data Word is in DAC1H[3:0], while the least significant byte is in DAC1L.

| DAC1H | | | | | | | | DAC1L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MSB | | | | | | | | | | | LSB |

001: The most significant 5-bits of the DAC1 Data Word is in DAC1H[4:0], while the least significant 7-bits are in DAC1L[7:1].

| DAC1H | | | | | | | | DAC1L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | MSB | | | | | | | | | | | LSB | |

010: The most significant 6-bits of the DAC1 Data Word is in DAC1H[5:0], while the least significant 6-bits are in DAC1L[7:2].

| DAC1H | | | | | | | | DAC1L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSB | | | | | | | | | | | LSB | | |

011: The most significant 7-bits of the DAC1 Data Word is in DAC1H[6:0], while the least significant 5-bits are in DAC1L[7:3].

| DAC1H | | | | | | | | DAC1L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSB | | | | | | | | | | | LSB | | | |

1xx: The most significant 8-bits of the DAC1 Data Word is in DAC1H[7:0], while the least significant 4-bits are in DAC1L[7:4].

| DAC1H | | | | | | | | DAC1L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSB | | | | | | | | | | | LSB | | | | |

.

## Table 8.1. DAC Electrical Characteristics

$V_{DD}$ = 3.0 V, AV+ = 3.0 V, VREF = 2.40 V (REFBE = 0), No Output Load unless otherwise specified

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Static Performance** | | | | | |
| Resolution | | | 12 | | bits |
| Integral Nonlinearity | | — | ±1.5 | — | LSB |
| Differential Nonlinearity | | — | — | ±1 | LSB |
| Output Noise | No Output Filter<br>100 kHz Output Filter<br>10 kHz Output Filter | — | 250<br>128<br>41 | — | µVrms |
| Offset Error | Data Word = 0x014 | — | ±3 | ±30 | mV |
| Offset Tempco | | — | 6 | — | ppm/°C |
| Full-Scale Error | | — | ±20 | ±60 | mV |
| Full-Scale Error Tempco | | — | 10 | — | ppm/°C |
| $V_{DD}$ Power Supply Rejection Ratio | | — | –60 | — | dB |
| Output Impedance in Shutdown Mode | DACnEN = 0 | — | 100 | — | kΩ |
| Output Sink Current | | — | 300 | — | µA |
| Output Short-Circuit Current | Data Word = 0xFFF | — | 15 | — | mA |
| **Dynamic Performance** | | | | | |
| Voltage Output Slew Rate | Load = 40 pF | — | 0.44 | — | V/µs |
| Output Settling Time to 1/2 LSB | Load = 40 pF, Output swing from code 0xFFF to 0x014 | — | 10 | — | µs |
| Output Voltage Swing | | 0 | — | VREF-1LSB | V |
| Startup Time | | — | 10 | — | µs |
| **Analog Outputs** | | | | | |
| Load Regulation | $I_L$ = 0.01 mA to 0.3 mA at code 0xFFF | — | 60 | — | ppm |
| **Power Consumption (each DAC)** | | | | | |
| Power Supply Current (AV+ supplied to DAC) | Data Word = 0x7FF | — | 110 | 400 | µA |

## NOTES:

# 9. Voltage Reference

The voltage reference options available on the C8051F12x and C8051F13x device families vary according to the device capabilities.

All devices include an internal voltage reference circuit, consisting of a 1.2 V, 15 ppm/°C (typical) bandgap voltage reference generator and a gain-of-two output buffer amplifier. The internal reference may be routed via the VREF pin to external system components or to the voltage reference input pins. The maximum load seen by the VREF pin must be less than 200 µA to AGND. Bypass capacitors of 0.1 µF and 4.7 µF are recommended from the VREF pin to AGND.

The Reference Control Register, REF0CN enables/disables the internal reference generator and the internal temperature sensor on all devices. The BIASE bit in REF0CN enables the on-board reference generator while the REFBE bit enables the gain-of-two buffer amplifier which drives the VREF pin. When disabled, the supply current drawn by the bandgap and buffer amplifier falls to less than 1 µA (typical) and the output of the buffer amplifier enters a high impedance state. If the internal bandgap is used as the reference voltage generator, BIASE and REFBE must both be set to logic 1. If the internal reference is not used, REFBE may be set to logic 0. Note that the BIASE bit must be set to logic 1 if any DACs or ADCs are used, regardless of whether the voltage reference is derived from the on-chip reference or supplied by an off-chip source. If no ADCs or DACs are being used, both of these bits can be set to logic 0 to conserve power.

When enabled, the temperature sensor connects to the highest order input of the ADC0 input multiplexer. The TEMPE bit within REF0CN enables and disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state. Any ADC measurements performed on the sensor while disabled will result in undefined data.

The electrical specifications for the internal voltage reference are given in Table 9.1.

## 9.1. Reference Configuration on the C8051F120/2/4/6

On the C8051F120/2/4/6 devices, the REF0CN register also allows selection of the voltage reference source for ADC0 and ADC2, as shown in SFR Definition 9.1. Bits AD0VRS and AD2VRS in the REF0CN register select the ADC0 and ADC2 voltage reference sources, respectively. Three voltage reference input pins allow each ADC and the two DACs to reference an external voltage reference or the on-chip voltage reference output (with an external connection). ADC0 may also reference the DAC0 output internally, and ADC2 may reference the analog power supply voltage, via the VREF multiplexers shown in Figure 9.1.

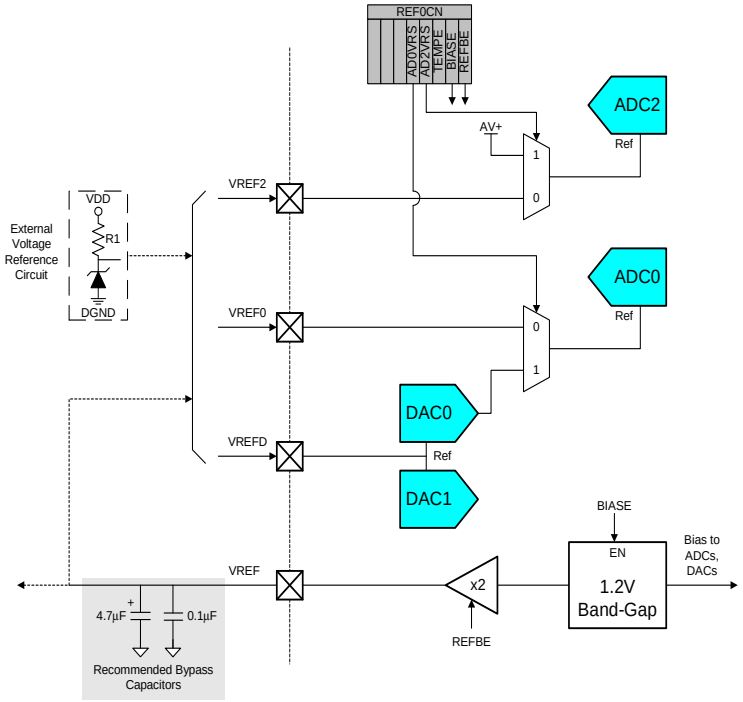



**Figure 9.1. Voltage Reference Functional Block Diagram (C8051F120/2/4/6)**

### SFR Definition 9.1. REF0CN: Reference Control (C8051F120/2/4/6)

SFR Page: 0
SFR Address: 0xD1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | AD0VRS | AD2VRS | TEMPE | BIASE | REFBE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bit4: AD0VRS: ADC0 Voltage Reference Select.
0: ADC0 voltage reference from VREF0 pin.
1: ADC0 voltage reference from DAC0 output.
Bit3: AD2VRS: ADC2 Voltage Reference Select.
0: ADC2 voltage reference from VREF2 pin.
1: ADC2 voltage reference from AV+.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC, DAC, or VREF).
0: Internal Bias Generator Off.
1: Internal Bias Generator On.
Bit0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer Off.
1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin.

## 9.2. Reference Configuration on the C8051F121/3/5/7

On the C8051F121/3/5/7 devices, the REF0CN register also allows selection of the voltage reference source for ADC0 and ADC2, as shown in SFR Definition 9.2. Bits AD0VRS and AD2VRS in the REF0CN register select the ADC0 and ADC2 voltage reference sources, respectively. The VREFA pin provides a voltage reference input for ADC0 and ADC2, which can be connected to an external precision reference or the internal voltage reference. ADC0 may also reference the DAC0 output internally, and ADC2 may reference the analog power supply voltage, via the VREF multiplexers shown in Figure 9.2.

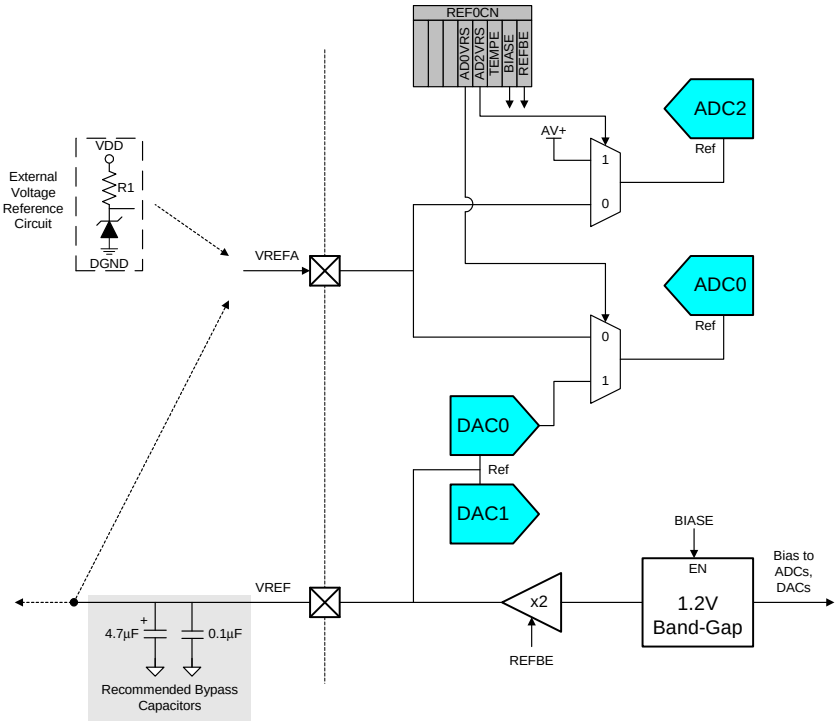



**Figure 9.2. Voltage Reference Functional Block Diagram (C8051F121/3/5/7)**

## SFR Definition 9.2. REF0CN: Reference Control (C8051F121/3/5/7)

SFR Page: 0
SFR Address: 0xD1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | AD0VRS | AD2VRS | TEMPE | BIASE | REFBE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bit4: AD0VRS: ADC0 Voltage Reference Select.
0: ADC0 voltage reference from VREFA pin.
1: ADC0 voltage reference from DAC0 output.
Bit3: AD2VRS: ADC2 Voltage Reference Select.
0: ADC2 voltage reference from VREFA pin.
1: ADC2 voltage reference from AV+.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC, DAC, or VREF).
0: Internal Bias Generator Off.
1: Internal Bias Generator On.
Bit0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer Off.
1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin.

## 9.3. Reference Configuration on the C8051F130/1/2/3

On the C8051F130/1/2/3 devices, the VREF0 pin provides a voltage reference input for ADC0, which can be connected to an external precision reference or the internal voltage reference, as shown in Figure 9.3. The REF0CN register for the C8051F130/1/2/3 is described in SFR Definition 9.3.

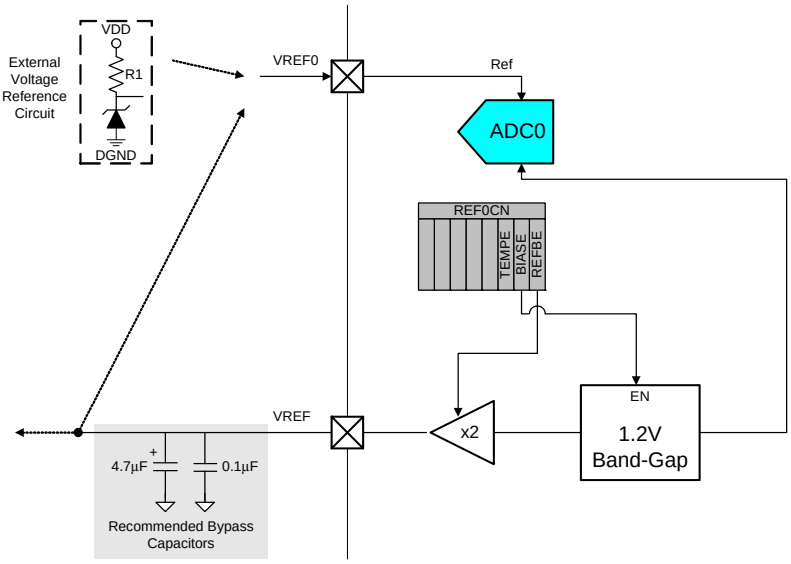



**Figure 9.3. Voltage Reference Functional Block Diagram (C8051F130/1/2/3)**

### SFR Definition 9.3. REF0CN: Reference Control (C8051F130/1/2/3)

SFR Page: 0
SFR Address: 0xD1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | Reserved | Reserved | TEMPE | BIASE | REFBE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bits4–3: Reserved: Must be written to 0.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC or VREF).
0: Internal Bias Generator Off.
1: Internal Bias Generator On.
Bit0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer Off.
1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin.

### Table 9.1. Voltage Reference Electrical Characteristics

$V_{DD}$ = 3.0 V, AV+ = 3.0 V, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Analog Bias Generator Power Supply Current | BIASE = 1 | — | 100 | — | µA |
| **Internal Reference (REFBE = 1)** | | | | | |
| Output Voltage | 25 °C ambient | 2.36 | 2.43 | 2.48 | V |
| VREF Short-Circuit Current | | — | — | 30 | mA |
| VREF Temperature Coefficient | | — | 15 | — | ppm/°C |
| Load Regulation | Load = 0 to 200 µA to AGND | — | 0.5 | — | ppm/µA |
| VREF Turn-on Time 1 | 4.7 µF tantalum, 0.1 µF ceramic bypass | — | 2 | — | ms |
| VREF Turn-on Time 2 | 0.1 µF ceramic bypass | — | 20 | — | µs |
| VREF Turn-on Time 3 | no bypass cap | — | 10 | — | µs |
| Reference Buffer Power Supply Current | | — | 40 | — | µA |
| Power Supply Rejection | | — | 140 | — | ppm/V |
| **External Reference (REFBE = 0)** | | | | | |
| Input Voltage Range | | 1.00 | — | (AV+) – 0.3 | V |
| Input Current | | — | 0 | 1 | µA |

## 10. Comparators

Two on-chip programmable voltage comparators are included, as shown in Figure 10.1. The inputs of each comparator are available at dedicated pins. The output of each comparator is optionally available at the package pins via the I/O crossbar. When assigned to package pins, each comparator output can be programmed to operate in open drain or push-pull modes. See **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238** for Crossbar and port initialization details.

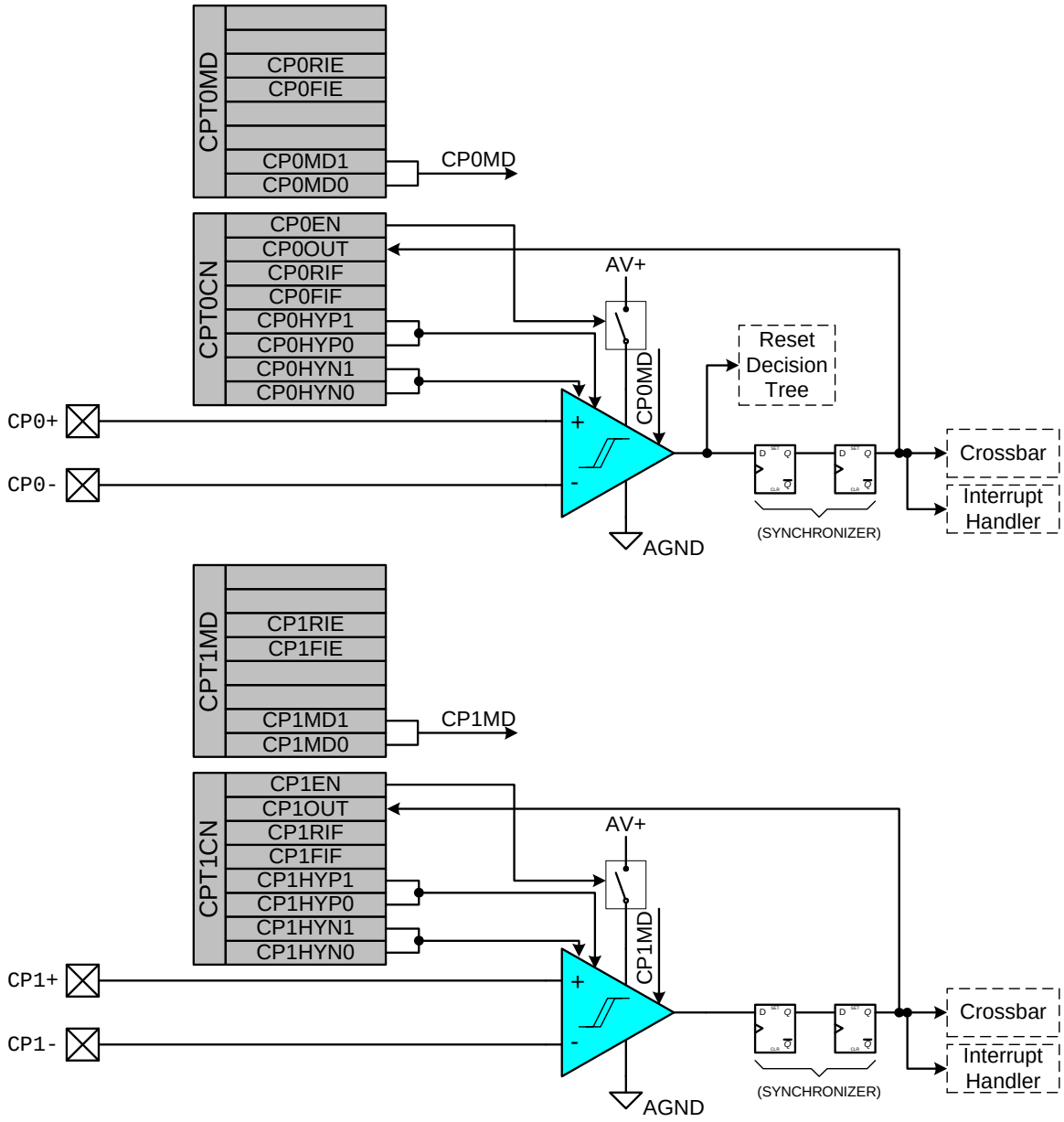



**Figure 10.1. Comparator Functional Block Diagram**

Comparator interrupts can be generated on rising-edge and/or falling-edge output transitions. (For interrupt enable and priority control, see **Section "11.3. Interrupt Handler" on page 154**). The CP0FIF flag is set upon a Comparator0 falling-edge interrupt, and the CP0RIF flag is set upon the Comparator0 rising-edge interrupt. Once set, these bits remain set until cleared by software. The Output State of Comparator0 can be obtained at any time by reading the CP0OUT bit. Comparator0 is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0. Comparator0 can also be programmed as a reset source; for details, see Section "**13.5. Comparator0 Reset**" on page **179**.

Note that after being enabled, there is a Power-Up time (listed in Table 10.1) during which the comparator outputs stabilize. The states of the Rising-Edge and Falling-Edge flags are indeterminant after comparator Power-Up and should be explicitly cleared before the comparator interrupts are enabled or the comparators are configured as a reset source.

Comparator0 response time may be configured in software via the CP0MD1-0 bits in register CPT0MD (see SFR Definition 10.2). Selecting a longer response time reduces the amount of current consumed by Comparator0. See Table 10.1 for complete timing and current consumption specifications.

The hysteresis of each comparator is software-programmable via its respective Comparator control register (CPT0CN and CPT1CN for Comparator0 and Comparator1, respectively). 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. The output of the comparator can be polled in software, or can be used as an interrupt source. Each comparator can be individually enabled or disabled (shutdown). When disabled, the comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, its interrupt capability is suspended and its supply current falls to less than 100 nA. Comparator inputs can be externally driven from –0.25 V to (AV+) + 0.25 V without damage or upset.

Comparator0 hysteresis is programmed using bits 3-0 in the Comparator0 Control Register CPT0CN (shown in SFR Definition 10.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in SFR Definition 10.1, the negative hysteresis can be programmed to three different settings, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

The operation of Comparator1 is identical to that of Comparator0, though Comparator1 may not be configured as a reset source. Comparator1 is controlled by the CPT1CN Register (SFR Definition 10.3) and the CPT1MD Register (SFR Definition 10.4). The complete electrical specifications for the Comparators are given in Table 10.1.

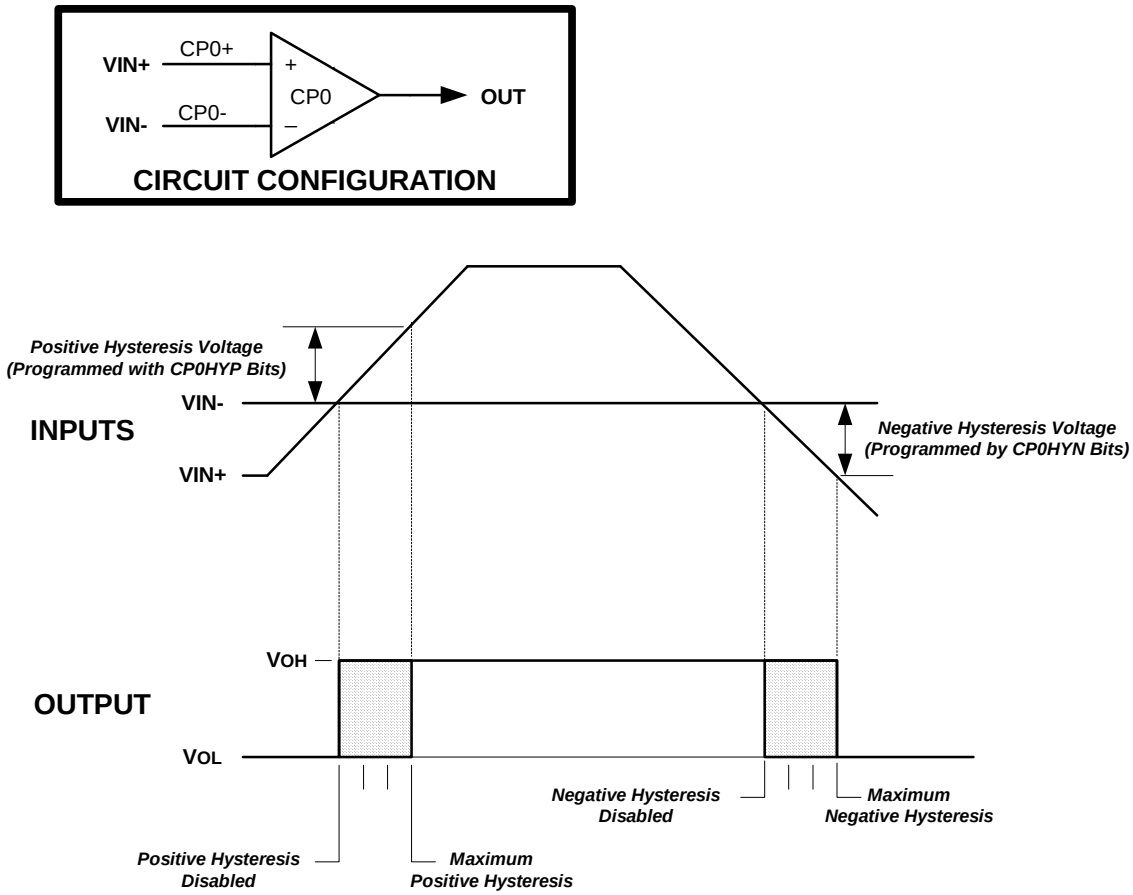



**Figure 10.2. Comparator Hysteresis Plot**

## SFR Definition 10.1. CPT0CN: Comparator0 Control

SFR Page: 1
SFR Address: 0x88

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CP0EN | CP0OUT | CP0RIF | CP0FIF | CP0HYP1 | CP0HYP0 | CP0HYN1 | CP0HYN0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit7: CP0EN: Comparator0 Enable Bit.
0: Comparator0 Disabled.
1: Comparator0 Enabled.

Bit6: CP0OUT: Comparator0 Output State Flag.
0: Voltage on CP0+ < CP0–.
1: Voltage on CP0+ > CP0–.

Bit5: CP0RIF: Comparator0 Rising-Edge Flag.
0: No Comparator0 Rising Edge has occurred since this flag was last cleared.
1: Comparator0 Rising Edge has occurred.

Bit4: CP0FIF: Comparator0 Falling-Edge Flag.
0: No Comparator0 Falling-Edge has occurred since this flag was last cleared.
1: Comparator0 Falling-Edge has occurred.

Bits3–2: CP0HYP1–0: Comparator0 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 15 mV.

Bits1–0: CP0HYN1–0: Comparator0 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 15 mV.

## SFR Definition 10.2. CPT0MD: Comparator0 Mode Selection

SFR Page: 1
SFR Address: 0x89

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | CP0RIE | CP0FIE | - | - | CP0MD1 | CP0MD0 | 00000010 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–6: UNUSED. Read = 00b, Write = don't care.
Bit 5: CP0RIE: Comparator 0 Rising-Edge Interrupt Enable Bit.
0: Comparator 0 rising-edge interrupt disabled.
1: Comparator 0 rising-edge interrupt enabled.
Bit 4: CP0FIE: Comparator 0 Falling-Edge Interrupt Enable Bit.
0: Comparator 0 falling-edge interrupt disabled.
1: Comparator 0 falling-edge interrupt enabled.
Bits3–2: UNUSED. Read = 00b, Write = don't care.
Bits1–0: CP0MD1–CP0MD0: Comparator0 Mode Select
These bits select the response time for Comparator0.

| Mode | CP0MD1 | CP0MD0 | Notes |
|---|---|---|---|
| 0 | 0 | 0 | Fastest Response Time |
| 1 | 0 | 1 | — |
| 2 | 1 | 0 | — |
| 3 | 1 | 1 | Lowest Power Consumption |

## SFR Definition 10.3. CPT1CN: Comparator1 Control

SFR Page: 2
SFR Address: 0x88

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CP1EN | CP1OUT | CP1RIF | CP1FIF | CP1HYP1 | CP1HYP0 | CP1HYN1 | CP1HYN0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit7: CP1EN: Comparator1 Enable Bit.
0: Comparator1 Disabled.
1: Comparator1 Enabled.

Bit6: CP1OUT: Comparator1 Output State Flag.
0: Voltage on CP1+ < CP1–.
1: Voltage on CP1+ > CP1–.

Bit5: CP1RIF: Comparator1 Rising-Edge Flag.
0: No Comparator1 Rising Edge has occurred since this flag was last cleared.
1: Comparator1 Rising Edge has occurred.

Bit4: CP1FIF: Comparator1 Falling-Edge Flag.
0: No Comparator1 Falling-Edge has occurred since this flag was last cleared.
1: Comparator1 Falling-Edge Interrupt has occurred.

Bits3–2: CP1HYP1–0: Comparator1 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 15 mV.

Bits1–0: CP1HYN1–0: Comparator1 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 15 mV.

## SFR Definition 10.4. CPT1MD: Comparator1 Mode Selection

SFR Page: 2
SFR Address: 0x89

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | CP1RIE | CP1FIE | - | - | CP1MD1 | CP1MD0 | 00000010 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits7–6: UNUSED. Read = 00b, Write = don't care.
Bit 5: CP1RIE: Comparator 1 Rising-Edge Interrupt Enable Bit.
0: Comparator 1 rising-edge interrupt disabled.
1: Comparator 1 rising-edge interrupt enabled.
Bit 4: CP1FIE: Comparator 0 Falling-Edge Interrupt Enable Bit.
0: Comparator 1 falling-edge interrupt disabled.
1: Comparator 1 falling-edge interrupt enabled.
Bits3–2: UNUSED. Read = 00b, Write = don't care.
Bits1–0: CP1MD1–CP1MD0: Comparator1 Mode Select
These bits select the response time for Comparator1.

| Mode | CP0MD1 | CP0MD0 | Notes |
|---|---|---|---|
| 0 | 0 | 0 | Fastest Response Time |
| 1 | 0 | 1 | — |
| 2 | 1 | 0 | — |
| 3 | 1 | 1 | Lowest Power Consumption |

## Table 10.1. Comparator Electrical Characteristics

$V_{DD}$ = 3.0 V, AV+ = 3.0 V, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Response Time: Mode 0, $V_{CM}$* = 1.5 V | CPn+ – CPn- = 100 mV | — | 100 | — | ns |
| | CPn+ – CPn– = –100 mV | — | 250 | — | ns |
| Response Time: Mode 1, $V_{CM}$* = 1.5 V | CPn+ – CPn– = 100 mV | — | 175 | — | ns |
| | CPn+ – CPn– = –100 mV | — | 500 | — | ns |
| Response Time: Mode 2, $V_{CM}$* = 1.5 V | CPn+ – CPn– = 100 mV | — | 320 | — | ns |
| | CPn+ – CPn– = –100 mV | — | 1100 | — | ns |
| Response Time: Mode 3, $V_{CM}$* = 1.5 V | CPn+ – CPn– = 100 mV | — | 1050 | — | ns |
| | CPn+ – CPn– = –100 mV | — | 5200 | — | ns |
| Common-Mode Rejection Ratio | | — | 1.5 | 4 | mV/V |
| Positive Hysteresis 1 | CPnHYP1-0 = 00 | — | 0 | 1 | mV |
| Positive Hysteresis 2 | CPnHYP1-0 = 01 | 2 | 4.5 | 7 | mV |
| Positive Hysteresis 3 | CPnHYP1-0 = 10 | 4 | 9 | 13 | mV |
| Positive Hysteresis 4 | CPnHYP1-0 = 11 | 10 | 17 | 25 | mV |
| Negative Hysteresis 1 | CPnHYN1-0 = 00 | — | 0 | 1 | mV |
| Negative Hysteresis 2 | CPnHYN1-0 = 01 | 2 | 4.5 | 7 | mV |
| Negative Hysteresis 3 | CPnHYN1-0 = 10 | 4 | 9 | 13 | mV |
| Negative Hysteresis 4 | CPnHYN1-0 = 11 | 10 | 17 | 25 | mV |
| Inverting or Non-Inverting Input Voltage Range | | –0.25 | — | (AV+) + 0.25 | V |
| Input Capacitance | | — | 7 | — | pF |
| Input Bias Current | | –5 | 0.001 | +5 | nA |
| Input Offset Voltage | | –10 | — | +10 | mV |
| **Power Supply** | | | | | |
| Power-Up Time | CPnEN from 0 to 1 | — | 20 | — | µs |
| Power Supply Rejection | | — | 0.1 | 1 | mV/V |
| Supply Current at DC (each comparator) | Mode 0 | — | 7.6 | — | µA |
| | Mode 1 | — | 3.2 | — | µA |
| | Mode 2 | — | 1.3 | — | µA |
| | Mode 3 | — | 0.4 | — | µA |

***Note:** $V_{CM}$ is the common-mode voltage on CPn+ and CPn-.

# 11. 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. Included are five 16-bit counter/timers (see description in **Section 23**), two full-duplex UARTs (see description in **Section 21** and **Section 22**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (see **Section 11.2.6**), and 8/4 byte-wide I/O Ports (see description in **Section 18**). The CIP-51 also includes on-chip debug hardware (see description in **Section 25**), and interfaces directly with the MCU's 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 11.1 for a block diagram).

- Fully Compatible with MCS-51 Instruction Set
- 100 or 50 MIPS Peak Using the On-Chip PLL
- 256 Bytes of Internal RAM
- 8/4 Byte-Wide I/O Ports
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

The CIP-51 includes the following features:

**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.

With the CIP-51's system clock running at 100 MHz, it has a peak throughput of 100 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 |

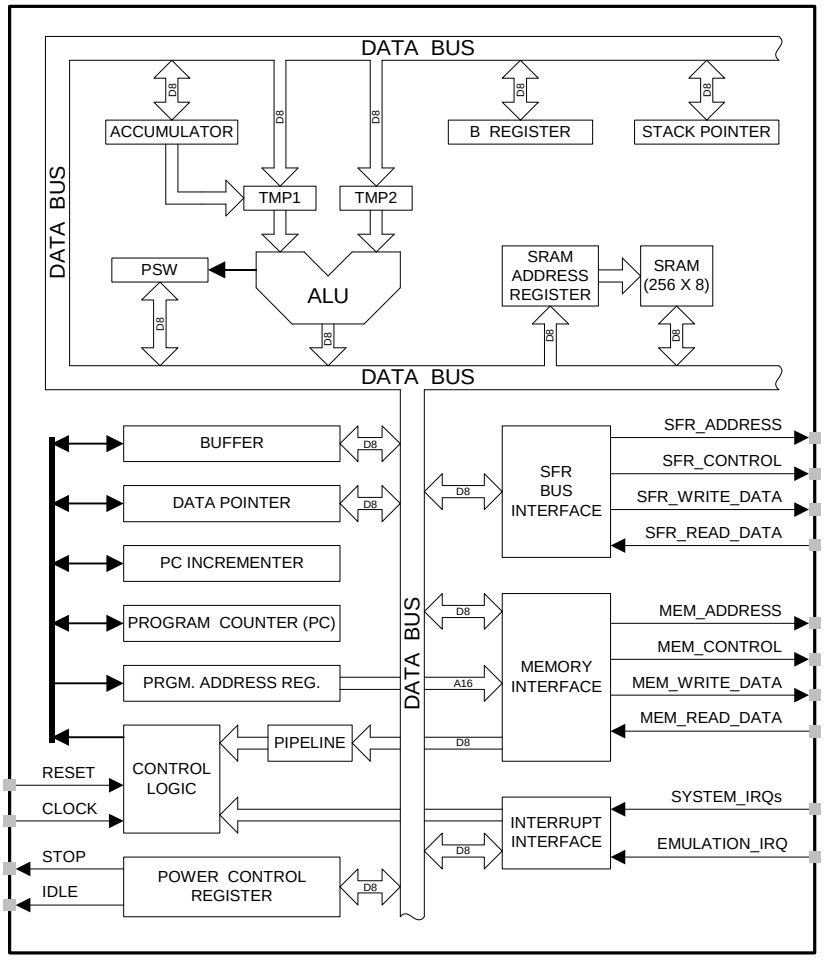



**Figure 11.1. CIP-51 Block Diagram**

**Programming and Debugging Support**

A JTAG-based serial interface is provided for in-system programming of the Flash program memory and communication with on-chip debug support logic. The re-programmable Flash can also be read and changed by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints and watch points, 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 debug is completely non-intrusive and non-invasive, requiring no RAM, Stack, timers, or other on-chip resources.

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, macro assembler, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via its JTAG interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

## 11.1. 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.

### 11.1.1. 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 11.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

### 11.1.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip XRAM, and accessing on-chip program Flash memory. The Flash access feature provides a mechanism for user software to update program code and use the program memory space for non-volatile data storage (see **Section “15. Flash Memory” on page 199**). The External Memory Interface provides a fast access to off-chip XRAM (or memory-mapped peripherals) via the MOVX instruction. Refer to **Section “17. External Data Memory Interface and On-Chip XRAM” on page 219** for details.

**Table 11.1. 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 |

**Table 11.1. CIP-51 Instruction Set Summary (Continued)**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| 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 |
| 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 |

**Table 11.1. CIP-51 Instruction Set Summary (Continued)**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| 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 |
| 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* |

**Table 11.1. CIP-51 Instruction Set Summary (Continued)**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| 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 |
| * Branch instructions will incur a cache-miss penalty if the branch target location is not already stored in the Branch Target Cache. See **Section “16. Branch Target Cache” on page 211** for more details. | | | |

**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 (2s 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 2K-byte 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 64K-byte program memory space.

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

## 11.2. 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. There are 256 bytes of internal data memory and 128k bytes (C8051F12x and C8051F130/1) or 64k bytes (C8051F132/3) of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 11.2.

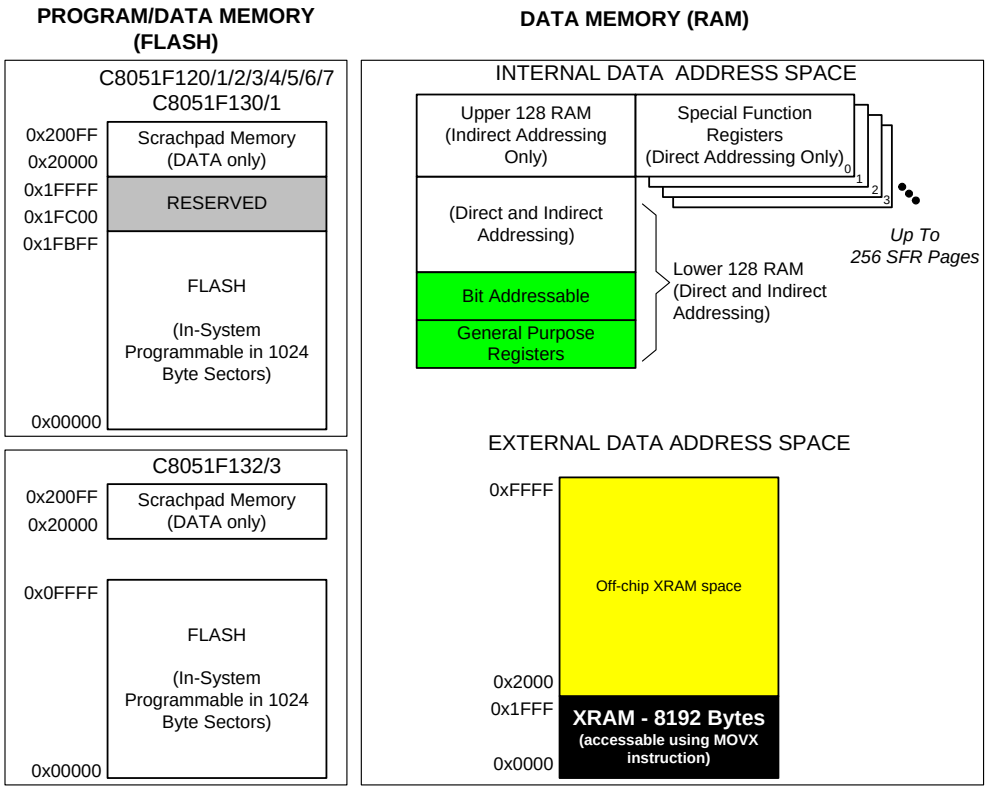



**Figure 11.2. Memory Map**

### 11.2.1. Program Memory

The C8051F12x and C8051F130/1 have a 128 kB program memory space. The MCU implements 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 three 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 11.1. 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. Note: 1024 bytes of the memory in Bank 3 (0x1FC00 to 0x1FFFF) are reserved and are not available for user program or data storage. The C8051F132/3 have a 64k byte program memory space implemented as in-system re-programmable Flash memory, and organized in a contiguous block from address 0x00000 to 0x0FFFF.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section “15. Flash Memory” on page 199** for further details.

## SFR Definition 11.1. PSBANK: Program Space Bank Select

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | COBANK | | - | - | IFBANK | | 00010001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xB1
SFR Page: All Pages

Bits 7–6: Reserved.
Bits 5–4: COBANK: Constant Operations Bank Select.
These bits select which Flash bank is targeted during constant operations (MOVC and Flash MOVX) involving addresses 0x8000 to 0xFFFF. These bits are ignored when accessing the Scratchpad memory areas (see **Section “15. Flash Memory” on page 199**).
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.
Bits 3–2: Reserved.
Bits 1–0: IFBANK: Instruction Fetch Operations Bank Select.
These bits select which Flash bank is used for instruction fetches involving addresses 0x8000 to 0xFFFF. These bits can only be changed from code in Bank 0 (see Figure 11.3).
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:** On the C8051F132/3, the COBANK and IFBANK bits should both remain set to the default setting of ‘01’ to ensure proper device functionality.

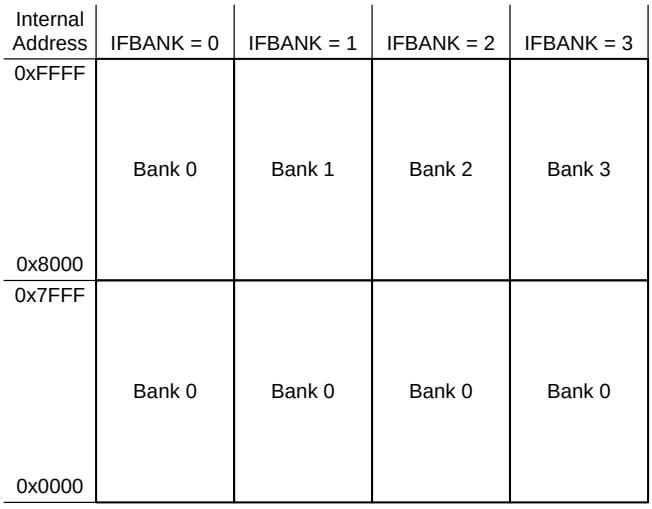



**Figure 11.3. Address Memory Map for Instruction Fetches (128 kB Flash Only)**

### 11.2.2. Data Memory

The CIP-51 implements 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 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 SFR's. 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 11.2 illustrates the data memory organization of the CIP-51.

### 11.2.3. 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 11.9). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

### 11.2.4. 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 bit 7 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.

### 11.2.5. 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, address 0x81) 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.

The MCUs also have built-in hardware for a stack record which is accessed by the debug logic. The stack record is a 32-bit shift register, where each PUSH or increment SP pushes one record bit onto the register, and each CALL pushes two record bits onto the register. (A POP or decrement SP pops one record bit,

and a RET pops two record bits, also.) The stack record circuitry can also detect an overflow or underflow on the 32-bit shift register, and can notify the debug software even with the MCU running at speed.

### 11.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFR's). The SFR's provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFR's found in a typical 8051 implementation as well as implementing additional SFR's used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 11.2 lists the SFR's implemented in the CIP-51 System Controller.

The SFR registers are accessed whenever the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFR's with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFR's are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the datasheet, as indicated in Table 11.3, for a detailed description of each register.

#### 11.2.6.1.SFR Paging

The CIP-51 features *SFR paging,* allowing the device to map many SFR's 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 SFR's. The C8051F12x family of devices utilizes five SFR pages: 0, 1, 2, 3, and F. 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).

#### 11.2.6.2.Interrupts and SFR Paging

When an interrupt occurs, the SFR Page Register will automatically switch to the SFR page containing the flag bit that caused the interrupt. The automatic SFR Page switch function conveniently removes the burden of switching SFR pages from the interrupt service routine. 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. On interrupt, the current SFRPAGE value is pushed to the SFRNEXT byte, and the value of SFRNEXT is pushed to SFRLAST. Hardware then loads SFRPAGE with the SFR Page containing the flag bit associated with the interrupt. 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.

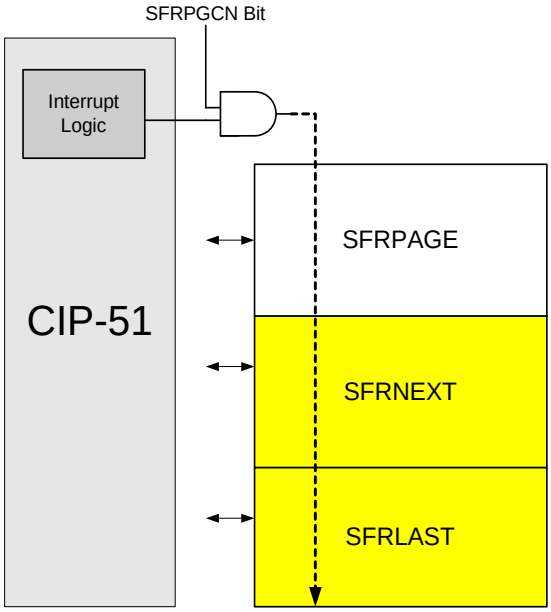



**Figure 11.4. SFR Page Stack**

Automatic hardware switching 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 (SFRPGCN). 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) is provided in Table 11.2. 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. Note that certain SFR’s 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 SFR’s are accessible from all SFR pages regardless of the SFRPAGE register value.

#### 11.2.6.3.SFR Page Stack Example

The following is an example that shows the operation of the SFR Page Stack during interrupts.

In this example, the SFR Page Control is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to Port 5 (SFR “P5”, located at address 0xD8 on SFR Page 0x0F). The device is also using the Programmable Counter Array (PCA) and the 10-bit ADC (ADC2) window comparator to monitor a voltage. The PCA is timing a critical control function in its interrupt service routine (ISR), so its interrupt is enabled and is set to *high* priority. The ADC2 is monitoring a voltage that is less important, but to minimize the software overhead its window comparator is being used with an associated ISR that is set to *low* priority. At this point, the SFR page is set to access the Port 5 SFR (SFRPAGE = 0x0F). See Figure 11.5 below.

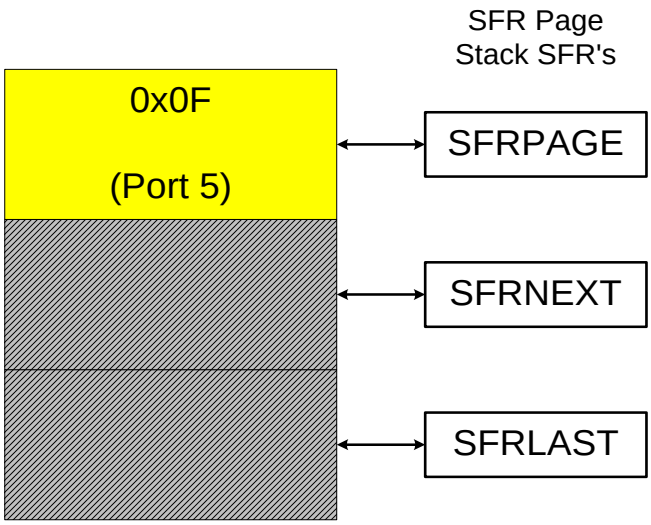



**Figure 11.5. SFR Page Stack While Using SFR Page 0x0F To Access Port 5**

While CIP-51 executes in-line code (writing values to Port 5 in this example), ADC2 Window Comparator Interrupt occurs. The CIP-51 vectors to the ADC2 Window Comparator ISR and pushes the current SFR Page value (SFR Page 0x0F) into SFRNEXT in the SFR Page Stack. The SFR page needed to access ADC2’s SFR’s is then automatically placed in the SFRPAGE register (SFR Page 0x02). SFRPAGE is considered the “top” of the SFR Page Stack. Software can now access the ADC2 SFR’s. Software may switch to any SFR Page by writing a new value to the SFRPAGE register at any time during the ADC2 ISR to access SFR’s that are not on SFR Page 0x02. See Figure 11.6 below.

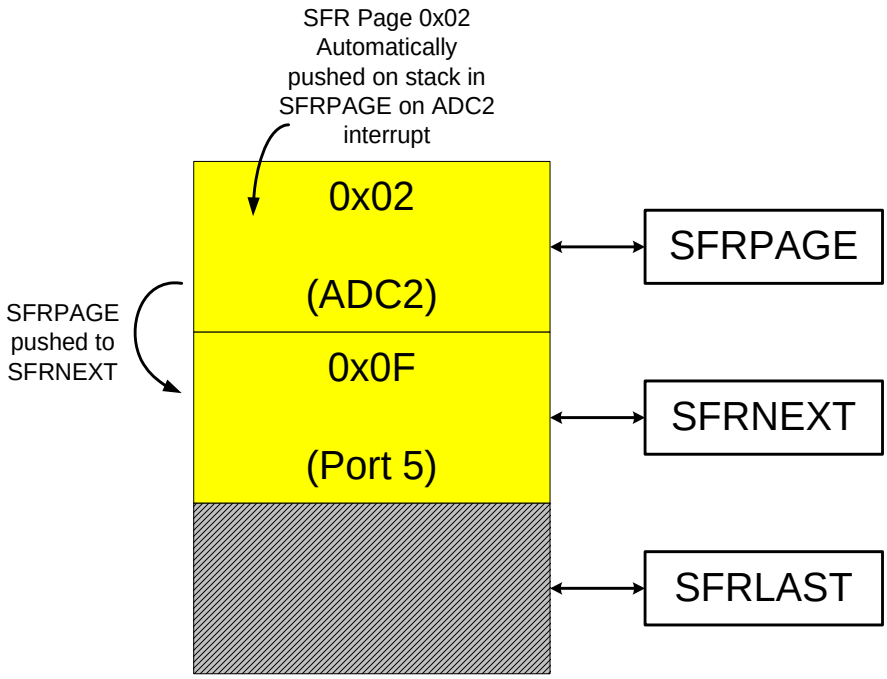



**Figure 11.6. SFR Page Stack After ADC2 Window Comparator Interrupt Occurs**

While in the ADC2 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a *high* priority interrupt, while the ADC2 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place the SFR page needed to access the PCA's special function registers into the SFRPAGE register, SFR Page 0x00. The value that was in the SFRPAGE register before the PCA interrupt (SFR Page 2 for ADC2) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR Page 0x0F for Port 5) is pushed down to the SFRLAST register, the "bottom" of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 11.7 below.

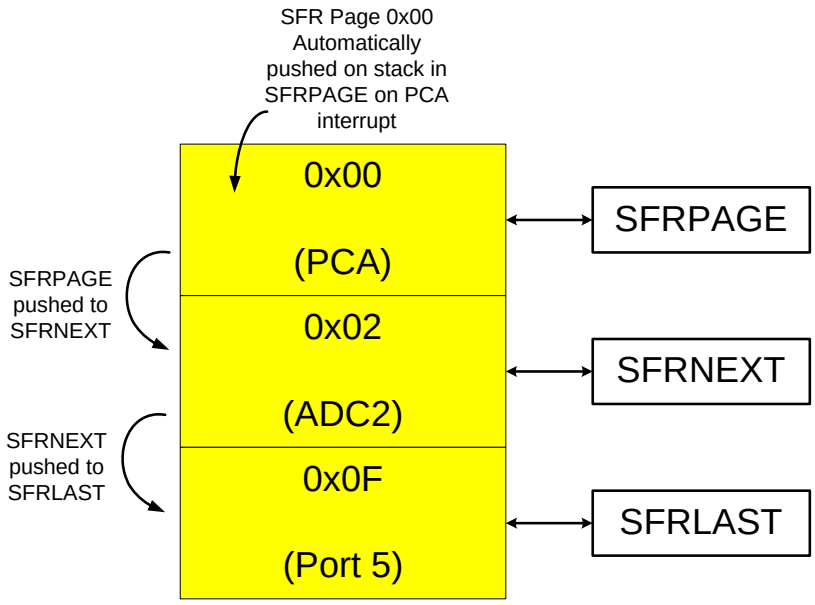



**Figure 11.7. SFR Page Stack Upon PCA Interrupt Occurring During an ADC2 ISR**

On exit from the PCA interrupt service routine, the CIP-51 will return to the ADC2 Window Comparator ISR. On execution of the RETI instruction, SFR Page 0x00 used to access the PCA registers will be automatically popped off of the SFR Page Stack, and the contents of the SFRNEXT register will be moved to the SFRPAGE register. Software in the ADC2 ISR can continue to access SFR's as it did prior to the PCA interrupt. Likewise, the contents of SFRLAST are moved to the SFRNEXT register. Recall this was the SFR Page value 0x0F being used to access Port 5 before the ADC2 interrupt occurred. See Figure 11.8 below.

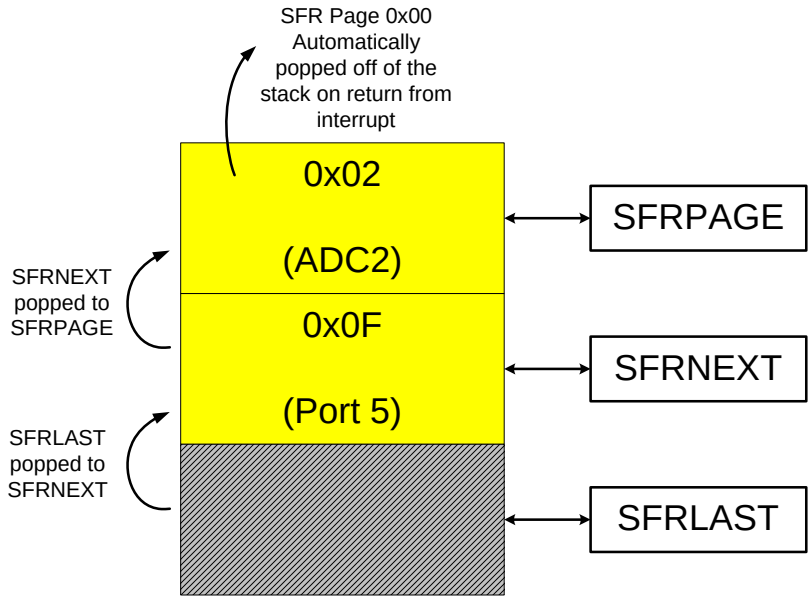



**Figure 11.8. SFR Page Stack Upon Return From PCA Interrupt**

On the execution of the RETI instruction in the ADC2 Window Comparator ISR, the value in SFRPAGE register is overwritten with the contents of SFRNEXT. The CIP-51 may now access the Port 5 SFR bits as it did prior to the interrupts occurring. See Figure 11.9 below.

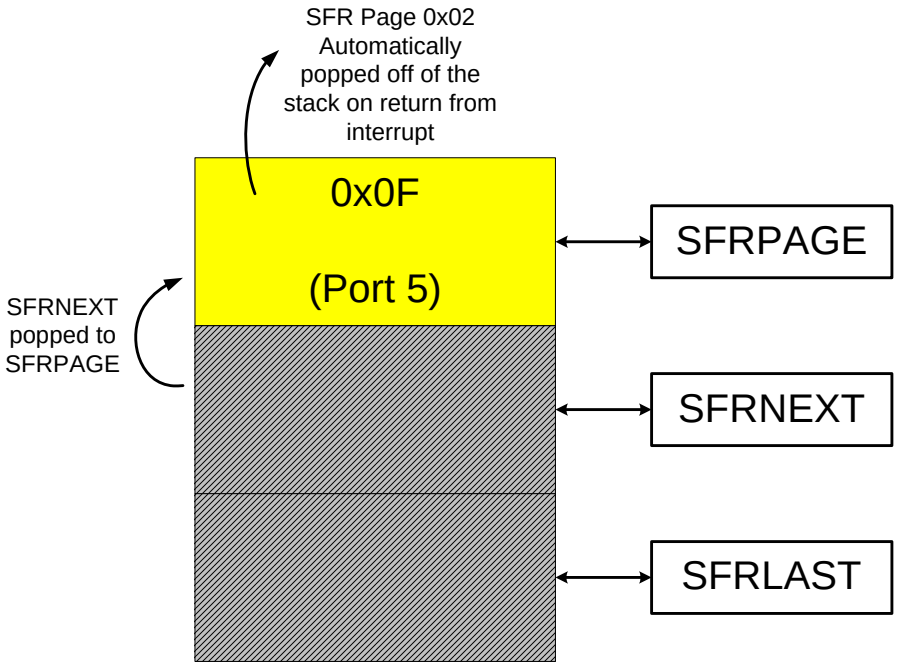



**Figure 11.9. SFR Page Stack Upon Return From ADC2 Window Interrupt**

Note that in the above example, all three bytes in the SFR Page Stack are accessible via the SFRPAGE, SFRNEXT, and SFRLAST special function registers. If the stack is altered while servicing an interrupt, it is possible to return to a different SFR Page upon interrupt exit than selected prior to the interrupt call. Direct access to the SFR Page stack can be useful to enable real-time operating systems to control and manage context switching between multiple tasks.

Push operations on the SFR Page Stack only occur on interrupt service, and pop operations only occur on interrupt exit (execution on the RETI instruction). The automatic switching of the SFRPAGE and operation of the SFR Page Stack as described above can be disabled in software by clearing the SFR Automatic Page Enable Bit (SFRPGEN) in the SFR Page Control Register (SFRPGCN). See SFR Definition 11.2.

## SFR Definition 11.2. SFRPGCN: SFR Page Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | SFRPGEN | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x96
SFR Page: F

Bits7–1: Reserved.

Bit0: SFRPGEN: SFR Automatic Page Control Enable.
Upon interrupt, the C8051 Core will vector to the specified interrupt service routine and automatically switch the SFR page to the corresponding peripheral or function's SFR page. This bit is used to control this autopaging function.
0: SFR Automatic Paging disabled. C8051 core will not automatically change to the appropriate SFR page (i.e., the SFR page that contains the SFR's for the peripheral/function that was the source of the interrupt).
1: SFR Automatic Paging enabled. Upon interrupt, the C8051 will switch the SFR page to the page that contains the SFR's for the peripheral or function that is the source of the interrupt.

## SFR Definition 11.3. SFRPAGE: SFR Page

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x84
SFR Page: All Pages

Bits7–0: SFR Page Bits: Byte Represents the SFR Page the C8051 MCU uses when reading or modifying SFR's.
Write: Sets the SFR Page.
Read: Byte is the SFR page the C8051 MCU is using.

When enabled in the SFR Page Control Register (SFRPGCN), the C8051 will automatically switch to the SFR Page that contains the SFR's 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 11.4. SFRNEXT: SFR Next Register

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x85
SFR Page: All Pages

Bits7–0: SFR Page Stack Bits: 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.

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. This is the value that will go to the SFR Page register upon a return from interrupt.

## SFR Definition 11.5. SFRLAST: SFR Last Register

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x86
SFR Page: All Pages

Bits7–0: SFR Page Stack Bits: 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.

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.

**Table 11.2. Special Function Register (SFR) Memory Map**

| ADDRESS | SFR Page | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |
|---|---|---|---|---|---|---|---|---|---|
| F8 | 0<br>1<br>2<br>3<br>F | SPI0CN<br><br><br><br>P7 | PCA0L | PCA0H | PCA0CPL0 | PCA0CPH0 | PCA0CPL1 | PCA0CPH1 | **WDTCN (ALL PAGES)** |
| F0 | 0<br>1<br>2<br>3<br>F | **B (ALL PAGES)** | | | | | | **EIP1 (ALL PAGES)** | **EIP2 (ALL PAGES)** |
| E8 | 0<br>1<br>2<br>3<br>F | ADC0CN<br><br>ADC2CN<br><br>P6 | PCA0CPL2 | PCA0CPH2 | PCA0CPL3 | PCA0CPH3 | PCA0CPL4 | PCA0CPH4 | RSTSRC |
| E0 | 0<br>1<br>2<br>3<br>F | **ACC (ALL PAGES)** | PCA0CPL5<br><br><br><br>XBR0 | PCA0CPH5<br><br><br><br>XBR1 | <br><br><br><br>XBR2 | | | **EIE1 (ALL PAGES)** | **EIE2 (ALL PAGES)** |
| D8 | 0<br>1<br>2<br>3<br>F | PCA0CN<br><br><br><br>P5 | PCA0MD | PCA0CPM0 | PCA0CPM1 | PCA0CPM2 | PCA0CPM3 | PCA0CPM4 | PCA0CPM5 |
| D0 | 0<br>1<br>2<br>3<br>F | **PSW (ALL PAGES)** | REF0CN | DAC0L<br>DAC1L | DAC0H<br>DAC1H | DAC0CN<br>DAC1CN | | | |
| C8 | 0<br>1<br>2<br>3<br>F | TMR2CN<br>TMR3CN<br>TMR4CN<br><br>P4 | TMR2CF<br>TMR3CF<br>TMR4CF | RCAP2L<br>RCAP3L<br>RCAP4L | RCAP2H<br>RCAP3H<br>RCAP4H | TMR2L<br>TMR3L<br>TMR4L | TMR2H<br>TMR3H<br>TMR4H | <br><br><br><br>MAC0RNDL | SMB0CR<br><br><br><br>MAC0RNDH |
| C0 | 0<br>1<br>2<br>3<br>F | SMB0CN<br><br><br><br>MAC0STA | SMB0STA<br><br><br><br>MAC0AL | SMB0DAT<br><br><br><br>MAC0AH | SMB0ADR<br><br><br><br>MAC0CF | ADC0GTL<br><br>ADC2GT | ADC0GTH | ADC0LTL<br><br>ADC2LT | ADC0LTH |
| | | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

**Table 11.2. Special Function Register (SFR) Memory Map (Continued)**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| B8 | 0<br>1<br>2<br>3<br>F | **IP (ALL PAGES)** | SADEN0 | AMX0CF<br><br>AMX2CF | AMX0SL<br><br>AMX2SL | ADC0CF<br><br>ADC2CF | | ADC0L<br><br>ADC2 | ADC0H |
| B0 | 0<br>1<br>2<br>3<br>F | **P3 (ALL PAGES)** | **PSBANK (ALL PAGES)** | | | | | | FLSCL<br><br><br><br>FLACL |
| A8 | 0<br>1<br>2<br>3<br>F | **IE (ALL PAGES)** | SADDR0 | | | | P1MDIN | | |
| A0 | 0<br>1<br>2<br>3<br>F | **P2 (ALL PAGES)** | EMI0TC<br><br><br><br>CCH0CN | EMI0CN<br><br><br><br>CCH0TN | EMI0CF<br><br><br><br>CCH0LC | P0MDOUT | P1MDOUT | P2MDOUT | P3MDOUT |
| 98 | 0<br>1<br>2<br>3<br>F | SCON0<br>SCON1 | SBUF0<br>SBUF1 | SPI0CFG<br><br><br><br>CCH0MA | SPI0DAT | P4MDOUT | SPI0CKR<br><br><br><br>P5MDOUT | P6MDOUT | P7MDOUT |
| 90 | 0<br>1<br>2<br>3<br>F | **P1 (ALL PAGES)** | SSTA0<br><br><br><br>MAC0BL | MAC0BH | MAC0ACC0 | MAC0ACC1 | MAC0ACC2 | MAC0ACC3<br>SFRPGCN | MAC0OVR<br>CLKSEL |
| 88 | 0<br>1<br>2<br>3<br>F | TCON<br>CPT0CN<br>CPT1CN<br><br>FLSTAT | TMOD<br>CPT0MD<br>CPT1MD<br><br>PLL0CN | TL0<br><br><br><br>OSCICN | TL1<br><br><br><br>OSCICL | TH0<br><br><br><br>OSCXCN | TH1<br><br><br><br>PLL0DIV | CKCON<br><br><br><br>PLL0MUL | PSCTL<br><br><br><br>PLL0FLT |
| 80 | 0<br>1<br>2<br>3<br>F | **P0 (ALL PAGES)** | **SP (ALL PAGES)** | **DPL (ALL PAGES)** | **DPH (ALL PAGES)** | **SFRPAGE (ALL PAGES)** | **SFRNEXT (ALL PAGES)** | **SFRLAST (ALL PAGES)** | **PCON (ALL PAGES)** |
| | | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

## Table 11.3. Special Function Registers

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| ACC | 0xE0 | All Pages | Accumulator | page 153 |
| ADC0CF | 0xBC | 0 | ADC0 Configuration | page 62[1], page 80[2] |
| ADC0CN | 0xE8 | 0 | ADC0 Control | page 63[1], page 81[2] |
| ADC0GTH | 0xC5 | 0 | ADC0 Greater-Than High Byte | page 66[1], page 84[2] |
| ADC0GTL | 0xC4 | 0 | ADC0 Greater-Than Low Byte | page 66[1], page 84[2] |
| ADC0H | 0xBF | 0 | ADC0 Data Word High Byte | page 64[1], page 82[2] |
| ADC0L | 0xBE | 0 | ADC0 Data Word Low Byte | page 64[1], page 82[2] |
| ADC0LTH | 0xC7 | 0 | ADC0 Less-Than High Byte | page 67[1], page 85[2] |
| ADC0LTL | 0xC6 | 0 | ADC0 Less-Than Low Byte | page 67[1], page 85[2] |
| ADC2 | 0xBE | 2 | ADC2 Data Word | page 99[3] |
| ADC2CF | 0xBC | 2 | ADC2 Configuration | page 97[3] |
| ADC2CN | 0xE8 | 2 | ADC2 Control | page 98[3] |
| ADC2GT | 0xC4 | 2 | ADC2 Greater-Than | page 102[3] |
| ADC2LT | 0xC6 | 2 | ADC2 Less-Than | page 102[3] |
| AMX0CF | 0xBA | 0 | ADC0 Multiplexer Configuration | page 60[1], page 78[2] |
| AMX0SL | 0xBB | 0 | ADC0 Multiplexer Channel Select | page 61[1], page 79[2] |
| AMX2CF | 0xBA | 2 | ADC2 Multiplexer Configuration | page 95[3] |
| AMX2SL | 0xBB | 2 | ADC2 Multiplexer Channel Select | page 96[3] |
| B | 0xF0 | All Pages | B Register | page 153 |
| CCH0CN | 0xA1 | F | Cache Control | page 215 |
| CCH0LC | 0xA3 | F | Cache Lock | page 216 |
| CCH0MA | 0x9A | F | Cache Miss Accumulator | page 217 |
| CCH0TN | 0xA2 | F | Cache Tuning | page 216 |
| CKCON | 0x8E | 0 | Clock Control | page 315 |
| CLKSEL | 0x97 | F | System Clock Select | page 188 |
| CPT0CN | 0x88 | 1 | Comparator 0 Control | page 123 |
| CPT0MD | 0x89 | 1 | Comparator 0 Configuration | page 123 |
| CPT1CN | 0x88 | 2 | Comparator 1 Control | page 124 |
| CPT1MD | 0x89 | 2 | Comparator 1 Configuration | page 125 |
| DAC0CN | 0xD4 | 0 | DAC0 Control | page 108[3] |
| DAC0H | 0xD3 | 0 | DAC0 High Byte | page 107[3] |
| DAC0L | 0xD2 | 0 | DAC0 Low Byte | page 107[3] |
| DAC1CN | 0xD4 | 1 | DAC1 Control | page 110[3] |
| DAC1H | 0xD3 | 1 | DAC1 High Byte | page 109[3] |
| DAC1L | 0xD2 | 1 | DAC1 Low Byte | page 109[3] |
| DPH | 0x83 | All Pages | Data Pointer High Byte | page 151 |
| DPL | 0x82 | All Pages | Data Pointer Low Byte | page 151 |

## Table 11.3. Special Function Registers (Continued)

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| EIE1 | 0xE6 | All Pages | Extended Interrupt Enable 1 | page 159 |
| EIE2 | 0xE7 | All Pages | Extended Interrupt Enable 2 | page 160 |
| EIP1 | 0xF6 | All Pages | Extended Interrupt Priority 1 | page 161 |
| EIP2 | 0xF7 | All Pages | Extended Interrupt Priority 2 | page 162 |
| EMI0CF | 0xA3 | 0 | EMIF Configuration | page 221 |
| EMI0CN | 0xA2 | 0 | EMIF Control | page 220 |
| EMI0TC | 0xA1 | 0 | EMIF Timing Control | page 226 |
| FLACL | 0xB7 | F | Flash Access Limit | page 206 |
| FLSCL | 0xB7 | 0 | Flash Scale | page 208 |
| FLSTAT | 0x88 | F | Flash Status | page 217 |
| IE | 0xA8 | All Pages | Interrupt Enable | page 157 |
| IP | 0xB8 | All Pages | Interrupt Priority | page 158 |
| MAC0ACC0 | 0x93 | 3 | MAC0 Accumulator Byte 0 (LSB) | page 174[4] |
| MAC0ACC1 | 0x94 | 3 | MAC0 Accumulator Byte 1 | page 173[4] |
| MAC0ACC2 | 0x95 | 3 | MAC0 Accumulator Byte 2 | page 173[4] |
| MAC0ACC3 | 0x96 | 3 | MAC0 Accumulator Byte 3 (MSB) | page 173[4] |
| MAC0AH | 0xC2 | 3 | MAC0 A Register High Byte | page 171[4] |
| MAC0AL | 0xC1 | 3 | MAC0 A Register Low Byte | page 172[4] |
| MAC0BH | 0x92 | 3 | MAC0 B Register High Byte | page 172[4] |
| MAC0BL | 0x91 | 3 | MAC0 B Register Low Byte | page 172[4] |
| MAC0CF | 0xC3 | 3 | MAC0 Configuration | page 170[4] |
| MAC0OVR | 0x97 | 3 | MAC0 Accumulator Overflow | page 174[4] |
| MAC0RNDH | 0xCF | 3 | MAC0 Rounding Register High Byte | page 174[4] |
| MAC0RNDL | 0xCE | 3 | MAC0 Rounding Register Low Byte | page 175[4] |
| MAC0STA | 0xC0 | 3 | MAC0 Status Register | page 171[4] |
| OSCICL | 0x8B | F | Internal Oscillator Calibration | page 186 |
| OSCICN | 0x8A | F | Internal Oscillator Control | page 186 |
| OSCXCN | 0x8C | F | External Oscillator Control | page 189 |
| P0 | 0x80 | All Pages | Port 0 Latch | page 248 |
| P0MDOUT | 0xA4 | F | Port 0 Output Mode Configuration | page 248 |
| P1 | 0x90 | All Pages | Port 1 Latch | page 249 |
| P1MDIN | 0xAD | F | Port 1 Input Mode | page 249 |
| P1MDOUT | 0xA5 | F | Port 1 Output Mode Configuration | page 250 |
| P2 | 0xA0 | All Pages | Port 2 Latch | page 250 |
| P2MDOUT | 0xA6 | F | Port 2 Output Mode Configuration | page 251 |
| P3 | 0xB0 | All Pages | Port 3 Latch | page 251 |
| P3MDOUT | 0xA7 | F | Port 3 Output Mode Configuration | page 252 |
| P4 | 0xC8 | F | Port 4 Latch | page 254 |
| P4MDOUT | 0x9C | F | Port 4 Output Mode Configuration | page 254 |
| P5 | 0xD8 | F | Port 5 Latch | page 255 |
| P5MDOUT | 0x9D | F | Port 5 Output Mode Configuration | page 255 |

**Table 11.3. Special Function Registers (Continued)**

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| P6 | 0xE8 | F | Port 6 Latch | page 256 |
| P6MDOUT | 0x9E | F | Port 6 Output Mode Configuration | page 256 |
| P7 | 0xF8 | F | Port 7 Latch | page 257 |
| P7MDOUT | 0x9F | F | Port 7 Output Mode Configuration | page 257 |
| PCA0CN | 0xD8 | 0 | PCA Control | page 335 |
| PCA0CPH0 | 0xFC | 0 | PCA Module 0 Capture/Compare High Byte | page 339 |
| PCA0CPH1 | 0xFE | 0 | PCA Module 1 Capture/Compare High Byte | page 339 |
| PCA0CPH2 | 0xEA | 0 | PCA Module 2 Capture/Compare High Byte | page 339 |
| PCA0CPH3 | 0xEC | 0 | PCA Module 3 Capture/Compare High Byte | page 339 |
| PCA0CPH4 | 0xEE | 0 | PCA Module 4 Capture/Compare High Byte | page 339 |
| PCA0CPH5 | 0xE2 | 0 | PCA Module 5 Capture/Compare High Byte | page 339 |
| PCA0CPL0 | 0xFB | 0 | PCA Module 0 Capture/Compare Low Byte | page 338 |
| PCA0CPL1 | 0xFD | 0 | PCA Module 1 Capture/Compare Low Byte | page 338 |
| PCA0CPL2 | 0xE9 | 0 | PCA Module 2 Capture/Compare Low Byte | page 338 |
| PCA0CPL3 | 0xEB | 0 | PCA Module 3 Capture/Compare Low Byte | page 338 |
| PCA0CPL4 | 0xED | 0 | PCA Module 4 Capture/Compare Low Byte | page 338 |
| PCA0CPL5 | 0xE1 | 0 | PCA Module 5 Capture/Compare Low Byte | page 338 |
| PCA0CPM0 | 0xDA | 0 | PCA Module 0 Mode | page 337 |
| PCA0CPM1 | 0xDB | 0 | PCA Module 1 Mode | page 337 |
| PCA0CPM2 | 0xDC | 0 | PCA Module 2 Mode | page 337 |
| PCA0CPM3 | 0xDD | 0 | PCA Module 3 Mode | page 337 |
| PCA0CPM4 | 0xDE | 0 | PCA Module 4 Mode | page 337 |
| PCA0CPM5 | 0xDF | 0 | PCA Module 5 Mode | page 337 |
| PCA0H | 0xFA | 0 | PCA Counter High Byte | page 338 |
| PCA0L | 0xF9 | 0 | PCA Counter Low Byte | page 338 |
| PCA0MD | 0xD9 | 0 | PCA Mode | page 336 |
| PCON | 0x87 | All Pages | Power Control | page 164 |
| PLL0CN | 0x89 | F | PLL Control | page 193 |
| PLL0DIV | 0x8D | F | PLL Divider | page 194 |
| PLL0FLT | 0x8F | F | PLL Filter | page 195 |
| PLL0MUL | 0x8E | F | PLL Multiplier | page 194 |
| PSBANK | 0xB1 | All Pages | Flash Bank Select | page 134 |
| PSCTL | 0x8F | 0 | Flash Write/Erase Control | page 209 |
| PSW | 0xD0 | All Pages | Program Status Word | page 152 |
| RCAP2H | 0xCB | 0 | Timer/Counter 2 Capture/Reload High Byte | page 323 |
| RCAP2L | 0xCA | 0 | Timer/Counter 2 Capture/Reload Low Byte | page 323 |
| RCAP3H | 0xCB | 1 | Timer 3 Capture/Reload High Byte | page 323 |
| RCAP3L | 0xCA | 1 | Timer 3 Capture/Reload Low Byte | page 323 |
| RCAP4H | 0xCB | 2 | Timer/Counter 4 Capture/Reload High Byte | page 323 |
| RCAP4L | 0xCA | 2 | Timer/Counter 4 Capture/Reload Low Byte | page 323 |

## Table 11.3. Special Function Registers (Continued)

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| REF0CN | 0xD1 | 0 | Voltage Reference Control | page 114[5], page 116[6], page 117[7] |
| RSTSRC | 0xEF | 0 | Reset Source | page 182 |
| SADDR0 | 0xA9 | 0 | UART 0 Slave Address | page 298 |
| SADEN0 | 0xB9 | 0 | UART 0 Slave Address Mask | page 298 |
| SBUF0 | 0x99 | 0 | UART 0 Data Buffer | page 298 |
| SBUF1 | 0x99 | 1 | UART 1 Data Buffer | page 305 |
| SCON0 | 0x98 | 0 | UART 0 Control | page 296 |
| SCON1 | 0x98 | 1 | UART 1 Control | page 304 |
| SFRLAST | 0x86 | All Pages | SFR Stack Last Page | page 143 |
| SFRNEXT | 0x85 | All Pages | SFR Stack Next Page | page 143 |
| SFRPAGE | 0x84 | All Pages | SFR Page Select | page 142 |
| SFRPGCN | 0x96 | F | SFR Page Control | page 142 |
| SMB0ADR | 0xC3 | 0 | SMBus Slave Address | page 269 |
| SMB0CN | 0xC0 | 0 | SMBus Control | page 266 |
| SMB0CR | 0xCF | 0 | SMBus Clock Rate | page 267 |
| SMB0DAT | 0xC2 | 0 | SMBus Data | page 268 |
| SMB0STA | 0xC1 | 0 | SMBus Status | page 269 |
| SP | 0x81 | All Pages | Stack Pointer | page 151 |
| SPI0CFG | 0x9A | 0 | SPI Configuration | page 280 |
| SPI0CKR | 0x9D | 0 | SPI Clock Rate Control | page 282 |
| SPI0CN | 0xF8 | 0 | SPI Control | page 281 |
| SPI0DAT | 0x9B | 0 | SPI Data | page 282 |
| SSTA0 | 0x91 | 0 | UART 0 Status | page 297 |
| TCON | 0x88 | 0 | Timer/Counter Control | page 313 |
| TH0 | 0x8C | 0 | Timer/Counter 0 High Byte | page 316 |
| TH1 | 0x8D | 0 | Timer/Counter 1 High Byte | page 316 |
| TL0 | 0x8A | 0 | Timer/Counter 0 Low Byte | page 315 |
| TL1 | 0x8B | 0 | Timer/Counter 1 Low Byte | page 316 |
| TMOD | 0x89 | 0 | Timer/Counter Mode | page 314 |
| TMR2CF | 0xC9 | 0 | Timer/Counter 2 Configuration | page 324 |
| TMR2CN | 0xC8 | 0 | Timer/Counter 2 Control | page 324 |
| TMR2H | 0xCD | 0 | Timer/Counter 2 High Byte | page 324 |
| TMR2L | 0xCC | 0 | Timer/Counter 2 Low Byte | page 323 |
| TMR3CF | 0xC9 | 1 | Timer 3 Configuration | page 324 |
| TMR3CN | 0xC8 | 1 | Timer 3 Control | page 324 |
| TMR3H | 0xCD | 1 | Timer 3 High Byte | page 324 |
| TMR3L | 0xCC | 1 | Timer 3 Low Byte | page 323 |
| TMR4CF | 0xC9 | 2 | Timer/Counter 4 Configuration | page 324 |
| TMR4CN | 0xC8 | 2 | Timer/Counter 4 Control | page 324 |
| TMR4H | 0xCD | 2 | Timer/Counter 4 High Byte | page 324 |

**Table 11.3. Special Function Registers (Continued)**

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| TMR4L | 0xCC | 2 | Timer/Counter 4 Low Byte | page 323 |
| WDTCN | 0xFF | All Pages | Watchdog Timer Control | page 181 |
| XBR0 | 0xE1 | F | Port I/O Crossbar Control 0 | page 245 |
| XBR1 | 0xE2 | F | Port I/O Crossbar Control 1 | page 246 |
| XBR2 | 0xE3 | F | Port I/O Crossbar Control 2 | page 247 |

**Notes:**

1. Refers to a register in the C8051F120/1/4/5 only.
2. Refers to a register in the C8051F122/3/6/7 and C8051F130/1/2/3 only.
3. Refers to a register in the C8051F120/1/2/3/4/5/6/7 only.
4. Refers to a register in the C8051F120/1/2/3 and C8051F130/1/2/3 only.
5. Refers to a register in the C8051F120/2/4/6 only.
6. Refers to a register in the C8051F121/3/5/7 only.
7. Refers to a register in the C8051F130/1/2/3 only.

### 11.2.7. 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 datasheet associated with their corresponding system function.

**SFR Definition 11.6. SP: Stack Pointer**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x81
SFR Page: All Pages

Bits7–0: SP: 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 11.7. DPL: Data Pointer Low Byte**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x82
SFR Page: All Pages

Bits7–0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

**SFR Definition 11.8. DPH: Data Pointer High Byte**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x83
SFR Page: All Pages

Bits7–0: DPH: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

## SFR Definition 11.9. PSW: Program Status Word

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CY | AC | F0 | RS1 | RS0 | OV | F1 | PARITY | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xD0
SFR Page: All Pages

Bit7: 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 0 by all other arithmetic operations.

Bit6: 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 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: RS1–RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

| RS1 | RS0 | Register Bank | Address |
|---|---|---|---|
| 0 | 0 | 0 | 0x00–0x07 |
| 0 | 1 | 1 | 0x08–0x0F |
| 1 | 0 | 2 | 0x10–0x17 |
| 1 | 1 | 3 | 0x18–0x1F |

Bit2: 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.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

## SFR Definition 11.10. ACC: Accumulator

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xE0
SFR Page: All Pages

Bits7–0: ACC: Accumulator.
This register is the accumulator for arithmetic operations.

## SFR Definition 11.11. B: B Register

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xF0
SFR Page: All Pages

Bits7–0: B: B Register.
This register serves as a second accumulator for certain arithmetic operations.

## 11.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 20 interrupt sources with 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. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an 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.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE, EIE1, or EIE2). 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:** Any instruction that clears the EA bit should be immediately followed by an instruction that has two or more opcode bytes. For example:

```
// in 'C':
EA = 0; // clear EA bit.
EA = 0; // this is a dummy instruction with two-byte opcode.

; in assembly:
CLR EA ; clear EA bit.
CLR EA ; this is a dummy instruction with two-byte opcode.
```

If an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction which clears the EA bit), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. However, a read of the EA bit will return a '0' inside the interrupt service routine. When the "CLR EA" opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

Some interrupt-pending flags are automatically cleared by the 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.

### 11.3.1. MCU Interrupt Sources and Vectors

The MCUs support 20 interrupt sources. Software can simulate an interrupt event 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. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 11.4. Refer to the datasheet 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).

### 11.3.2. External Interrupts

Two of the external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of bits IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flag for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /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 follows the state of the external interrupt's input pin. 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.

**Table 11.4. Interrupt Summary**

| Interrupt Source | Interrupt Vector | Priority Order | Pending Flags | Bit addressable? | Cleared by HW? | SFRPAGE (SFRPGEN = 1) | Enable Flag | Priority Control |
|---|---|---|---|---|---|---|---|---|
| Reset | 0x0000 | Top | None | N/A | N/A | 0 | Always Enabled | Always Highest |
| External Interrupt 0 (/INT0) | 0x0003 | 0 | IE0 (TCON.1) | Y | Y | 0 | EX0 (IE.0) | PX0 (IP.0) |
| Timer 0 Overflow | 0x000B | 1 | TF0 (TCON.5) | Y | Y | 0 | ET0 (IE.1) | PT0 (IP.1) |
| External Interrupt 1 (/INT1) | 0x0013 | 2 | IE1 (TCON.3) | Y | Y | 0 | EX1 (IE.2) | PX1 (IP.2) |
| Timer 1 Overflow | 0x001B | 3 | TF1 (TCON.7) | Y | Y | 0 | ET1 (IE.3) | PT1 (IP.3) |
| UART0 | 0x0023 | 4 | RI0 (SCON0.0)<br>TI0 (SCON0.1) | Y | | 0 | ES0 (IE.4) | PS0 (IP.4) |
| Timer 2 | 0x002B | 5 | TF2 (TMR2CN.7)<br>EXF2 (TMR2CN.6) | Y | | 0 | ET2 (IE.5) | PT2 (IP.5) |
| Serial Peripheral Interface | 0x0033 | 6 | SPIF (SPI0CN.7)<br>WCOL (SPI0CN.6)<br>MODF (SPI0CN.5)<br>RXOVRN (SPI0CN.4) | Y | | 0 | ESPI0 (EIE1.0) | PSPI0 (EIP1.0) |
| SMBus Interface | 0x003B | 7 | SI (SMB0CN.3) | Y | | 0 | ESMB0 (EIE1.1) | PSMB0 (EIP1.1) |
| ADC0 Window Comparator | 0x0043 | 8 | AD0WINT (ADC0CN.1) | Y | | 0 | EWADC0 (EIE1.2) | PWADC0 (EIP1.2) |
| Programmable Counter Array | 0x004B | 9 | CF (PCA0CN.7)<br>CCFn (PCA0CN.n) | Y | | 0 | EPCA0 (EIE1.3) | PPCA0 (EIP1.3) |
| Comparator 0 Falling Edge | 0x0053 | 10 | CP0FIF (CPT0CN.4) | Y | | 1 | ECP0F (EIE1.4) | PCP0F (EIP1.4) |
| Comparator 0 Rising Edge | 0x005B | 11 | CP0RIF (CPT0CN.5) | Y | | 1 | ECP0R (EIE1.5) | PCP0R (EIP1.5) |
| Comparator 1 Falling Edge | 0x0063 | 12 | CP1FIF (CPT1CN.4) | Y | | 2 | ECP1F (EIE1.6) | PCP1F (EIP1.6) |

**Table 11.4. Interrupt Summary (Continued)**

| Interrupt Source | Interrupt Vector | Priority Order | Pending Flags | Bit addressable? | Cleared by HW? | SFRPAGE (SFRPGEN = 1) | Enable Flag | Priority Control |
|---|---|---|---|---|---|---|---|---|
| Comparator 1 Rising Edge | 0x006B | 13 | CP1RIF (CPT1CN.5) | Y | | 2 | ECP1R (EIE1.7) | PCP1F (EIP1.7) |
| Timer 3 | 0x0073 | 14 | TF3 (TMR3CN.7)<br>EXF3 (TMR3CN.6) | Y | | 1 | ET3 (EIE2.0) | PT3 (EIP2.0) |
| ADC0 End of Conversion | 0x007B | 15 | AD0INT (ADC0CN.5) | Y | | 0 | EADC0 (EIE2.1) | PADC0 (EIP2.1) |
| Timer 4 | 0x0083 | 16 | TF4 (TMR4CN.7)<br>EXF4 (TMR4CN.7) | Y | | 2 | ET4 (EIE2.2) | PT4 (EIP2.2) |
| ADC2 Window Comparator | 0x008B | 17 | AD2WINT (ADC2CN.0) | Y | | 2 | EWADC2 (EIE2.3) | PWADC2 (EIP2.3) |
| ADC2 End of Conversion | 0x0093 | 18 | AD2INT (ADC2CN.5) | Y | | 2 | EADC2 (EIE2.4) | PADC2 (EIP2.4) |
| RESERVED | 0x009B | 19 | N/A | N/A | N/A | N/A | N/A | N/A |
| UART1 | 0x00A3 | 20 | RI1 (SCON1.0)<br>TI1 (SCON1.1) | Y | | 1 | ES1 (EIE2.6) | PS1 (EIP2.6) |

### 11.3.3. 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. Each interrupt has an associated interrupt priority bit in an SFR (IP-EIP2) 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, given in Table 11.4.

### 11.3.4. 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 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. Additional clock cycles will be required if a cache miss occurs (see **Section “16. Branch Target Cache” on page 211** for more details). 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) is when the CPU is performing an RETI instruction followed by a DIV as the next instruction, and a cache miss event also occurs. 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.

### 11.3.5. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described below. Refer to the datasheet 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 11.12. IE: Interrupt Enable**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| EA | IEGF0 | ET2 | ES0 | ET1 | EX1 | ET0 | EX0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xA8
SFR Page: All Pages

Bit7: EA: Enable All Interrupts.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit6: IEGF0: General Purpose Flag 0.
This is a general purpose flag for use under software control.

Bit5: ET2: Enabler Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable Timer 2 interrupt.

Bit4: ES0: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit3: ET1: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable Timer 1 interrupt.
1: Enable Timer 1 interrupt.

Bit2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable External Interrupt 1.
1: Enable External Interrupt 1.

Bit1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable Timer 0 interrupts.
1: Enable Timer 0 interrupts.

Bit0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable External Interrupt 0.
1: Enable External Interrupt 0.

## SFR Definition 11.13. IP: Interrupt Priority

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | PT2 | PS0 | PT1 | PX1 | PT0 | PX0 | 11000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xB8
SFR Page: All Pages

Bits7–6: UNUSED. Read = 11b, Write = don't care.
Bit5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt set to low priority.
1: Timer 2 interrupt set to high priority.
Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority.
1: UART0 interrupts set to high priority.
Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority.
1: Timer 1 interrupts set to high priority.
Bit2: 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.
1: External Interrupt 1 set to high priority.
Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority.
1: Timer 0 interrupt set to high priority.
Bit0: 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.
1: External Interrupt 0 set to high priority.

## SFR Definition 11.14. EIE1: Extended Interrupt Enable 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| ECP1R | ECP1F | ECP0R | ECP0F | EPCA0 | EWADC0 | ESMB0 | ESPI0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE6
SFR Page: All Pages

Bit7: ECP1R: Enable Comparator1 (CP1) Rising Edge Interrupt.
This bit sets the masking of the CP1 rising edge interrupt.
0: Disable CP1 rising edge interrupts.
1: Enable CP1 rising edge interrupts.

Bit6: ECP1F: Enable Comparator1 (CP1) Falling Edge Interrupt.
This bit sets the masking of the CP1 falling edge interrupt.
0: Disable CP1 falling edge interrupts.
1: Enable CP1 falling edge interrupts.

Bit5: ECP0R: Enable Comparator0 (CP0) Rising Edge Interrupt.
This bit sets the masking of the CP0 rising edge interrupt.
0: Disable CP0 rising edge interrupts.
1: Enable CP0 rising edge interrupts.

Bit4: ECP0F: Enable Comparator0 (CP0) Falling Edge Interrupt.
This bit sets the masking of the CP0 falling edge interrupt.
0: Disable CP0 falling edge interrupts.
1: Enable CP0 falling edge interrupts.

Bit3: EPCA0: Enable Programmable Counter Array (PCA0) Interrupt.
This bit sets the masking of the PCA0 interrupts.
0: Disable PCA0 interrupts.
1: Enable PCA0 interrupts.

Bit2: EWADC0: Enable Window Comparison ADC0 Interrupt.
This bit sets the masking of ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison Interrupt.
1: Enable ADC0 Window Comparison Interrupt.

Bit1: ESMB0: Enable System Management Bus (SMBus0) Interrupt.
This bit sets the masking of the SMBus interrupt.
0: Disable SMBus interrupts.
1: Enable SMBus interrupts.

Bit0: ESPI0: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of SPI0 interrupt.
0: Disable SPI0 interrupts.
1: Enable SPI0 interrupts.

## SFR Definition 11.15. EIE2: Extended Interrupt Enable 2

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | ES1 | - | EADC2 | EWADC2 | ET4 | EADC0 | ET3 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE7
SFR Page: All Pages

Bit7: UNUSED. Read = 0b, Write = don't care.
Bit6: ES1: Enable UART1 Interrupt.
This bit sets the masking of the UART1 interrupt.
0: Disable UART1 interrupts.
1: Enable UART1 interrupts.
Bit5: UNUSED. Read = 0b, Write = don't care.
Bit4: EADC2: Enable ADC2 End Of Conversion Interrupt.
This bit sets the masking of the ADC2 End of Conversion interrupt.
0: Disable ADC2 End of Conversion interrupts.
1: Enable ADC2 End of Conversion Interrupts.
Bit3: EWADC2: Enable Window Comparison ADC2 Interrupt.
This bit sets the masking of ADC2 Window Comparison interrupt.
0: Disable ADC2 Window Comparison Interrupts.
1: Enable ADC2 Window Comparison Interrupts.
Bit2: ET4: Enable Timer 4 Interrupt
This bit sets the masking of the Timer 4 interrupt.
0: Disable Timer 4 interrupts.
1: Enable Timer 4 interrupts.
Bit1: EADC0: Enable ADC0 End of Conversion Interrupt.
This bit sets the masking of the ADC0 End of Conversion Interrupt.
0: Disable ADC0 End of Conversion Interrupts.
1: Enable ADC0 End of Conversion Interrupts.
Bit0: ET3: Enable Timer 3 Interrupt.
This bit sets the masking of the Timer 3 interrupt.
0: Disable Timer 3 interrupts.
1: Enable Timer 3 interrupts.

## SFR Definition 11.16. EIP1: Extended Interrupt Priority 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PCP1R | PCP1F | PCP0R | PCP0F | PPCA0 | PWADC0 | PSMB0 | PSPI0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xF6
SFR Page: All Pages

Bit7: PCP1R: Comparator1 (CP1) Rising Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 rising interrupt set to low priority.
1: CP1 rising interrupt set to high priority.

Bit6: PCP1F: Comparator1 (CP1) Falling Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 falling interrupt set to low priority.
1: CP1 falling interrupt set to high priority.

Bit5: PCP0R: Comparator0 (CP0) Rising Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 rising interrupt set to low priority.
1: CP0 rising interrupt set to high priority.

Bit4: PCP0F: Comparator0 (CP0) Falling Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 falling interrupt set to low priority.
1: CP0 falling interrupt set to high priority.

Bit3: PPCA0: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority.
1: PCA0 interrupt set to high priority.

Bit2: 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.
1: ADC0 Window interrupt set to high priority.

Bit1: PSMB0: System Management Bus (SMBus0) Interrupt Priority Control.
This bit sets the priority of the SMBus0 interrupt.
0: SMBus interrupt set to low priority.
1: SMBus interrupt set to high priority.

Bit0: PSPI0: Serial Peripheral Interface (SPI0) Interrupt Priority Control.
This bit sets the priority of the SPI0 interrupt.
0: SPI0 interrupt set to low priority.
1: SPI0 interrupt set to high priority.

## SFR Definition 11.17. EIP2: Extended Interrupt Priority 2

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | PS1 | - | PADC2 | PWADC2 | PT4 | PADC0 | PT3 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xF7
SFR Page: All Pages

Bit7: UNUSED. Read = 0b, Write = don't care.
Bit6: ES1: UART1 Interrupt Priority Control.
This bit sets the priority of the UART1 interrupt.
0: UART1 interrupt set to low priority.
1: UART1 interrupt set to high priority.
Bit5: UNUSED. Read = 0b, Write = don't care.
Bit4: PADC2: ADC2 End Of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC2 End of Conversion interrupt.
0: ADC2 End of Conversion interrupt set to low priority.
1: ADC2 End of Conversion interrupt set to high priority.
Bit3: PWADC2: ADC2 Window Compare Interrupt Priority Control.
This bit sets the priority of the ADC2 Window Compare interrupt.
0: ADC2 Window Compare interrupt set to low priority.
1: ADC2 Window Compare interrupt set to high priority.
Bit2: PT4: Timer 4 Interrupt Priority Control.
This bit sets the priority of the Timer 4 interrupt.
0: Timer 4 interrupt set to low priority.
1: Timer 4 interrupt set to high priority.
Bit1: PADC0: ADC0 End of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC0 End of Conversion Interrupt.
0: ADC0 End of Conversion interrupt set to low priority.
1: ADC0 End of Conversion interrupt set to high priority.
Bit0: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupts.
0: Timer 3 interrupt set to low priority.
1: Timer 3 interrupt set to high priority.

## 11.4. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the external peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the system clock is stopped. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. SFR Definition 11.18 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

Although the CIP-51 has Idle and Stop modes built in (as with any standard 8051 architecture), power management of the entire MCU is better accomplished by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and put into low power mode. Digital peripherals, such as timers or serial buses, draw little power whenever they are not in use. Turning off the Flash memory saves power, similar to entering Idle mode. Turning off the oscillator saves even more power, but requires a reset to restart the MCU.

### 11.4.1. 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. 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 or $\overline{\text{RST}}$ is asserted. 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 0x00000.

If enabled, the 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 13** for more information on the use and configuration of the WDT.

Note: Any instruction which sets the IDLE bit should be immediately followed by an instruction which has two or more opcode bytes. For example:

```
// in ‘C’:
PCON |= 0x01;    // Set IDLE bit
PCON = PCON;     // ... Followed by a 3-cycle Dummy Instruction

; in assembly:
ORL PCON, #01h   ; Set IDLE bit
MOV PCON, PCON   ; ... Followed by a 3-cycle Dummy Instruction
```

If the instruction following the write to the IDLE bit is a single-byte instruction and an interrupt occurs during the execution of the instruction of the instruction which sets the IDLE bit, the CPU may not wake from IDLE mode when a future interrupt occurs.

### 11.4.2. 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. In Stop mode, the CPU and oscillators are stopped, effectively shutting down all digital peripherals. Each analog peripheral must 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 0x00000.

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 sleep for longer than the MCD timeout of 100 µs.

**SFR Definition 11.18. PCON: Power Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | STOP | IDLE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x87
SFR Page: All Pages

Bits7–3: Reserved.
Bit1: STOP: STOP Mode Select.
Writing a ‘1’ to this bit will place the CIP-51 into STOP mode. This bit will always read ‘0’.
1: CIP-51 forced into power-down mode. (Turns off oscillator).
Bit0: IDLE: IDLE Mode Select.
Writing a ‘1’ to this bit will place the CIP-51 into IDLE mode. This bit will always read ‘0’.
1: CIP-51 forced into IDLE mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, and all peripherals remain active.)

# 12. Multiply And Accumulate (MAC0)

The C8051F120/1/2/3 and C8051F130/1/2/3 devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. MAC0 contains a 16-by-16 bit multiplier and a 40-bit adder, which can perform integer or fractional multiply-accumulate and multiply operations on signed input values in two SYSCLK cycles. A rounding engine provides a rounded 16-bit fractional result after an additional (third) SYSCLK cycle. MAC0 also contains a 1-bit arithmetic shifter that will left or right-shift the contents of the 40-bit accumulator in a single SYSCLK cycle. Figure 12.1 shows a block diagram of the MAC0 unit and its associated Special Function Registers.

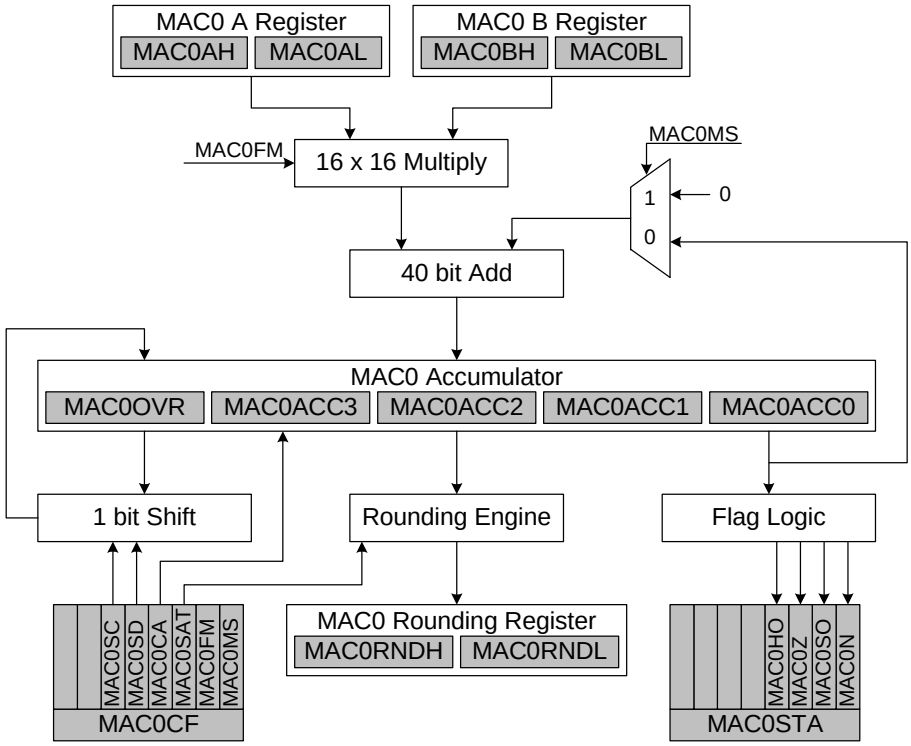



**Figure 12.1. MAC0 Block Diagram**

## 12.1. Special Function Registers

There are thirteen Special Function Register (SFR) locations associated with MAC0. Two of these registers are related to configuration and operation, while the other eleven are used to store multi-byte input and output data for MAC0. The Configuration register MAC0CF (SFR Definition 12.1) is used to configure and control MAC0. The Status register MAC0STA (SFR Definition 12.2) contains flags to indicate overflow conditions, as well as zero and negative results. The 16-bit MAC0A (MAC0AH:MAC0AL) and MAC0B (MAC0BH:MAC0BL) registers are used as inputs to the multiplier. The MAC0 Accumulator register is 40 bits long, and consists of five SFRs: MAC0OVR, MAC0ACC3, MAC0ACC2, MAC0ACC1, and MAC0ACC0. The primary results of a MAC0 operation are stored in the Accumulator registers. If they are needed, the rounded results are stored in the 16-bit Rounding Register MAC0RND (MAC0RNDH:MAC0RNDL).

## 12.2. Integer and Fractional Math

MAC0 is capable of interpreting the 16-bit inputs stored in MAC0A and MAC0B as signed integers or as signed fractional numbers. When the MAC0FM bit (MAC0CF.1) is cleared to '0', the inputs are treated as 16-bit, 2's complement, integer values. After the operation, the accumulator will contain a 40-bit, 2's complement, integer value. Figure 12.2 shows how integers are stored in the SFRs.

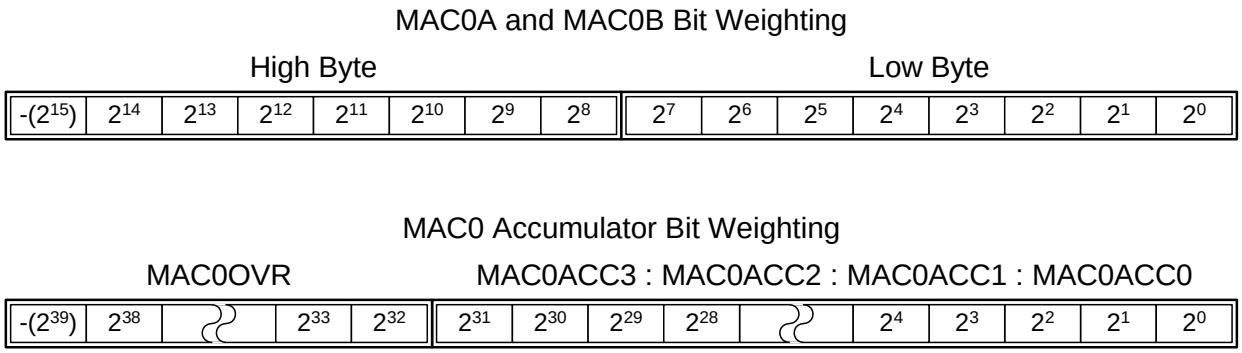
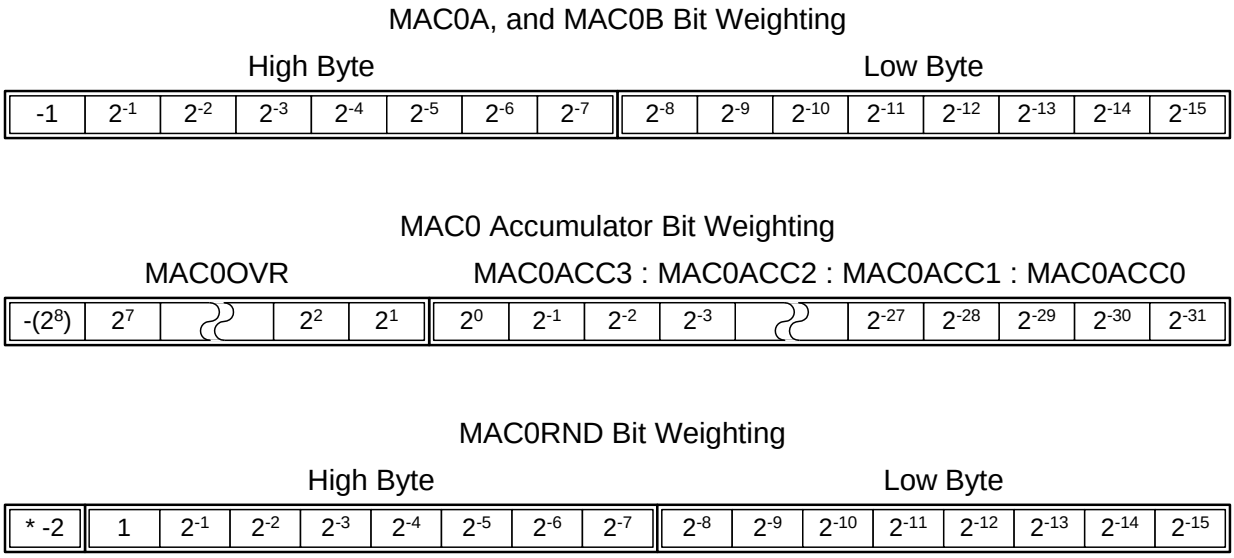



**Figure 12.2. Integer Mode Data Representation**

When the MAC0FM bit is set to '1', the inputs are treated at 16-bit, 2's complement, fractional values. The decimal point is located between bits 15 and 14 of the data word. After the operation, the accumulator will contain a 40-bit, 2's complement, fractional value, with the decimal point located between bits 31 and 30. Figure 12.3 shows how fractional numbers are stored in the SFRs.




* The MAC0RND register contains the 16 LSBs of a two's complement number. The MAC0N Flag can be used to determine the sign of the MAC0RND register.

**Figure 12.3. Fractional Mode Data Representation**

## 12.3. Operating in Multiply and Accumulate Mode

MAC0 operates in Multiply and Accumulate (MAC) mode when the MAC0MS bit (MAC0CF.0) is cleared to '0'. When operating in MAC mode, MAC0 performs a 16-by-16 bit multiply on the contents of the MAC0A and MAC0B registers, and adds the result to the contents of the 40-bit MAC0 accumulator. Figure 12.4 shows the MAC0 pipeline. There are three stages in the pipeline, each of which takes exactly one SYSCLK cycle to complete. The MAC operation is initiated with a write to the MAC0BL register. After the MAC0BL register is written, MAC0A and MAC0B are multiplied on the first SYSCLK cycle. During the second stage of the MAC0 pipeline, the results of the multiplication are added to the current accumulator contents, and the result of the addition is stored in the MAC0 accumulator. The status flags in the MAC0STA register are set after the end of the second pipeline stage. During the second stage of the pipeline, the next multiplication can be initiated by writing to the MAC0BL register, if it is desired. The rounded (and optionally, saturated) result is available in the MAC0RNDH and MAC0RNDL registers at the end of the third pipeline stage. If the MAC0CA bit (MAC0CF.3) is set to '1' when the MAC operation is initiated, the accumulator and all MAC0STA flags will be cleared during the next cycle of the controller's clock (SYSCLK). The MAC0CA bit will clear itself to '0' when the clear operation is complete.

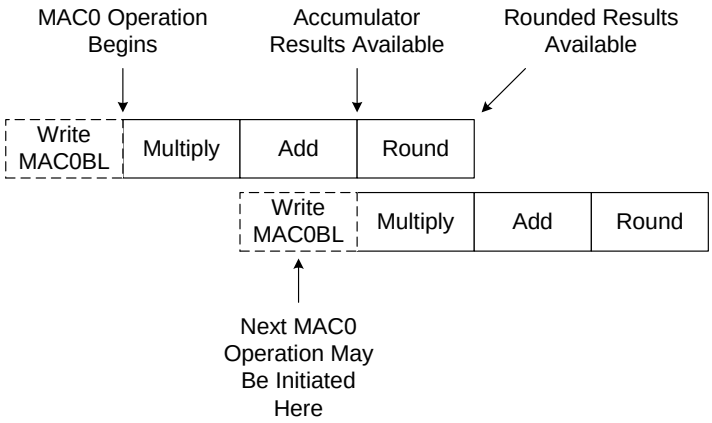



**Figure 12.4. MAC0 Pipeline**

## 12.4. Operating in Multiply Only Mode

MAC0 operates in Multiply Only mode when the MAC0MS bit (MAC0CF.0) is set to '1'. Multiply Only mode is identical to Multiply and Accumulate mode, except that the multiplication result is added with a value of zero before being stored in the MAC0 accumulator (i.e. it overwrites the current accumulator contents). The result of the multiplication is available in the MAC0 accumulator registers at the end of the second MAC0 pipeline stage (two SYSCLKs after writing to MAC0BL). As in MAC mode, the rounded result is available in the MAC0 Rounding Registers after the third pipeline stage. Note that in Multiply Only mode, the MAC0HO flag is not affected.

## 12.5. Accumulator Shift Operations

MAC0 contains a 1-bit arithmetic shift function which can be used to shift the contents of the 40-bit accumulator left or right by one bit. The accumulator shift is initiated by writing a '1' to the MAC0SC bit (MAC0CF.5), and takes one SYSCLK cycle (the rounded result is available in the MAC0 Rounding Registers after a second SYSCLK cycle, and MAC0SC is cleared to '0'). The direction of the arithmetic shift is controlled by the MAC0SD bit (MAC0CF.4). When this bit is cleared to '0', the MAC0 accumulator will shift left. When the MAC0SD bit is set to '1', the MAC0 accumulator will shift right. Right-shift operations are sign-extended with the current value of bit 39. Note that the status flags in the MAC0STA register are not affected by shift operations.

## 12.6. Rounding and Saturation

A Rounding Engine is included, which can be used to provide a rounded result when operating on fractional numbers. MAC0 uses an unbiased rounding algorithm to round the data stored in bits 31–16 of the accumulator, as shown in Table 12.1. Rounding occurs during the third stage of the MAC0 pipeline, after any shift operation, or on a write to the LSB of the accumulator. The rounded results are stored in the rounding registers: MAC0RNDH (SFR Definition 12.12) and MAC0RNDL (SFR Definition 12.13). The accumulator registers are not affected by the rounding engine. Although rounding is primarily used for fractional data, the data in the rounding registers is updated in the same way when operating in integer mode.

**Table 12.1. MAC0 Rounding (MAC0SAT = 0)**

| Accumulator Bits 15–0 (MAC0ACC1:MAC0ACC0) | Accumulator Bits 31–16 (MAC0ACC3:MAC0ACC2) | Rounding Direction | Rounded Results (MAC0RNDH:MAC0RNDL) |
|---|---|---|---|
| Greater Than 0x8000 | Anything | Up | (MAC0ACC3:MAC0ACC2) + 1 |
| Less Than 0x8000 | Anything | Down | (MAC0ACC3:MAC0ACC2) |
| Equal To 0x8000 | Odd (LSB = 1) | Up | (MAC0ACC3:MAC0ACC2) + 1 |
| Equal To 0x8000 | Even (LSB = 0) | Down | (MAC0ACC3:MAC0ACC2) |

The rounding engine can also be used to saturate the results stored in the rounding registers. If the MAC0SAT bit is set to '1' and the rounding register overflows, the rounding registers will saturate. When a positive overflow occurs, the rounding registers will show a value of 0x7FFF when saturated. For a negative overflow, the rounding registers will show a value of 0x8000 when saturated. If the MAC0SAT bit is cleared to '0', the rounding registers will not saturate.

## 12.7. Usage Examples

This section details some software examples for using MAC0. **Section 12.7.1** shows a series of two MAC operations using fractional numbers. **Section 12.7.2** shows a single operation in Multiply Only mode with integer numbers. The last example, shown in **Section 12.7.3**, demonstrates how the left-shift and right-shift operations can be used to modify the accumulator. All of the examples assume that all of the flags in the MAC0STA register are initially set to '0'.

### 12.7.1. Multiply and Accumulate Example

The example below implements the equation:

$$(0.5 \times 0.25) + (0.5 \times -0.25) = 0.125 - 0.125 = 0.0$$

```
MOV   MAC0CF, #0Ah    ; Set to Clear Accumulator, Use fractional numbers
MOV   MAC0AH, #40h    ; Load MAC0A register with 4000 hex = 0.5 decimal
MOV   MAC0AL, #00h
MOV   MAC0BH, #20h    ; Load MAC0B register with 2000 hex = 0.25 decimal
MOV   MAC0BL, #00h    ; This line initiates the first MAC operation
MOV   MAC0BH, #E0h    ; Load MAC0B register with E000 hex = -0.25 decimal
MOV   MAC0BL, #00h    ; This line initiates the second MAC operation
NOP
NOP                   ; After this instruction, the Accumulator should be equal to 0,
                      ; and the MAC0STA register should be 0x04, indicating a zero
NOP                   ; After this instruction, the Rounding register is updated
```

### 12.7.2. Multiply Only Example

The example below implements the equation:

$$4660 \times -292 = -1360720$$

```
MOV   MAC0CF, #01h    ; Use integer numbers, and multiply only mode (add to zero)
MOV   MAC0AH, #12h    ; Load MAC0A register with 1234 hex = 4660 decimal
MOV   MAC0AL, #34h
MOV   MAC0BH, #FEh    ; Load MAC0B register with FEDC hex = -292 decimal
MOV   MAC0BL, #DCh    ; This line initiates the Multiply operation
NOP
NOP                   ; After this instruction, the Accumulator should be equal to
                      ; FFFFEB3CB0 hex = -1360720 decimal. The MAC0STA register should
                      ; be 0x01, indicating a negative result.
NOP                   ; After this instruction, the Rounding register is updated
```

### 12.7.3. MAC0 Accumulator Shift Example

The example below shifts the MAC0 accumulator left one bit, and then right two bits:

```
MOV   MAC0OVR, #40h   ; The next few instructions load the accumulator with the value
MOV   MAC0ACC3, #88h  ; 4088442211 Hex.
MOV   MAC0ACC2, #44h
MOV   MAC0ACC1, #22h
MOV   MAC0ACC0, #11h
MOV   MAC0CF, #20h    ; Initiate a Left-shift
NOP                   ; After this instruction, the accumulator should be 0x8110884422
NOP                   ; The rounding register is updated after this instruction
MOV   MAC0CF, #30h    ; Initiate a Right-shift
MOV   MAC0CF, #30h    ; Initiate a second Right-shift
NOP                   ; After this instruction, the accumulator should be 0xE044221108
NOP                   ; The rounding register is updated after this instruction
```

## SFR Definition 12.1. MAC0CF: MAC0 Configuration

| R | R | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | MAC0SC | MAC0SD | MAC0CA | MAC0SAT | MAC0FM | MAC0MS | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC3
SFR Page: 3

Bits 7–6: UNUSED: Read = 00b, Write = don't care.

Bit 5: MAC0SC: Accumulator Shift Control.
When set to 1, the 40-bit MAC0 Accumulator register will be shifted during the next SYSCLK cycle. The direction of the shift (left or right) is controlled by the MAC0RS bit.
This bit is cleared to '0' by hardware when the shift is complete.

Bit 4: MAC0SD: Accumulator Shift Direction.
This bit controls the direction of the accumulator shift activated by the MAC0SC bit.
0: MAC0 Accumulator will be shifted left.
1: MAC0 Accumulator will be shifted right.

Bit 3: MAC0CA: Clear Accumulator.
This bit is used to reset MAC0 before the next operation.
When set to '1', the MAC0 Accumulator will be cleared to zero and the MAC0 Status register will be reset during the next SYSCLK cycle.
This bit will be cleared to '0' by hardware when the reset is complete.

Bit 2: MAC0SAT: Saturate Rounding Register.
This bit controls whether the Rounding Register will saturate. If this bit is set and a Soft Overflow occurs, the Rounding Register will saturate. This bit does not affect the operation of the MAC0 Accumulator. See Section 12.6 for more details about rounding and saturation.
0: Rounding Register will not saturate.
1: Rounding Register will saturate.

Bit 1: MAC0FM: Fractional Mode.
This bit selects between Integer Mode and Fractional Mode for MAC0 operations.
0: MAC0 operates in Integer Mode.
1: MAC0 operates in Fractional Mode.

Bit 0: MAC0MS: Mode Select
This bit selects between MAC Mode and Multiply Only Mode.
0: MAC (Multiply and Accumulate) Mode.
1: Multiply Only Mode.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 12.2. MAC0STA: MAC0 Status

| R | R | R | R | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | MAC0HO | MAC0Z | MAC0SO | MAC0N | 00000100 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xC0
SFR Page: 3

Bits 7–4: UNUSED: Read = 0000b, Write = don't care.

Bit 3: MAC0HO: Hard Overflow Flag.
This bit is set to '1' whenever an overflow out of the MAC0OVR register occurs during a MAC operation (i.e. when MAC0OVR changes from 0x7F to 0x80 or from 0x80 to 0x7F). The hard overflow flag must be cleared in software by directly writing it to '0', or by resetting the MAC logic using the MAC0CA bit in register MAC0CF.

Bit 2: MAC0Z: Zero Flag.
This bit is set to '1' if a MAC0 operation results in an Accumulator value of zero. If the result is non-zero, this bit will be cleared to '0'.

Bit 1: MAC0SO: Soft Overflow Flag.
This bit is set to '1' when a MAC operation causes an overflow into the sign bit (bit 31) of the MAC0 Accumulator. If the overflow condition is corrected after a subsequent MAC operation, this bit is cleared to '0'.

Bit 0: MAC0N: Negative Flag.
If the MAC Accumulator result is negative, this bit will be set to '1'. If the result is positive or zero, this flag will be cleared to '0'.

***Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 12.3. MAC0AH: MAC0 A High Byte

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC2
SFR Page: 3

Bits 7–0: High Byte (bits 15–8) of MAC0 A Register.

### SFR Definition 12.4. MAC0AL: MAC0 A Low Byte

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC1
SFR Page: 3

Bits 7–0: Low Byte (bits 7–0) of MAC0 A Register.

### SFR Definition 12.5. MAC0BH: MAC0 B High Byte

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x92
SFR Page: 3

Bits 7–0: High Byte (bits 15–8) of MAC0 B Register.

### SFR Definition 12.6. MAC0BL: MAC0 B Low Byte

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x91
SFR Page: 3

Bits 7–0: Low Byte (bits 7–0) of MAC0 B Register.
A write to this register initiates a Multiply or Multiply and Accumulate operation.

***Note:** The contents of this register should not be changed by software during the first MAC0 pipeline stage.

## SFR Definition 12.7. MAC0ACC3: MAC0 Accumulator Byte 3

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x96
SFR Page: 3

Bits 7–0: Byte 3 (bits 31–24) of MAC0 Accumulator.

***Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 12.8. MAC0ACC2: MAC0 Accumulator Byte 2

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x95
SFR Page: 3

Bits 7–0: Byte 2 (bits 23–16) of MAC0 Accumulator.

***Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

## SFR Definition 12.9. MAC0ACC1: MAC0 Accumulator Byte 1

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x94
SFR Page: 3

Bits 7–0: Byte 1 (bits 15–8) of MAC0 Accumulator.

***Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

### SFR Definition 12.10. MAC0ACC0: MAC0 Accumulator Byte 0

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x93
SFR Page: 3

Bits 7–0: Byte 0 (bits 7–0) of MAC0 Accumulator.

***Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

### SFR Definition 12.11. MAC0OVR: MAC0 Accumulator Overflow

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x97
SFR Page: 3

Bits 7–0: MAC0 Accumulator Overflow Bits (bits 39–32).

***Note:** The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

### SFR Definition 12.12. MAC0RNDH: MAC0 Rounding Register High Byte

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xCF
SFR Page: 3

Bits 7–0: High Byte (bits 15–8) of MAC0 Rounding Register.

## SFR Definition 12.13. MAC0RNDL: MAC0 Rounding Register Low Byte

| R | R | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xCE
SFR Page: 3

Bits 7–0: Low Byte (bits 7–0) of MAC0 Rounding Register.

# 13. 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 configuration.
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost even though the data on the stack are not altered.

The I/O port latches are reset to 0xFF (all logic 1's), activating internal weak pullups during and after the reset. For $V_{DD}$ Monitor resets, the $\overline{RST}$ pin is driven low until the end of the $V_{DD}$ reset timeout.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator running at its lowest frequency. Refer to Section "**14. Oscillators**" on page **185** for information on selecting and configuring the system clock source. The Watchdog Timer is enabled using its longest timeout interval (see Section "**13.7. Watchdog Timer Reset**" on page **179**). Once the system clock source is stable, program execution begins at location 0x0000.

There are seven sources for putting the MCU into the reset state: power-on, power-fail, external $\overline{RST}$ pin, external CNVSTR0 signal, software command, Comparator0, Missing Clock Detector, and Watchdog Timer. Each reset source is described in the following sections.

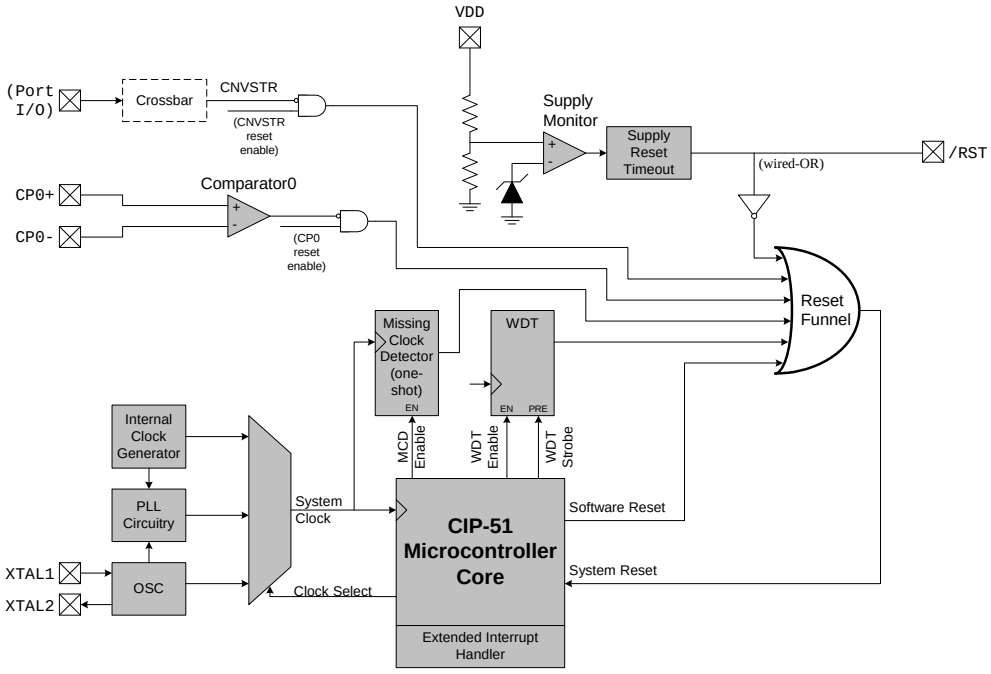



**Figure 13.1. Reset Sources**

## 13.1. Power-on Reset

The C8051F120/1/2/3/4/5/6/7 family incorporates a power supply monitor that holds the MCU in the reset state until $V_{DD}$ rises above the $V_{RST}$ level during power-up. See Figure 13.2 for timing diagram, and refer to Table 13.1 for the Electrical Characteristics of the power supply monitor circuit. The $\overline{RST}$ pin is asserted low until the end of the 100 ms $V_{DD}$ Monitor timeout in order to allow the $V_{DD}$ supply to stabilize. The $V_{DD}$ Monitor reset is enabled and disabled using the external $V_{DD}$ monitor enable pin (MONEN). When the $V_{DD}$ Monitor is enabled, it is selected as a reset source using the PORSF bit. If the RSTSRC register is written by firmware, PORSF (RSTSRC.1) must be written to '1' for the $V_{DD}$ Monitor to be effective.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. 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.

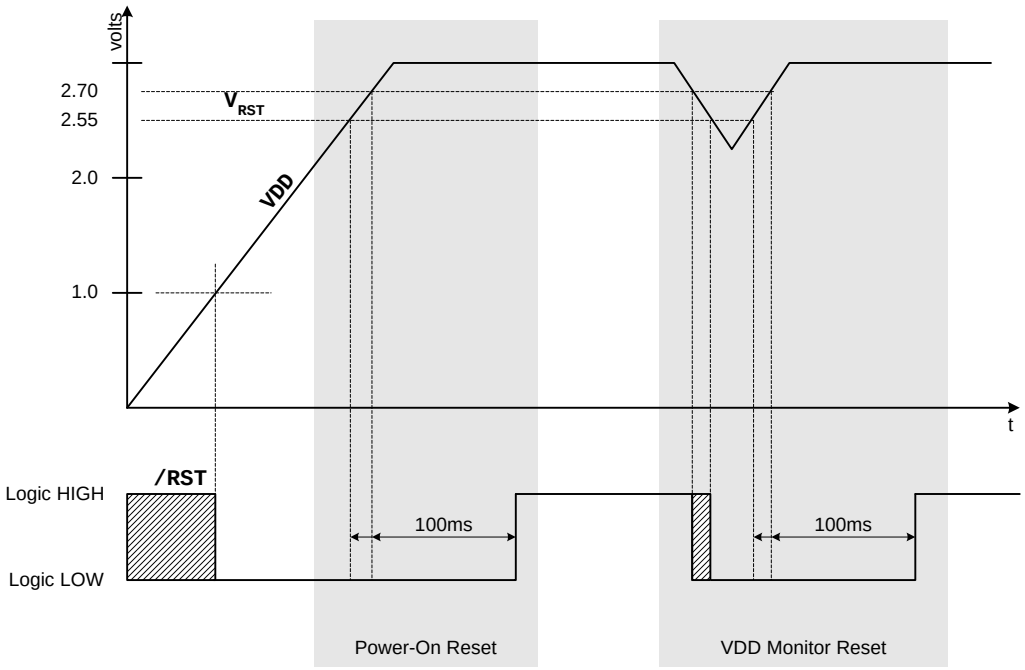



**Figure 13.2. Reset Timing**

## 13.2. Power-fail Reset

When a power-down transition or power irregularity causes $V_{DD}$ to drop below $V_{RST}$, the power supply monitor will drive the $\overline{RST}$ pin low and return the CIP-51 to the reset state. When $V_{DD}$ returns to a level above VRST, the CIP-51 will leave the reset state in the same manner as that for the power-on reset (see Figure 13.2). Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if $V_{DD}$ dropped below the level required for data retention. If the PORSF flag is set to logic 1, the data may no longer be valid.

## 13.3. External Reset

The external RST pin provides a means for external circuitry to force the MCU into a reset state. Asserting the RST pin low will cause the MCU to enter the reset state. It may be desirable to provide an external pull-up and/or decoupling of the RST pin to avoid erroneous noise-induced resets. The MCU will remain in reset until at least 12 clock cycles after the active-low RST signal is removed. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

## 13.4. Missing Clock Detector Reset

The Missing Clock Detector is essentially a one-shot circuit that is triggered by the MCU system clock. If the system clock goes away for more than 100 µs, the one-shot will time out and generate a reset. After a Missing Clock Detector reset, the MCDRSF flag (RSTSRC.2) will be set, signifying the MSD as the reset source; otherwise, this bit reads '0'. The state of the RST pin is unaffected by this reset. Setting the MCDRSF bit, RSTSRC.2 (see Section "**14. Oscillators**" on page **185**) enables the Missing Clock Detector.

## 13.5. Comparator0 Reset

Comparator0 can be configured as a reset input by writing a '1' to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled using CPT0CN.7 (see Section "**10. Comparators**" on page **119**) 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 (CP0+ pin) is less than the inverting input voltage (CP0- pin), the MCU 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 state of the RST pin is unaffected by this reset.

## 13.6. External CNVSTR0 Pin Reset

The external CNVSTR0 signal can be configured as a reset input by writing a '1' to the CNVRSEF flag (RSTSRC.6). The CNVSTR0 signal can appear on any of the P0, P1, P2 or P3 I/O pins as described in Section "**18.1. Ports 0 through 3 and the Priority Crossbar Decoder**" on page **238**. Note that the Crossbar must be configured for the CNVSTR0 signal to be routed to the appropriate Port I/O. The Crossbar should be configured and enabled before the CNVRSEF is set. When configured as a reset, CNVSTR0 is active-low and level sensitive. CNVSTR0 cannot be used to start ADC0 conversions when it is configured as a reset source. After a CNVSTR0 reset, the CNVRSEF flag (RSTSRC.6) will read '1' signifying CNVSTR0 as the reset source; otherwise, this bit reads '0'. The state of the /RST pin is unaffected by this reset.

## 13.7. Watchdog Timer Reset

The MCU includes a programmable Watchdog Timer (WDT) running off the system clock. A WDT overflow will force the MCU into the reset state. To prevent the reset, the WDT must be restarted by application software before overflow. If the system experiences a software or hardware malfunction preventing the software from restarting the WDT, the WDT will overflow and cause a reset. This should prevent the system from running out of control.

Following a reset the WDT is automatically enabled and running with the default maximum time interval. If desired the WDT can be disabled by system software or locked on to prevent accidental disabling. Once locked, the WDT cannot be disabled until the next system reset. The state of the RST pin is unaffected by this reset.

The WDT consists of a 21-bit timer running from the programmed system clock. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT

reset is generated. The WDT can be enabled and disabled as needed in software, or can be permanently enabled if desired. Watchdog features are controlled via the Watchdog Timer Control Register (WDTCN) shown in SFR Definition 13.1.

### 13.7.1. Enable/Reset WDT

The watchdog timer is both enabled and reset by writing 0xA5 to the WDTCN register. The user's application software should include periodic writes of 0xA5 to WDTCN as needed to prevent a watchdog timer overflow. The WDT is enabled and reset as a result of any system reset.

### 13.7.2. Disable WDT

Writing 0xDE followed by 0xAD to the WDTCN register disables the WDT. The following code segment illustrates disabling the WDT:

```
CLR    EA            ; disable all interrupts
MOV    WDTCN,#0DEh   ; disable software watchdog timer
MOV    WDTCN,#0ADh
SETB   EA            ; re-enable interrupts
```

The writes of 0xDE and 0xAD must occur within 4 clock cycles of each other, or the disable operation is ignored. This means that the prefetch engine should be enabled and interrupts should be disabled during this procedure to avoid any delay between the two writes.

### 13.7.3. Disable WDT Lockout

Writing 0xFF to WDTCN locks out the disable feature. Once locked out, the disable operation is ignored until the next system reset. Writing 0xFF does not enable or reset the watchdog timer. Applications always intending to use the watchdog should write 0xFF to WDTCN in the initialization code.

### 13.7.4. Setting WDT Interval

WDTCN.[2:0] control the watchdog timeout interval. The interval is given by the following equation:

$4^{3 + WDTCN[2-0]} \times T_{sysclk}$ ; where $T_{sysclk}$ is the system clock period.

For a 3 MHz system clock, this provides an interval range of 0.021 ms to 349.5 ms. WDTCN.7 must be logic 0 when setting this interval. Reading WDTCN returns the programmed interval. WDTCN.[2:0] reads 111b after a system reset.

## SFR Definition 13.1. WDTCN: Watchdog Timer Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | xxxxx111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xFF
SFR Page: All Pages

Bits7–0: WDT Control
Writing 0xA5 both enables and reloads the WDT.
Writing 0xDE followed within 4 system clocks by 0xAD disables the WDT.
Writing 0xFF locks out the disable feature.

Bit4: Watchdog Status Bit (when Read)
Reading the WDTCN.[4] bit indicates the Watchdog Timer Status.
0: WDT is inactive
1: WDT is active

Bits2–0: Watchdog Timeout Interval Bits
The WDTCN.[2:0] bits set the Watchdog Timeout Interval. When writing these bits, WDTCN.7 must be set to 0.

## SFR Definition 13.2. RSTSRC: Reset Source

| R | R/W | R/W | R/W | R | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | CNVRSEF | C0RSEF | SWRSEF | WDTRSF | MCDRSF | PORSF | PINRSF | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xEF
SFR Page: 0

Bit7: Reserved.

Bit6: CNVRSEF: Convert Start 0 Reset Source Enable and Flag
Write: 0: CNVSTR0 is not a reset source.
1: CNVSTR0 is a reset source (active low).
Read: 0: Source of prior reset was not CNVSTR0.
1: Source of prior reset was CNVSTR0.

Bit5: C0RSEF: Comparator0 Reset Enable and Flag.
Write: 0: Comparator0 is not a reset source.
1: Comparator0 is a reset source (active low).
Read: 0: Source of last reset was not Comparator0.
1: Source of last reset was Comparator0.

Bit4: SWRSF: Software Reset Force and Flag.
Write: 0: No effect.
1: Forces an internal reset. $\overline{\text{RST}}$ pin is not effected.
Read: 0: Source of last reset was not a write to the SWRSF bit.
1: Source of last reset was a write to the SWRSF bit.

Bit3: WDTRSF: Watchdog Timer Reset Flag.
0: Source of last reset was not WDT timeout.
1: Source of last reset was WDT timeout.

Bit2: MCDRSF: Missing Clock Detector Flag.
Write: 0: Missing Clock Detector disabled.
1: Missing Clock Detector enabled; triggers a reset if a missing clock condition is detected.
Read: 0: Source of last reset was not a Missing Clock Detector timeout.
1: Source of last reset was a Missing Clock Detector timeout.

Bit1: PORSF: Power-On Reset Flag.
Write: If the $V_{DD}$ monitor circuitry is enabled (by tying the MONEN pin to a logic high state), this bit can be written to select or de-select the $V_{DD}$ monitor as a reset source.
0: De-select the $V_{DD}$ monitor as a reset source.
1: Select the $V_{DD}$ monitor as a reset source.
**Important: At power-on, the $V_{DD}$ monitor is enabled/disabled using the external $V_{DD}$ monitor enable pin (MONEN). The PORSF bit does not disable or enable the $V_{DD}$ monitor circuit. It simply selects the $V_{DD}$ monitor as a reset source.**
Read: This bit is set whenever a power-on reset occurs. This may be due to a true power-on reset or a $V_{DD}$ monitor reset. In either case, data memory should be considered indeterminate following the reset.
0: Source of last reset was not a power-on or $V_{DD}$ monitor reset.
1: Source of last reset was a power-on or $V_{DD}$ monitor reset.
**Note: When this flag is read as '1', all other reset flags are indeterminate.**

Bit0: PINRSF: HW Pin Reset Flag.
Write: 0: No effect.
1: Forces a Power-On Reset. $\overline{\text{RST}}$ is driven low.
Read: 0: Source of prior reset was not $\overline{\text{RST}}$ pin.
1: Source of prior reset was $\overline{\text{RST}}$ pin.

### Table 13.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $\overline{\text{RST}}$ Output Low Voltage | $I_{OL}$ = 8.5 mA, $V_{DD}$ = 2.7 to 3.6 V | — | — | 0.6 | V |
| $\overline{\text{RST}}$ Input High Voltage | | 0.7 x $V_{DD}$ | — | — | V |
| $\overline{\text{RST}}$ Input Low Voltage | | — | — | 0.3 x $V_{DD}$ | |
| $\overline{\text{RST}}$ Input Leakage Current | $\overline{\text{RST}}$ = 0.0 V | — | 50 | — | µA |
| $V_{DD}$ for $\overline{\text{RST}}$ Output Valid | | 1.0 | — | — | V |
| AV+ for $\overline{\text{RST}}$ Output Valid | | 1.0 | — | — | V |
| $V_{DD}$ POR Threshold ($V_{RST}$)* | | 2.40 | 2.55 | 2.70 | V |
| Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset | | 10 | — | — | ns |
| Reset Time Delay | $\overline{\text{RST}}$ rising edge after $V_{DD}$ crosses $V_{RST}$ threshold | 80 | 100 | 120 | ms |
| Missing Clock Detector Timeout | Time from last system clock to reset initiation | 100 | 220 | 500 | µs |
| ***Note:** When operating at frequencies above 50 MHz, minimum $V_{DD}$ supply Voltage is 3.0 V. | | | | | |

## NOTES:

# 14. Oscillators

The devices include a programmable internal oscillator and an external oscillator drive circuit. The internal oscillator can be enabled, disabled, and calibrated using the OSCICN and OSCICL registers, as shown in Figure 14.1. The system clock can be sourced by the external oscillator circuit, the internal oscillator, or the on-chip phase-locked loop (PLL). The internal oscillator's electrical specifications are given in Table 14.1 on page 185.

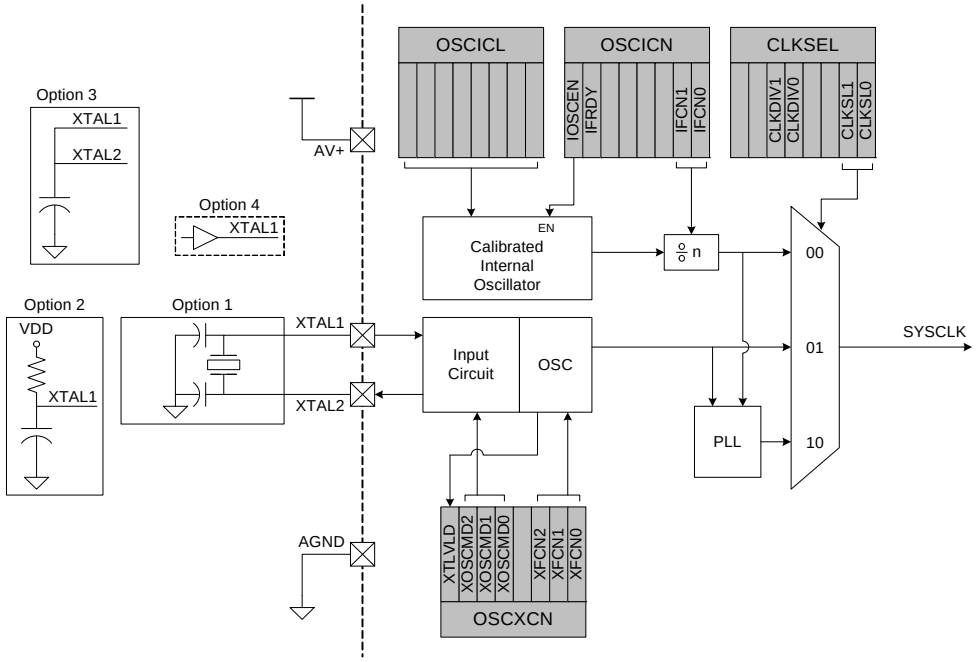



**Figure 14.1. Oscillator Diagram**

### Table 14.1. Oscillator Electrical Characteristics

–40°C to +85°C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Calibrated Internal Oscillator Frequency | | 24 | 24.5 | 25 | MHz |
| Internal Oscillator Supply Current (from $V_{DD}$) | OSCICN.7 = 1 | — | 400 | — | µA |
| External Clock Frequency | | 0 | — | 30 | MHz |
| $T_{XCH}$ (External Clock High Time) | | 15 | — | — | ns |
| $T_{XCL}$ (External Clock Low Time) | | 15 | — | — | ns |

## 14.1. Internal Calibrated Oscillator

All devices include a calibrated internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register as defined by SFR Definition 14.1. OSCICL is factory calibrated to obtain a 24.5 MHz frequency.

Electrical specifications for the precision internal oscillator are given in Table 14.1. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN.

### SFR Definition 14.1. OSCICL: Internal Oscillator Calibration.

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Variable |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8B
SFR Page: F

Bits 7–0: OSCICL: Internal Oscillator Calibration Register.
This register calibrates the internal oscillator period. The reset value for OSCICL defines the internal oscillator base frequency. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

### SFR Definition 14.2. OSCICN: Internal Oscillator Control

| R/W | R | R/W | R | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| IOSCEN | IFRDY | - | - | - | - | IFCN1 | IFCN0 | 11000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8A
SFR Page: F

Bit 7: IOSCEN: Internal Oscillator Enable Bit.
0: Internal Oscillator Disabled.
1: Internal Oscillator Enabled.
Bit 6: IFRDY: Internal Oscillator Frequency Ready Flag.
0: Internal Oscillator not running at programmed frequency.
1: Internal Oscillator running at programmed frequency.
Bits 5–2: Reserved.
Bits 1–0: IFCN1-0: Internal Oscillator Frequency Control Bits.
00: Internal Oscillator is divided by 8.
01: Internal Oscillator is divided by 4.
10: Internal Oscillator is divided by 2.
11: Internal Oscillator is divided by 1.

## 14.2. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 14.1. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 and/or XTAL1 pin(s) as shown in Option 2, 3, or 4 of Figure 14.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 14.4).

## 14.3. System Clock Selection

The CLKSL1-0 bits in register CLKSEL select which oscillator source generates the system clock. CLKSL1-0 must be set to ‘01’ for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals, such as the timers and PCA, when the internal oscillator or the PLL is selected as the system clock. The system clock may be switched on-the-fly between the internal and external oscillators or the PLL, so long as the selected oscillator source is enabled and settled. The internal oscillator requires little start-up time, and may be enabled and selected as the system clock in the same write to OSCICN. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use as the system clock. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to ‘1’ by hardware when the external oscillator is settled. To avoid reading a false XTLVLD, in crystal mode software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD. RC and C modes typically require no startup time. The PLL also requires time to lock onto the desired frequency, and the PLL Lock Flag (PLLLCK in register PLL0CN) is set to ‘1’ by hardware once the PLL is locked on the correct frequency.

## SFR Definition 14.3. CLKSEL: System Clock Selection

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | CLKDIV1 | CLKDIV0 | - | - | CLKSL1 | CLKSL0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x97
SFR Page: F

Bits 7–6: Reserved.
Bits 5–4: CLKDIV1–0: Output SYSCLK Divide Factor.
These bits can be used to pre-divide SYSCLK before it is output to a port pin through the crossbar.
00: Output will be SYSCLK.
01: Output will be SYSCLK/2.
10: Output will be SYSCLK/4.
11: Output will be SYSCLK/8.
See **Section “18. Port Input/Output” on page 235** for more details about routing this output to a port pin.
Bits 3–2: Reserved.
Bits 1–0: CLKSL1–0: System Clock Source Select Bits.
00: SYSCLK derived from the Internal Oscillator, and scaled as per the IFCN bits in OSCICN.
01: SYSCLK derived from the External Oscillator circuit.
10: SYSCLK derived from the PLL.
11: Reserved.

## SFR Definition 14.4. OSCXCN: External Oscillator Control

| R | R/W | R/W | R/W | R | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| XTLVLD | XOSCMD2 | XOSCMD1 | XOSCMD0 | - | XFCN2 | XFCN1 | XFCN0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8C
SFR Page: F

Bit7: XTLVLD: Crystal Oscillator Valid Flag.
(**Valid only when XOSCMD = 11x.**)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits6–4: XOSCMD2–0: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode (External CMOS Clock input on XTAL1 pin).
011: External CMOS Clock Mode with divide by 2 stage (External CMOS Clock input on XTAL1 pin).
10x: RC/C Oscillator Mode with divide by 2 stage.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit3: RESERVED. Read = 0, Write = don't care.

Bits2–0: XFCN2–0: External Oscillator Frequency Control Bits.
000-111: see table below:

| XFCN | Crystal (XOSCMD = 11x) | RC (XOSCMD = 10x) | C (XOSCMD = 10x) |
|---|---|---|---|
| 000 | f ≤ 32 kHz | f ≤ 25 kHz | K Factor = 0.87 |
| 001 | 32 kHz < f ≤ 84 kHz | 25 kHz < f ≤ 50 kHz | K Factor = 2.6 |
| 010 | 84 kHz < f ≤ 225 kHz | 50 kHz < f ≤ 100 kHz | K Factor = 7.7 |
| 011 | 225 kHz < f ≤ 590 kHz | 100 kHz < f ≤ 200 kHz | K Factor = 22 |
| 100 | 590 kHz < f ≤ 1.5 MHz | 200 kHz < f ≤ 400 kHz | K Factor = 65 |
| 101 | 1.5 MHz < f ≤ 4 MHz | 400 kHz < f ≤ 800 kHz | K Factor = 180 |
| 110 | 4 MHz < f ≤ 10 MHz | 800 kHz < f ≤ 1.6 MHz | K Factor = 664 |
| 111 | 10 MHz < f ≤ 30 MHz | 1.6 MHz < f ≤ 3.2 MHz | K Factor = 1590 |

**CRYSTAL MODE** (Circuit from Figure 14.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal frequency.

**RC MODE** (Circuit from Figure 14.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
**f** = **1.23(10$^3$) / (R * C)**, where
f = frequency of oscillation in MHz
C = capacitor value in pF
R = Pullup resistor value in kΩ

**C MODE** (Circuit from Figure 14.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
**f** = **KF / (C *** $V_{DD}$**)**, where
f = frequency of oscillation in MHz
C = capacitor value on XTAL1, XTAL2 pins in pF
$V_{DD}$ = Power Supply on MCU in Volts

## 14.4. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 14.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 14.4 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is enabled, the oscillator amplitude detection circuit requires a settle time to achieve proper bias. Waiting at least 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Enable the external oscillator.
Step 2. Wait at least 1 ms.
Step 3. Poll for XTLVLD => '1'.
Step 4. Switch the system clock to the external oscillator.

**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.

## 14.5. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 14.1, Option 2. 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. 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. If the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

$f = 1.23(10^3)/RC = 1.23\ (10^3)/[246 \times 50] = 0.1$ MHz = 100 kHz
Referring to the table in SFR Definition 14.4, the required XFCN setting is 010.

## 14.6. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 14.1, Option 3. 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. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation from the equations below. Assume $V_{DD}$ = 3.0 V and C = 50 pF:

$f = KF/( C \times V_{DD} ) = KF/( 50 \times 3 )$
$f = KF/150$
If a frequency of roughly 50 kHz is desired, select the K Factor from the table in SFR Definition 14.4 as KF = 7.7:

f = 7.7/150 = 0.051 MHz, or 51 kHz

Therefore, the XFCN value to use in this example is 010.

## 14.7. Phase-Locked Loop (PLL)

A Phase-Locked-Loop (PLL) is included, which is used to multiply the internal oscillator or an external clock source to achieve higher CPU operating frequencies. The PLL circuitry is designed to produce an output frequency between 25 MHz and 100 MHz, from a divided reference frequency between 5 MHz and 30 MHz. A block diagram of the PLL is shown in Figure 14.2.

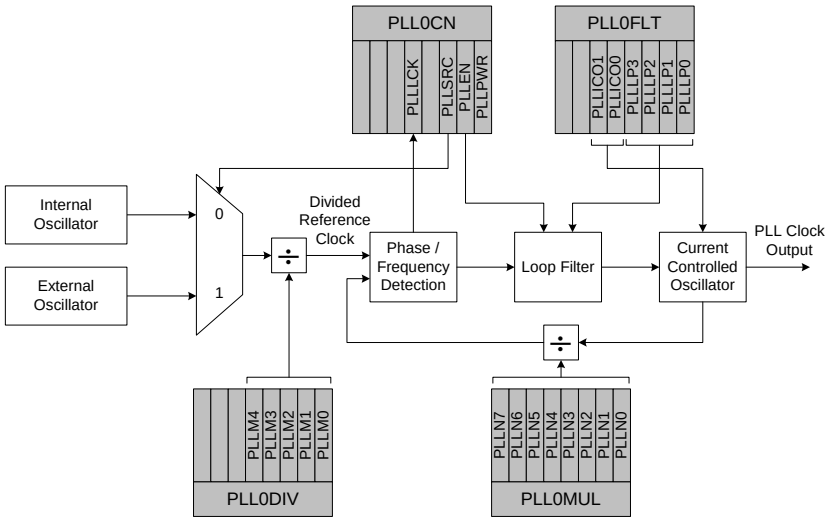



**Figure 14.2. PLL Block Diagram**

### 14.7.1. PLL Input Clock and Pre-divider

The PLL circuitry can derive its reference clock from either the internal oscillator or an external clock source. The PLLSRC bit (PLL0CN.2) controls which clock source is used for the reference clock (see SFR Definition 14.5). If PLLSRC is set to ‘0’, the internal oscillator source is used. Note that the internal oscillator divide factor (as specified by bits IFCN1-0 in register OSCICN) will also apply to this clock. When PLLSRC is set to ‘1’, an external oscillator source will be used. The external oscillator should be active and settled before it is selected as a reference clock for the PLL circuit. The reference clock is divided down prior to the PLL circuit, according to the contents of the PLLM4-0 bits in the PLL Pre-divider Register (PLL0DIV), shown in SFR Definition 14.6.

### 14.7.2. PLL Multiplication and Output Clock

The PLL circuitry will multiply the divided reference clock by the multiplication factor stored in the PLL0MUL register shown in SFR Definition 14.7. To accomplish this, it uses a feedback loop consisting of a phase/frequency detector, a loop filter, and a current-controlled oscillator (ICO). It is important to configure the loop filter and the ICO for the correct frequency ranges. The PLLLP3–0 bits (PLL0FLT.3–0) should be set according to the divided reference clock frequency. Likewise, the PLLICO1–0 bits (PLL0FLT.5–4) should be set according to the desired output frequency range. SFR Definition 14.8 describes the proper settings to use for the PLLLP3–0 and PLLICO1–0 bits. When the PLL is locked and stable at the desired frequency, the PLLLCK bit (PLL0CN.5) will be set to a ‘1’. The resulting PLL frequency will be set according to the equation:
Where “Reference Frequency” is the selected source clock frequency, PLLN is the PLL Multiplier, and PLLM is the PLL Pre-divider.

$$\text{PLL Frequency} = \text{Reference Frequency} \times \frac{\text{PLLN}}{\text{PLLM}}$$

### 14.7.3. Powering on and Initializing the PLL

To set up and use the PLL as the system clock after power-up of the device, the following procedure should be implemented:

Step 1. Ensure that the reference clock to be used (internal or external) is running and stable.
Step 2. Set the PLLSRC bit (PLL0CN.2) to select the desired clock source for the PLL.
Step 3. Program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see **Section “15. Flash Memory” on page 199**).
Step 4. Enable power to the PLL by setting PLLPWR (PLL0CN.0) to ‘1’.
Step 5. Program the PLL0DIV register to produce the divided reference frequency to the PLL.
Step 6. Program the PLLLP3–0 bits (PLL0FLT.3–0) to the appropriate range for the divided reference frequency.
Step 7. Program the PLLICO1–0 bits (PLL0FLT.5–4) to the appropriate range for the PLL output frequency.
Step 8. Program the PLL0MUL register to the desired clock multiplication factor.
Step 9. Wait at least 5 µs, to provide a fast frequency lock.
Step 10. Enable the PLL by setting PLLEN (PLL0CN.1) to ‘1’.
Step 11. Poll PLLLCK (PLL0CN.4) until it changes from ‘0’ to ‘1’.
Step 12. Switch the System Clock source to the PLL using the CLKSEL register.

If the PLL characteristics need to be changed when the PLL is already running, the following procedure should be implemented:

Step 1. The system clock should first be switched to either the internal oscillator or an external clock source that is running and stable, using the CLKSEL register.
Step 2. Ensure that the reference clock to be used for the new PLL setting (internal or external) is running and stable.
Step 3. Set the PLLSRC bit (PLL0CN.2) to select the new clock source for the PLL.
Step 4. If moving to a faster frequency, program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see **Section “15. Flash Memory” on page 199**).
Step 5. Disable the PLL by setting PLLEN (PLL0CN.1) to ‘0’.
Step 6. Program the PLL0DIV register to produce the divided reference frequency to the PLL.
Step 7. Program the PLLLP3–0 bits (PLL0FLT.3–0) to the appropriate range for the divided reference frequency.
Step 8. Program the PLLICO1-0 bits (PLL0FLT.5–4) to the appropriate range for the PLL output frequency.
Step 9. Program the PLL0MUL register to the desired clock multiplication factor.
Step 10. Enable the PLL by setting PLLEN (PLL0CN.1) to ‘1’.
Step 11. Poll PLLLCK (PLL0CN.4) until it changes from ‘0’ to ‘1’.
Step 12. Switch the System Clock source to the PLL using the CLKSEL register.
Step 13. If moving to a slower frequency, program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see **Section “15. Flash Memory” on**

page 199). **Important Note: Cache reads, cache writes, and the prefetch engine should be disabled whenever the FLRT bits are changed to a lower setting.**

To shut down the PLL, the system clock should be switched to the internal oscillator or a stable external clock source, using the CLKSEL register. Next, disable the PLL by setting PLLEN (PLL0CN.1) to '0'. Finally, the PLL can be powered off, by setting PLLPWR (PLL0CN.0) to '0'. Note that the PLLEN and PLLPWR bits can be cleared at the same time.

### SFR Definition 14.5. PLL0CN: PLL Control

| R/W | R/W | R/W | R | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | PLLLCK | 0 | PLLSRC | PLLEN | PLLPWR | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x89
SFR Page: F

Bits 7–5: UNUSED: Read = 000b; Write = don't care.
Bit 4: PLLCK: PLL Lock Flag.
0: PLL Frequency is not locked.
1: PLL Frequency is locked.
Bit 3: RESERVED. Must write to '0'.
Bit 2: PLLSRC: PLL Reference Clock Source Select Bit.
0: PLL Reference Clock Source is Internal Oscillator.
1: PLL Reference Clock Source is External Oscillator.
Bit 1: PLLEN: PLL Enable Bit.
0: PLL is held in reset.
1: PLL is enabled. PLLPWR must be '1'.
Bit 0: PLLPWR: PLL Power Enable.
0: PLL bias generator is de-activated. No static power is consumed.
1: PLL bias generator is active. Must be set for PLL to operate.

### SFR Definition 14.6. PLL0DIV: PLL Pre-divider

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | PLLM4 | PLLM3 | PLLM2 | PLLM1 | PLLM0 | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8D
SFR Page: F

Bits 7–5: UNUSED: Read = 000b; Write = don't care.
Bits 4–0: PLLM4–0: PLL Reference Clock Pre-divider.
These bits select the pre-divide value of the PLL reference clock. When set to any non-zero value, the reference clock will be divided by the value in PLLM4–0. When set to '00000b', the reference clock will be divided by 32.

### SFR Definition 14.7. PLL0MUL: PLL Clock Scaler

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PLLN7 | PLLN6 | PLLN5 | PLLN4 | PLLN3 | PLLN2 | PLLN1 | PLLN0 | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8E
SFR Page: F

Bits 7–0: PLLN7–0: PLL Multiplier.
These bits select the multiplication factor of the divided PLL reference clock. When set to any non-zero value, the multiplication factor will be equal to the value in PLLN7-0. When set to '00000000b', the multiplication factor will be equal to 256.

### SFR Definition 14.8. PLL0FLT: PLL Filter

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | PLLICO1 | PLLICO0 | PLLLP3 | PLLLP2 | PLLLP1 | PLLLP0 | 00110001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8F
SFR Page: F

Bits 7–6: UNUSED: Read = 00b; Write = don't care.
Bits 5–4: PLLICO1-0: PLL Current-Controlled Oscillator Control Bits.
Selection is based on the desired output frequency, according to the following table:

| PLL Output Clock | PLLICO1-0 |
|---|---|
| 65–100 MHz | 00 |
| 45–80 MHz | 01 |
| 30–60 MHz | 10 |
| 25–50 MHz | 11 |

Bits 3–0: PLLLP3-0: PLL Loop Filter Control Bits.
Selection is based on the divided PLL reference clock, according to the following table:

| Divided PLL Reference Clock | PLLLP3-0 |
|---|---|
| 19–30 MHz | 0001 |
| 12.2–19.5 MHz | 0011 |
| 7.8–12.5 MHz | 0111 |
| 5–8 MHz | 1111 |

### Table 14.2. PLL Frequency Characteristics

–40 to +85 °C unless otherwise specified

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Input Frequency (Divided Reference Frequency) | | 5 | | 30 | MHz |
| PLL Output Frequency | | 25 | | 100* | MHz |
| ***Note:** The maximum operating frequency of the C8051F124/5/6/7 is 50 MHz | | | | | |

**Table 14.3. PLL Lock Timing Characteristics**

–40 to +85 °C unless otherwise specified

| Input Frequency | Multiplier (Pll0mul) | Pll0flt Setting | Output Frequency | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|---|
| 5 MHz | 20 | 0x0F | 100 MHz | | 202 | | µs |
| | 13 | 0x0F | 65 MHz | | 115 | | µs |
| | 16 | 0x1F | 80 MHz | | 241 | | µs |
| | 9 | 0x1F | 45 MHz | | 116 | | µs |
| | 12 | 0x2F | 60 MHz | | 258 | | µs |
| | 6 | 0x2F | 30 MHz | | 112 | | µs |
| | 10 | 0x3F | 50 MHz | | 263 | | µs |
| | 5 | 0x3F | 25 MHz | | 113 | | µs |
| 25 MHz | 4 | 0x01 | 100 MHz | | 42 | | µs |
| | 2 | 0x01 | 50 MHz | | 33 | | µs |
| | 3 | 0x11 | 75 MHz | | 48 | | µs |
| | 2 | 0x11 | 50 MHz | | 17 | | µs |
| | 2 | 0x21 | 50 MHz | | 42 | | µs |
| | 1 | 0x21 | 25 MHz | | 33 | | µs |
| | 2 | 0x31 | 50 MHz | | 60 | | µs |
| | 1 | 0x31 | 25 MHz | | 25 | | µs |

## NOTES:

# 15. Flash Memory

All devices include either 128 kB (C8051F12x and C8051F130/1) or 64 kB (C8051F132/3) of on-chip, reprogrammable Flash memory for program code or non-volatile data storage. An additional 256-byte page of Flash is also included for non-volatile data storage. The Flash memory can be programmed in-system through the JTAG interface, or by software using the MOVX write instructions. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Bytes should be erased (set to 0xFF) before being reprogrammed. Flash write and erase operations are automatically timed by hardware for proper execution. During a Flash erase or write, the FLBUSY bit in the FLSTAT register is set to '1' (see SFR Definition 16.5). During this time, instructions that are located in the prefetch buffer or the branch target cache can be executed, but the processor will stall until the erase or write is completed if instruction data must be fetched from Flash memory. Interrupts that have been pre-loaded into the branch target cache can also be serviced at this time, if the current code is also executing from the prefetch engine or cache memory. Any interrupts that are not pre-loaded into cache, or that occur while the core is halted, will be held in a pending state during the Flash write/erase operation, and serviced in priority order once the Flash operation has completed. Refer to Table 15.1 for the electrical characteristics of the Flash memory.

## 15.1. Programming the Flash Memory

The simplest means of programming the Flash memory is through the JTAG interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the JTAG commands to program Flash memory, see **Section "25. JTAG (IEEE 1149.1)" on page 341**.

The Flash memory can be programmed from software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1. This directs the MOVX writes to Flash memory instead of to XRAM, which is the default target. The PSWE bit remains set until cleared by software. To avoid errant Flash writes, it is recommended that interrupts be disabled while the PSWE bit is logic 1.

Flash memory is read using the MOVC instruction. MOVX reads are always directed to XRAM, regardless of the state of PSWE.

On the devices with 128 kB of Flash, the COBANK bits in the PSBANK register (SFR Definition 11.1) determine which of the upper three Flash banks are mapped to the address range 0x08000 to 0x0FFFF for Flash writes, reads and erases.

For devices with 64 kB of Flash. the COBANK bits should always remain set to '01' to ensure that Flash write, erase, and read operations are valid.

**NOTE: To ensure the integrity of Flash memory contents, it is strongly recommended that the on-chip $V_{DD}$ monitor be enabled by connecting the $V_{DD}$ monitor enable pin (MONEN) to $V_{DD}$ and setting the PORSF bit in the RSTSRC register to '1' in any system that writes and/or erases Flash memory from software. See "Reset Sources" on page 177 for more information.**

A write to Flash memory can clear bits but cannot set them; only an erase operation can set bits in Flash. **A byte location to be programmed must be erased before a new value can be written**.

Write/Erase timing is automatically controlled by hardware. Note that on the 128 k Flash versions, 1024 bytes beginning at location 0x1FC00 are reserved. Flash writes and erases targeting the reserved area should be avoided.

**Table 15.1. Flash Electrical Characteristics**

$V_{DD}$ = 2.7 to 3.6 V; –40 to +85 °C

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Flash Size[1] | C8051F12x and C8051F130/1 | | 131328[2] | | Bytes |
| Flash Size[1] | C8051F132/3 | | 65792 | | Bytes |
| Endurance | | 20k | 100k | | Erase/Write |
| Erase Cycle Time | | 10 | 12 | 14 | ms |
| Write Cycle Time | | 40 | 50 | 60 | µs |

**Notes:**
1. Includes 256-byte Scratch Pad Area
2. 1024 Bytes at location 0x1FC00 to 0x1FFFF are reserved.

### 15.1.1. 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 and erased using the MOVX write instruction (as described in **Section 15.1.2** and **Section 15.1.3**) and read using the MOVC instruction. The COBANK bits in register PSBANK (SFR Definition 11.1) control which portion of the Flash memory is targeted by writes and erases of addresses above 0x07FFF. For devices with 64 kB of Flash. the COBANK bits should always remain set to '01' to ensure that Flash write, erase, and read operations are valid.

Two additional 128-byte sectors (256 bytes total) of Flash memory are included for non-volatile data storage. The smaller sector size makes them particularly well suited as general purpose, non-volatile scratch-pad memory. Even though Flash memory can be written a single byte at a time, an entire sector must be erased first. In order to change a single byte of a multi-byte data set, the data must be moved to temporary storage. The 128-byte sector-size facilitates updating data without wasting program memory or RAM space. The 128-byte sectors are double-mapped over the normal Flash memory for MOVC reads and MOVX writes only; their addresses range from 0x00 to 0x7F and from 0x80 to 0xFF (see Figure 15.2). To access the 128-byte sectors, the SFLE bit in PSCTL must be set to logic 1. Code execution from the 128-byte Scratchpad areas is not permitted. The 128-byte sectors can be erased individually, or both at the same time. To erase both sectors simultaneously, the address 0x0400 should be targeted during the erase operation with SFLE set to '1'. See Figure 15.1 for the memory map under different COBANK and SFLE settings.

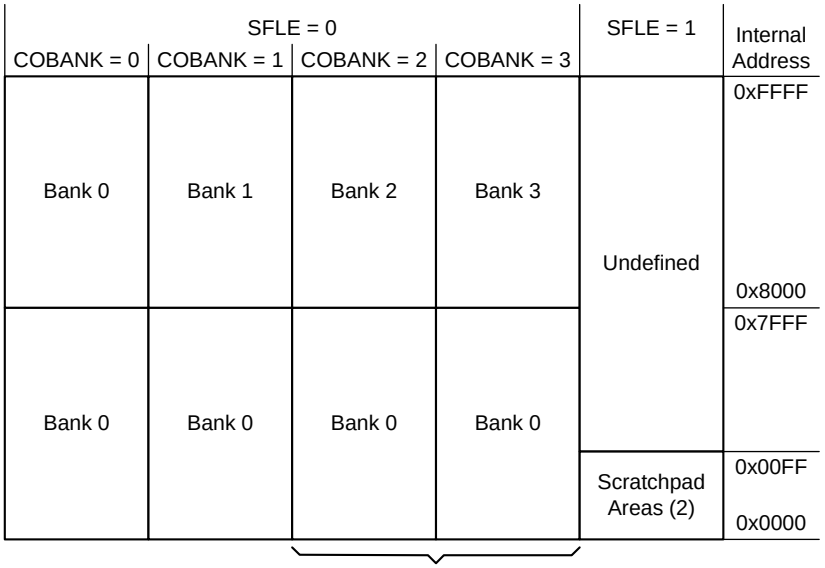



**Figure 15.1. Flash Memory Map for MOVC Read and MOVX Write Operations**

### 15.1.2. Erasing Flash Pages From Software

When erasing Flash memory, an entire page is erased (all bytes in the page are set to 0xFF). The Flash memory is organized in 1024-byte pages. The 256 bytes of Scratchpad area (addresses 0x20000 to 0x200FF) consists of two 128 byte pages. To erase any Flash page, the FLWE, PSWE, and PSEE bits must be set to '1', and a byte must be written using a MOVX instruction to any address within that page. The following is the recommended procedure for erasing a Flash page from software:

Step 1. Disable interrupts.
Step 2. If erasing a page in Bank 1, Bank 2, or Bank 3, set the COBANK bits (PSBANK.5-4) for the appropriate bank.
Step 3. If erasing a page in the Scratchpad area, set the SFLE bit (PSCTL.2).
Step 4. Set FLWE (FLSCL.0) to enable Flash writes/erases via user software.
Step 5. Set PSEE (PSCTL.1) to enable Flash erases.
Step 6. Set PSWE (PSCTL.0) to redirect MOVX commands to write to Flash.
Step 7. Use the MOVX instruction to write a data byte to any location within the page to be erased.
Step 8. Clear PSEE to disable Flash erases.
Step 9. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 10. Clear the FLWE bit, to disable Flash writes/erases.
Step 11. If erasing a page in the Scratchpad area, clear the SFLE bit.
Step 12. Re-enable interrupts.

### 15.1.3. Writing Flash Memory From Software

Bytes in Flash memory can be written one byte at a time, or in small blocks. The CHBLKW bit in register CCH0CN (SFR Definition 16.1) controls whether a single byte or a block of bytes is written to Flash during a write operation. When CHBLKW is cleared to '0', the Flash will be written one byte at a time. When CHBLKW is set to '1', the Flash will be written in blocks of four bytes for addresses in code space, or blocks of two bytes for addresses in the Scratchpad area. Block writes are performed in the same amount of time as single byte writes, which can save time when storing large amounts of data to Flash memory.

For single-byte writes to Flash, bytes are written individually, and the Flash write is performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is as follows:

Step 1. Disable interrupts.
Step 2. Clear CHBLKW (CCH0CN.0) to select single-byte write mode.
Step 3. If writing to bytes in Bank 1, Bank 2, or Bank 3, set the COBANK bits (PSBANK.5-4) for the appropriate bank.
Step 4. If writing to bytes in the Scratchpad area, set the SFLE bit (PSCTL.2).
Step 5. Set FLWE (FLSCL.0) to enable Flash writes/erases via user software.
Step 6. Set PSWE (PSCTL.0) to redirect MOVX commands to write to Flash.
Step 7. Use the MOVX instruction to write a data byte to the desired location (repeat as necessary).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Clear the FLWE bit, to disable Flash writes/erases.
Step 10. If writing to bytes in the Scratchpad area, clear the SFLE bit.
Step 11. Re-enable interrupts.

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. When writing to addresses located in any of the four code banks, a Flash write block is four bytes long, from addresses ending in 00b to addresses ending in 11b. Writes must be performed sequentially (i.e. addresses ending in 00b, 01b, 10b, and 11b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 11b. When writing to addresses located in the Flash Scratchpad area, a Flash block is two bytes long, from addresses ending in 0b to addresses ending in 1b. The Flash write will be performed following the MOVX write that targets the address ending in 1b. If any bytes in the block do not need to be updated in Flash, they should be written to 0xFF. The recommended procedure for writing Flash in blocks is as follows:

Step 1. Disable interrupts.
Step 2. Set CHBLKW (CCH0CN.0) to select block write mode.
Step 3. If writing to bytes in Bank 1, Bank 2, or Bank 3, set the COBANK bits (PSBANK.5-4) for the appropriate bank.
Step 4. If writing to bytes in the Scratchpad area, set the SFLE bit (PSCTL.2).
Step 5. Set FLWE (FLSCL.0) to enable Flash writes/erases via user software.
Step 6. Set PSWE (PSCTL.0) to redirect MOVX commands to write to Flash.
Step 7. Use the MOVX instruction to write data bytes to the desired block. The data bytes must be written sequentially, and the last byte written must be the high byte of the block (see text for details, repeat as necessary).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Clear the FLWE bit, to disable Flash writes/erases.
Step 10. If writing to bytes in the Scratchpad area, clear the SFLE bit.
Step 11. Re-enable interrupts.

## 15.2. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as prevent the viewing of proprietary program code and constants. The Program Store Write Enable (PSCTL.0), Program Store Erase Enable (PSCTL.1), and Flash Write/Erase Enable (FLACL.0) bits protect the Flash memory from accidental modification by software. These bits must be explicitly set to logic 1 before software can write or erase the Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the JTAG interface or by software running on the system controller.

A set of security lock bytes protect the Flash program memory from being read or altered across the JTAG interface. Each bit in a security lock-byte protects one 16k-byte block of memory. Clearing a bit to logic 0 in the Read Lock Byte prevents the corresponding block of Flash memory from being read across the JTAG interface. Clearing a bit in the Write/Erase Lock Byte protects the block from JTAG erasures and/or writes. The Scratchpad area is read or write/erase locked when all bits in the corresponding security byte are cleared to logic 0.

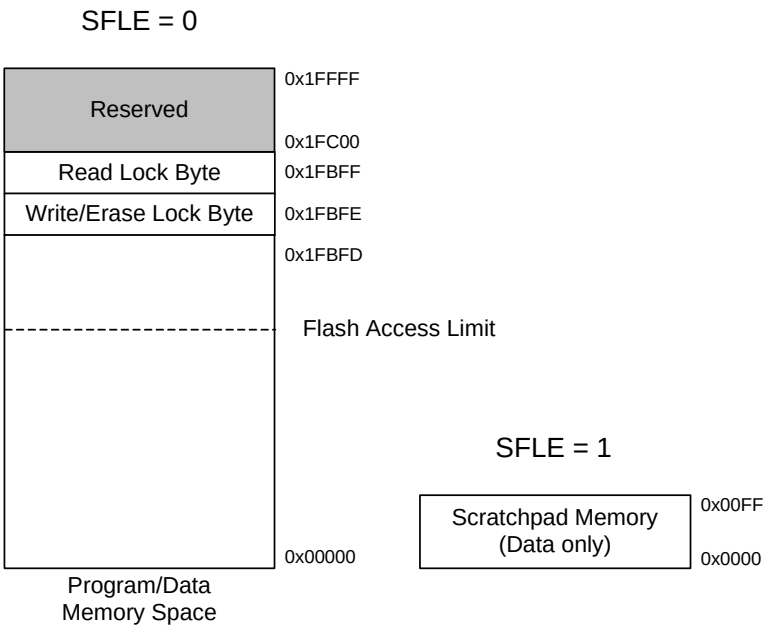
On the C8051F12x and C8051F130/1, the security lock bytes are located at 0x1FBFE (Write/Erase Lock) and 0x1FBFF (Read Lock), as shown in Figure 15.2. On the C8051F132/3, the security lock bytes are located at 0x0FFFE (Write/Erase Lock) and 0x0FFFF (Read Lock), as shown in Figure 15.3. The 1024-byte sector containing the lock bytes can be written to, but not erased, by software. An attempted read of a read-locked byte returns undefined data. Debugging code in a read-locked sector is not possible through the JTAG interface. The lock bits can always be read from and written to logic 0 regardless of the security setting applied to the block containing the security bytes. This allows additional blocks to be protected after the block containing the security bytes has been locked.

**Important Note: To ensure protection from external access, the block containing the lock bytes must be Write/Erase locked. On the 128 kB devices (C8051F12x and C8051F130/1), the block containing the security bytes is 0x18000-0x1BFFF, and is locked by clearing bit 7 of the Write/Erase Lock Byte. On the 64 kB devices (C8051F132/3), the block containing the security bytes is 0x0C000-0x0FFFF, and is locked by clearing bit 3 of the Write/Erase Lock Byte. If the page containing the security bytes is not Write/Erase locked, it is still possible to erase this page of Flash memory through the JTAG port and reset the security bytes.**

**When the page containing the security bytes has been Write/Erase locked, a JTAG full device erase must be performed to unlock any areas of Flash protected by the security bytes. A JTAG full device erase is initiated by performing a normal JTAG erase operation on either of the security byte locations. This operation must be initiated through the JTAG port, and cannot be performed from firmware running on the device.**

Read and Write/Erase Security Bits.
(Bit 7 is MSB.)

| Bit | Memory Block |
|---|---|
| 7 | 0x1C000 - 0x1FBFD |
| 6 | 0x18000 - 0x1BFFF |
| 5 | 0x14000 - 0x17FFF |
| 4 | 0x10000 - 0x13FFF |
| 3 | 0x0C000 - 0x0FFFF |
| 2 | 0x08000 - 0x0BFFF |
| 1 | 0x04000 - 0x07FFF |
| 0 | 0x00000 - 0x03FFF |




**Flash Read Lock Byte**

Bits7–0: Each bit locks a corresponding block of memory. (Bit7 is MSB).

0: Read operations are locked (disabled) for corresponding block across the JTAG interface.

1: Read operations are unlocked (enabled) for corresponding block across the JTAG interface.

**Flash Write/Erase Lock Byte**

Bits7–0: Each bit locks a corresponding block of memory.

0: Write/Erase operations are locked (disabled) for corresponding block across the JTAG interface.

1: Write/Erase operations are unlocked (enabled) for corresponding block across the JTAG interface.

NOTE: When the highest block is locked, the security bytes may be written but not erased.

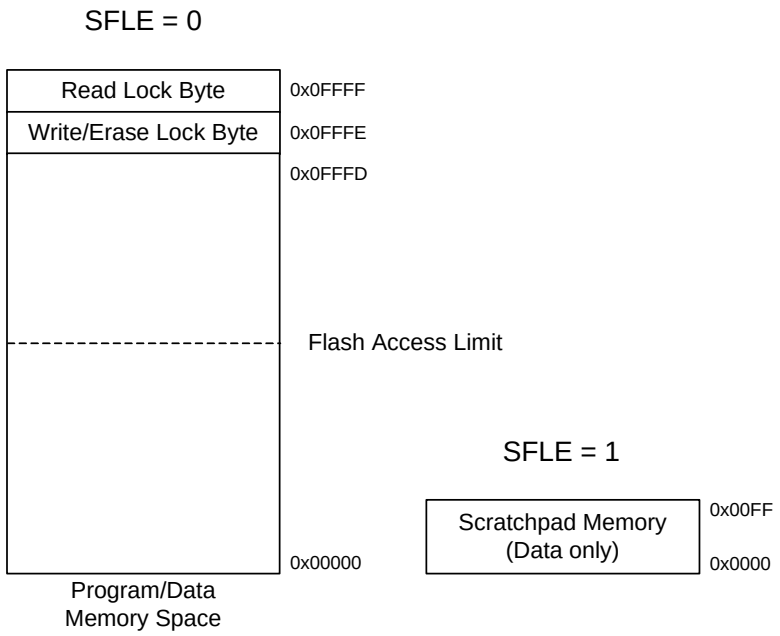
**Flash access Limit Register (FLACL)**

The Flash Access Limit is defined by the setting of the FLACL register, as described in SFR Definition 15.1. Firmware running at or above this address is prohibited from using the MOVX and MOVC instructions to read, write, or erase Flash locations below this address.

**Figure 15.2. 128 kB Flash Memory Map and Security Bytes**

Read and Write/Erase Security Bits.
(Bit 7 is MSB.)

| Bit | Memory Block |
|---|---|
| 7 | N/A |
| 6 | N/A |
| 5 | N/A |
| 4 | N/A |
| 3 | 0x0C000 - 0x0FFFF |
| 2 | 0x08000 - 0x0BFFF |
| 1 | 0x04000 - 0x07FFF |
| 0 | 0x00000 - 0x03FFF |




**Flash Read Lock Byte**

Bits7–0: Each bit locks a corresponding block of memory. (Bit7 is MSB).

0: Read operations are locked (disabled) for corresponding block across the JTAG interface.

1: Read operations are unlocked (enabled) for corresponding block across the JTAG interface.

**Flash Write/Erase Lock Byte**

Bits7–0: Each bit locks a corresponding block of memory.

0: Write/Erase operations are locked (disabled) for corresponding block across the JTAG interface.

1: Write/Erase operations are unlocked (enabled) for corresponding block across the JTAG interface.

NOTE: When the highest block is locked, the security bytes may be written but not erased.

**Flash access Limit Register (FLACL)**

The Flash Access Limit is defined by the setting of the FLACL register, as described in SFR Definition 15.1. Firmware running at or above this address is prohibited from using the MOVX and MOVC instructions to read, write, or erase Flash locations below this address.

**Figure 15.3. 64 kB Flash Memory Map and Security Bytes**

The Flash Access Limit security feature (see SFR Definition 15.1) protects proprietary program code and data from being read by software running on the device. This feature provides support for OEMs that wish to program the MCU with proprietary value-added firmware before distribution. The value-added firmware can be protected while allowing additional code to be programmed in remaining program memory space later.

The Flash Access Limit (FAL) is a 17-bit address that establishes two logical partitions in the program memory space. The first is an upper partition consisting of all the program memory locations at or above the FAL address, and the second is a lower partition consisting of all the program memory locations starting at 0x00000 up to (but excluding) the FAL address. Software in the upper partition can execute code in the lower partition, but is prohibited from reading locations in the lower partition using the MOVC instruction. (Executing a MOVC instruction from the upper partition with a source address in the lower partition will return indeterminate data.) Software running in the lower partition can access locations in both the upper and lower partition without restriction.

The Value-added firmware should be placed in the lower partition. On reset, control is passed to the value-added firmware via the reset vector. Once the value-added firmware completes its initial execution, it branches to a predetermined location in the upper partition. If entry points are published, software running in the upper partition may execute program code in the lower partition, but it cannot read or change the contents of the lower partition. Parameters may be passed to the program code running in the lower partition either through the typical method of placing them on the stack or in registers before the call or by placing them in prescribed memory locations in the upper partition.

The FAL address is specified using the contents of the Flash Access Limit Register. The 8 MSBs of the 17-bit FAL address are determined by the setting of the FLACL register. Thus, the FAL can be located on 512-byte boundaries anywhere in program memory space. However, the 1024-byte erase sector size essentially requires that a 1024 boundary be used. The contents of a non-initialized FLACL security byte are 0x00, thereby setting the FAL address to 0x00000 and allowing read access to all locations in program memory space by default.

### SFR Definition 15.1. FLACL: Flash Access Limit

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |

SFR Address: 0xB7
SFR Page: F

Bits 7–0: FLACL: Flash Access Limit.
This register holds the most significant 8 bits of the 17-bit program memory read/write/erase limit address. The lower 9 bits of the read/write/erase limit are always set to 0. A write to this register sets the Flash Access Limit. This register can only be written once after any reset. Any subsequent writes are ignored until the next reset. **To fully protect all addresses below this limit, bit 0 of FLACL should be set to '0' to align the FAL on a 1024-byte Flash page boundary.**

### 15.2.1. Summary of Flash Security Options

There are three Flash access methods supported on the C8051F12x and C8051F13x devices; 1) Accessing Flash through the JTAG debug interface, 2) Accessing Flash from firmware residing below the Flash Access Limit, and 3) Accessing Flash from firmware residing at or above the Flash Access Limit.

**Accessing Flash through the JTAG debug interface:**

1. The Read and Write/Erase Lock bytes (security bytes) provide security for Flash access through the JTAG interface.
2. Any unlocked page may be read from, written to, or erased.
3. Locked pages cannot be read from, written to, or erased.
4. Reading the security bytes is always permitted.
5. Locking additional pages by writing to the security bytes is always permitted.
6. If the page containing the security bytes is **unlocked**, it can be directly erased. **Doing so will reset the security bytes and unlock all pages of Flash.**
7. If the page containing the security bytes is **locked**, it cannot be directly erased. **To unlock the page containing the security bytes, a full JTAG device erase is required.** A full JTAG device erase will erase all Flash pages, including the page containing the security bytes and the security bytes themselves.
8. The Reserved Area cannot be read from, written to, or erased at any time.

**Accessing Flash from firmware residing below the Flash Access Limit:**

1. The Read and Write/Erase Lock bytes (security bytes) do not restrict Flash access from user firmware.
2. Any page of Flash except the page containing the security bytes may be read from, written to, or erased.
3. **The page containing the security bytes cannot be erased.** Unlocking pages of Flash can only be performed via the JTAG interface.
4. The page containing the security bytes may be read from or written to. Pages of Flash can be locked from JTAG access by writing to the security bytes.
5. The Reserved Area cannot be read from, written to, or erased at any time.

**Accessing Flash from firmware residing at or above the Flash Access Limit:**

1. The Read and Write/Erase Lock bytes (security bytes) do not restrict Flash access from user firmware.
2. Any page of Flash at or above the Flash Access Limit except the page containing the security bytes may be read from, written to, or erased.
3. Any page of Flash below the Flash Access Limit cannot be read from, written to, or erased.
4. Code branches to locations below the Flash Access Limit are permitted.
5. **The page containing the security bytes cannot be erased.** Unlocking pages of Flash can only be performed via the JTAG interface.
6. The page containing the security bytes may be read from or written to. Pages of Flash can be locked from JTAG access by writing to the security bytes.
7. The Reserved Area cannot be read from, written to, or erased at any time.

## SFR Definition 15.2. FLSCL: Flash Memory Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | FLRT | | Reserved | Reserved | Reserved | FLWE | 10000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |

SFR Address: 0xB7
SFR Page: 0

Bits 7–6: Unused.
Bits 5–4: FLRT: Flash Read Time.
These bits should be programmed to the smallest allowed value, according to the system clock speed.
00: SYSCLK ≤ 25 MHz.
01: SYSCLK ≤ 50 MHz.
10: SYSCLK ≤ 75 MHz.
11: SYSCLK ≤ 100 MHz.
Bits 3–1: RESERVED. Read = 000b. Must Write 000b.
Bit 0: FLWE: Flash Write/Erase Enable.
This bit must be set to allow Flash writes/erasures from user software.
0: Flash writes/erases disabled.
1: Flash writes/erases enabled.

**Important Note: When changing the FLRT bits to a lower setting (e.g. when changing from a value of 11b to 00b), cache reads, cache writes, and the prefetch engine should be disabled using the CCH0CN register (see SFR Definition 16.1).**

## SFR Definition 15.3. PSCTL: Program Store Read/Write Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | SFLE | PSEE | PSWE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |

SFR Address: 0x8F
SFR Page: 0

Bits 7–3: UNUSED. Read = 00000b, Write = don't care.

Bit 2: SFLE: Scratchpad Flash Memory Access Enable
When this bit is set, Flash MOVC reads and writes from user software are directed to the two 128-byte Scratchpad Flash sectors. When SFLE is set to logic 1, Flash accesses out of the address range 0x00-0xFF should not be attempted (with the exception of address 0x400, which can be used to simultaneously erase both Scratchpad areas). Reads/Writes out of this range will yield undefined results.
0: Flash access from user software directed to the Program/Data Flash sector.
1: Flash access from user software directed to the two 128 byte Scratchpad sectors.

Bit 1: PSEE: Program Store Erase Enable.
Setting this bit allows an entire page of the Flash program memory to be erased provided the PSWE bit is also set. After setting this bit, 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. **Note: The Flash page containing the Read Lock Byte and Write/Erase Lock Byte cannot be erased by software.**
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit 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 location must be erased prior to writing data.
0: Write to Flash program memory disabled. MOVX write operations target External RAM.
1: Write to Flash program memory enabled. MOVX write operations target Flash memory.

## NOTES:

# 16. Branch Target Cache

The C8051F12x and C8051F13x device families incorporate a 63x4 byte branch target cache with a 4-byte prefetch engine. Because the access time of the Flash memory is 40 Flashns, and the minimum instruction time is 10ns (C8051F120/1/2/3 and C8051F130/1/2/3) or 20 ns (C8051F124/5/6/7), the branch target cache and prefetch engine are necessary for full-speed code execution. Instructions are read from Flash memory four bytes at a time by the prefetch engine, and given to the CIP-51 processor core to execute. When running linear code (code without any jumps or branches), the prefetch engine alone allows instructions to be executed at full speed. When a code branch occurs, a search is performed for the branch target (destination address) in the cache. If the branch target information is found in the cache (called a "cache hit"), the instruction data is read from the cache and immediately returned to the CIP-51 with no delay in code execution. If the branch target is not found in the cache (called a "cache miss"), the processor may be stalled for up to four clock cycles while the next set of four instructions is retrieved from Flash memory. Each time a cache miss occurs, the requested instruction data is written to the cache if allowed by the current cache settings. A data flow diagram of the interaction between the CIP-51 and the Branch Target Cache and Prefetch Engine is shown in Figure 16.1.

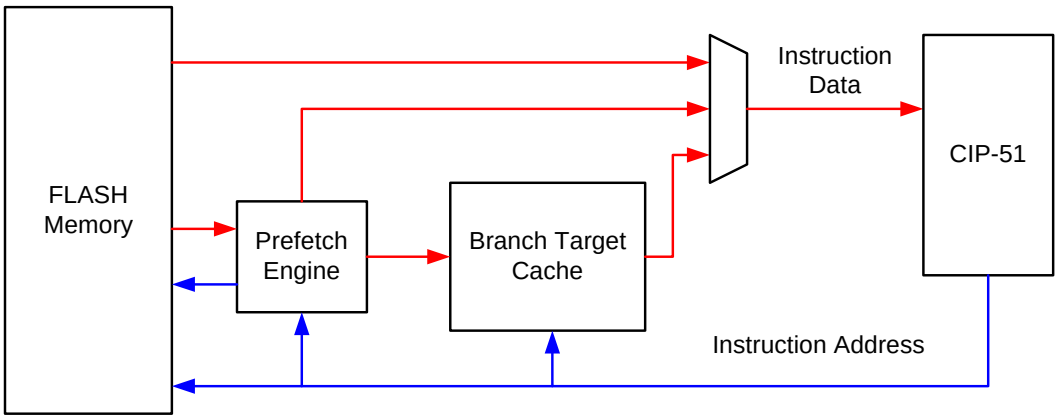



**Figure 16.1. Branch Target Cache Data Flow**

## 16.1. Cache and Prefetch Operation

The branch target cache maintains two sets of memory locations: "slots" and "tags". A slot is where the cached instruction data from Flash is stored. Each slot holds four consecutive code bytes. A tag contains the 15 most significant bits of the corresponding Flash address for each four-byte slot. Thus, instruction data is always cached along four-byte boundaries in code space. A tag also contains a "valid bit", which indicates whether a cache location contains valid instruction data. A special cache location (called the linear tag and slot), is reserved for use by the prefetch engine. The cache organization is shown in Figure 16.2. Each time a Flash read is requested, the address is compared with all valid cache tag locations (including the linear tag). If any of the tag locations match the requested address, the data from that slot is immediately provided to the CIP-51. If the requested address matches a location that is currently being read by the prefetch engine, the CIP-51 will be stalled until the read is complete. If a match is not found, the current prefetch operation is abandoned, and a new prefetch operation is initiated for the requested instruction data. When the prefetch operation is finished, the CIP-51 begins executing the instructions that were retrieved, and the prefetch engine begins reading the next four-byte word from Flash memory. If the newly-fetched data also meets the criteria necessary to be cached, it will be written to the cache in the slot indicated by the current replacement algorithm.

The replacement algorithm is selected with the Cache Algorithm bit, CHALGM (CCH0TN.3). When CHALGM is cleared to '0', the cache will use the rebound algorithm to replace cache locations. The rebound algorithm replaces locations in order from the beginning of cache memory to the end, and then from the end of cache memory to the beginning. When CHALGM is set to '1', the cache will use the pseudo-random algorithm to replace cache locations. The pseudo-random algorithm uses a pseudo-random number to determine which cache location to replace. The cache can be manually emptied by writing a '1' to the CHFLUSH bit (CCH0CN.4).

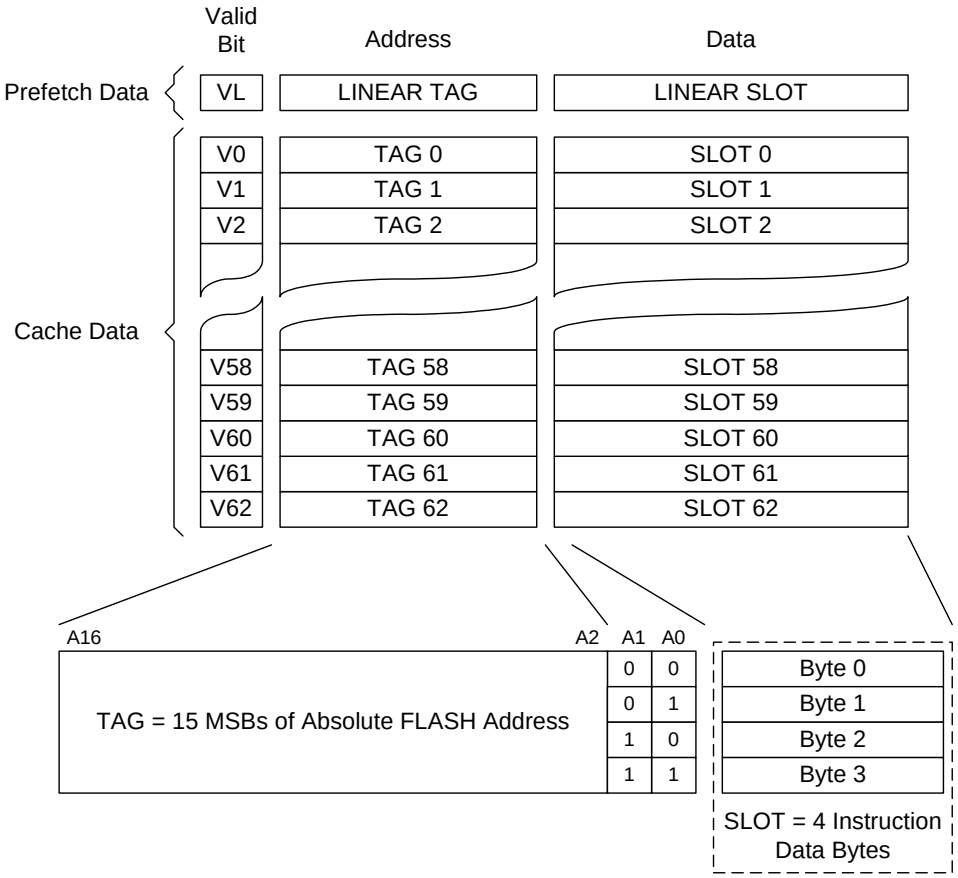



**Figure 16.2. Branch Target Cache Organiztion**

## 16.2. Cache and Prefetch Optimization

By default, the branch target cache is configured to provide code speed improvements for a broad range of circumstances. **In most applications, the cache control registers should be left in their reset states.** Sometimes it is desirable to optimize the execution time of a specific routine or critical timing loop. The branch target cache includes options to exclude caching of certain types of data, as well as the ability to pre-load and lock time-critical branch locations to optimize execution speed.

The most basic level of cache control is implemented with the Cache Miss Penalty Threshold bits, CHMSTH (CCH0TN.1-0). If the processor is stalled during a prefetch operation for more clock cycles than the number stored in CHMSTH, the requested data will be cached when it becomes available. The CHMSTH bits are set to zero by default, meaning that any time the processor is stalled, the new data will be cached. If, for example, CHMSTH is equal to 2, any cache miss causing a delay of 3 or 4 clock cycles will be cached, while a cache miss causing a delay of 1-2 clock cycles will not be cached.

Certain types of instruction data or certain blocks of code can also be excluded from caching. The destinations of RETI instructions are, by default, excluded from caching. To enable caching of RETI destinations, the CHRETI bit (CCH0CN.3) can be set to '1'. It is generally not beneficial to cache RETI destinations unless the same instruction is likely to be interrupted repeatedly (such as a code loop that is waiting for an interrupt to happen). Instructions that are part of an interrupt service routine (ISR) can also be excluded from caching. By default, ISR instructions are cached, but this can be disabled by clearing the CHISR bit (CCH0CN.2) to '0'. The other information that can be explicitly excluded from caching are the data returned by MOVC instructions. Clearing the CHMOV bit (CCH0CN.1) to '0' will disable caching of MOVC data. If MOVC caching is allowed, it can be restricted to only use slot 0 for the MOVC information (excluding cache push operations). The CHFIXM bit (CCH0TN.2) controls this behavior.

Further cache control can be implemented by disabling all cache writes. Cache writes can be disabled by clearing the CHWREN bit (CCH0CN.7) to '0'. Although normal cache writes (such as those after a cache miss) are disabled, data can still be written to the cache with a cache push operation. Disabling cache writes can be used to prevent a non-critical section of code from changing the cache contents. Note that regardless of the value of CHWREN, a Flash write or erase operation automatically removes the affected bytes from the cache. Cache reads and the prefetch engine can also be individually disabled. Disabling cache reads forces all instructions data to execute from Flash memory or from the prefetch engine. To disable cache reads, the CHRDEN bit (CCH0CN.6) can be cleared to '0'. Note that when cache reads are disabled, cache writes will still occur (if CHWREN is set to '1'). Disabling the prefetch engine is accomplished using the CHPFEN bit (CCH0CN.5). When this bit is cleared to '0', the prefetch engine will be disabled. If both CHPFEN and CHRDEN are '0', code will execute at a fixed rate, as instructions become available from the Flash memory.

Cache locations can also be pre-loaded and locked with time-critical branch destinations. For example, in a system with an ISR that must respond as fast as possible, the entry point for the ISR can be locked into a cache location to minimize the response latency of the ISR. Up to 61 locations can be locked into the cache at one time. Instructions are locked into cache by enabling cache push operations with the CHPUSH bit (CCH0LC.7). When CHPUSH is set to '1', a MOVC instruction will cause the four-byte segment containing the data byte to be written to the cache slot location indicated by CHSLOT (CCH0LC.5-0). CHSLOT is them decremented to point to the next lockable cache location. This process is called a cache push operation. Cache locations that are above CHSLOT are "locked", and cannot be changed by the processor core, as shown in Figure 16.3. Cache locations can be unlocked by using a cache pop operation. A cache pop is performed by writing a '1' to the CHPOP bit (CCH0LC.6). When a cache pop is initiated, the value of CHSLOT is incremented. This unlocks the most recently locked cache location, but does not remove the information from the cache. Note that a cache pop should not be initiated if CHSLOT is equal to 111110b. Doing so may have an adverse effect on cache performance. **Important: Although locking cache location 1 is not explicitly disabled by hardware, the entire cache will be unlocked when CHSLOT is equal to 000000b. Therefore, cache locations 1 and 0 must remain unlocked at all times.**

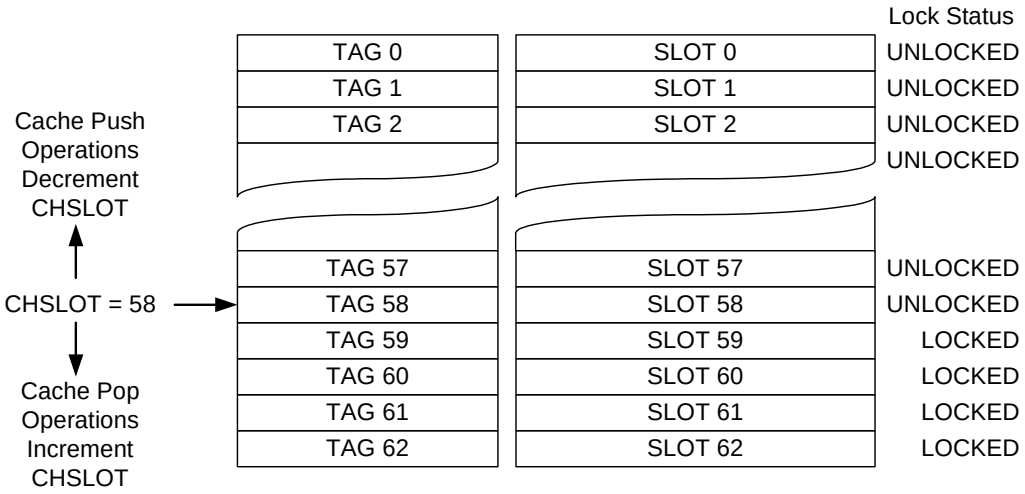



**Figure 16.3. Cache Lock Operation**

## SFR Definition 16.1. CCH0CN: Cache Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CHWREN | CHRDEN | CHPFEN | CHFLSH | CHRETI | CHISR | CHMOVC | CHBLKW | 11100110 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA1
SFR Page: F

Bit 7: CHWREN: Cache Write Enable.
This bit enables the processor to write to the cache memory.
0: Cache contents are not allowed to change, except during Flash writes/erasures or cache locks.
1: Writes to cache memory are allowed.

Bit 6: CHRDEN: Cache Read Enable.
This bit enables the processor to read instructions from the cache memory.
0: All instruction data comes from Flash memory or the prefetch engine.
1: Instruction data is obtained from cache (when available).

Bit 5: CHPFEN: Cache Prefetch Enable.
This bit enables the prefetch engine.
0: Prefetch engine is disabled.
1: Prefetch engine is enabled.

Bit 4: CHFLSH: Cache Flush.
When written to a ‘1’, this bit clears the cache contents. This bit always reads ‘0’.

Bit 3: CHRETI: Cache RETI Destination Enable.
This bit enables the destination of a RETI address to be cached.
0: Destinations of RETI instructions will not be cached.
1: RETI destinations will be cached.

Bit 2: CHISR: Cache ISR Enable.
This bit allows instructions which are part of an Interrupt Service Rountine (ISR) to be cached.
0: Instructions in ISRs will not be loaded into cache memory.
1: Instructions in ISRs can be cached.

Bit 1: CHMOVC: Cache MOVC Enable.
This bit allows data requested by a MOVC instruction to be loaded into the cache memory.
0: Data requested by MOVC instructions will not be cached.
1: Data requested by MOVC instructions will be loaded into cache memory.

Bit 0: CHBLKW: Block Write Enable.
This bit allows block writes to Flash memory from software.
0: Each byte of a software Flash write is written individually.
1: Flash bytes are written in groups of four (for code space writes) or two (for scratchpad writes).

## SFR Definition 16.2. CCH0TN: Cache Tuning

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CHMSCTL | | | | CHALGM | CHFIXM | CHMSTH | | 00000100 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA2
SFR Page: F

Bits 7–4: CHMSCTL: Cache Miss Penalty Accumulator (Bits 4–1).
These are bits 4-1 of the Cache Miss Penalty Accumulator. To read these bits, they must first be latched by reading the CHMSCTH bits in the CCH0MA Register (See SFR Definition 16.4).

Bit 3: CHALGM: Cache Algorithm Select.
This bit selects the cache replacement algorithm.
0: Cache uses Rebound algorithm.
1: Cache uses Pseudo-random algorithm.

Bit 2: CHFIXM: Cache Fix MOVC Enable.
This bit forces MOVC writes to the cache memory to use slot 0.
0: MOVC data is written according to the current algorithm selected by the CHALGM bit.
1: MOVC data is always written to cache slot 0.

Bits 1–0: CHMSTH: Cache Miss Penalty Threshold.
These bits determine when missed instruction data will be cached.
If data takes longer than CHMSTH clocks to obtain, it will be cached.

## SFR Definition 16.3. CCH0LC: Cache Lock Control

| R/W | R/W | R | R | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CHPUSH | CHPOP | CHSLOT | | | | | | 00111110 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA3
SFR Page: F

Bit 7: CHPUSH: Cache Push Enable.
This bit enables cache push operations, which will lock information in cache slots using MOVC instructions.
0: Cache push operations are disabled.
1: Cache push operations are enabled. When a MOVC read is executed, the requested 4-byte segment containing the data is locked into the cache at the location indicated by CHSLOT, and CHSLOT is decremented.
Note that no more than 61 cache slots should be locked at one time, since the entire cache will be unlocked when CHSLOT is equal to 0.

Bit 6: CHPOP: Cache Pop.
Writing a ‘1’ to this bit will increment CHSLOT and then unlock that location. This bit always reads ‘0’. Note that Cache Pop operations should not be performed while CHSLOT = 111110b. “Pop”ing more Cache slots than have been “Push”ed will have indeterminate results on the Cache performance.

Bits 5–0: CHSLOT: Cache Slot Pointer.
These read-only bits are the pointer into the cache lock stack. Locations above CHSLOT are locked, and will not be changed by the processor, except when CHSLOT equals 0.

## SFR Definition 16.4. CCH0MA: Cache Miss Accumulator

| R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CHMSOV | CHMSCTH | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9A
SFR Page: F

Bit 7: CHMSOV: Cache Miss Penalty Overflow.
This bit indicates when the Cache Miss Penalty Accumulator has overflowed since it was last written.
0: The Cache Miss Penalty Accumulator has not overflowed since it was last written.
1: An overflow of the Cache Miss Penalty Accumulator has occurred since it was last written.

Bits 6–0: CHMSCTH: Cache Miss Penalty Accumulator (bits 11–5)
These are bits 11-5 of the Cache Miss Penalty Accumulator. The next four bits (bits 4-1) are stored in CHMSCTL in the CCH0TN register.
The Cache Miss Penalty Accumulator is incremented every clock cycle that the processor is delayed due to a cache miss. This is primarily used as a diagnostic feature, when optimizing code for execution speed.
Writing to CHMSCTH clears the lower 5 bits of the Cache Miss Penalty Accumulator.
Reading from CHMSCTH returns the current value of CHMSTCH, and latches bits 4-1 into CHMSTCL so that they can be read. Because bit 0 of the Cache Miss Penalty Accumulator is not available, the Cumulative Miss Penalty is equal to 2 * (CCHMSTCH:CCHMSTCL).

## SFR Definition 16.5. FLSTAT: Flash Status

| R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | FLBUSY | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x88
SFR Page: F

Bit 7–1: Reserved.

Bit 0: FLBUSY: Flash Busy
This bit indicates when a Flash write or erase operation is in progress.
0: Flash is idle or reading.
1: Flash write/erase operation is currently in progress.

## NOTES:

# 17. External Data Memory Interface and On-Chip XRAM

There are 8 kB of on-chip RAM mapped into the external data memory space (XRAM), as well as an External Data Memory Interface which can be used to access off-chip 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 17.1). Note: the MOVX instruction can also be used for writing to the Flash memory. See **Section “15. Flash Memory” on page 199** for details. The MOVX instruction accesses XRAM by default. The EMIF can be configured to appear on the lower GPIO Ports (P0–P3) or the upper GPIO Ports (P4–P7).

## 17.1. 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.

### 17.1.1. 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.

### 17.1.2. 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
```

## 17.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Select EMIF on Low Ports (P3, P2, P1, and P0) or High Ports (P7, P6, P5, and P4).
2. Configure the Output Modes of the port pins as either push-pull or open-drain (push-pull is most common).
3. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic ‘1’).
4. Select Multiplexed mode or Non-multiplexed mode.

5. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
6. 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 17.2.

## 17.3. Port Selection and Configuration

The External Memory Interface can appear on Ports 3, 2, 1, and 0 (All Devices) or on Ports 7, 6, 5, and 4 (100-pin TQFP devices only), depending on the state of the PRTSEL bit (EMI0CF.5). If the lower Ports are selected, the EMIFLE bit (XBR2.1) must be set to a '1' so that the Crossbar will skip over P0.7 (/WR), P0.6 (/RD), and if multiplexed mode is selected P0.5 (ALE). For more information about the configuring the Crossbar, see **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238**.

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 (on Ports 3, 2, 1, and 0). See **Section "18. Port Input/ Output" on page 235** 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. See"Configuring the Output Modes of the Port Pins" on page 239.

### SFR Definition 17.1. EMI0CN: External Memory Interface Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PGSEL7 | PGSEL6 | PGSEL5 | PGSEL4 | PGSEL3 | PGSEL2 | PGSEL1 | PGSEL0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA2
SFR Page: 0

Bits7–0: 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 17.2. EMI0CF: External Memory Configuration

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | PRTSEL | EMD2 | EMD1 | EMD0 | EALE1 | EALE0 | 00000011 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA3
SFR Page: 0

Bits7–6: Unused. Read = 00b. Write = don't care.

Bit5: PRTSEL: EMIF Port Select.
0: EMIF active on P0–P3.
1: EMIF active on P4–P7.

Bit4: EMD2: EMIF Multiplex Mode Select.
0: EMIF operates in multiplexed address/data mode.
1: EMIF operates in non-multiplexed mode (separate address and data pins).

Bits3–2: EMD1-0: EMIF Operating Mode Select.
These bits control the operating mode of the External Memory Interface.
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 k boundary are directed on-chip. Accesses above the 8 k boundary are directed off-chip. 8-bit off-chip MOVX operations use the current contents of the Address High port latches to resolve upper address byte. Note that in order 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 k boundary are directed on-chip. Accesses above the 8k boundary are directed off-chip. 8-bit off-chip MOVX operations use 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.

Bits1–0: EALE1–0: ALE Pulse-Width Select Bits (only has 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.

## 17.4. 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.

### 17.4.1. 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 17.1.

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 /RD or /WR is asserted.

See **Section "17.6.2. Multiplexed Mode" on page 230** for more information.

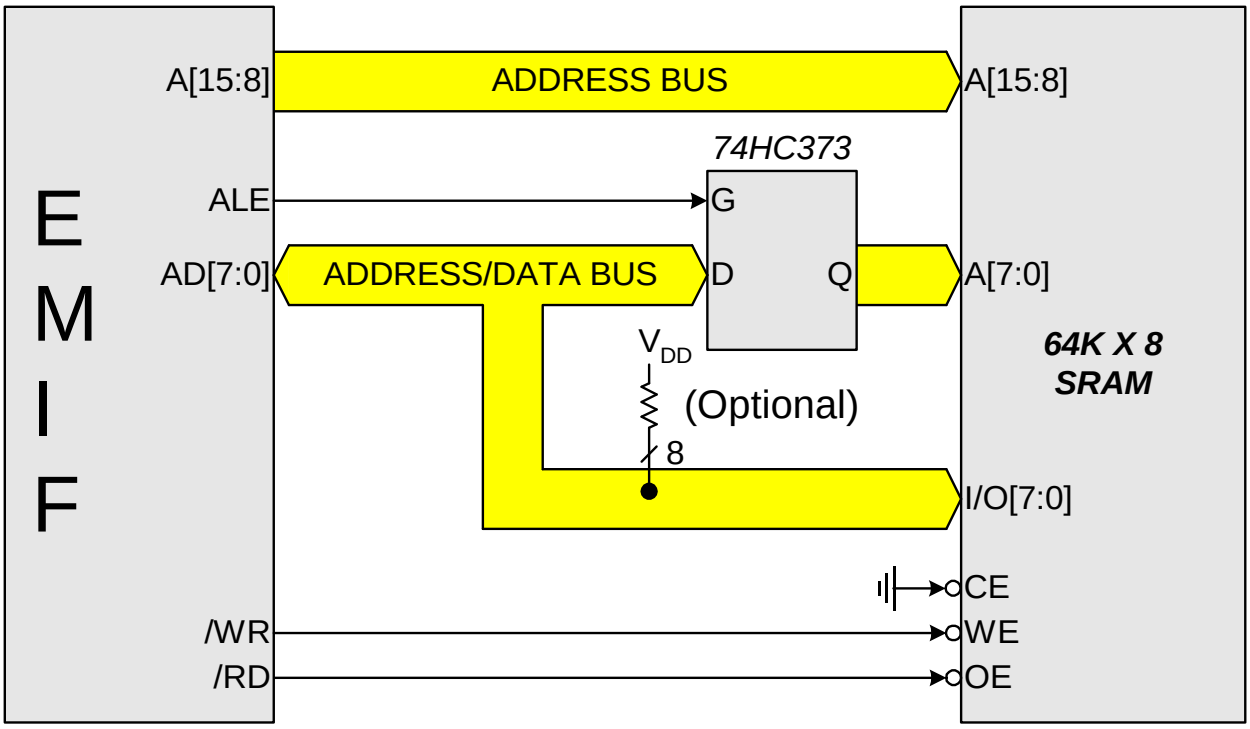



**Figure 17.1. Multiplexed Configuration Example**

### 17.4.2. 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 17.2. See **Section "17.6.1. Non-multiplexed Mode" on page 227** for more information about Non-multiplexed operation.

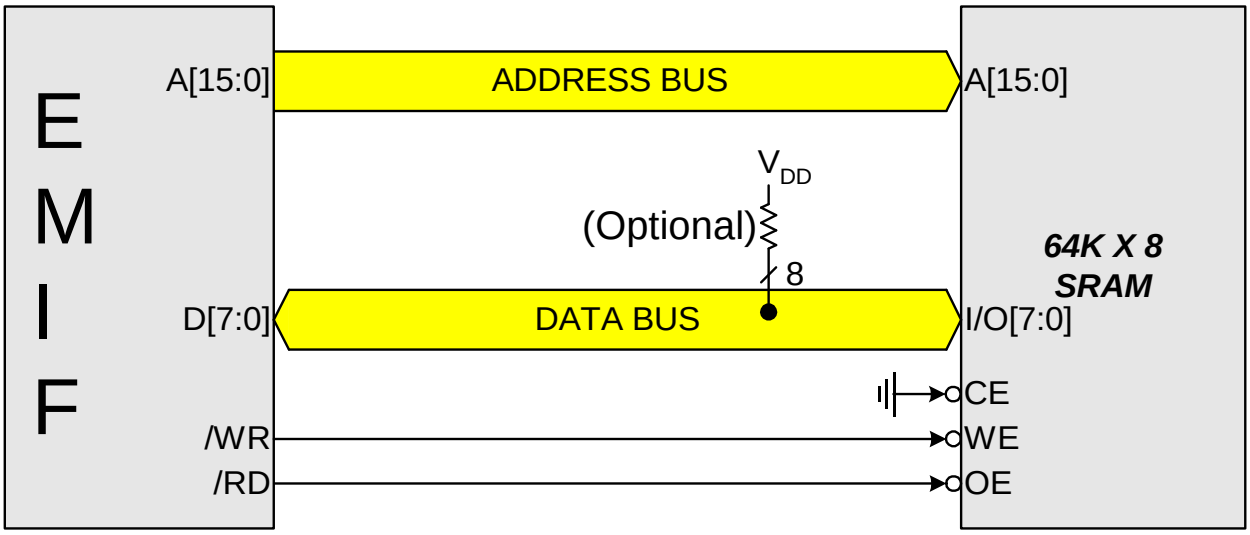



**Figure 17.2. Non-multiplexed Configuration Example**

## 17.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 17.3, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 17.2). These modes are summarized below. More information about the different modes can be found in **Section "SFR Definition 17.3. EMI0TC: External Memory Timing Control" on page 226**.

### 17.5.1. Internal XRAM Only

When 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 k 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.

### 17.5.2. Split Mode without Bank Select

When 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 8 k boundary will access on-chip XRAM space.
- Effective addresses above the 8 k 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.

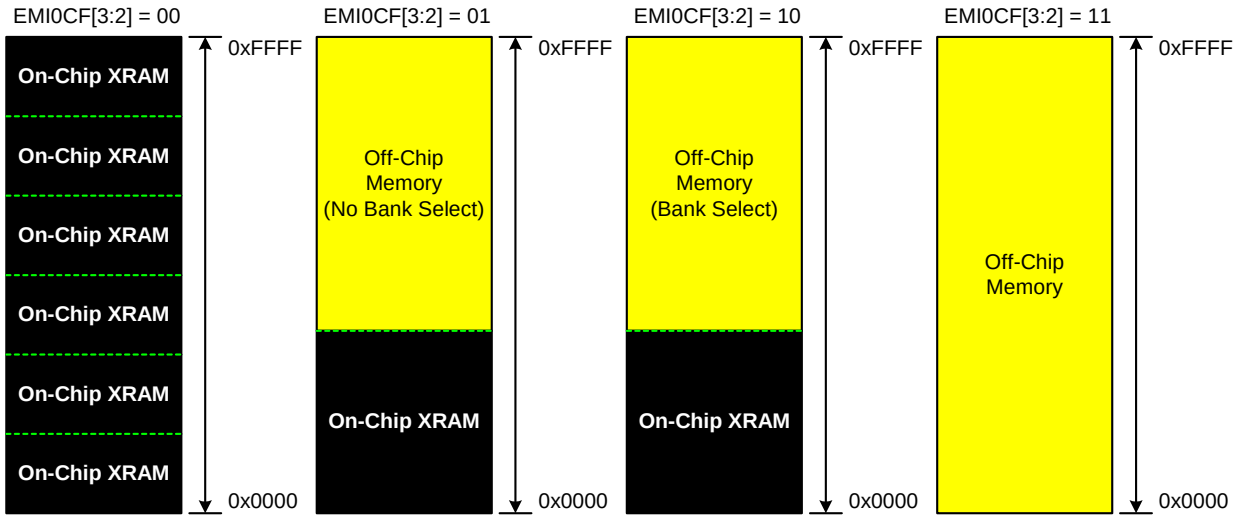



**Figure 17.3. EMIF Operating Modes**

### 17.5.3. 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 8k boundary will access on-chip XRAM space.
- Effective addresses above the 8k 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.

### 17.5.4. 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 8k 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.

## 17.6. EMIF 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, /RD and /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 17.3, 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 /RD or /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 /RD or /WR + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 17.1 lists the ac parameters for the External Memory Interface, and Figure 17.4 through Figure 17.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

## SFR Definition 17.3. EMI0TC: External Memory Timing Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| EAS1 | EAS0 | ERW3 | EWR2 | EWR1 | EWR0 | EAH1 | EAH0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA1
SFR Page: 0

Bits7–6: EAS1–0: 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.

Bits5–2: EWR3–0: EMIF /WR and /RD Pulse-Width Control Bits.
0000: /WR and /RD pulse width = 1 SYSCLK cycle.
0001: /WR and /RD pulse width = 2 SYSCLK cycles.
0010: /WR and /RD pulse width = 3 SYSCLK cycles.
0011: /WR and /RD pulse width = 4 SYSCLK cycles.
0100: /WR and /RD pulse width = 5 SYSCLK cycles.
0101: /WR and /RD pulse width = 6 SYSCLK cycles.
0110: /WR and /RD pulse width = 7 SYSCLK cycles.
0111: /WR and /RD pulse width = 8 SYSCLK cycles.
1000: /WR and /RD pulse width = 9 SYSCLK cycles.
1001: /WR and /RD pulse width = 10 SYSCLK cycles.
1010: /WR and /RD pulse width = 11 SYSCLK cycles.
1011: /WR and /RD pulse width = 12 SYSCLK cycles.
1100: /WR and /RD pulse width = 13 SYSCLK cycles.
1101: /WR and /RD pulse width = 14 SYSCLK cycles.
1110: /WR and /RD pulse width = 15 SYSCLK cycles.
1111: /WR and /RD pulse width = 16 SYSCLK cycles.

Bits1–0: EAH1–0: 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.

### 17.6.1. Non-multiplexed Mode

#### 17.6.1.1.16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’

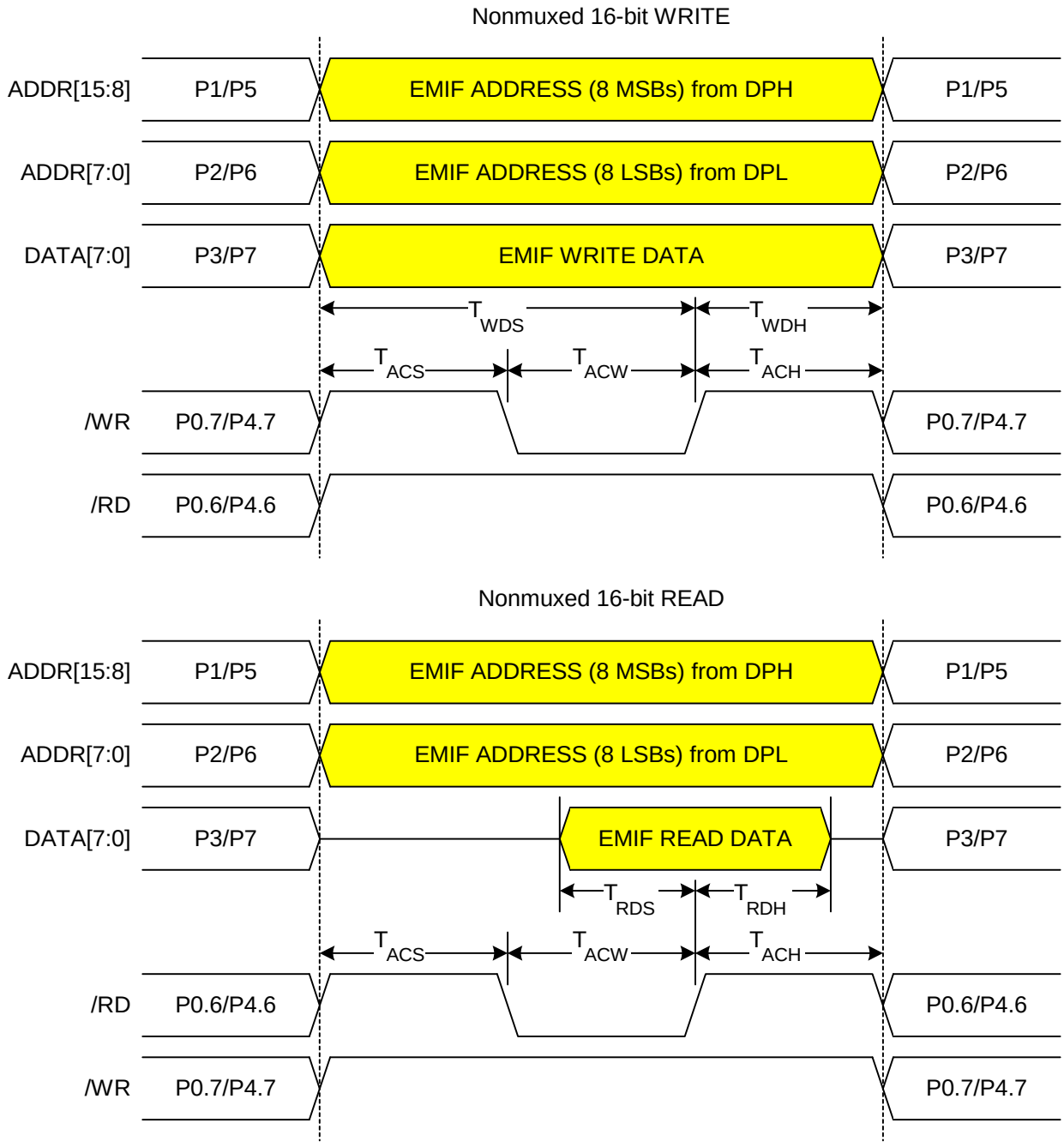



**Figure 17.4. Non-multiplexed 16-bit MOVX Timing**

**17.6.1.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘101’ or ‘111’.**

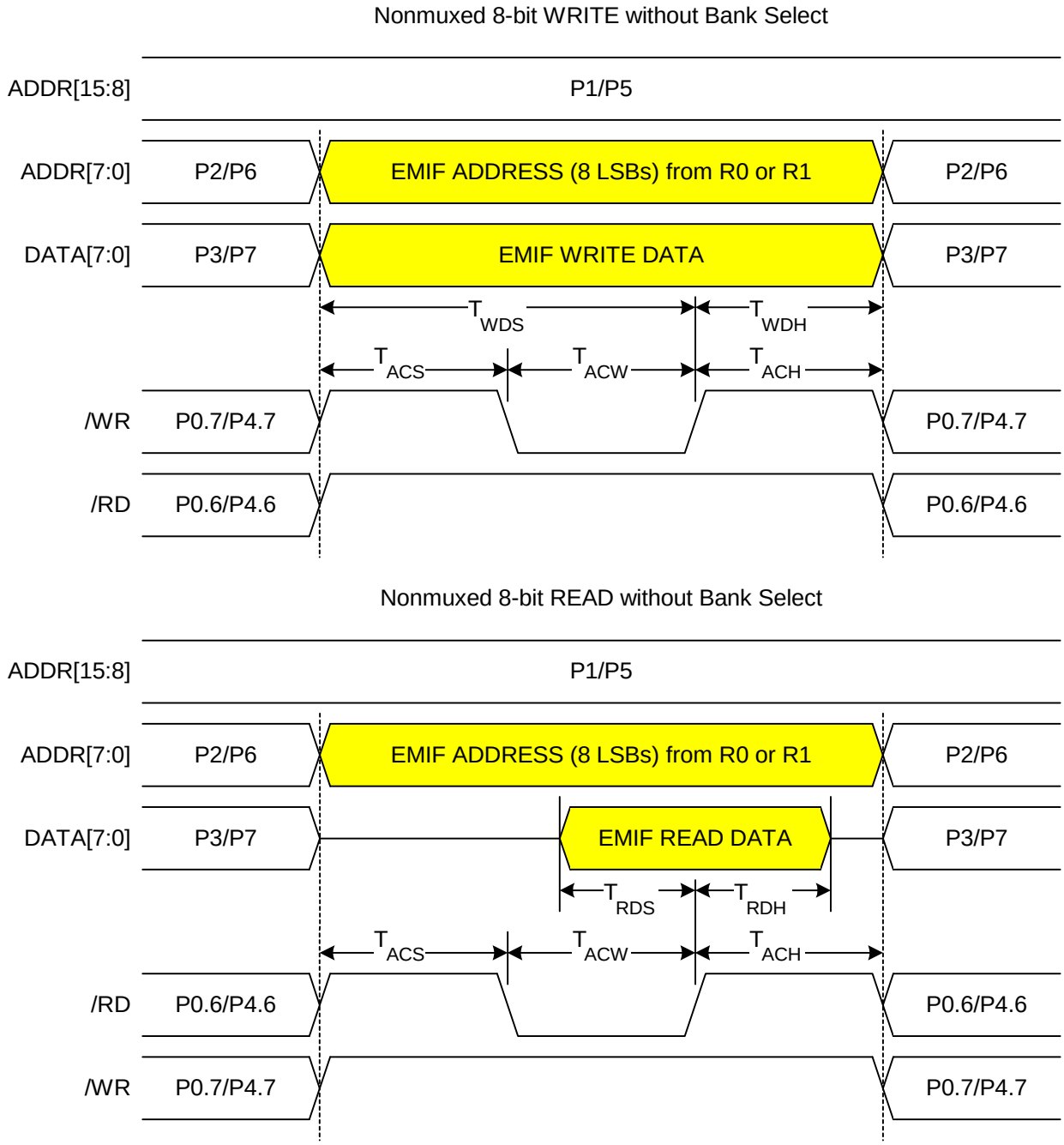



**Figure 17.5. Non-multiplexed 8-bit MOVX without Bank Select Timing**

**17.6.1.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = '110'.**

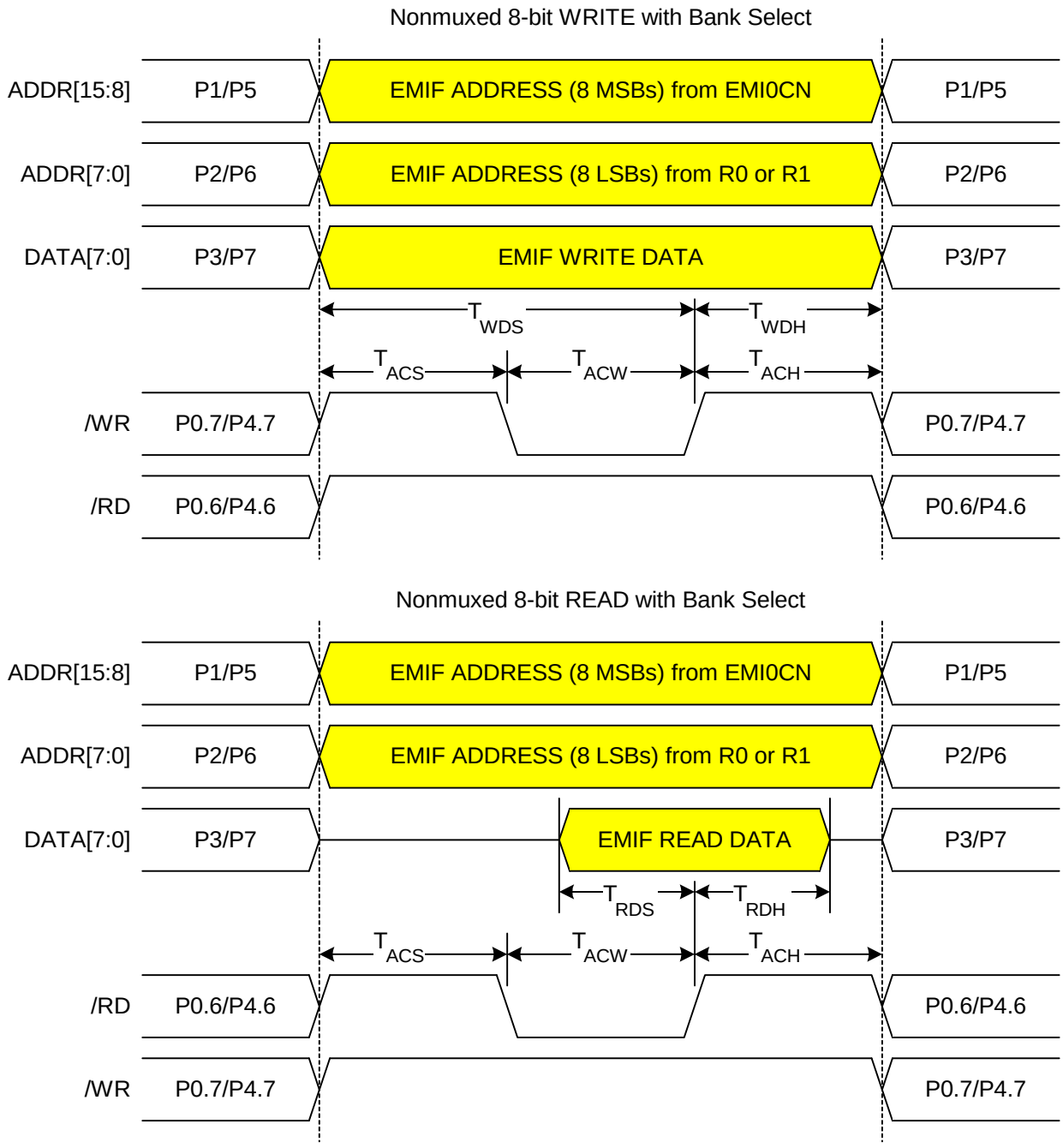



**Figure 17.6. Non-multiplexed 8-bit MOVX with Bank Select Timing**

### 17.6.2. Multiplexed Mode

#### 17.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’

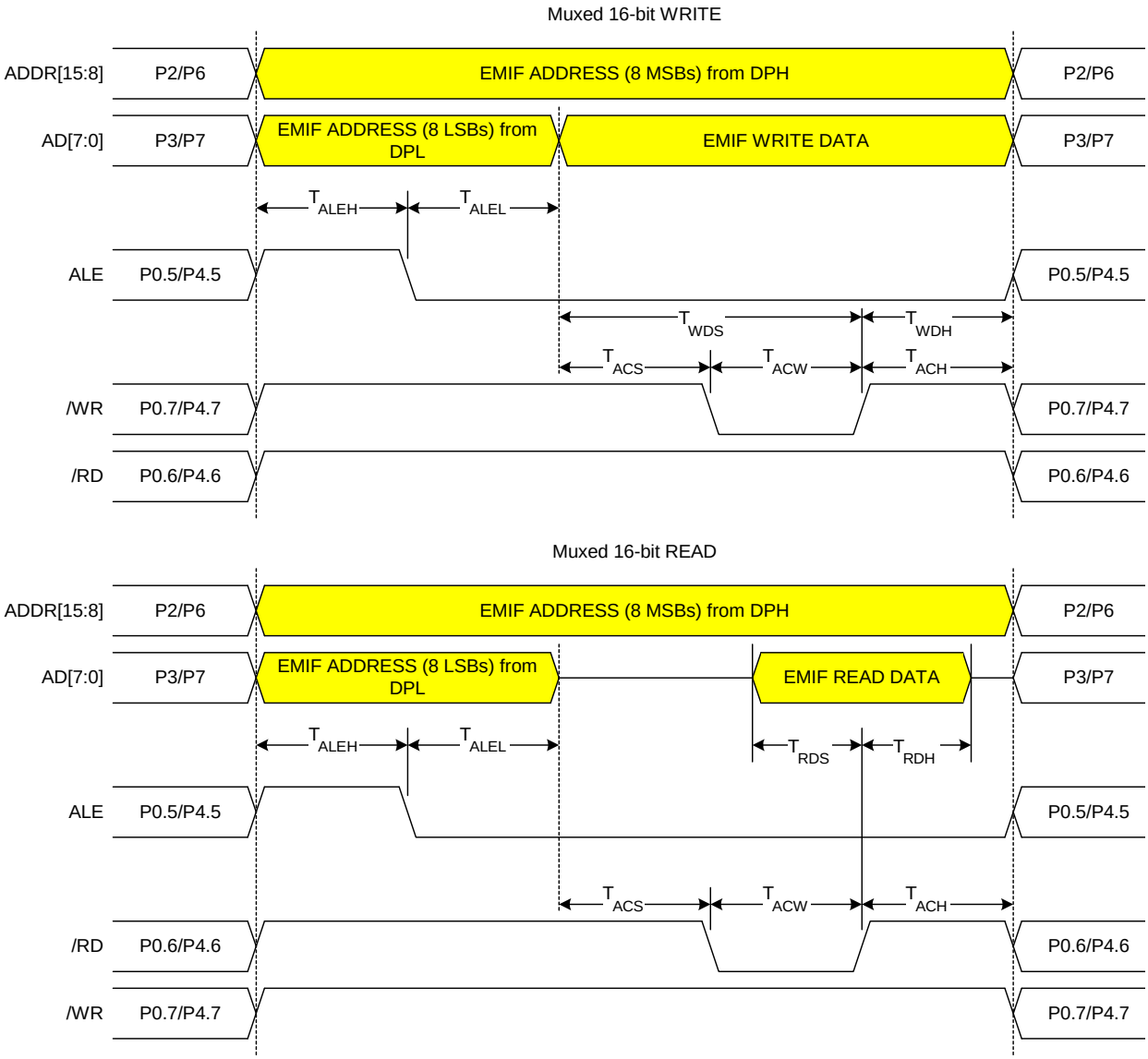



**Figure 17.7. Multiplexed 16-bit MOVX Timing**

### 17.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

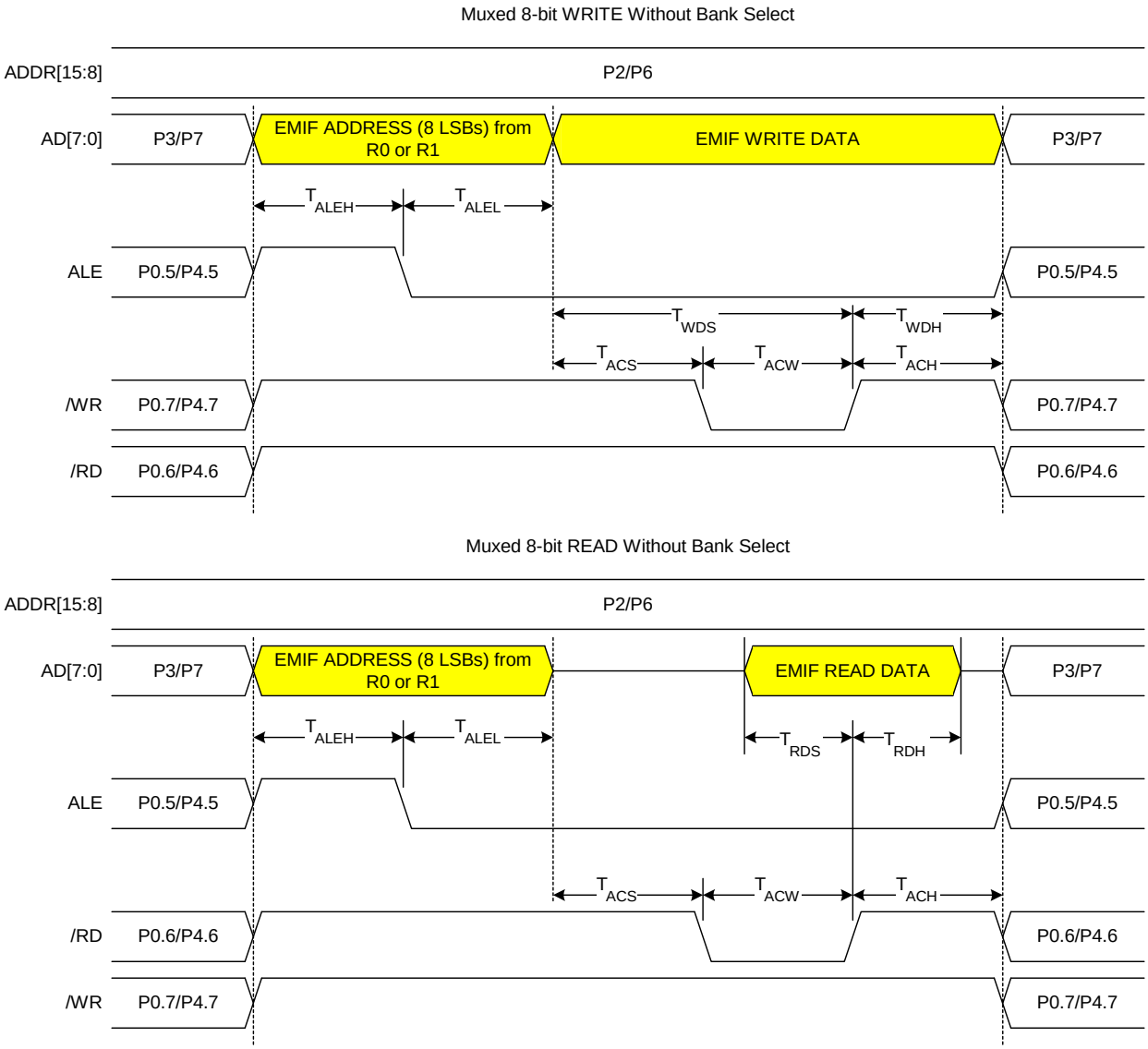



**Figure 17.8. Multiplexed 8-bit MOVX without Bank Select Timing**

### 17.6.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘010’.

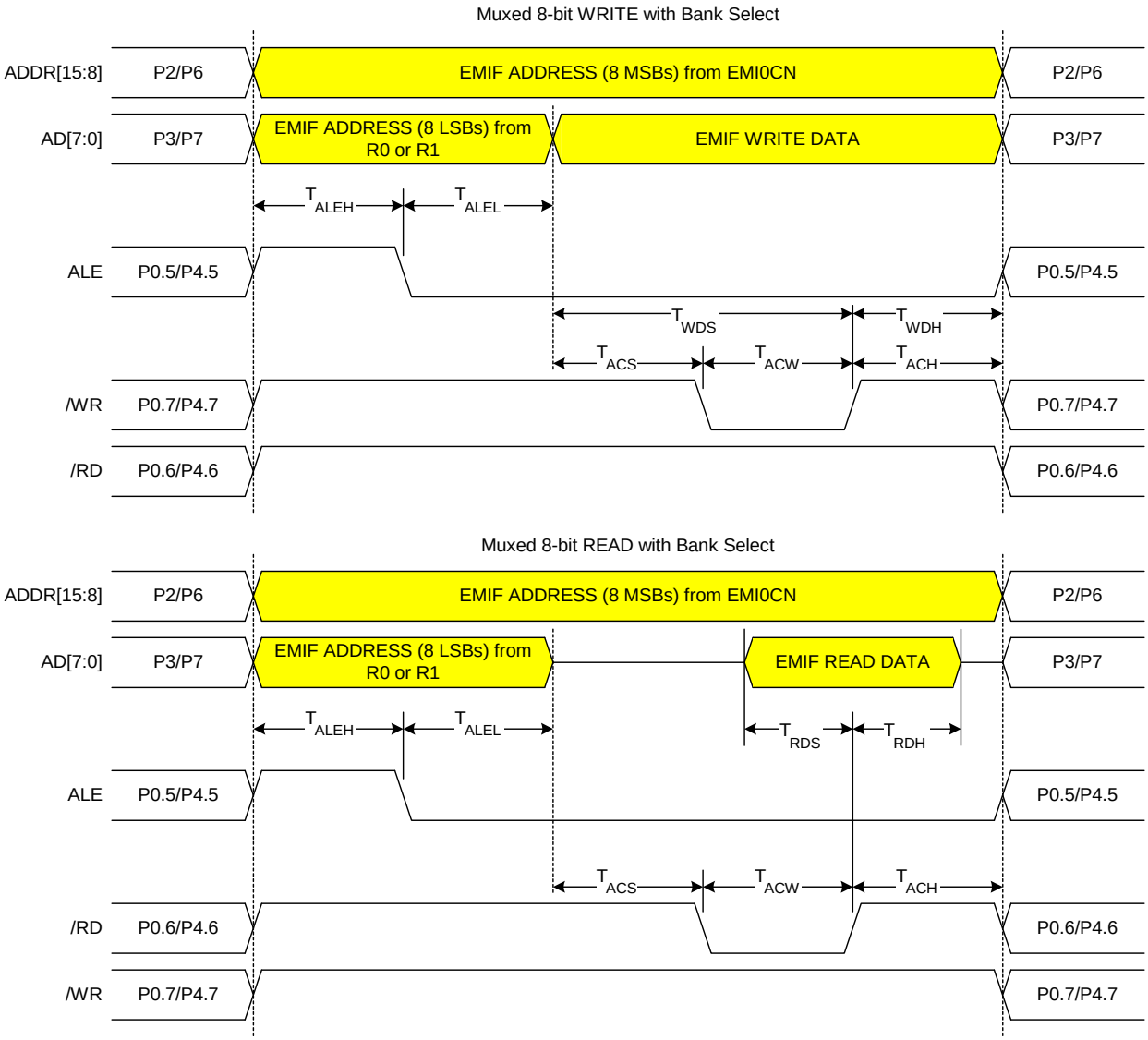



**Figure 17.9. Multiplexed 8-bit MOVX with Bank Select Timing**

**Table 17.1. 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).

## NOTES:

## 18. Port Input/Output

The devices are fully integrated mixed-signal System on a Chip MCUs with 64 digital I/O pins (100-pin TQFP packaging) or 32 digital I/O pins (64-pin TQFP packaging), organized as 8-bit Ports. All ports are both bit- and byte-addressable through their corresponding Port Data registers. All Port pins are 5 V-tolerant, and all support configurable Open-Drain or Push-Pull output modes and weak pullups. A block diagram of the Port I/O cell is shown in Figure 18.1. Complete Electrical Specifications for the Port I/O pins are given in Table 18.1.

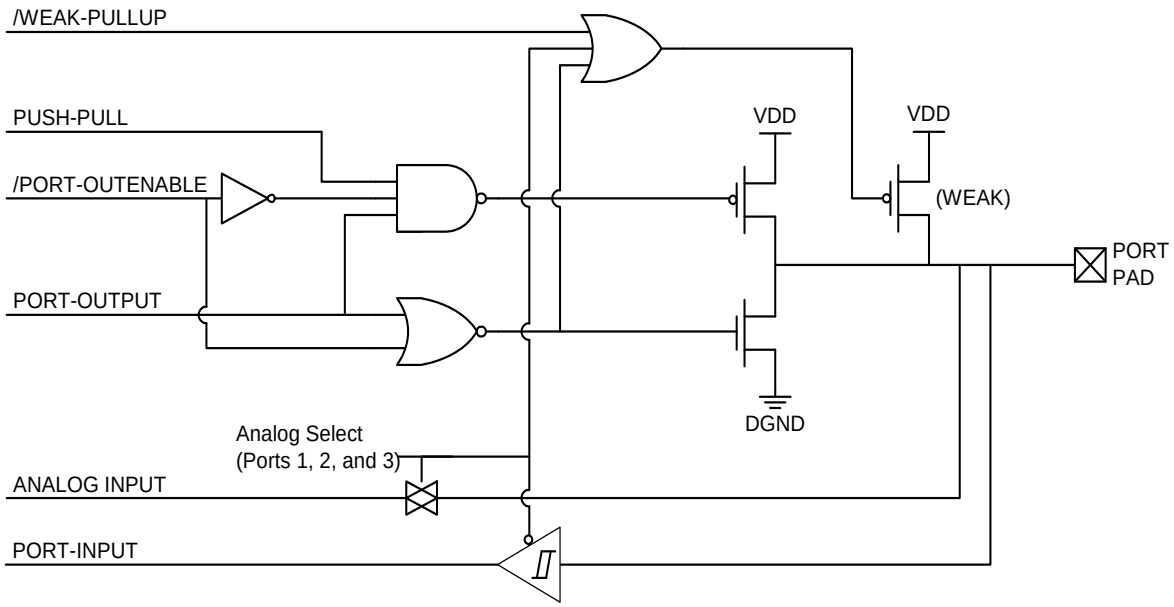



**Figure 18.1. Port I/O Cell Block Diagram**

### Table 18.1. Port I/O DC Electrical Characteristics

$V_{DD}$ = 2.7 to 3.6 V, –40 to +85 °C unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Output High Voltage ($V_{OH}$) | $I_{OH}$ = -3 mA, Port I/O Push-Pull<br>$I_{OH}$ = -10 µA, Port I/O Push-Pull<br>$I_{OH}$ = -10 mA, Port I/O Push-Pull | $V_{DD}$ – 0.7<br>$V_{DD}$ – 0.1 | <br><br>$V_{DD}$ – 0.8 | | V |
| Output Low Voltage ($V_{OL}$) | $I_{OL}$ = 8.5 mA<br>$I_{OL}$ = 10 µA<br>$I_{OL}$ = 25 mA | | <br><br>1.0 | 0.6<br>0.1 | V |
| Input High Voltage (VIH) | | 0.7 x $V_{DD}$ | | | |
| Input Low Voltage (VIL) | | | | 0.3 x $V_{DD}$ | |
| Input Leakage Current | DGND < Port Pin < $V_{DD}$, Pin Tri-state<br>Weak Pullup Off<br>Weak Pullup On | | <br><br>10 | <br>± 1 | µA |
| Input Capacitance | | | 5 | | pF |

A wide array of digital resources is available through the four lower I/O Ports: P0, P1, P2, and P3. Each of the pins on P0, P1, P2, and P3, can be defined as a General-Purpose I/O (GPIO) pin or can be controlled by a digital peripheral or function (like UART0 or /INT1 for example), as shown in Figure 18.2. The system designer controls which digital functions are assigned pins, limited only by the number of pins available. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read from its associated Data register regardless of whether that pin has been assigned to a digital peripheral or behaves as GPIO. The Port pins on Port 1 can be used as Analog Inputs to ADC2.

An External Memory Interface which is active during the execution of an off-chip MOVX instruction can be active on either the lower Ports or the upper Ports. See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

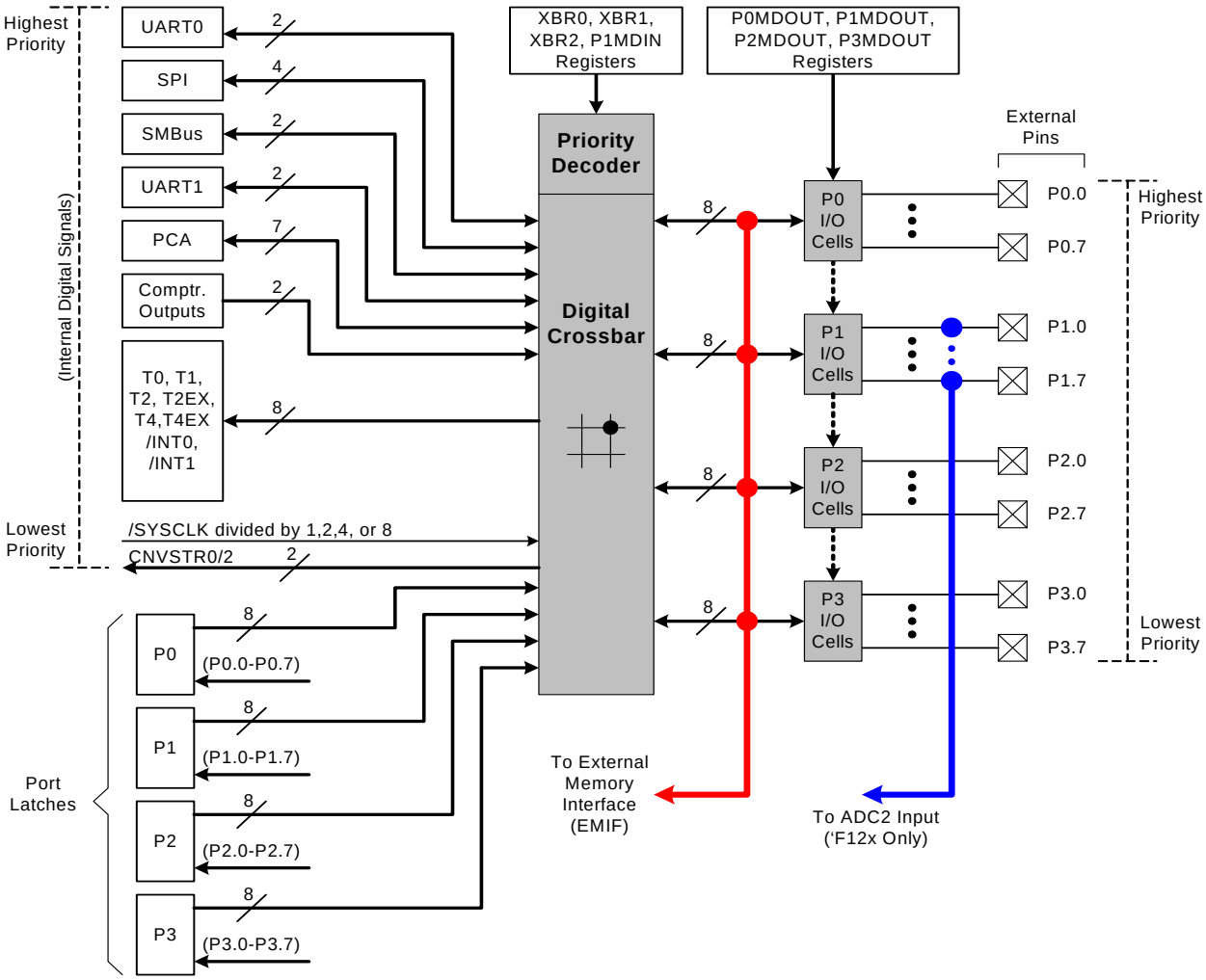



**Figure 18.2. Port I/O Functional Block Diagram**

## 18.1. Ports 0 through 3 and the Priority Crossbar Decoder

The Priority Crossbar Decoder, or “Crossbar”, allocates and assigns Port pins on Port 0 through Port 3 to the digital peripherals (UARTs, SMBus, PCA, Timers, etc.) on the device using a priority order. The Port pins are allocated in order starting with P0.0 and continue through P3.7 if necessary. The digital peripherals are assigned Port pins in a priority order which is listed in Figure 18.3, with UART0 having the highest priority and CNVSTR2 having the lowest priority.

### 18.1.1. Crossbar Pin Assignment and Allocation

The Crossbar assigns Port pins to a peripheral if the corresponding enable bits of the peripheral are set to a logic 1 in the Crossbar configuration registers XBR0, XBR1, and XBR2, shown in SFR Definition 18.1, SFR Definition 18.2, and SFR Definition 18.3. For example, if the UART0EN bit (XBR0.2) is set to a logic 1, the TX0 and RX0 pins will be mapped to P0.0 and P0.1 respectively.

| | P0 | | | | | | | | P1 | | | | | | | | P2 | | | | | | | | P3 | | | | | | | | Crossbar Register Bits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIN I/O | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| TX0 | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UART0EN: XBR0.2 |
| RX0 | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SCK | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SPI0EN: XBR0.1 |
| MISO | | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MOSI | | | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NSS | | | | ● | | ● | NSS is not assigned to a port pin when the SPI is placed in 3-wire mode | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SDA | ● | | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | | | | | | SMB0EN: XBR0.0 |
| SCL | | ● | | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | | | | | |
| TX1 | ● | | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | | | | UART1EN: XBR2.2 |
| RX1 | | ● | | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | | | |
| CEX0 | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | | PCA0ME: XBR0.[5:3] |
| CEX1 | | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | |
| CEX2 | | | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | |
| CEX3 | | | | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | |
| CEX4 | | | | | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | |
| CEX5 | | | | | | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | |
| ECI | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | ECI0E: XBR0.6 |
| CP0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | CP0E: XBR0.7 |
| CP1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | CP1E: XBR1.0 |
| T0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | T0E: XBR1.1 |
| /INT0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | INT0E: XBR1.2 |
| T1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | T1E: XBR1.3 |
| /INT1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | INT1E: XBR1.4 |
| T2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | T2E: XBR1.5 |
| T2EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | T2EXE: XBR1.6 |
| T4 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | T4E: XBR2.3 |
| T4EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | T4EXE: XBR2.4 |
| /SYSCLK | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | SYSCKE: XBR1.7 |
| CNVSTR0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | CNVSTE0: XBR2.0 |
| CNVSTR2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | CNVSTE2: XBR2.5 |
| | | | | | | ALE | /RD | /WR | AIN2.0/A8 | AIN2.1/A9 | **AIN2.2/A10** | **AIN2.3/A11** | **AIN2.4/A12** | AIN2.5/A13 | AIN2.6/A14 | AIN2.7/A15 | A8m/A0 | A9m/A1 | A10m/A2 | A11m/A3 | A12m/A4 | A13m/A5 | A14m/A6 | A15m/A7 | AD0/D0 | AD1/D1 | AD2/D2 | AD3/D3 | AD4/D4 | AD5/D5 | AD6/D6 | AD7/D7 | |
| | | | | | | | | | AIN2 Inputs/Non-muxed Addr H | | | | | | | | Muxed Addr H/Non-muxed Addr L | | | | | | | | Muxed Data/Non-muxed Data | | | | | | | | |

**Figure 18.3. Priority Crossbar Decode Table (EMIFLE = 0; P1MDIN = 0xFF)**

Because UART0 has the highest priority, its pins will always be mapped to P0.0 and P0.1 when UART0EN is set to a logic 1. If a digital peripheral’s enable bits are not set to a logic 1, then its ports are not accessible at the Port pins of the device. Also note that the Crossbar assigns pins to all associated functions when a serial communication peripheral is selected (i.e. SMBus, SPI, UART). It would be impossible, for exam-

ple, to assign TX0 to a Port pin without assigning RX0 as well. Each combination of enabled peripherals results in a unique device pinout.

All Port pins on Ports 0 through 3 that are not allocated by the Crossbar can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See SFR Definition 18.4, SFR Definition 18.6, SFR Definition 18.9, and SFR Definition 18.11), a set of SFR's which are both byte- and bit-addressable. The output states of Port pins that are allocated by the Crossbar are controlled by the digital peripheral that is mapped to those pins. Writes to the Port Data registers (or associated Port bits) will have no effect on the states of these pins.

A Read of a Port Data register (or Port bit) will always return the logic state present at the pin itself, regardless of whether the Crossbar has allocated the pin for peripheral use or not. An exception to this occurs during the execution of a *read-modify-write* instruction (ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR, SETB, and the bitwise MOV write operation). During the *read* cycle of the *read-modify-write* instruction, it is the contents of the Port Data register, not the state of the Port pins themselves, which is read. Note that at clock rates above 50 MHz, when a pin is written and then immediately read (i.e. a write instruction followed immediately by a read instruction), the propagation delay of the port drivers may cause the read instruction to return the previous logic level of the pin.

Because the Crossbar registers affect the pinout of the peripherals of the device, they are typically configured in the initialization code of the system before the peripherals themselves are configured. Once configured, the Crossbar registers are typically left alone.

Once the Crossbar registers have been properly configured, the Crossbar is enabled by setting XBARE (XBR2.4) to a logic 1. **Until XBARE is set to a logic 1, the output drivers on Ports 0 through 3 are explicitly disabled in order to prevent possible contention on the Port pins while the Crossbar registers and other registers which can affect the device pinout are being written**.

The output drivers on Crossbar-assigned input signals (like RX0, for example) are explicitly disabled; thus the values of the Port Data registers and the PnMDOUT registers have no effect on the states of these pins.

### 18.1.2. Configuring the Output Modes of the Port Pins

The output drivers on Ports 0 through 3 remain disabled until the Crossbar is enabled by setting XBARE (XBR2.4) to a logic 1.

The output mode of each port pin can be configured to be either Open-Drain or Push-Pull. In the Push-Pull configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and writing a logic 1 will cause the Port pin to be driven to $V_{DD}$. In the Open-Drain configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire (like the SDA signal on an SMBus connection).

The output modes of the Port pins on Ports 0 through 3 are determined by the bits in the associated PnMDOUT registers (See SFR Definition 18.5, SFR Definition 18.8, SFR Definition 18.10, and SFR Definition 18.12). For example, a logic 1 in P3MDOUT.7 will configure the output mode of P3.7 to Push-Pull; a logic 0 in P3MDOUT.7 will configure the output mode of P3.7 to Open-Drain. All Port pins default to Open-Drain output.

The PnMDOUT registers control the output modes of the port pins regardless of whether the Crossbar has allocated the Port pin for a digital peripheral or not. The exceptions to this rule are: the Port pins connected to SDA, SCL, RX0 (if UART0 is in Mode 0), and RX1 (if UART1 is in Mode 0) are always configured as Open-Drain outputs, regardless of the settings of the associated bits in the PnMDOUT registers.

### 18.1.3. Configuring Port Pins as Digital Inputs

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic 1 to the associated bit in the Port Data register. For example, P3.7 is configured as a digital input by setting P3MDOUT.7 to a logic 0 and P3.7 to a logic 1.

If the Port pin has been assigned to a digital peripheral by the Crossbar and that pin functions as an input (for example RX0, the UART0 receive pin), then the output drivers on that pin are automatically disabled.

### 18.1.4. Weak Pullups

By default, each Port pin has an internal weak pullup device enabled which provides a resistive connection (about 100 kΩ) between the pin and $V_{DD}$. The weak pullup devices can be globally disabled by writing a logic 1 to the Weak Pullup Disable bit, (WEAKPUD, XBR2.7). The weak pullup is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pullup device. The weak pullup device can also be explicitly disabled on any Port 1 pin by configuring the pin as an Analog Input, as described below.

### 18.1.5. Configuring Port 1 Pins as Analog Inputs

The pins on Port 1 can serve as analog inputs to the ADC2 analog MUX on the C8051F12x devices. A Port pin is configured as an Analog Input by writing a logic 0 to the associated bit in the PnMDIN registers. All Port pins default to a Digital Input mode. Configuring a Port pin as an analog input:

1. Disables the digital input path from the pin. This prevents additional power supply current from being drawn when the voltage at the pin is near $V_{DD}$ / 2. A read of the Port Data bit will return a logic 0 regardless of the voltage at the Port pin.
2. Disables the weak pullup device on the pin.
3. Causes the Crossbar to “skip over” the pin when allocating Port pins for digital peripherals.

Note that the output drivers on a pin configured as an Analog Input are not explicitly disabled. Therefore, the associated P1MDOUT bits of pins configured as Analog Inputs should explicitly be set to logic 0 (Open-Drain output mode), and the associated Port1 Data bits should be set to logic 1 (high-impedance). Also note that it is not required to configure a Port pin as an Analog Input in order to use it as an input to ADC2, however, it is strongly recommended. See the ADC2 section in this datasheet for further information.

### 18.1.6. External Memory Interface Pin Assignments

If the External Memory Interface (EMIF) is enabled on the Low ports (Ports 0 through 3), EMIFLE (XBR2.5) should be set to a logic 1 so that the Crossbar will not assign peripherals to P0.7 (/WR), P0.6 (/RD), and if the External Memory Interface is in Multiplexed mode, P0.5 (ALE). Figure 18.4 shows an example Crossbar Decode Table with EMIFLE=1 and the EMIF in Multiplexed mode. Figure 18.5 shows an example Crossbar Decode Table with EMIFLE=1 and the EMIF in Non-multiplexed mode.

If the External Memory Interface is enabled on the Low ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Crossbar registers or the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus. See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

| | P0 | | | | | | | | P1 | | | | | | | | P2 | | | | | | | | P3 | | | | | | | | Crossbar Register Bits |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIN I/O | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| TX0 | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UART0EN: XBR0.2 |
| RX0 | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SCK | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SPI0EN: XBR0.1 |
| MISO | | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MOSI | | | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NSS | | | | ● | | | | | ● NSS is not assigned to a port pin when the SPI is placed in 3-wire mode | | | | | | | | | | | | | | | | | | | | | | | | |
| SDA | ● | | ● | ● | ● | | | | ● | ● | | | | | | | | | | | | | | | | | | | | | | | SMB0EN: XBR0.0 |
| SCL | | ● | | ● | ● | | | | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | | |
| TX1 | ● | | ● | ● | ● | | | | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | | UART1EN: XBR2.2 |
| RX1 | | ● | | ● | ● | | | | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | | |
| CEX0 | ● | | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | | PCA0ME: XBR0.[5:3] |
| CEX1 | | ● | | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | | |
| CEX2 | | | ● | | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | | |
| CEX3 | | | | ● | | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | | |
| CEX4 | | | | | ● | | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | | |
| CEX5 | | | | | | | | | ● | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | | |
| ECI | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | ECI0E: XBR0.6 |
| CP0 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | CP0E: XBR0.7 |
| CP1 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | CP1E: XBR1.0 |
| T0 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | T0E: XBR1.1 |
| /INT0 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | INT0E: XBR1.2 |
| T1 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | T1E: XBR1.3 |
| /INT1 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | INT1E: XBR1.4 |
| T2 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | T2E: XBR1.5 |
| T2EX | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | T2EXE: XBR1.6 |
| T4 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | T4E: XBR2.3 |
| T4EX | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | T4EXE: XBR2.4 |
| /SYSCLK | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | SYSCKE: XBR1.7 |
| CNVSTR0 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | CNVSTE0: XBR2.0 |
| CNVSTR2 | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | CNVSTE2: XBR2.5 |
| | | | | | | ALE | /RD | /WR | AIN2.0/A8 | AIN2.1/A9 | **AIN2.2/A10** | **AIN2.3/A11** | **AIN2.4/A12** | AIN2.5/A13 | AIN2.6/A14 | AIN2.7/A15 | A8m/A0 | A9m/A1 | A10m/A2 | A11m/A3 | A12m/A4 | A13m/A5 | A14m/A6 | A15m/A7 | AD0/D0 | AD1/D1 | AD2/D2 | AD3/D3 | AD4/D4 | AD5/D5 | AD6/D6 | AD7/D7 | |
| | | | | | | | | | AIN2 Inputs/Non-muxed Addr H | | | | | | | | Muxed Addr H/Non-muxed Addr L | | | | | | | | Muxed Data/Non-muxed Data | | | | | | | | |

**Figure 18.4. Priority Crossbar Decode Table (EMIFLE = 1; EMIF in Multiplexed Mode; P1MDIN = 0xFF)**

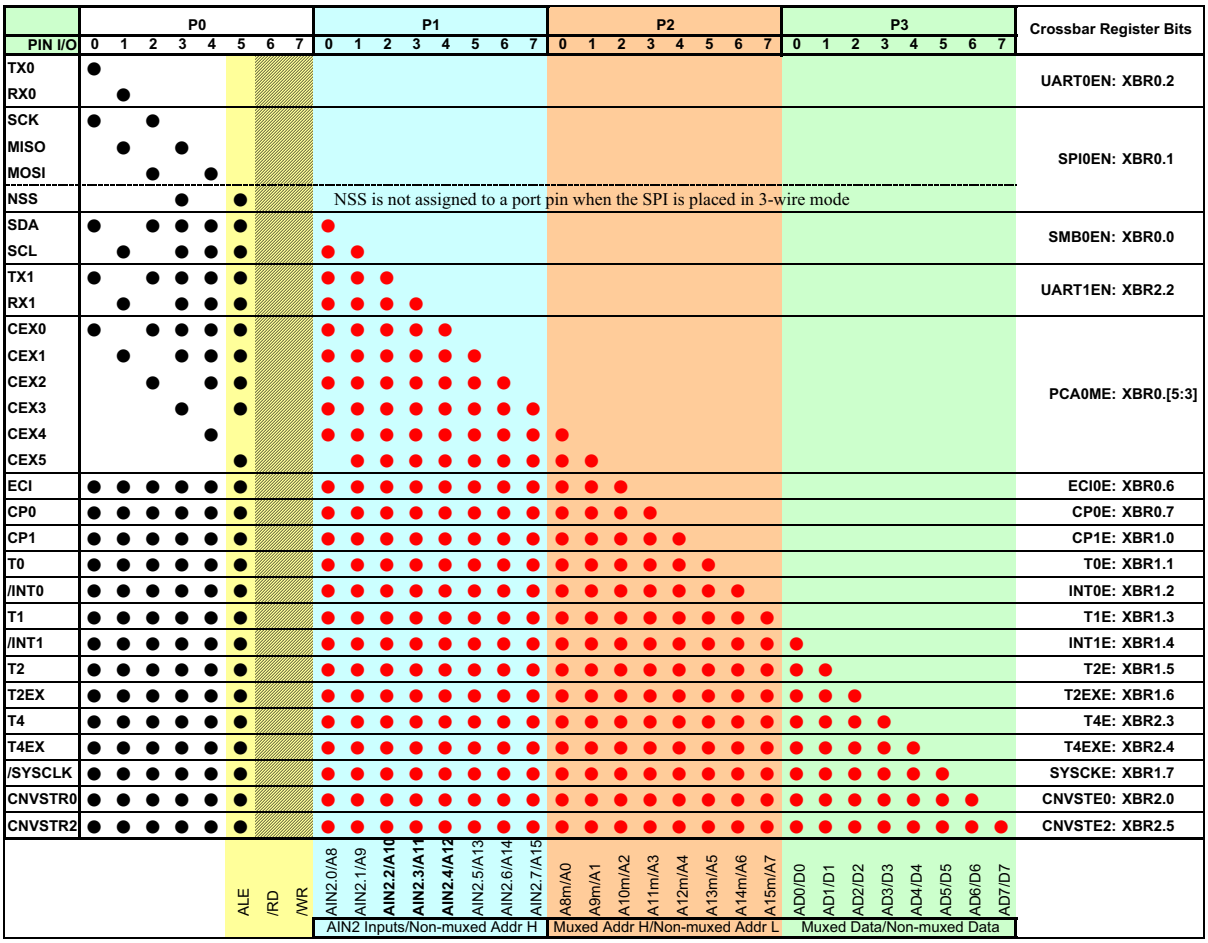
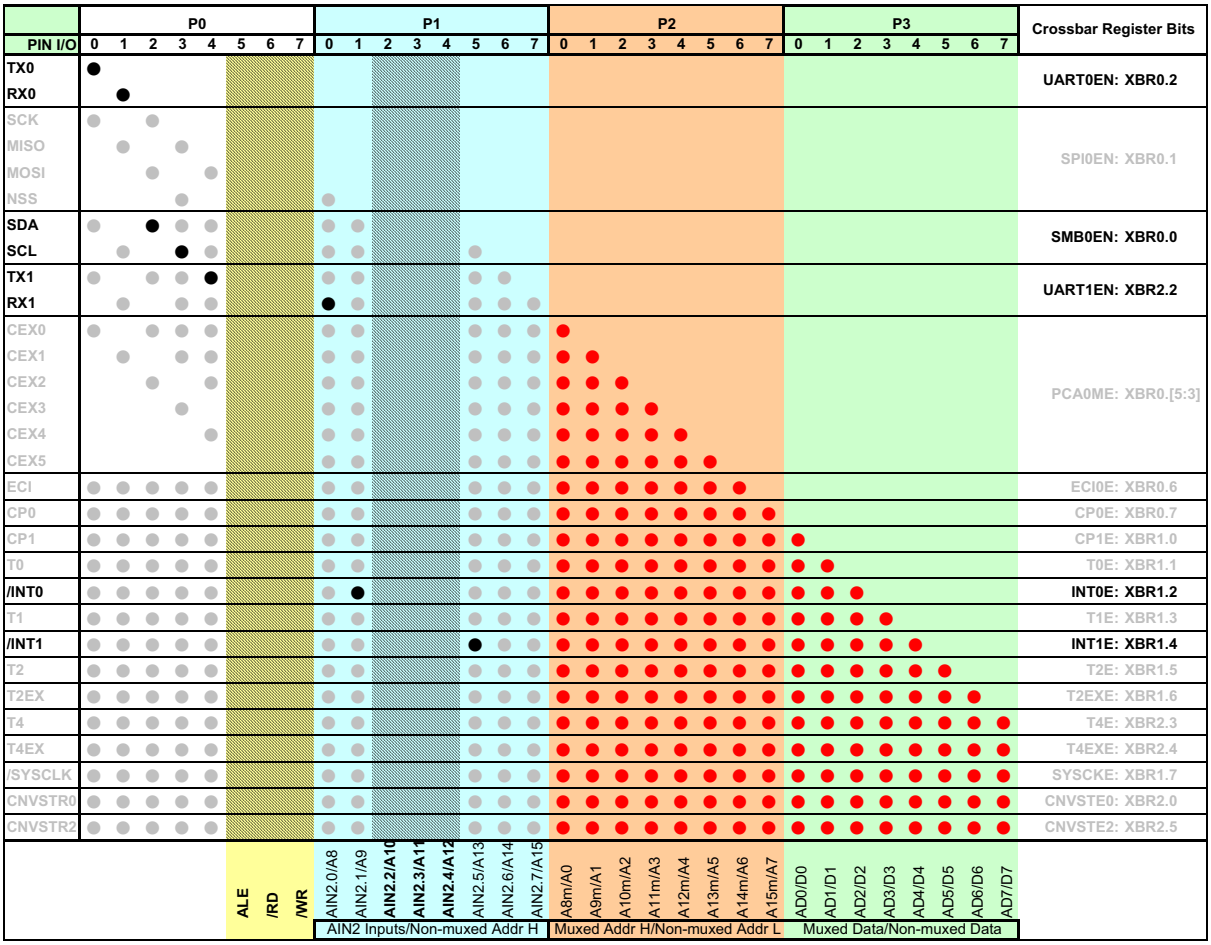



**Figure 18.5. Priority Crossbar Decode Table (EMIFLE = 1; EMIF in Non-Multiplexed Mode; P1MDIN = 0xFF)**

### 18.1.7. Crossbar Pin Assignment Example

In this example (Figure 18.6), we configure the Crossbar to allocate Port pins for UART0, the SMBus, UART1, /INT0, and /INT1 (8 pins total). Additionally, we configure the External Memory Interface to operate in Multiplexed mode and to appear on the Low ports. Further, we configure P1.2, P1.3, and P1.4 for Analog Input mode so that the voltages at these pins can be measured by ADC2. The configuration steps are as follows:

1. XBR0, XBR1, and XBR2 are set such that UART0EN = 1, SMB0EN = 1, INT0E = 1, INT1E = 1, and EMIFLE = 1. Thus: XBR0 = 0x05, XBR1 = 0x14, and XBR2 = 0x02.
2. We configure the External Memory Interface to use Multiplexed mode and to appear on the Low ports. PRTSEL = 0, EMD2 = 0.
3. We configure the desired Port 1 pins to Analog Input mode by setting P1MDIN to 0xE3 (P1.4, P1.3, and P1.2 are Analog Inputs, so their associated P1MDIN bits are set to logic 0).
4. We enable the Crossbar by setting XBARE = 1: XBR2 = 0x42.
   - UART0 has the highest priority, so P0.0 is assigned to TX0, and P0.1 is assigned to RX0.
   - The SMBus is next in priority order, so P0.2 is assigned to SDA, and P0.3 is assigned to SCL.
   - UART1 is next in priority order, so P0.4 is assigned to TX1. Because the External Memory Interface is selected on the lower Ports, EMIFLE = 1, which causes the Crossbar to skip P0.6 (/RD) and P0.7 (/WR). Because the External Memory Interface is configured in Multiplexed mode, the Crossbar will also skip P0.5 (ALE). RX1 is assigned to the next non-skipped pin, which in this case is P1.0.
   - /INT0 is next in priority order, so it is assigned to P1.1.
   - P1MDIN is set to 0xE3, which configures P1.2, P1.3, and P1.4 as Analog Inputs, causing the Crossbar to skip these pins.
   - /INT1 is next in priority order, so it is assigned to the next non-skipped pin, which is P1.5.
   - The External Memory Interface will drive Ports 2 and 3 (denoted by red dots in Figure 18.6) during the execution of an off-chip MOVX instruction.
5. We set the UART0 TX pin (TX0, P0.0) and UART1 TX pin (TX1, P0.4) outputs to Push-Pull by setting P0MDOUT = 0x11.
6. We configure all EMIF-controlled pins to push-pull output mode by setting P0MDOUT |= 0xE0; P2MDOUT = 0xFF; P3MDOUT = 0xFF.
7. We explicitly disable the output drivers on the 3 Analog Input pins by setting P1MDOUT = 0x00 (configure outputs to Open-Drain) and P1 = 0xFF (a logic 1 selects the high-impedance state).




**(EMIFLE = 1; EMIF in Multiplexed Mode; P1MDIN = 0xE3;
XBR0 = 0x05; XBR1 = 0x14; XBR2 = 0x42)**

**Figure 18.6. Crossbar Example**

## SFR Definition 18.1. XBR0: Port I/O Crossbar Register 0

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CP0E | ECI0E | PCA0ME | | | UART0EN | SPI0EN | SMB0EN | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE1
SFR Page: F

Bit7: CP0E: Comparator 0 Output Enable Bit.
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.

Bit6: ECI0E: PCA0 External Counter Input Enable Bit.
0: PCA0 External Counter Input unavailable at Port pin.
1: PCA0 External Counter Input (ECI0) routed to Port pin.

Bits5–3: PCA0ME: PCA0 Module I/O Enable Bits.
000: All PCA0 I/O unavailable at port pins.
001: CEX0 routed to port pin.
010: CEX0, CEX1 routed to 2 port pins.
011: CEX0, CEX1, and CEX2 routed to 3 port pins.
100: CEX0, CEX1, CEX2, and CEX3 routed to 4 port pins.
101: CEX0, CEX1, CEX2, CEX3, and CEX4 routed to 5 port pins.
110: CEX0, CEX1, CEX2, CEX3, CEX4, and CEX5 routed to 6 port pins.

Bit2: UART0EN: UART0 I/O Enable Bit.
0: UART0 I/O unavailable at Port pins.
1: UART0 TX routed to P0.0, and RX routed to P0.1.

Bit1: SPI0EN: SPI0 Bus I/O Enable Bit.
0: SPI0 I/O unavailable at Port pins.
1: SPI0 SCK, MISO, MOSI, and NSS routed to 4 Port pins. Note that the NSS signal is not assigned to a port pin if the SPI is in 3-wire mode. See Section "**17. External Data Memory Interface and On-Chip XRAM**" on page **219** for more information.

Bit0: SMB0EN: SMBus0 Bus I/O Enable Bit.
0: SMBus0 I/O unavailable at Port pins.
1: SMBus0 SDA and SCL routed to 2 Port pins.

## SFR Definition 18.2. XBR1: Port I/O Crossbar Register 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SYSCKE | T2EXE | T2E | INT1E | T1E | INT0E | T0E | CP1E | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE2
SFR Page: F

Bit7: SYSCKE: /SYSCLK Output Enable Bit.
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK (divided by 1, 2, 4, or 8) routed to Port pin. divide factor is determined by the CLKDIV1–0 bits in register CLKSEL (See **Section “14. Oscillators” on page 185**).

Bit6: T2EXE: T2EX Input Enable Bit.
0: T2EX unavailable at Port pin.
1: T2EX routed to Port pin.

Bit5: T2E: T2 Input Enable Bit.
0: T2 unavailable at Port pin.
1: T2 routed to Port pin.

Bit4: INT1E: /INT1 Input Enable Bit.
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port pin.

Bit3: T1E: T1 Input Enable Bit.
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.

Bit2: INT0E: /INT0 Input Enable Bit.
0: /INT0 unavailable at Port pin.
1: /INT0 routed to Port pin.

Bit1: T0E: T0 Input Enable Bit.
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.

Bit0: CP1E: CP1 Output Enable Bit.
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

## SFR Definition 18.3. XBR2: Port I/O Crossbar Register 2

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| WEAKPUD | XBARE | CNVST2E | T4EXE | T4E | UART1E | EMIFLE | CNVST0E | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE3
SFR Page: F

Bit7: WEAKPUD: Weak Pullup Disable Bit.
0: Weak pullups globally enabled.
1: Weak pullups globally disabled.

Bit6: XBARE: Crossbar Enable Bit.
0: Crossbar disabled. All pins on Ports 0, 1, 2, and 3, are forced to Input mode.
1: Crossbar enabled.

Bit5: CNVST2E: External Convert Start 2 Input Enable Bit.
0: CNVSTR2 unavailable at Port pin.
1: CNVSTR2 routed to Port pin.

Bit4: T4EXE: T4EX Input Enable Bit.
0: T4EX unavailable at Port pin.
1: T4EX routed to Port pin.

Bit3: T4E: T4 Input Enable Bit.
0: T4 unavailable at Port pin.
1: T4 routed to Port pin.

Bit2: UART1E: UART1 I/O Enable Bit.
0: UART1 I/O unavailable at Port pins.
1: UART1 TX and RX routed to 2 Port pins.

Bit1: EMIFLE: External Memory Interface Low-Port Enable Bit.
0: P0.7, P0.6, and P0.5 functions are determined by the Crossbar or the Port latches.
1: If EMI0CF.4 = ‘0’ (External Memory Interface is in Multiplexed mode)
P0.7 (/WR), P0.6 (/RD), and P0.5 (ALE) are ‘skipped’ by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.
1: If EMI0CF.4 = ‘1’ (External Memory Interface is in Non-multiplexed mode)
P0.7 (/WR) and P0.6 (/RD) are ‘skipped’ by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.

Bit0: CNVST0E: ADC0 External Convert Start Input Enable Bit.
0: CNVST0 for ADC0 unavailable at Port pin.
1: CNVST0 for ADC0 routed to Port pin.

### SFR Definition 18.4. P0: Port0 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x80
SFR Page: All Pages

Bits7–0: P0.[7:0]: Port0 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P0MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P0.n pin is logic low.
1: P0.n pin is logic high.

**Note:** P0.7 (/WR), P0.6 (/RD), and P0.5 (ALE) can be driven by the External Data Memory Interface. See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information. See also SFR Definition 18.3 for information about configuring the Crossbar for External Memory accesses.

### SFR Definition 18.5. P0MDOUT: Port0 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA4
SFR Page: F

Bits7–0: P0MDOUT.[7:0]: Port0 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

**Note:** SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.

## SFR Definition 18.6. P1: Port1 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x90
SFR Page: All Pages

Bits7–0: P1.[7:0]: Port1 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P1MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P1.n pin is logic low.
1: P1.n pin is logic high.

**Notes:**

1. On C8051F12x devices, P1.[7:0] can be configured as inputs to ADC2 as AIN2.[7:0], in which case they are 'skipped' by the Crossbar assignment process and their digital input paths are disabled, depending on P1MDIN (See SFR Definition 18.7**)**. Note that in analog mode, the output mode of the pin is determined by the Port 1 latch and P1MDOUT (SFR Definition 18.8). See **Section "7. ADC2 (8-Bit ADC, C8051F12x Only)" on page 91** for more information about ADC2.
2. P1.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

## SFR Definition 18.7. P1MDIN: Port1 Input Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xAD
SFR Page: F

Bits7–0: P1MDIN.[7:0]: Port 1 Input Mode Bits.
0: Port Pin is configured in Analog Input mode. The digital input path is disabled (a read from the Port bit will always return '0'). The weak pullup on the pin is disabled.
1: Port Pin is configured in Digital Input mode. A read from the Port bit will return the logic level at the Pin. When configured as a digital input, the state of the weak pullup for the port pin is determined by the WEAKPUD bit (XBR2.7, see SFR Definition 18.3).

### SFR Definition 18.8. P1MDOUT: Port1 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA5
SFR Page: F

Bits7–0: P1MDOUT.[7:0]: Port1 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

**Note:** SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.

### SFR Definition 18.9. P2: Port2 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xA0
SFR Page: All Pages

Bits7–0: P2.[7:0]: Port2 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P2MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P2.n pin is logic low.
1: P2.n pin is logic high.

**Note:** P2.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Multiplexed mode, or as Address[7:0] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

## SFR Definition 18.10. P2MDOUT: Port2 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA6
SFR Page: F

Bits7–0: P2MDOUT.[7:0]: Port2 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

**Note:** SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.

## SFR Definition 18.11. P3: Port3 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xB0
SFR Page: All Pages

Bits7–0: P3.[7:0]: Port3 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P3MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P3.n pin is logic low.
1: P3.n pin is logic high.

**Note:** P3.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

**SFR Definition 18.12. P3MDOUT: Port3 Output Mode**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA7
SFR Page: F

Bits7–0: P3MDOUT.[7:0]: Port3 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

## 18.2. Ports 4 through 7 (100-pin TQFP devices only)

All Port pins on Ports 4 through 7 can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See SFR Definition 18.13, SFR Definition 18.15, SFR Definition 18.17, and SFR Definition 18.19), a set of SFR's which are both bit and byte-addressable. Note also that the Port 4, 5, 6, and 7 registers are located on SFR Page F. The SFRPAGE register must be set to 0x0F to access these Port registers.

A Read of a Port Data register (or Port bit) will always return the logic state present at the pin itself, regardless of whether the Crossbar has allocated the pin for peripheral use or not. An exception to this occurs during the execution of a *read-modify-write* instruction (ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR, SETB, and the bitwise MOV write operation). During the *read* cycle of the *read-modify-write* instruction, it is the contents of the Port Data register, not the state of the Port pins themselves, which is read. Note that at clock rates above 50 MHz, when a pin is written and then immediately read (i.e. a write instruction followed immediately by a read instruction), the propagation delay of the port drivers may cause the read instruction to return the previous logic level of the pin.

### 18.2.1. Configuring Ports which are not Pinned Out

Although P4, P5, P6, and P7 are not brought out to pins on the 64-pin TQFP devices, the Port Data registers are still present and can be used by software. Because the digital input paths also remain active, it is recommended that these pins not be left in a 'floating' state in order to avoid unnecessary power dissipation arising from the inputs floating to non-valid logic levels. This condition can be prevented by any of the following:

1. Leave the weak pullup devices enabled by setting WEAKPUD (XBR2.7) to a logic 0.
2. Configure the output modes of P4, P5, P6, and P7 to "Push-Pull" by writing PnMDOUT = 0xFF.
3. Force the output states of P4, P5, P6, and P7 to logic 0 by writing zeros to the Port Data registers: P4 = 0x00, P5 = 0x00, P6= 0x00, and P7 = 0x00.

### 18.2.2. Configuring the Output Modes of the Port Pins

The output mode of each port pin can be configured to be either Open-Drain or Push-Pull. In the Push-Pull configuration, a logic 0 in the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to be driven to $V_{DD}$. In the Open-Drain configuration, a logic 0 in the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire.

The output modes of the Port pins on Ports 4 through 7 are determined by the bits in their respective PnMDOUT Output Mode Registers. Each bit in PnMDOUT controls the output mode of its corresponding port pin (see SFR Definition 18.14, SFR Definition 18.16, SFR Definition 18.18, and SFR Definition 18.20). For example, to place Port pin 4.3 in push-pull mode (digital output), set P4MDOUT.3 to logic 1. All port pins default to open-drain mode upon device reset.

### 18.2.3. Configuring Port Pins as Digital Inputs

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic 1 to the associated bit in the Port Data register. For example, P7.7 is configured as a digital input by setting P7MDOUT.7 to a logic 0 and P7.7 to a logic 1.

### 18.2.4. Weak Pullups

By default, each Port pin has an internal weak pullup device enabled which provides a resistive connection (about 100 kΩ) between the pin and $V_{DD}$. The weak pullup devices can be globally disabled by writing a logic 1 to the Weak Pullup Disable bit, (WEAKPUD, XBR2.7). The weak pullup is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pullup device.

### 18.2.5. External Memory Interface

If the External Memory Interface (EMIF) is enabled on the High ports (Ports 4 through 7), EMIFLE (XBR2.5) should be set to a logic 0.

If the External Memory Interface is enabled on the High ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus during the MOVX execution. See **Section “17. External Data Memory Interface and On-Chip XRAM” on page 219** for more information about the External Memory Interface.

### SFR Definition 18.13. P4: Port4 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P4.7 | P4.6 | P4.5 | P4.4 | P4.3 | P4.2 | P4.1 | P4.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xC8
SFR Page: F

Bits7–0: P4.[7:0]: Port4 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P4MDOUT.n bit = 0). See SFR Definition 18.14.
Read - Returns states of I/O pins.
0: P4.n pin is logic low.
1: P4.n pin is logic high.

**Note:** P4.7 (/WR), P4.6 (/RD), and P4.5 (ALE) can be driven by the External Data Memory Interface. See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information.

### SFR Definition 18.14. P4MDOUT: Port4 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9C
SFR Page: F

Bits7–0: P4MDOUT.[7:0]: Port4 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

## SFR Definition 18.15. P5: Port5 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P5.7 | P5.6 | P5.5 | P5.4 | P5.3 | P5.2 | P5.1 | P5.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xD8
SFR Page: F

Bits7–0: P5.[7:0]: Port5 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P5MDOUT bit = 0). See SFR Definition 18.16.
Read - Returns states of I/O pins.
0: P5.n pin is logic low.
1: P5.n pin is logic high.

**Note:** P5.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

## SFR Definition 18.16. P5MDOUT: Port5 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9D
SFR Page: F

Bits7–0: P5MDOUT.[7:0]: Port5 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

### SFR Definition 18.17. P6: Port6 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P6.7 | P6.6 | P6.5 | P6.4 | P6.3 | P6.2 | P6.1 | P6.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xE8
SFR Page: F

Bits7–0: P6.[7:0]: Port6 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P6MDOUT bit = 0). See SFR Definition 18.18.
Read - Returns states of I/O pins.
0: P6.n pin is logic low.
1: P6.n pin is logic high.

**Note:** P6.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Multiplexed mode, or as Address[7:0] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

### SFR Definition 18.18. P6MDOUT: Port6 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9E
SFR Page: F

Bits7–0: P6MDOUT.[7:0]: Port6 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

## SFR Definition 18.19. P7: Port7 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P7.7 | P7.6 | P7.5 | P7.4 | P7.3 | P7.2 | P7.1 | P7.0 | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xF8
SFR Page: F

Bits7–0: P7.[7:0]: Port7 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P7MDOUT bit = 0). See SFR Definition 18.20.
Read - Returns states of I/O pins.
0: P7.n pin is logic low.
1: P7.n pin is logic high.

**Note:** P7.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

## SFR Definition 18.20. P7MDOUT: Port7 Output Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9F
SFR Page: F

Bits7–0: P7MDOUT.[7:0]: Port7 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

## NOTES:

## 19. System Management Bus / I2C Bus (SMBus0)

The SMBus0 I/O interface is a two-wire, bi-directional serial bus. SMBus0 is compliant with the System Management Bus Specification, version 1.1, and compatible with the I2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus0 interface autonomously controlling the serial transfer of the data. A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

SMBus0 may operate as a master and/or slave, and may function on a bus with multiple masters. SMBus0 provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation.

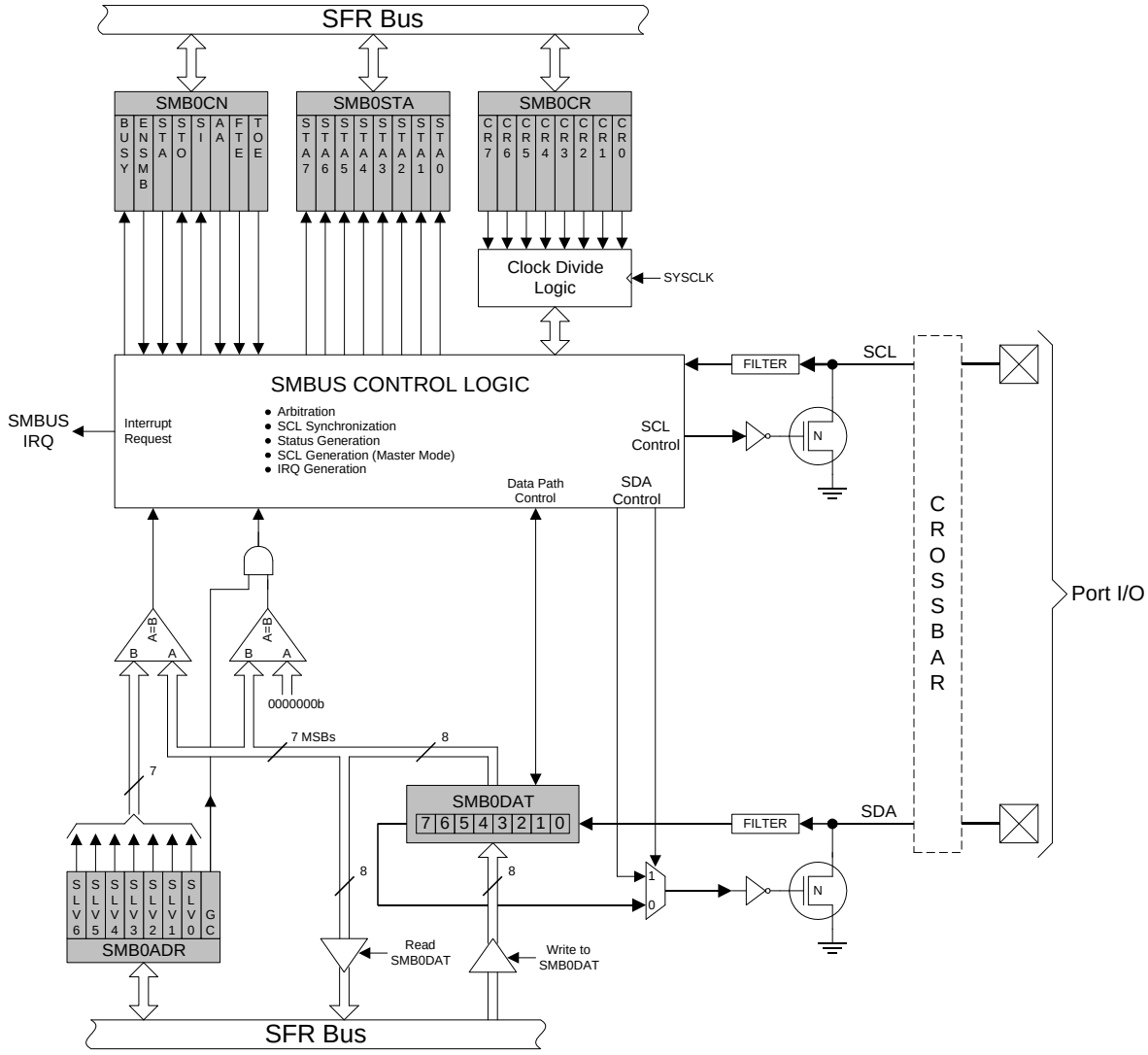



**Figure 19.1. SMBus0 Block Diagram**

Figure 19.2 shows a typical SMBus configuration. The SMBus0 interface will work at any voltage between 3.0 and 5.0 V and 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 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 will not exceed 300 ns and 1000 ns, respectively.

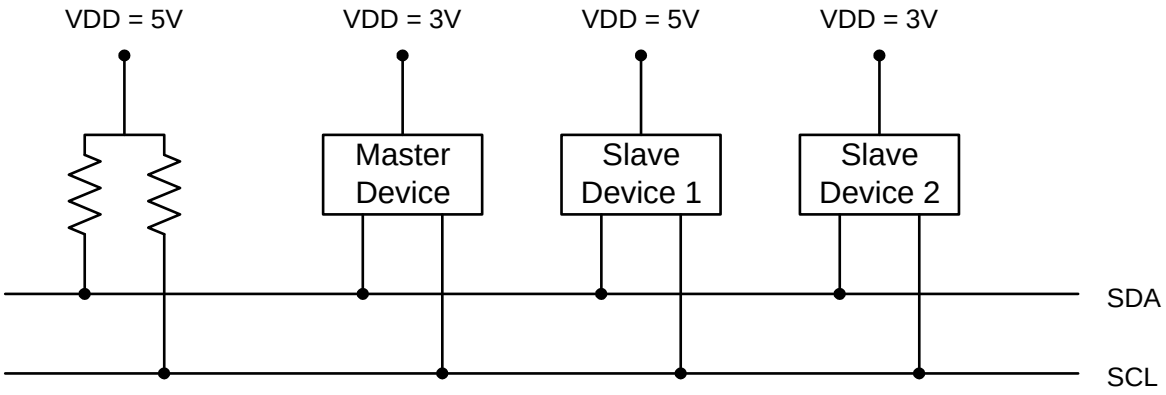



**Figure 19.2. Typical SMBus Configuration**

## 19.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I2C-bus and how to use it (including specifications), Philips Semiconductor.
2. The I2C-Bus Specification -- Version 2.0, Philips Semiconductor.
3. System Management Bus Specification -- Version 1.1, SBS Implementers Forum.

## 19.2. SMBus Protocol

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. Note: 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 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. Each byte that is received (by a master or slave) must be acknowledged (ACK) with a low SDA during a high SCL (see Figure 19.3). If the receiving device does not ACK, the transmitting device will read a “not acknowledge” (NACK), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address. 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 19.3 illustrates a typical SMBus transaction.

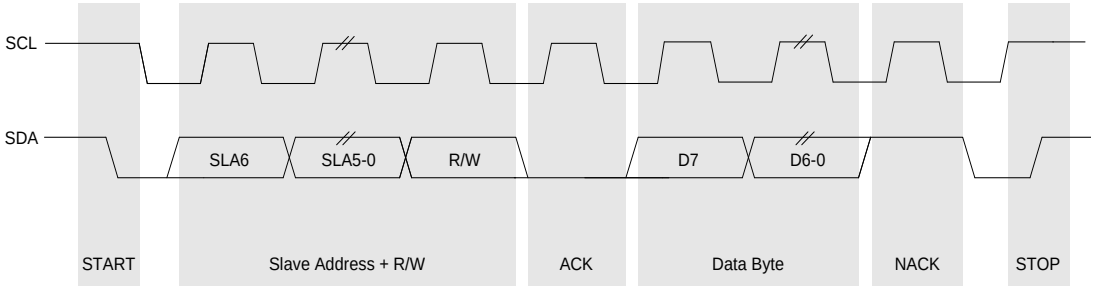



**Figure 19.3. SMBus Transaction**

### 19.2.1. 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 19.2.4**). 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 give up the bus. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

### 19.2.2. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, 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.

### 19.2.3. 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.

### 19.2.4. 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. If an SMBus device is waiting to generate a Master START, the START will be generated following the bus free timeout.

## 19.3. SMBus Transfer Modes

The SMBus0 interface may be configured to operate as a master and/or a slave. At any particular time, the interface will be operating in one of the following modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. See Table 19.1 for transfer mode status decoding using the SMB0STA status register. The following mode descriptions illustrate an interrupt-driven SMBus0 application; SMBus0 may alternatively be operated in polled mode.

### 19.3.1. Master Transmitter Mode

Serial data is transmitted on SDA while the serial clock is output on SCL. SMBus0 generates a START condition and then transmits the first byte containing the address of the target slave device and the data direction bit. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. The SMBus0 interface transmits one or more bytes of serial data, waiting for an acknowledge (ACK) from the slave after each byte. To indicate the end of the serial transfer, SMBus0 generates a STOP condition.

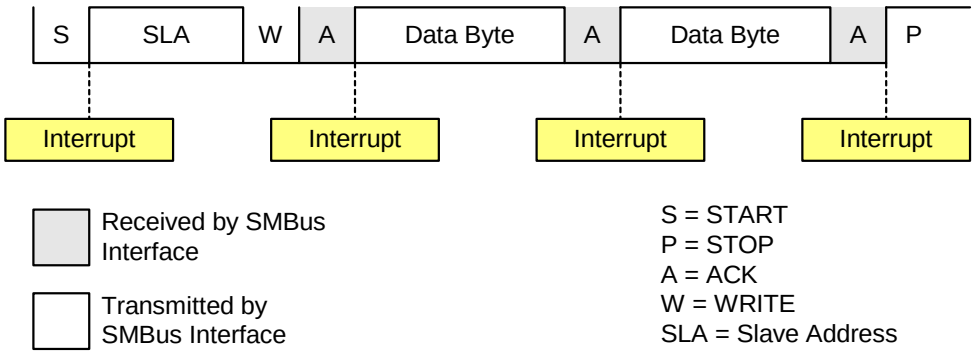



**Figure 19.4. Typical Master Transmitter Sequence**

### 19.3.2. Master Receiver Mode

Serial data is received on SDA while the serial clock is output on SCL. The SMBus0 interface generates a START followed by the first data 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 to indicate a "READ" operation. The SMBus0 interface receives serial data from the slave and generates the clock on SCL. After each byte is received, SMBus0 generates an ACK or NACK depending on the state of the AA bit in register SMB0CN. SMBus0 generates a STOP condition to indicate the end of the serial transfer.

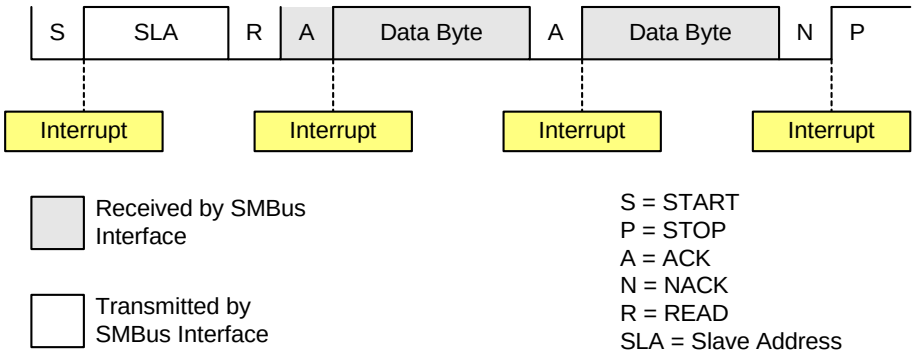



**Figure 19.5. Typical Master Receiver Sequence**

### 19.3.3. Slave Transmitter Mode

Serial data is transmitted on SDA while the serial clock is received on SCL. The SMBus0 interface receives a START followed by data byte containing the slave address and direction bit. If the received slave address matches the address held in register SMB0ADR, the SMBus0 interface generates an ACK. SMBus0 will also ACK if the general call address (0x00) is received and the General Call Address Enable bit (SMB0ADR.0) is set to logic 1. In this case the data direction bit (R/W) will be logic 1 to indicate a "READ" operation. The SMBus0 interface receives the clock on SCL and transmits one or more bytes of serial data, waiting for an ACK from the master after each byte. SMBus0 exits slave mode after receiving a STOP condition from the master.

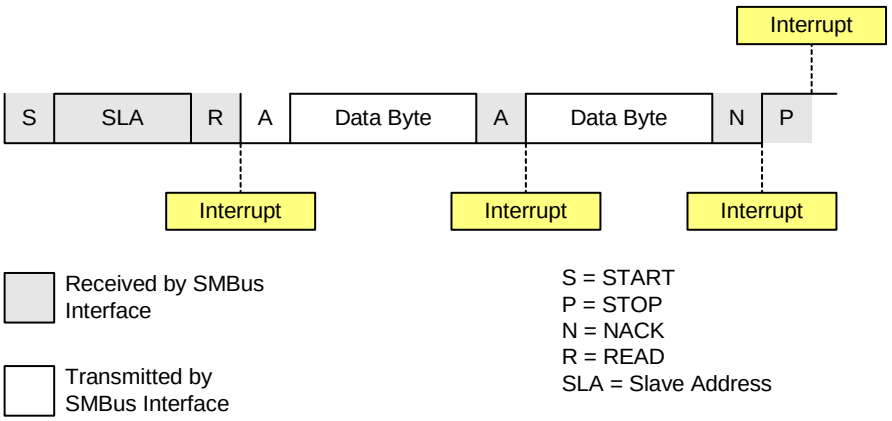



**Figure 19.6. Typical Slave Transmitter Sequence**

### 19.3.4. Slave Receiver Mode

Serial data is received on SDA while the serial clock is received on SCL. The SMBus0 interface receives a START followed by data byte containing the slave address and direction bit. If the received slave address matches the address held in register SMB0ADR, the interface generates an ACK. SMBus0 will also ACK if the general call address (0x00) is received and the General Call Address Enable bit (SMB0ADR.0) is set to logic 1. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. The SMBus0 interface receives one or more bytes of serial data; after each byte is received, the interface transmits an ACK or NACK depending on the state of the AA bit in SMB0CN. SMBus0 exits Slave Receiver Mode after receiving a STOP condition from the master.

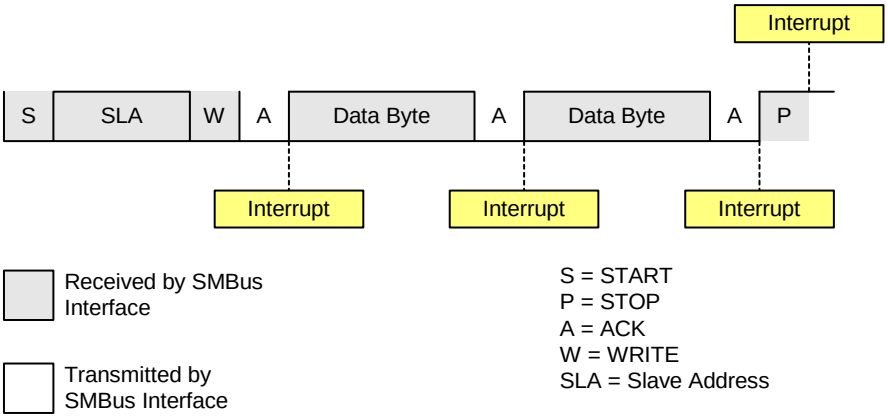



**Figure 19.7. Typical Slave Receiver Sequence**

## 19.4. SMBus Special Function Registers

The SMBus0 serial interface is accessed and controlled through five SFR's: SMB0CN Control Register, SMB0CR Clock Rate Register, SMB0ADR Address Register, SMB0DAT Data Register and SMB0STA Status Register. The five special function registers related to the operation of the SMBus0 interface are described in the following sections.

### 19.4.1. Control Register

The SMBus0 Control register SMB0CN is used to configure and control the SMBus0 interface. All of the bits in the register can be read or written by software. Two of the control bits are also affected by the SMBus0 hardware. The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by the hardware when a valid serial interrupt condition occurs. It can only be cleared by software. The Stop flag (STO, SMB0CN.4) is set to logic 1 by software. It is cleared to logic 0 by hardware when a STOP condition is detected on the bus.

Setting the ENSMB flag to logic 1 enables the SMBus0 interface. Clearing the ENSMB flag to logic 0 disables the SMBus0 interface and removes it from the bus. Momentarily clearing the ENSMB flag and then resetting it to logic 1 will reset SMBus0 communication. However, ENSMB should not be used to temporarily remove a device from the bus since the bus state information will be lost. Instead, the Assert Acknowledge (AA) flag should be used to temporarily remove the device from the bus (see description of AA flag below).

Setting the Start flag (STA, SMB0CN.5) to logic 1 will put SMBus0 in a master mode. If the bus is free, SMBus0 will generate a START condition. If the bus is not free, SMBus0 waits for a STOP condition to free the bus and then generates a START condition after a 5 µs delay per the SMB0CR value (In accordance with the SMBus protocol, the SMBus0 interface also considers the bus free if the bus is idle for 50 µs and no STOP condition was recognized). If STA is set to logic 1 while SMBus0 is in master mode and one or more bytes have been transferred, a repeated START condition will be generated.

When the Stop flag (STO, SMB0CN.4) is set to logic 1 while the SMBus0 interface is in master mode, the interface generates a STOP condition. In a slave mode, the STO flag may be used to recover from an error condition. In this case, a STOP condition is not generated on the bus, but the SMBus hardware behaves as if a STOP condition has been received and enters the "not addressed" slave receiver mode. Note that this simulated STOP will not cause the bus to appear free to SMBus0. The bus will remain occupied until a STOP appears on the bus or a Bus Free Timeout occurs. Hardware automatically clears the STO flag to logic 0 when a STOP condition is detected on the bus.

The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by hardware when the SMBus0 interface enters one of 27 possible states. If interrupts are enabled for the SMBus0 interface, an interrupt request is generated when the SI flag is set. The SI flag must be cleared by software.

**Important Note:** If SI is set to logic 1 while the SCL line is low, the clock-low period of the serial clock will be stretched and the serial transfer is suspended until SI is cleared to logic 0. A high level on SCL is not affected by the setting of the SI flag.

The Assert Acknowledge flag (AA, SMB0CN.2) is used to set the level of the SDA line during the acknowledge clock cycle on the SCL line. Setting the AA flag to logic 1 will cause an ACK (low level on SDA) to be sent during the acknowledge cycle if the device has been addressed. Setting the AA flag to logic 0 will cause a NACK (high level on SDA) to be sent during acknowledge cycle. After the transmission of a byte in slave mode, the slave can be temporarily removed from the bus by clearing the AA flag. The slave's own address and general call address will be ignored. To resume operation on the bus, the AA flag must be reset to logic 1 to allow the slave's address to be recognized.

Setting the SMBus0 Free Timer Enable bit (FTE, SMB0CN.1) to logic 1 enables the timer in SMB0CR. When SCL goes high, the timer in SMB0CR counts up. A timer overflow indicates a free bus timeout: if SMBus0 is waiting to generate a START, it will do so after this timeout. The bus free period should be less than 50 µs (see SFR Definition 19.2, SMBus0 Clock Rate Register).

When the TOE bit in SMB0CN is set to logic 1, Timer 3 is used to detect SCL low timeouts. If Timer 3 is enabled (see **Section “23.2. Timer 2, Timer 3, and Timer 4” on page 317**), Timer 3 is forced to reload when SCL is high, and forced to count when SCL is low. With Timer 3 enabled and configured to overflow after 25 ms (and TOE set), a Timer 3 overflow indicates a SCL low timeout; the Timer 3 interrupt service routine can then be used to reset SMBus0 communication in the event of an SCL low timeout.

## SFR Definition 19.1. SMB0CN: SMBus0 Control

| R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| BUSY | ENSMB | STA | STO | SI | AA | FTE | TOE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xC0
SFR Page: 0

Bit7: BUSY: Busy Status Flag.
0: SMBus0 is free
1: SMBus0 is busy

Bit6: ENSMB: SMBus Enable.
This bit enables/disables the SMBus serial interface.
0: SMBus0 disabled.
1: SMBus0 enabled.

Bit5: STA: SMBus Start Flag.
0: No START condition is transmitted.
1: When operating as a master, a START condition is transmitted if the bus is free. (If the bus is not free, the START is transmitted after a STOP is received.) If STA is set after one or more bytes have been transmitted or received and before a STOP is received, a repeated START condition is transmitted.

Bit4: STO: SMBus Stop Flag.
0: No STOP condition is transmitted.
1: Setting STO to logic 1 causes a STOP condition to be transmitted. When a STOP condition is received, hardware clears STO to logic 0. If both STA and STO are set, a STOP condition is transmitted followed by a START condition. In slave mode, setting the STO flag causes SMBus to behave as if a STOP condition was received.

Bit3: SI: SMBus Serial Interrupt Flag.
This bit is set by hardware when one of 27 possible SMBus0 states is entered. (Status code 0xF8 does not cause SI to be set.) When the SI interrupt is enabled, setting this bit causes the CPU to vector to the SMBus interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: AA: SMBus Assert Acknowledge Flag.
This bit defines the type of acknowledge returned during the acknowledge cycle on the SCL line.
0: A "not acknowledge" (high level on SDA) is returned during the acknowledge cycle.
1: An "acknowledge" (low level on SDA) is returned during the acknowledge cycle.

Bit1: FTE: SMBus Free Timer Enable Bit
0: No timeout when SCL is high
1: Timeout when SCL high time exceeds limit specified by the SMB0CR value.

Bit0: TOE: SMBus Timeout Enable Bit
0: No timeout when SCL is low.
1: Timeout when SCL low time exceeds limit specified by Timer 3, if enabled.

### 19.4.2. Clock Rate Register

**SFR Definition 19.2. SMB0CR: SMBus0 Clock Rate**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xCF
SFR Page: 0

Bits7–0: SMB0CR.[7:0]: SMBus0 Clock Rate Preset
The SMB0CR Clock Rate register controls the frequency of the serial clock SCL in master mode. The 8-bit word stored in the SMB0CR Register preloads a dedicated 8-bit timer. The timer counts up, and when it rolls over to 0x00, the SCL logic state toggles.

The SMB0CR setting should be bounded by the following equation , where *SMB0CR* is the unsigned 8-bit value in register SMB0CR, and *SYSCLK* is the system clock frequency in MHz:

$$SMB0CR < \left(288 - 0.85 \cdot \frac{SYSCLK}{4}\right) / 1.125$$

The resulting SCL signal high and low times are given by the following equations, where SYSCLK is the system clock frequency in Hz:

$$T_{LOW} = 4 \times (256 - SMB0CR) / SYSCLK$$

$$T_{HIGH} \cong 4 \times (258 - SMB0CR) / SYSCLK + 625ns$$

Using the same value of SMB0CR from above, the Bus Free Timeout period is given in the following equation:

$$T_{BFT} \cong 10 \times \frac{4 \times (256 - SMB0CR) + 1}{SYSCLK}$$

### 19.4.3. Data Register

The SMBus0 Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software can read or write to this register while the SI flag is set to logic 1; software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag reads logic 0 since the hardware may be in the process of shifting a byte of data in 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. Therefore, 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 in SMB0DAT.

**SFR Definition 19.3. SMB0DAT: SMBus0 Data**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC2
SFR Page: 0

Bits7–0: SMB0DAT: SMBus0 Data.
The SMB0DAT register contains a byte of data to be transmitted on the SMBus0 serial interface or a byte that has just been received on the SMBus0 serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.3) is set to logic 1. When the SI flag is not set, the system may be in the process of shifting data and the CPU should not attempt to access this register.

### 19.4.4. Address Register

The SMB0ADR Address register holds the slave address for the SMBus0 interface. In slave mode, the seven most-significant bits hold the 7-bit slave address. The least significant bit (Bit0) is used to enable the recognition of the general call address (0x00). If Bit0 is set to logic 1, the general call address will be recognized. Otherwise, the general call address is ignored. The contents of this register are ignored when SMBus0 is operating in master mode.

### SFR Definition 19.4. SMB0ADR: SMBus0 Address

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SLV6 | SLV5 | SLV4 | SLV3 | SLV2 | SLV1 | SLV0 | GC | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC3
SFR Page: 0

Bits7–1: SLV6–SLV0: SMBus0 Slave Address.
These bits are loaded with the 7-bit slave address to which SMBus0 will respond when operating as a slave transmitter or slave receiver. SLV6 is the most significant bit of the address and corresponds to the first bit of the address byte received.

Bit0: GC: General Call Address Enable.
This bit is used to enable general call address (0x00) recognition.
0: General call address is ignored.
1: General call address is recognized.

### 19.4.5. Status Register

The SMB0STA Status register holds an 8-bit status code indicating the current state of the SMBus0 interface. There are 28 possible SMBus0 states, each with a corresponding unique status code. The five most significant bits of the status code vary while the three least-significant bits of a valid status code are fixed at zero when SI = '1'. Therefore, all possible status codes are multiples of eight. This facilitates the use of status codes in software as an index used to branch to appropriate service routines (allowing 8 bytes of code to service the state or jump to a more extensive service routine).

For the purposes of user software, the contents of the SMB0STA register is only defined when the SI flag is logic 1. Software should never write to the SMB0STA register; doing so will yield indeterminate results. The 28 SMBus0 states, along with their corresponding status codes, are given in Table 1.1.

### SFR Definition 19.5. SMB0STA: SMBus0 Status

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| STA7 | STA6 | STA5 | STA4 | STA3 | STA2 | STA1 | STA0 | 11111000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC1
SFR Page: 0

Bits7–3: STA7–STA3: SMBus0 Status Code.
These bits contain the SMBus0 Status Code. There are 28 possible status codes; each status code corresponds to a single SMBus state. A valid status code is present in SMB0STA when the SI flag (SMB0CN.3) is set to logic 1. The content of SMB0STA is not defined when the SI flag is logic 0. Writing to the SMB0STA register at any time will yield indeterminate results.

Bits2–0: STA2–STA0: The three least significant bits of SMB0STA are always read as logic 0 when the SI flag is logic 1.

**Table 19.1. SMB0STA Status Codes and States**

| Mode | Status Code | SMBus State | Typical Action |
|---|---|---|---|
| MT/ MR | 0x08 | START condition transmitted. | Load SMB0DAT with Slave Address + R/W. Clear STA. |
| | 0x10 | Repeated START condition transmitted. | Load SMB0DAT with Slave Address + R/W. Clear STA. |
| Master Transmitter | 0x18 | Slave Address + W transmitted. ACK received. | Load SMB0DAT with data to be transmitted. |
| | 0x20 | Slave Address + W transmitted. NACK received. | Acknowledge poll to retry. Set STO + STA. |
| | 0x28 | Data byte transmitted. ACK received. | 1) Load SMB0DAT with next byte, OR<br>2) Set STO, OR<br>3) Clear STO then set STA for repeated START. |
| | 0x30 | Data byte transmitted. NACK received. | 1) Retry transfer OR<br>2) Set STO. |
| | 0x38 | Arbitration Lost. | Save current data. |
| Master Receiver | 0x40 | Slave Address + R transmitted. ACK received. | If only receiving one byte, clear AA (send NACK after received byte). Wait for received data. |
| | 0x48 | Slave Address + R transmitted. NACK received. | Acknowledge poll to retry. Set STO + STA. |
| | 0x50 | Data byte received. ACK transmitted. | Read SMB0DAT. Wait for next byte. If next byte is last byte, clear AA. |
| | 0x58 | Data byte received. NACK transmitted. | Set STO. |

**Table 19.1. SMB0STA Status Codes and States (Continued)**

| Mode | Status Code | SMBus State | Typical Action |
|---|---|---|---|
| Slave Receiver | 0x60 | Own slave address + W received. ACK transmitted. | Wait for data. |
| | 0x68 | Arbitration lost in sending SLA + R/W as master. Own address + W received. ACK transmitted. | Save current data for retry when bus is free. Wait for data. |
| | 0x70 | General call address received. ACK transmitted. | Wait for data. |
| | 0x78 | Arbitration lost in sending SLA + R/W as master. General call address received. ACK transmitted. | Save current data for retry when bus is free. |
| | 0x80 | Data byte received. ACK transmitted. | Read SMB0DAT. Wait for next byte or STOP. |
| | 0x88 | Data byte received. NACK transmitted. | Set STO to reset SMBus. |
| | 0x90 | Data byte received after general call address. ACK transmitted. | Read SMB0DAT. Wait for next byte or STOP. |
| | 0x98 | Data byte received after general call address. NACK transmitted. | Set STO to reset SMBus. |
| | 0xA0 | STOP or repeated START received. | No action necessary. |
| Slave Transmitter | 0xA8 | Own address + R received. ACK transmitted. | Load SMB0DAT with data to transmit. |
| | 0xB0 | Arbitration lost in transmitting SLA + R/W as master. Own address + R received. ACK transmitted. | Save current data for retry when bus is free. Load SMB0DAT with data to transmit. |
| | 0xB8 | Data byte transmitted. ACK received. | Load SMB0DAT with data to transmit. |
| | 0xC0 | Data byte transmitted. NACK received. | Wait for STOP. |
| | 0xC8 | Last data byte transmitted (AA=0). ACK received. | Set STO to reset SMBus. |
| Slave | 0xD0 | SCL Clock High Timer per SMB0CR timed out | Set STO to reset SMBus. |
| All | 0x00 | Bus Error (illegal START or STOP) | Set STO to reset SMBus. |
| | 0xF8 | Idle | State does not set SI. |

## NOTES:

# 20. 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.

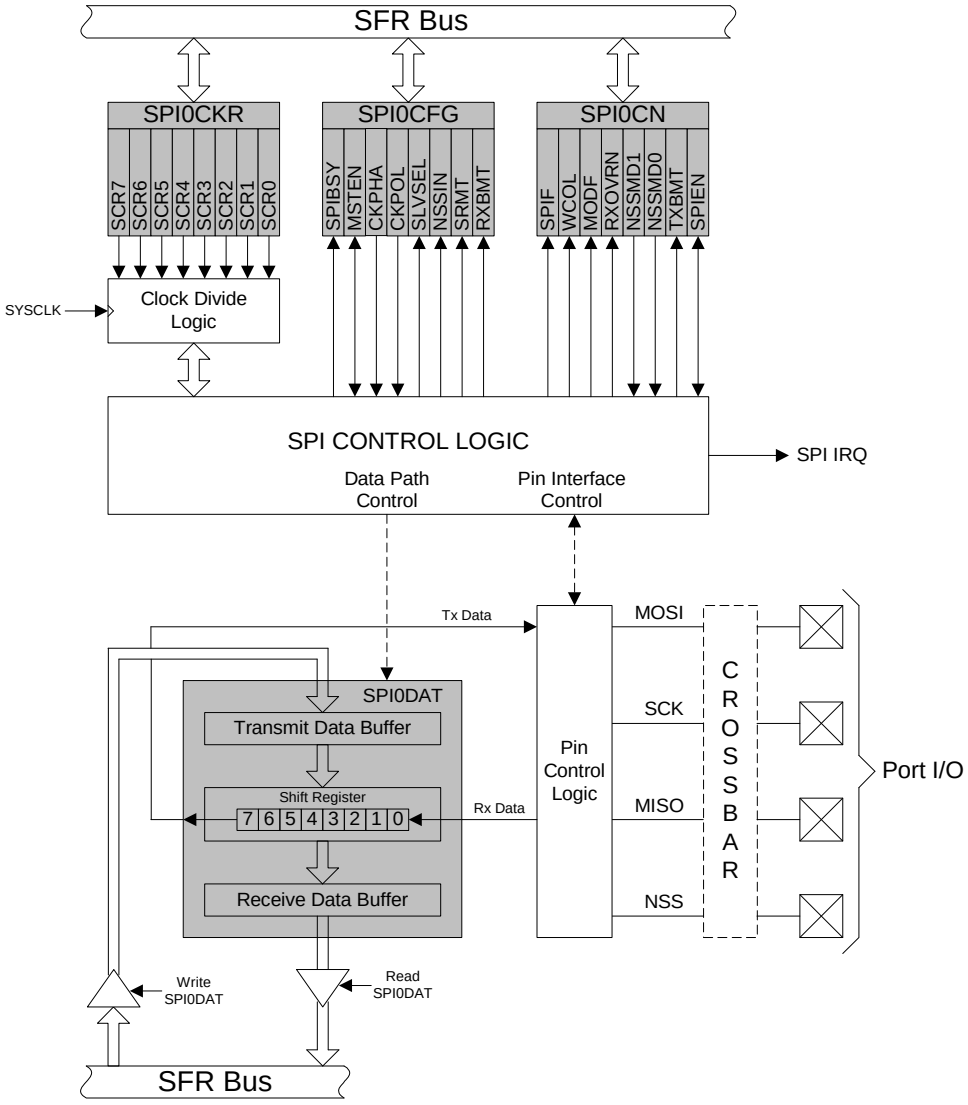



**Figure 20.1. SPI Block Diagram**

## 20.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 20.1.1. 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.

### 20.1.2. 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.

### 20.1.3. 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.

### 20.1.4. 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 20.2, Figure 20.3, and Figure 20.4 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 "**18. Port Input/Output**" on page **235** for general purpose port I/O and crossbar information.

## 20.2. 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 20.2 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 20.3 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 20.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

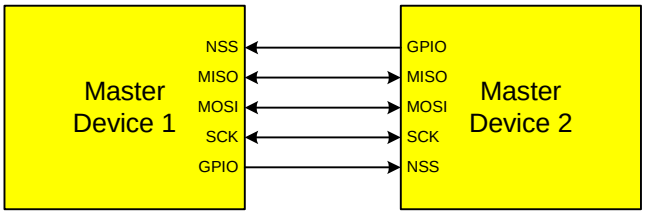



**Figure 20.2. Multiple-Master Mode Connection Diagram**

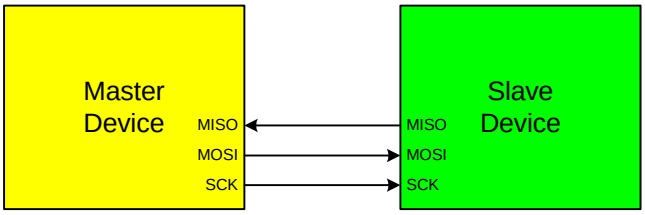



**Figure 20.3. 3-Wire Single Master and Slave Mode Connection Diagram**

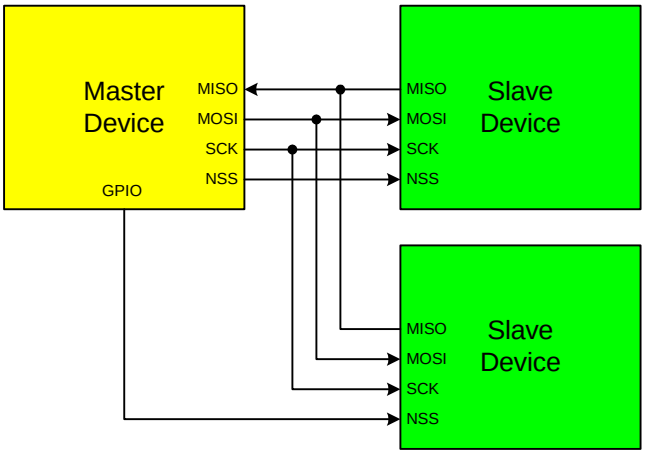



**Figure 20.4. 4-Wire Single Master and Slave Mode Connection Diagram**

## 20.3. 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 20.4 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 20.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

## 20.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

*Note that all of the following bits must be cleared by software.*

1. 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.
2. 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.
3. 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.
4. 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.

## 20.5. Serial Clock Timing

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 20.5. For slave mode, the clock and data relationships are shown in Figure 20.6 and Figure 20.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x/13x, C8051F31x, C8051F32x, and C8051F33x

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 20.3 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.

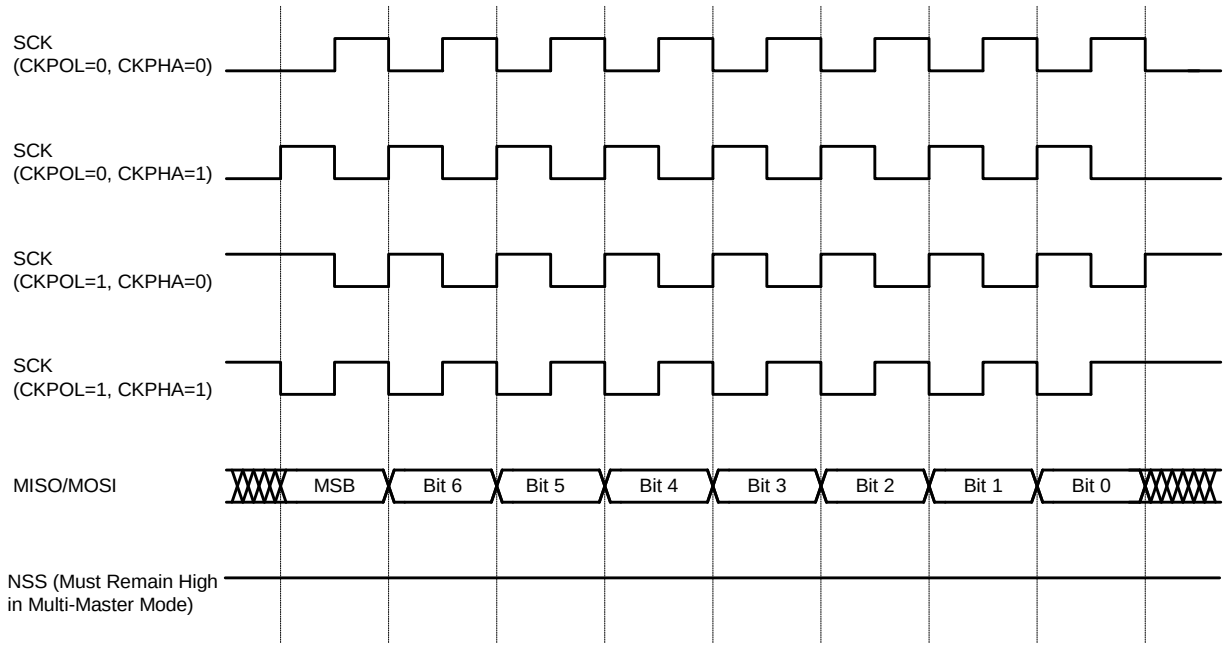



**Figure 20.5. Master Mode Data/Clock Timing**

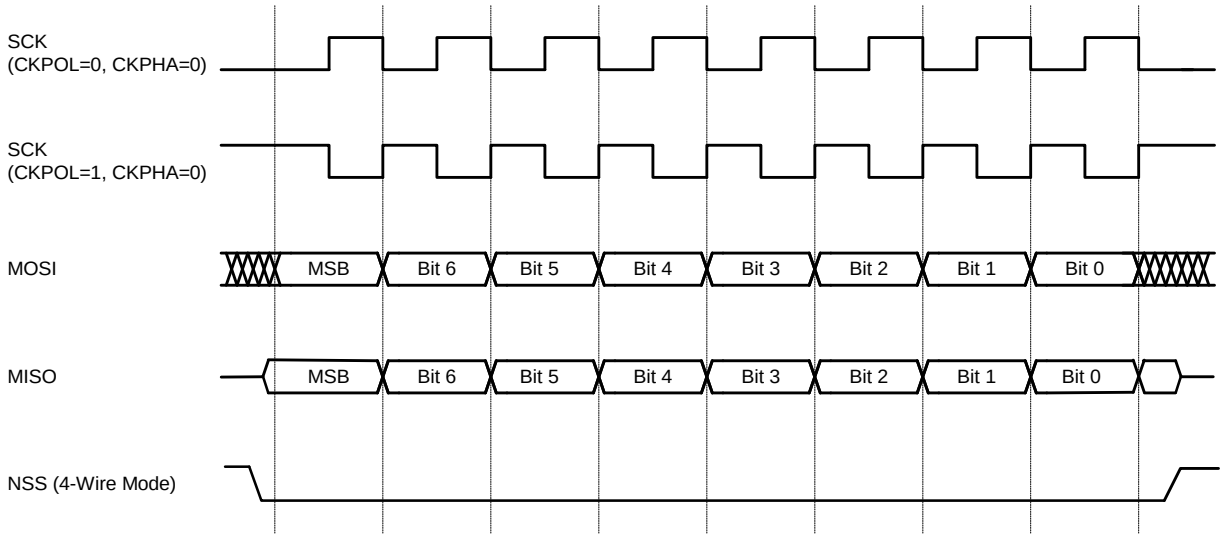



**Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)**

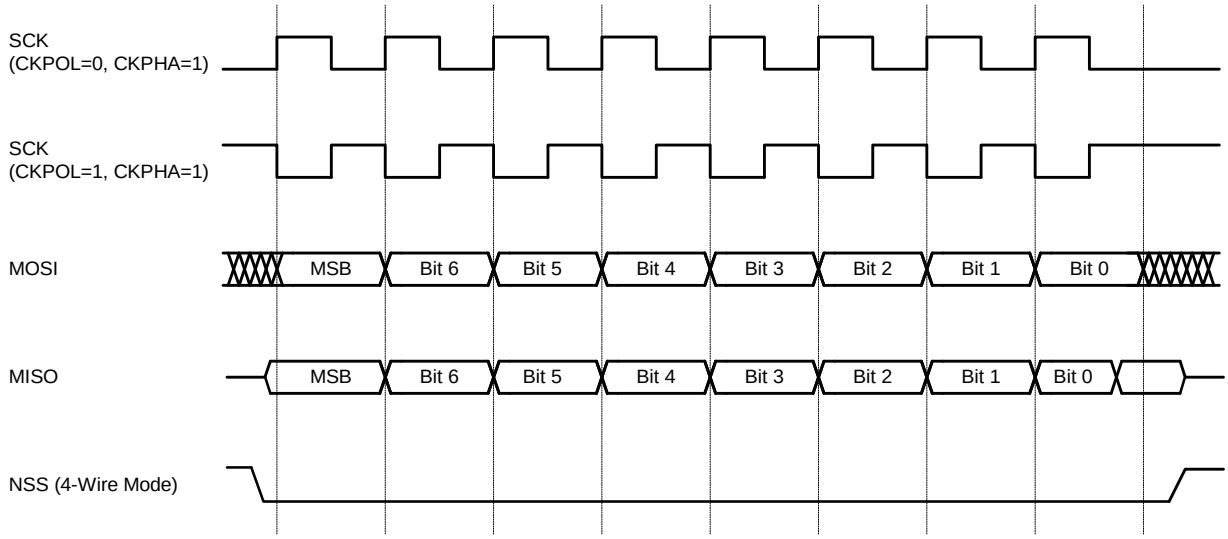



**Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)**

## 20.6. 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 20.1. SPI0CFG: SPI0 Configuration

| R | R/W | R/W | R/W | R | R | R | R | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SPIBSY | MSTEN | CKPHA | CKPOL | SLVSEL | NSSIN | SRMT | RXBMT | 00000111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9A
SFR Page: 0

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*

Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: SLVSEL: Slave Selected Flag (read only).
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.

Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
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.

Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read 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.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read 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.
NOTE: 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 20.1 for timing parameters.

## SFR Definition 20.2. SPI0CN: SPI0 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SPIF | WCOL | MODF | RXOVRN | NSSMD1 | NSSMD0 | TXBMT | SPIEN | 00000110 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xF8
SFR Page: 0

Bit 7: SPIF: SPI0 Interrupt Flag.
This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 6: WCOL: Write Collision Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. It must be cleared by software.

Bit 5: MODF: Mode Fault Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 4: RXOVRN: Receive Overrun Flag (Slave Mode only).
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) 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. This bit is not automatically cleared by hardware. It must be cleared by software.

Bits 3–2: NSSMD1–NSSMD0: Slave Select Mode.
Selects between the following NSS operation modes:
(See **Section “20.2. SPI0 Master Mode Operation” on page 275** and **Section “20.3. SPI0 Slave Mode Operation” on page 277**).
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 always 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.

Bit 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.

Bit 0: SPIEN: SPI0 Enable.
This bit enables/disables the SPI.
0: SPI disabled.
1: SPI enabled.

### SFR Definition 20.3. SPI0CKR: SPI0 Clock Rate

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SCR7 | SCR6 | SCR5 | SCR4 | SCR3 | SCR2 | SCR1 | SCR0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x9D
SFR Page: 0

Bits 7–0: SCR7–SCR0: 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.

$$f_{SCK} = \frac{\text{SYSCLK}}{2 \times (\text{SPI0CKR} + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

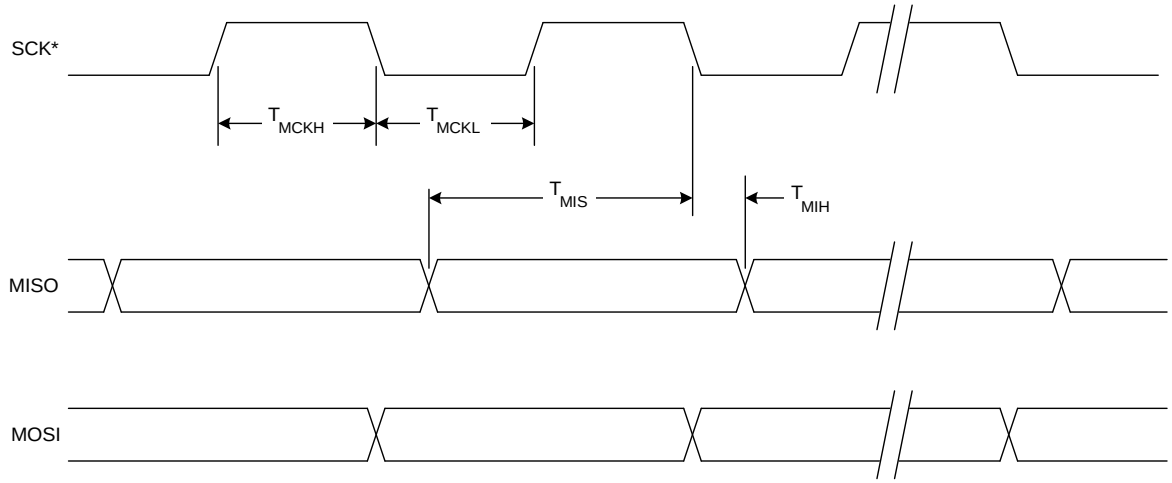
$$f_{SCK} = 200kHz$$

### SFR Definition 20.4. SPI0DAT: SPI0 Data

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

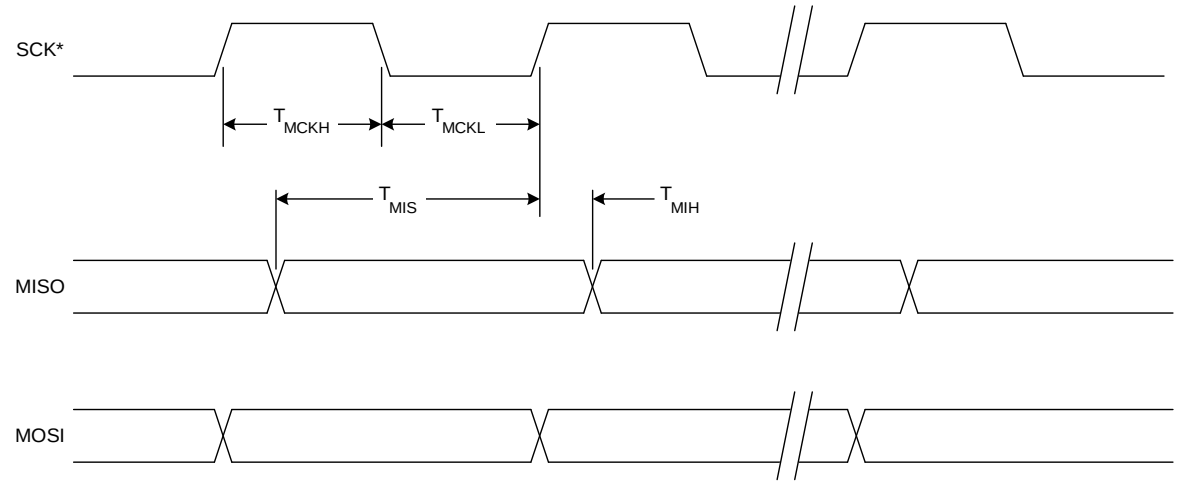
SFR Address: 0x9B
SFR Page: 0

Bits 7–0: SPI0DAT: 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.



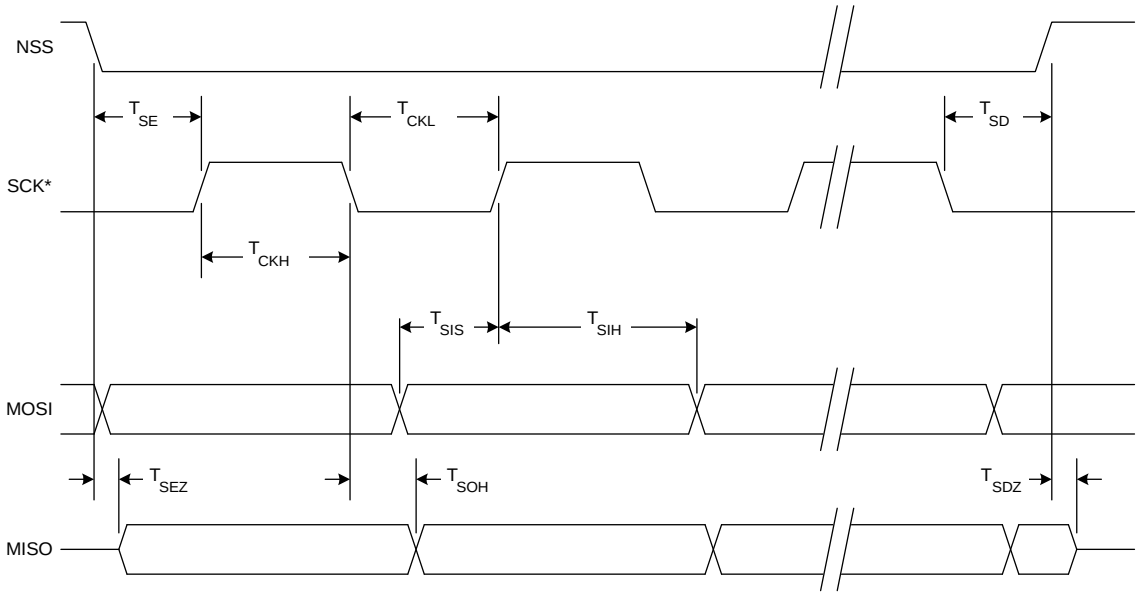

* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 20.8. SPI Master Timing (CKPHA = 0)**



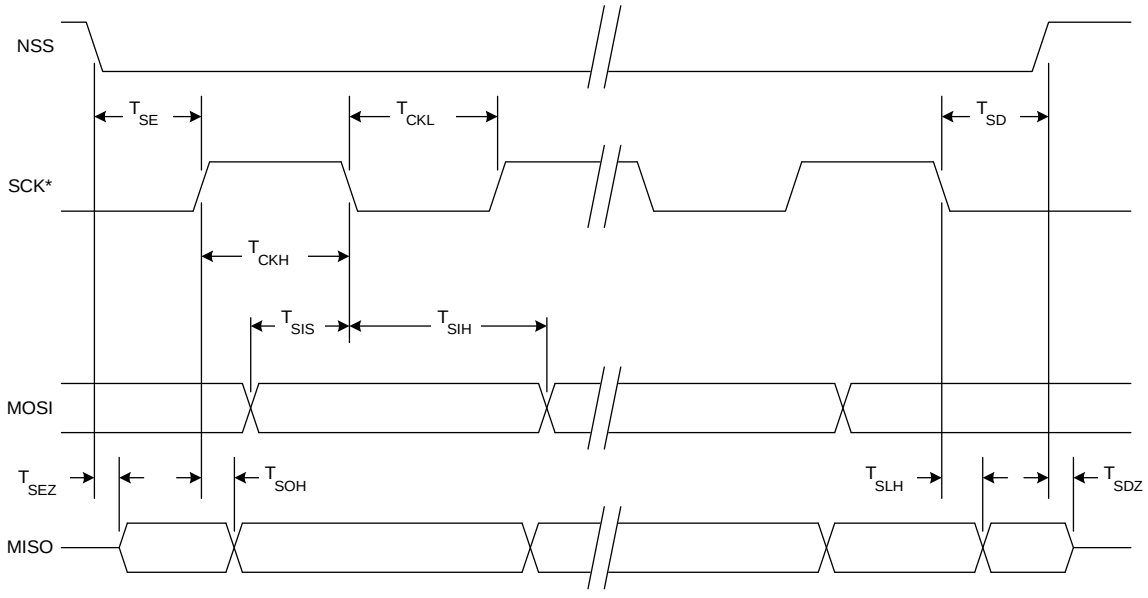

* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 20.9. SPI Master Timing (CKPHA = 1)**




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 20.10. SPI Slave Timing (CKPHA = 0)**




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 20.11. SPI Slave Timing (CKPHA = 1)**

**Table 20.1. SPI Slave Timing Parameters**

| Parameter | Description | Min | Max | Units |
|---|---|---|---|---|
| **Master Mode Timing*** (See Figure 20.8 and Figure 20.9) | | | | |
| $T_{MCKH}$ | 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 20.10 and Figure 20.11) | | | | |
| $T_{SE}$ | 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).

## NOTES:

# 21. UART0

UART0 is an enhanced serial port with frame error detection and address recognition hardware. UART0 may operate in full-duplex asynchronous or half-duplex synchronous modes, and mutiproccessor communication is fully supported. Receive data is buffered in a holding register, allowing UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte. A Receive Overrun bit indicates when new received data is latched into the receive buffer before the previously received byte has been read.

UART0 is accessed via its associated SFR's, Serial Control (SCON0) and Serial Data Buffer (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. Reading SCON0 accesses the Receive register and writing SCON0 accesses the Transmit register.

UART0 may be operated in polled or interrupt mode. UART0 has two sources of interrupts: a Transmit Interrupt flag, TI0 (SCON0.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI0 (SCON0.0) set when reception of a data byte is complete. UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine; they must be cleared manually by software. This allows software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

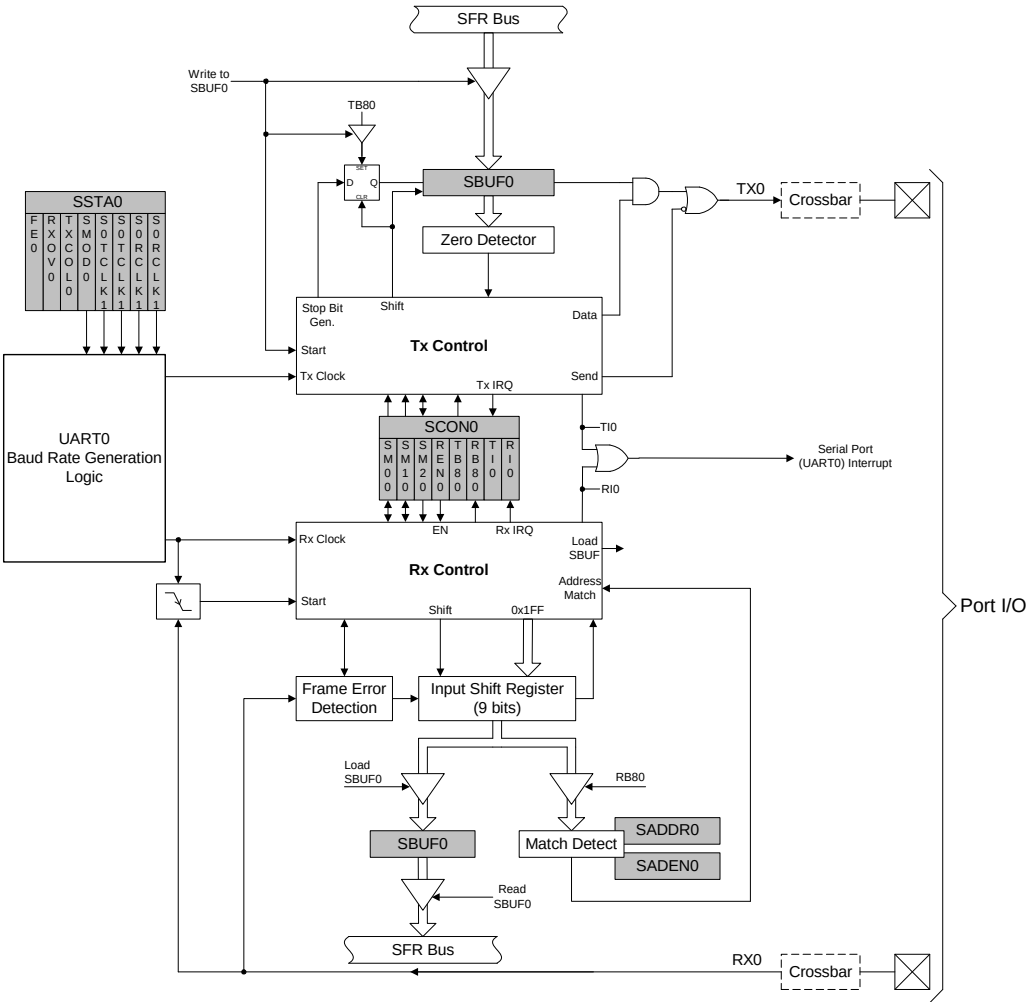



**Figure 21.1. UART0 Block Diagram**

## 21.1. UART0 Operational Modes

UART0 provides four operating modes (one synchronous and three asynchronous) selected by setting configuration bits in the SCON0 register. These four modes offer different baud rates and communication protocols. The four modes are summarized in Table 21.1.

**Table 21.1. UART0 Modes**

| Mode | Synchronization | Baud Clock | Data Bits | Start/Stop Bits |
|---|---|---|---|---|
| 0 | Synchronous | SYSCLK / 12 | 8 | None |
| 1 | Asynchronous | Timer 1, 2, 3, or 4 Overflow | 8 | 1 Start, 1 Stop |
| 2 | Asynchronous | SYSCLK / 32 or SYSCLK / 64 | 9 | 1 Start, 1 Stop |
| 3 | Asynchronous | Timer 1, 2, 3, or 4 Overflow | 9 | 1 Start, 1 Stop |

### 21.1.1. Mode 0: Synchronous Mode

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX0 pin. The TX0 pin provides the shift clock for both transmit and receive. The MCU must be the master since it generates the shift clock for transmission in both directions (see the interconnect diagram in Figure 21.3).

Data transmission begins when an instruction writes a data byte to the SBUF0 register. Eight data bits are transferred LSB first (see the timing diagram in Figure 21.2), and the TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the eighth bit time. Data reception begins when the REN0 Receive Enable bit (SCON0.4) is set to logic 1 and the RI0 Receive Interrupt Flag (SCON0.0) is cleared. One cycle after the eighth bit is shifted in, the RI0 flag is set and reception stops until software clears the RI0 bit. An interrupt will occur if enabled when either TI0 or RI0 are set.

The Mode 0 baud rate is SYSCLK / 12. RX0 is forced to open-drain in Mode 0, and an external pullup will typically be required.

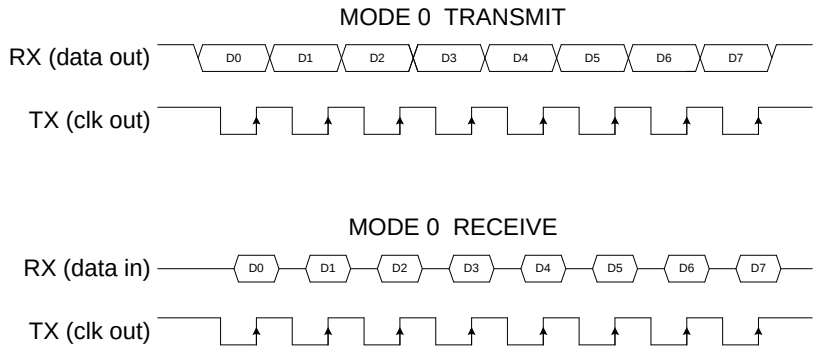



**Figure 21.2. UART0 Mode 0 Timing Diagram**

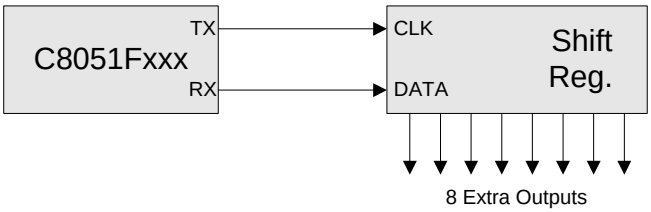



**Figure 21.3. UART0 Mode 0 Interconnect**

### 21.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

Mode 1 provides standard asynchronous, full duplex communication using a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted 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 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 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 SM20 is logic 1, the stop bit must be logic 1.

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 are set.

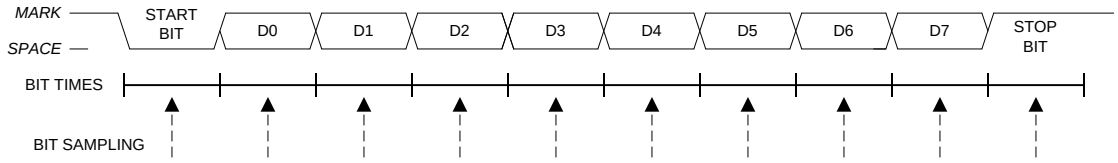



**Figure 21.4. UART0 Mode 1 Timing Diagram**

The baud rate generated in Mode 1 is a function of timer overflow. UART0 can use Timer 1 operating in *8-Bit Auto-Reload Mode*, or Timer 2, 3, or 4 operating in *Auto-reload Mode* to generate the baud rate (note that the TX and RX clocks are selected separately). On each timer overflow event (a rollover from all ones - (0xFF for Timer 1, 0xFFFF for Timer 2, 3, or 4) - to zero) a clock is sent to the baud rate logic.

Timers 1, 2, 3, or 4 are selected as the baud rate source with bits in the SSTA0 register (see SFR Definition 21.2). The transmit baud rate clock is selected using the S0TCLK1 and S0TCLK0 bits, and the receive baud rate clock is selected using the S0RCLK1 and S0RCLK0 bits.

When Timer 1 is selected as a baud rate source, the SMOD0 bit (SSTA0.4) selects whether or not to divide the Timer 1 overflow rate by two. On reset, the SMOD0 bit is logic 0, thus selecting the lower speed baud rate by default. The SMOD0 bit affects the baud rate generated by Timer 1 as shown in Equation 21.1.

The Mode 1 baud rate equations are shown below, where T1M is bit4 of register CKCON, TH1 is the 8-bit reload register for Timer 1, and [RCAPnH , RCAPnL] is the 16-bit reload register for Timer 2, 3, or 4.

**Equation 21.1. Mode 1 Baud Rate using Timer 1**

When SMOD0 = 0:

$$\text{Mode1_BaudRate} = 1/32 \cdot \text{Timer1_OverflowRate}$$

When SMOD0 = 1:

$$\text{Mode1_BaudRate} = 1/16 \cdot \text{Timer1_OverflowRate}$$

The Timer 1 overflow rate is determined by the Timer 1 clock source (T1CLK) and reload value (TH1). The frequency of T1CLK is selected as described in **Section "23.1. Timer 0 and Timer 1" on page 309**. The Timer 1 overflow rate is calculated as shown in Equation 21.2.

### Equation 21.2. Timer 1 Overflow Rate

$$\text{Timer1_OverflowRate} = \text{T1CLK} / (256 - \text{TH1})$$

When Timers 2, 3, or 4 are selected as a baud rate source, the baud rate is generated as shown in Equation 21.3.

### Equation 21.3. Mode 1 Baud Rate using Timer 2, 3, or 4

$$\text{Mode1_BaudRate} = 1/16 \cdot \text{Timer234_OverflowRate}$$

The overflow rate for Timer 2, 3, or 4 is determined by the clock source for the timer (TnCLK) and the 16-bit reload value stored in the RCAPn register (n = 2, 3, or 4), as shown in Equation 21.4.

### Equation 21.4. Timer 2, 3, or 4 Overflow Rate

$$\text{Timer234_OverflowRate} = \text{TnCLK} / (65536 - \text{RCAPn})$$

### 21.1.3. Mode 2: 9-Bit UART, Fixed Baud Rate

Mode 2 provides asynchronous, full-duplex communication using 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. Mode 2 supports multiprocessor communications and hardware address recognition (see **Section 21.2**). On transmit, the ninth data bit is determined by the value in TB80 (SCON0.3). It can be assigned the value of the parity flag P in the PSW 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 logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if RI0 is logic 0 and one of the following requirements are met:

1. SM20 is logic 0
2. SM20 is logic 1, the received 9th bit is logic 1, and the received address matches the UART0 address as described in **Section 21.2**.

If the above conditions are satisfied, the eight bits of data are stored in SBUF0, the ninth 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 are set.

The baud rate in Mode 2 is either SYSCLK / 32 or SYSCLK / 64, according to the value of the SMOD0 bit in register SSTA0.

**Equation 21.5. Mode 2 Baud Rate**

$$BaudRate = 2^{SMOD0} \times \left(\frac{SYSCLK}{64}\right)$$

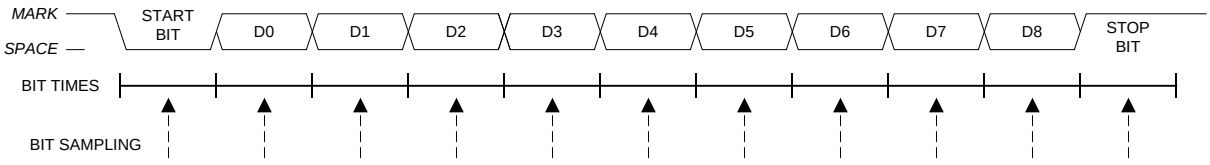



**Figure 21.5. UART0 Modes 2 and 3 Timing Diagram**

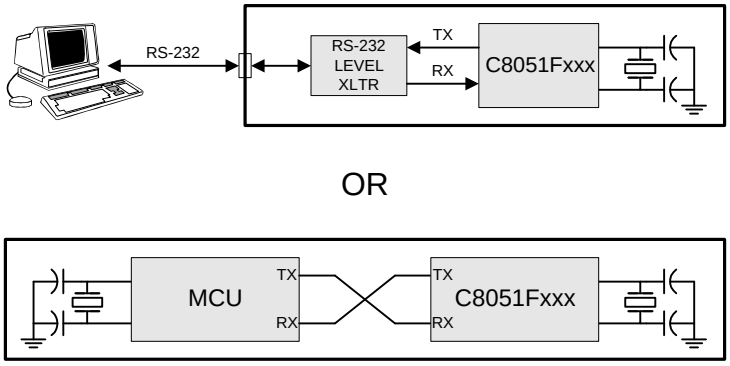



**Figure 21.6. UART0 Modes 1, 2, and 3 Interconnect Diagram**

### 21.1.4. Mode 3: 9-Bit UART, Variable Baud Rate

Mode 3 uses the Mode 2 transmission protocol with the Mode 1 baud rate generation. Mode 3 operation transmits 11 bits: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The baud rate is derived from Timer 1 or Timer 2, 3, or 4 overflows, as defined by Equation 21.1 and Equation 21.3. Multiprocessor communications and hardware address recognition are supported, as described in **Section 21.2**.

## 21.2. Multiprocessor Communications

Modes 2 and 3 support multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit and the built-in UART0 address recognition hardware. 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. UART0 will recognize as “valid” (i.e., capable of causing an interrupt) **two** types of addresses: (1) a *masked* address and (2) a *broadcast* address **at any given time**. Both are described below.

### 21.2.1. Configuration of a Masked Address

The UART0 address is configured via two SFR’s: SADDR0 (Serial Address) and SADEN0 (Serial Address Enable). SADEN0 sets the bit mask for the address held in SADDR0: bits set to logic 1 in SADEN0 correspond to bits in SADDR0 that are checked against the received address byte; bits set to logic 0 in SADEN0 correspond to “don’t care” bits in SADDR0.

| **Example 1, SLAVE #1** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 00001111 |
| UART0 Address | = xxxx0101 |

| **Example 2, SLAVE #2** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 11110011 |
| UART0 Address | = 0011xx01 |

| **Example 3, SLAVE #3** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 11000000 |
| UART0 Address | = 00xxxxxx |

Setting the SM20 bit (SCON0.5) configures UART0 such that when a stop bit is received, UART0 will generate an interrupt only if the ninth bit is logic 1 (RB80 = ‘1’) and the received data byte matches the UART0 slave address. Following the received address interrupt, the slave will clear its SM20 bit to enable interrupts on the reception of the following data byte(s). Once the entire message is received, the addressed slave resets its SM20 bit to ignore all transmissions until it receives the next address byte. While SM20 is logic 1, UART0 ignores all bytes that do not match the UART0 address and include a ninth bit that is logic 1.

### 21.2.2. Broadcast Addressing

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 broadcast address is the logical OR of registers SADDR0 and SADEN0, and ‘0’s of the result are treated as “don’t cares”. Typically a broadcast address of 0xFF (hexadecimal) is acknowledged by all slaves, assuming “don’t care” bits as ‘1’s. 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)..

| **Example 4, SLAVE #1** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 00001111 |
| Broadcast Address | = 00111111 |

| **Example 5, SLAVE #2** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 11110011 |
| Broadcast Address | = 11110111 |

| **Example 6, SLAVE #3** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 11000000 |
| Broadcast Address | = 11110101 |

Where all ZEROES in the Broadcast address are don’t cares.

Note in the above examples 4, 5, and 6, each slave would recognize as “valid” an address of 0xFF as a broadcast address. Also note that examples 4, 5, and 6 uses the same SADDR0 and SADEN0 register values as shown in the examples 1, 2, and 3 respectively (slaves #1, 2, and 3). Thus, a master could address each slave device individually using a masked address, and also broadcast to all three slave devices. For example, if a Master were to send an address “11110101”, only slave #1 would recognize the address as valid. If a master were to then send an address of “11111111”, all three slave devices would recognize the address as a valid broadcast address.

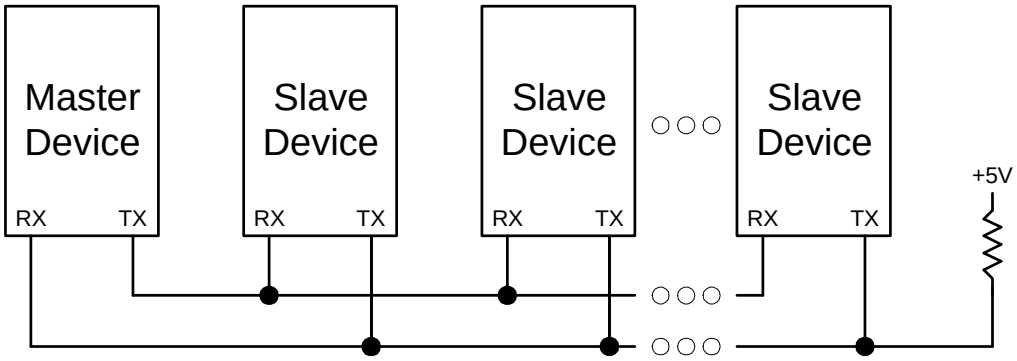



**Figure 21.7. UART Multi-Processor Mode Interconnect Diagram**

## 21.3. Frame and Transmission Error Detection

**All Modes:**
The Transmit Collision bit (TXCOL0 bit in register SSTA0) reads ‘1’ if user software writes data to the SBUF0 register while a transmit is in progress.

**Modes 1, 2, and 3:**
The Receive Overrun bit (RXOV0 in register SSTA0) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. The Frame Error bit (FE0 in register SSTA0) reads ‘1’ if an invalid (low) STOP bit is detected.

**Table 21.2. Oscillator Frequencies for Standard Baud Rates**

| System Clock Frequency (MHz) | Divide Factor | Timer 1 Reload Value[1] | Timer 2, 3, or 4 Reload Value | Resulting Baud Rate (Hz)[2] |
|---|---|---|---|---|
| 100.0 | 864 | 0xCA | 0xFFCA | 115200 (115741) |
| 99.5328 | 864 | 0xCA | 0xFFCA | 115200 |
| 50.0 | 432 | 0xE5 | 0xFFE5 | 115200 (115741) |
| 49.7664 | 432 | 0xE5 | 0xFFE5 | 115200 |
| 24.0 | 208 | 0xF3 | 0xFFF3 | 115200 (115384) |
| 22.1184 | 192 | 0xF4 | 0xFFF4 | 115200 |
| 18.432 | 160 | 0xF6 | 0xFFF6 | 115200 |
| 11.0592 | 96 | 0xFA | 0xFFFA | 115200 |
| 3.6864 | 32 | 0xFE | 0xFFFE | 115200 |
| 1.8432 | 16 | 0xFF | 0xFFFF | 115200 |
| 100.0 | 3472 | 0x27 | 0xFF27 | 28800 (28802) |
| 99.5328 | 3456 | 0x28 | 0xFF28 | 28800 |
| 50.0 | 1744 | 0x93 | 0xFF93 | 28800 (28670) |
| 49.7664 | 1728 | 0x94 | 0xFF94 | 28800 |
| 24.0 | 832 | 0xCC | 0xFFCC | 28800 (28846) |
| 22.1184 | 768 | 0xD0 | 0xFFD0 | 28800 |
| 18.432 | 640 | 0xD8 | 0xFFD8 | 28800 |
| 11.0592 | 348 | 0xE8 | 0xFFE8 | 28800 |
| 3.6864 | 128 | 0xF8 | 0xFFF8 | 28800 |
| 1.8432 | 64 | 0xFC | 0xFFFC | 28800 |
| 100.0 | 10416 | - | 0xFD75 | 9600 (9601) |
| 99.5328 | 10368 | - | 0xFD78 | 9600 |
| 50.0 | 5216 | - | 0xFEBA | 9600 (9586) |
| 49.7664 | 5184 | - | 0xFEBC | 9600 |
| 24.0 | 2496 | 0x64 | 0xFF64 | 9600 (9615) |
| 22.1184 | 2304 | 0x70 | 0xFF70 | 9600 |
| 18.432 | 1920 | 0x88 | 0xFF88 | 9600 |
| 11.0592 | 1152 | 0xB8 | 0xFFB8 | 9600 |
| 3.6864 | 384 | 0xE8 | 0xFFE8 | 9600 |
| 1.8432 | 192 | 0xF4 | 0xFFF4 | 9600 |

**Notes:**

1. Assumes SMOD0 = 1 and T1M = 1.
2. Numbers in parenthesis show the actual baud rate.

## SFR Definition 21.1. SCON0: UART0 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| SM00 | SM10 | SM20 | REN0 | TB80 | RB80 | TI0 | RI0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x98
SFR Page: 0

Bits7–6: SM00–SM10: Serial Port Operation Mode:
Write:
When written, these bits select the Serial Port Operation Mode as follows:

| SM00 | SM10 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: Synchronous Mode |
| 0 | 1 | Mode 1: 8-Bit UART, Variable Baud Rate |
| 1 | 0 | Mode 2: 9-Bit UART, Fixed Baud Rate |
| 1 | 1 | Mode 3: 9-Bit UART, Variable Baud Rate |

Reading these bits returns the current UART0 mode as defined above.

Bit5: SM20: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port Operation Mode.
Mode 0: No effect
Mode 1: 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.
Mode 2 and 3: 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 and the received address matches the UART0 address or the broadcast address.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART0 receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: TB80: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in Modes 2 and 3. It is not used in Modes 0 and 1. Set or cleared by software as required.

Bit2: RB80: Ninth Receive Bit.
The bit is assigned the logic level of the ninth bit received in Modes 2 and 3. In Mode 1, if SM20 is logic 0, RB80 is assigned the logic level of the received stop bit. RB8 is not used in Mode 0.

Bit1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in Mode 0, or at the beginning of the stop bit in other modes). 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

Bit0: RI0: Receive Interrupt Flag.
Set by hardware when a byte of data has been received by UART0 (as selected by the SM20 bit). 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.

## SFR Definition 21.2. SSTA0: UART0 Status and Clock Selection

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| FE0 | RXOV0 | TXCOL0 | SMOD0 | S0TCLK1 | S0TCLK0 | S0RCLK1 | S0RCLK0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x91
SFR Page: 0

Bit7: FE0: Frame Error Flag.*
This flag indicates if an invalid (low) STOP bit is detected.
0: Frame Error has not been detected
1: Frame Error has been detected.

Bit6: RXOV0: Receive Overrun Flag.*
This flag indicates new data has been latched into the receive buffer before software has read the previous byte.
0: Receive overrun has not been detected.
1: Receive Overrun has been detected.

Bit5: TXCOL0: Transmit Collision Flag.*
This flag indicates user software has written to the SBUF0 register while a transmission is in progress.
0: Transmission Collision has not been detected.
1: Transmission Collision has been detected.

Bit4: SMOD0: UART0 Baud Rate Doubler Enable.
This bit enables/disables the divide-by-two function of the UART0 baud rate logic for configurations described in the UART0 section.
0: UART0 baud rate divide-by-two enabled.
1: UART0 baud rate divide-by-two disabled.

Bits3–2: UART0 Transmit Baud Rate Clock Selection Bits

| S0TCLK1 | S0TCLK0 | **Serial Transmit Baud Rate Clock Source** |
|---|---|---|
| 0 | 0 | Timer 1 generates UART0 TX Baud Rate |
| 0 | 1 | Timer 2 Overflow generates UART0 TX baud rate |
| 1 | 0 | Timer 3 Overflow generates UART0 TX baud rate |
| 1 | 1 | Timer 4 Overflow generates UART0 TX baud rate |

Bits1–0: UART0 Receive Baud Rate Clock Selection Bits

| S0RCLK1 | S0RCLK0 | **Serial Receive Baud Rate Clock Source** |
|---|---|---|
| 0 | 0 | Timer 1 generates UART0 RX Baud Rate |
| 0 | 1 | Timer 2 Overflow generates UART0 RX baud rate |
| 1 | 0 | Timer 3 Overflow generates UART0 RX baud rate |
| 1 | 1 | Timer 4 Overflow generates UART0 RX baud rate |

***Note:** FE0, RXOV0, and TXCOL0 are flags only, and no interrupt is generated by these conditions.

### SFR Definition 21.3. SBUF0: UART0 Data Buffer

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x99
SFR Page: 0

Bits7–0: SBUF0.[7:0]: UART0 Buffer Bits 7–0 (MSB–LSB)
This is actually two registers; a transmit and a receive buffer register. When data is moved to SBUF0, it goes to the transmit buffer and is held for serial transmission. Moving a byte to SBUF0 is what initiates the transmission. When data is moved from SBUF0, it comes from the receive buffer.

### SFR Definition 21.4. SADDR0: UART0 Slave Address

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xA9
SFR Page: 0

Bits7–0: SADDR0.[7:0]: UART0 Slave Address
The contents of this register are used to define the UART0 slave address. Register SADEN0 is a bit mask to determine which bits of SADDR0 are checked against a received address: corresponding bits set to logic 1 in SADEN0 are checked; corresponding bits set to logic 0 are “don’t cares”.

### SFR Definition 21.5. SADEN0: UART0 Slave Address Enable

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xB9
SFR Page: 0

Bits7–0: SADEN0.[7:0]: UART0 Slave Address Enable
Bits in this register enable corresponding bits in register SADDR0 to determine the UART0 slave address.
0: Corresponding bit in SADDR0 is a “don’t care”.
1: Corresponding bit in SADDR0 is checked against a received address.

## 22. UART1

UART1 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 "22.1. Enhanced Baud Rate Generation" on page 300**). Received data buffering allows UART1 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART1 has two associated SFRs: Serial Control Register 1 (SCON1) and Serial Data Buffer 1 (SBUF1). The single SBUF1 location provides access to both transmit and receive registers. Reading SBUF1 accesses the buffered Receive register; writing SBUF1 accesses the Transmit register.

With UART1 interrupts enabled, an interrupt is generated each time a transmit is completed (TI1 is set in SCON1), or a data byte has been received (RI1 is set in SCON1). The UART1 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 UART1 interrupt (transmit complete or receive complete).

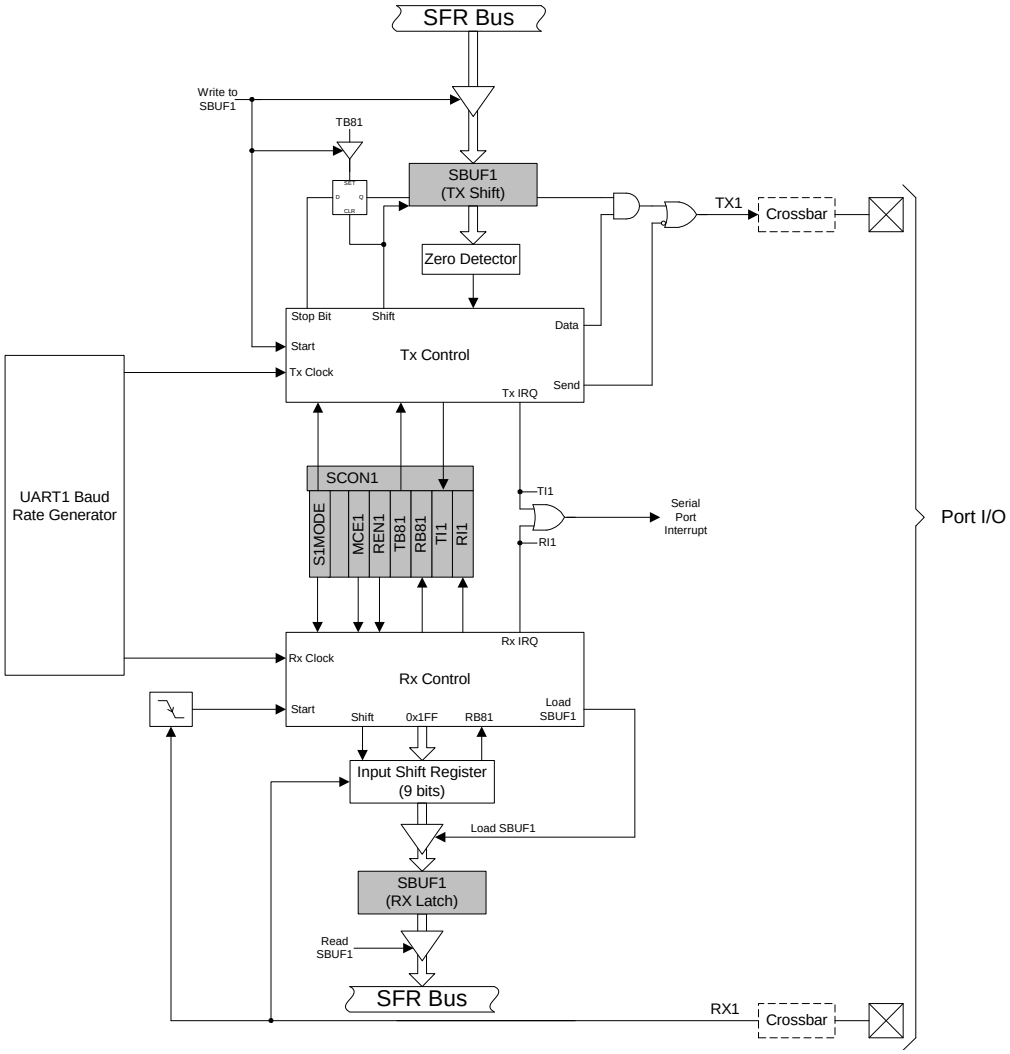



**Figure 22.1. UART1 Block Diagram**

## 22.1. Enhanced Baud Rate Generation

The UART1 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 22.2), 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.

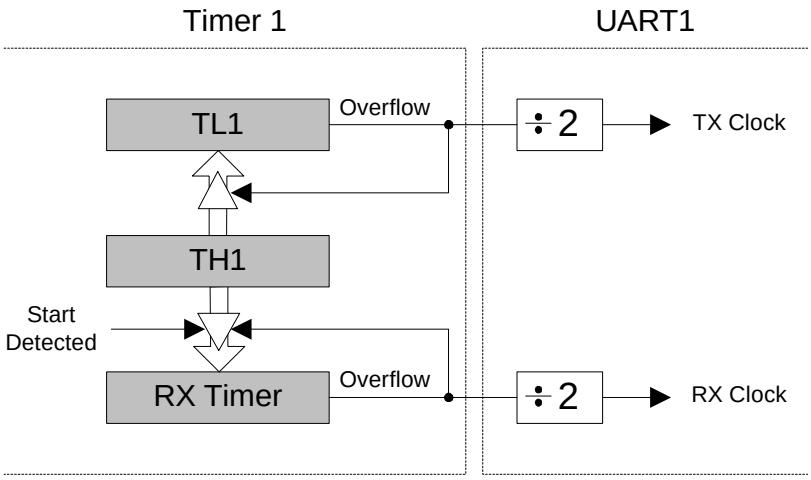



**Figure 22.2. UART1 Baud Rate Logic**

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see **Section "23.1.3. Mode 2: 8-bit Counter/ Timer with Auto-Reload" on page 311**). The Timer 1 reload value should be set so that overflows will occur at two times the desired baud rate. Note that Timer 1 may be clocked by one of five sources: SYSCLK, SYSCLK / 4, SYSCLK / 12, SYSCLK / 48, or the external oscillator clock / 8. For any given Timer 1 clock source, the UART1 baud rate is determined by Equation 22.1.

### Equation 22.1. UART1 Baud Rate

$$UARTBaudRate = \frac{T1_{CLK}}{(256 - T1H)} \times \frac{1}{2}$$

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 "23.1. Timer 0 and Timer 1" on page 309**. A quick reference for typical baud rates and system clock frequencies is given in Table 22.1 through Table 22.5. Note that the internal oscillator or PLL may still generate the system clock when the external oscillator is driving Timer 1 (see **Section "23.1. Timer 0 and Timer 1" on page 309** for more details).

## 22.2. Operational Modes

UART1 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S1MODE bit (SCON1.7). Typical UART connection options are shown below.

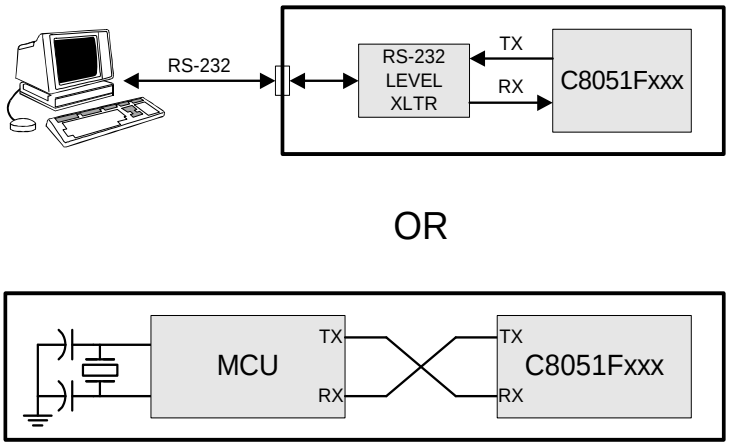



**Figure 22.3. UART Interconnect Diagram**

### 22.2.1. 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 TX1 pin and received at the RX1 pin. On receive, the eight data bits are stored in SBUF1 and the stop bit goes into RB81 (SCON1.2).

Data transmission begins when software writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: RI1 must be logic 0, and if MCE1 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 SBUF1 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF1, the stop bit is stored in RB81 and the RI1 flag is set. If these conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set. An interrupt will occur if enabled when either TI1 or RI1 is set.

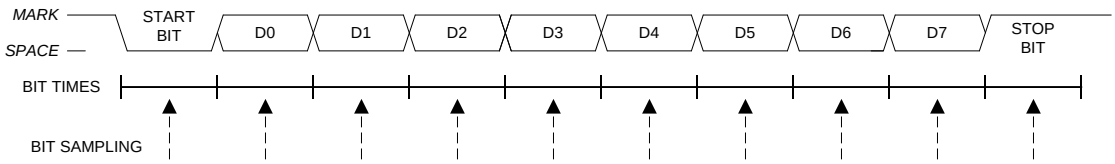



**Figure 22.4. 8-Bit UART Timing Diagram**

### 22.2.2. 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 TB81 (SCON1.3), 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 RB81 (SCON1.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: (1) RI1 must be logic 0, and (2) if MCE1 is logic 1, the 9th bit must be logic 1 (when MCE1 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 SBUF1, the ninth bit is stored in RB81, and the RI1 flag is set to '1'. If the above conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set to '1'. A UART1 interrupt will occur if enabled when either TI1 or RI1 is set to '1'.

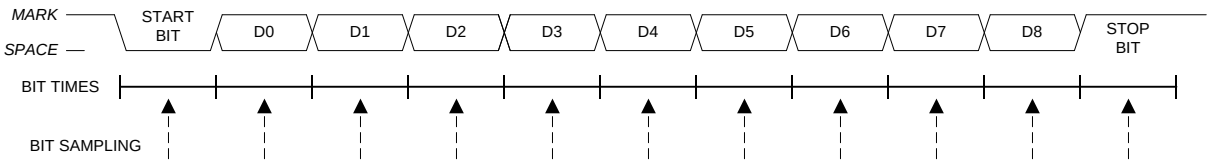



**Figure 22.5. 9-Bit UART Timing Diagram**

## 22.3. 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 MCE1 bit (SCON.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 one (RB81 = 1) signifying an address byte has been received. In the UART interrupt handler, software should compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave should clear its MCE1 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE1 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 should reset its MCE1 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).

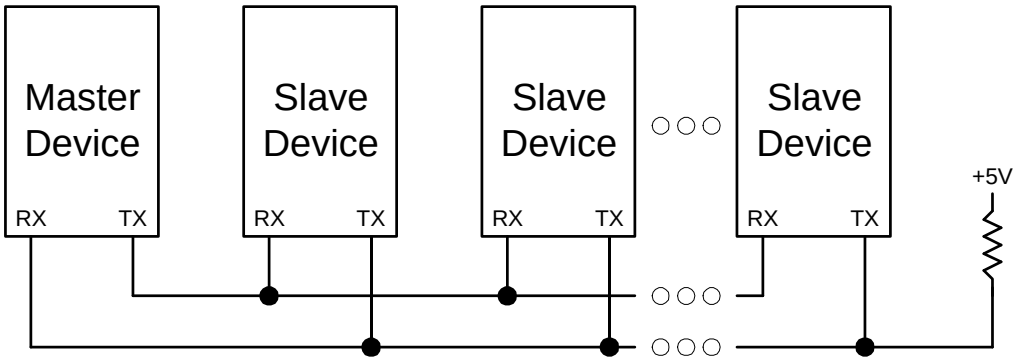



**Figure 22.6. UART Multi-Processor Mode Interconnect Diagram**

## SFR Definition 22.1. SCON1: Serial Port 1 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| S1MODE | - | MCE1 | REN1 | TB81 | RB81 | TI1 | RI1 | 01000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x98
SFR Page: 1

Bit7: S1MODE: Serial Port 1 Operation Mode.
This bit selects the UART1 Operation Mode.
0: Mode 0: 8-bit UART with Variable Baud Rate
1: Mode 1: 9-bit UART with Variable Baud Rate

Bit6: UNUSED. Read = 1b. Write = don't care.

Bit5: MCE1: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
Mode 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI1 will only be activated if stop bit is logic level 1.
Mode 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI1 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN1: Receive Enable.
This bit enables/disables the UART receiver.
0: UART1 reception disabled.
1: UART1 reception enabled.

Bit3: TB81: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB81: Ninth Receive Bit.
RB81 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI1: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART1 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART1 interrupt is enabled, setting this bit causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software

Bit0: RI1: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART1 (set at the STOP bit sampling time). When the UART1 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

### SFR Definition 22.2. SBUF1: Serial (UART1) Port Data Buffer

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x99
SFR Page: 1

Bits7–0: SBUF1[7:0]: 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 SBUF1, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF1 is what initiates the transmission. A read of SBUF1 returns the contents of the receive latch.

### Table 22.1. 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)* | T1M* | Timer 1 Reload Value (hex) |
| SYSCLK from Internal Osc. | 230400 | -0.32% | 106 | SYSCLK | XX | 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 |

X = Don't care

***Note:** SCA1-SCA0 and T1M bit definitions can be found in **Section 23.1**.

**Table 22.2. Timer Settings for Standard Baud Rates Using an External 25.0 MHz Oscillator**

| | Frequency: 25.0 MHz | | | | | | |
|---|---|---|---|---|---|---|---|
| | Target Baud Rate (bps) | Baud Rate % Error | Oscilla-tor Divide Factor | Timer Clock Source | SCA1-SCA0 (pre-scale select)* | T1M* | Timer 1 Reload Value (hex) |
| SYSCLK from External Osc. | 230400 | -0.47% | 108 | SYSCLK | XX | 1 | 0xCA |
| | 115200 | 0.45% | 218 | SYSCLK | XX | 1 | 0x93 |
| | 57600 | -0.01% | 434 | SYSCLK | XX | 1 | 0x27 |
| | 28800 | 0.45% | 872 | SYSCLK / 4 | 01 | 0 | 0x93 |
| | 14400 | -0.01% | 1736 | SYSCLK / 4 | 01 | 0 | 0x27 |
| | 9600 | 0.15% | 2608 | EXTCLK / 8 | 11 | 0 | 0x5D |
| | 2400 | 0.45% | 10464 | SYSCLK / 48 | 10 | 0 | 0x93 |
| | 1200 | -0.01% | 20832 | SYSCLK / 48 | 10 | 0 | 0x27 |
| SYSCLK from Internal Osc. | 57600 | -0.47% | 432 | EXTCLK / 8 | 11 | 0 | 0xE5 |
| | 28800 | -0.47% | 864 | EXTCLK / 8 | 11 | 0 | 0xCA |
| | 14400 | 0.45% | 1744 | EXTCLK / 8 | 11 | 0 | 0x93 |
| | 9600 | 0.15% | 2608 | EXTCLK / 8 | 11 | 0 | 0x5D |

X = Don't care

***Note:** SCA1-SCA0 and T1M bit definitions can be found in **Section 23.1**.

**Table 22.3. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator**

| | Frequency: 22.1184 MHz | | | | | | |
|---|---|---|---|---|---|---|---|
| | Target Baud Rate (bps) | Baud Rate % Error | Oscilla-tor Divide Factor | Timer Clock Source | SCA1-SCA0 (pre-scale select)* | T1M* | Timer 1 Reload Value (hex) |
| SYSCLK from External Osc. | 230400 | 0.00% | 96 | SYSCLK | XX | 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 |
| 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 |

X = Don't care

***Note:** SCA1-SCA0 and T1M bit definitions can be found in **Section 23.1**.

**Table 22.4. Timer Settings for Standard Baud Rates Using the PLL**

| Frequency: 50.0 MHz | | | | | | |
|---|---|---|---|---|---|---|
| **Target Baud Rate (bps)** | **Baud Rate % Error** | **Oscilla-tor Divide Factor** | **Timer Clock Source** | **SCA1-SCA0 (pre-scale select)*** | **T1M*** | **Timer 1 Reload Value (hex)** |
| 230400 | 0.45% | 218 | SYSCLK | XX | 1 | 0x93 |
| 115200 | -0.01% | 434 | SYSCLK | XX | 1 | 0x27 |
| 57600 | 0.45% | 872 | SYSCLK / 4 | 01 | 0 | 0x93 |
| 28800 | -0.01% | 1736 | SYSCLK / 4 | 01 | 0 | 0x27 |
| 14400 | 0.22% | 3480 | SYSCLK / 12 | 00 | 0 | 0x6F |
| 9600 | -0.01% | 5208 | SYSCLK / 12 | 00 | 0 | 0x27 |
| 2400 | -0.01% | 20832 | SYSCLK / 48 | 10 | 0 | 0x27 |

X = Don't care

***Note:** SCA1-SCA0 and T1M bit definitions can be found in **Section 23.1**.

**Table 22.5. Timer Settings for Standard Baud Rates Using the PLL**

| Frequency: 100.0 MHz | | | | | | |
|---|---|---|---|---|---|---|
| **Target Baud Rate (bps)** | **Baud Rate % Error** | **Oscilla-tor Divide Factor** | **Timer Clock Source** | **SCA1-SCA0 (pre-scale select)*** | **T1M*** | **Timer 1 Reload Value (hex)** |
| 230400 | -0.01% | 434 | SYSCLK | XX | 1 | 0x27 |
| 115200 | 0.45% | 872 | SYSCLK / 4 | 01 | 0 | 0x93 |
| 57600 | -0.01% | 1736 | SYSCLK / 4 | 01 | 0 | 0x27 |
| 28800 | 0.22% | 3480 | SYSCLK / 12 | 00 | 0 | 0x6F |
| 14400 | -0.47% | 6912 | SYSCLK / 48 | 10 | 0 | 0xB8 |
| 9600 | 0.45% | 10464 | SYSCLK / 48 | 10 | 0 | 0x93 |

X = Don't care

***Note:** SCA1-SCA0 and T1M bit definitions can be found in **Section 23.1**.

## NOTES:

# 23. Timers

Each MCU includes 5 counter/timers: Timer 0 and Timer 1 are 16-bit counter/timers compatible with those found in the standard 8051. Timer 2, Timer 3, and Timer 4 are 16-bit auto-reload and capture counter/timers for use with the ADCs, DACs, square-wave generation, 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 3 offers 16-bit auto-reload and capture. Timers 2 and 4 are identical, and offer not only 16-bit auto-reload and capture, but have the ability to produce a 50% duty-cycle square-wave (toggle output) at an external port pin.

| Timer 0 and Timer 1 Modes: | Timer 2, 3 and 4 Modes: |
|---|---|
| 13-bit counter/timer | 16-bit counter/timer with auto-reload |
| 16-bit counter/timer | 16-bit counter/timer with capture |
| 8-bit counter/timer with auto-reload | Toggle Output (Timer 2 and 4 only) |
| 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 by which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 23.3 for pre-scaled clock selection). Timers 0 and 1 can be configured to use either the pre-scaled clock signal or the system clock directly. Timers 2, 3, and 4 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

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. Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signal need not be periodic, but it should be held at a given logic level for at least two full system clock cycles to ensure the level is properly sampled.

## 23.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate 8-bit SFRs: 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 their status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (**Section “11.3.5. Interrupt Register Descriptions” on page 157**); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (**Section 11.3.5**). 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). Both timers can be configured independently.

### 23.1.1. 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 the TL0 register. 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 "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238** 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 23.3).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal /INT0 is logic-level 1. Setting GATE0 to '1' allows the timer to be controlled by the external input signal /INT0 (see **Section "11.3.5. Interrupt Register Descriptions" on page 157**), facilitating pulse width measurements.

| TR0 | GATE0 | /INT0 | Counter/Timer |
|---|---|---|---|
| 0 | X | X | Disabled |
| 1 | 0 | X | Enabled |
| 1 | 1 | 0 | Disabled |
| 1 | 1 | 1 | Enabled |

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 /INT1 is used with Timer 1.

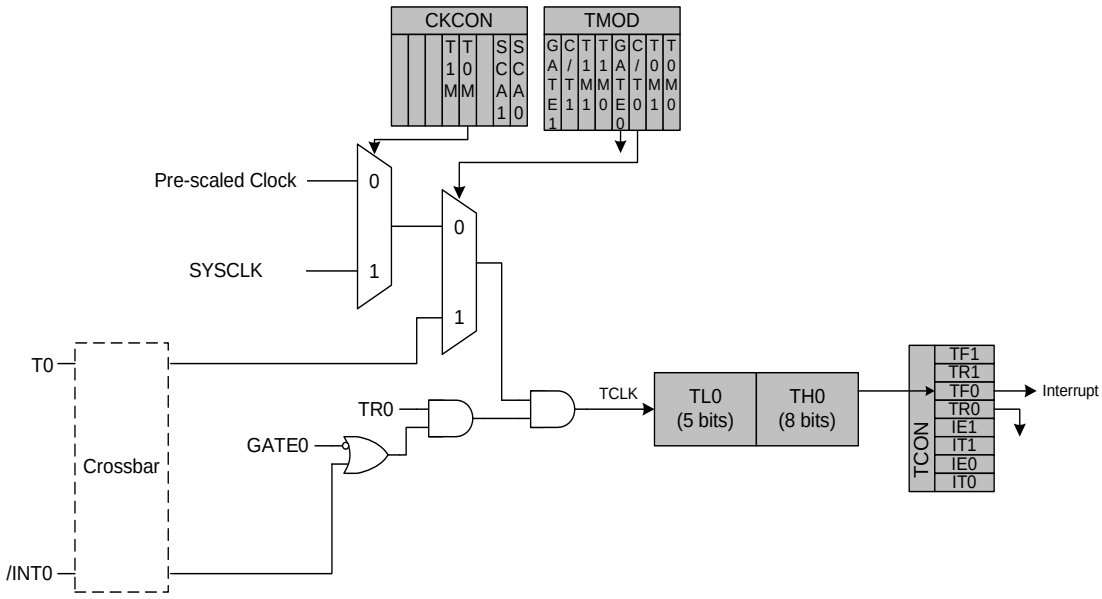



**Figure 23.1. T0 Mode 0 Block Diagram**

### 23.1.2. 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.

### 23.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 or 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 0xFF 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 low

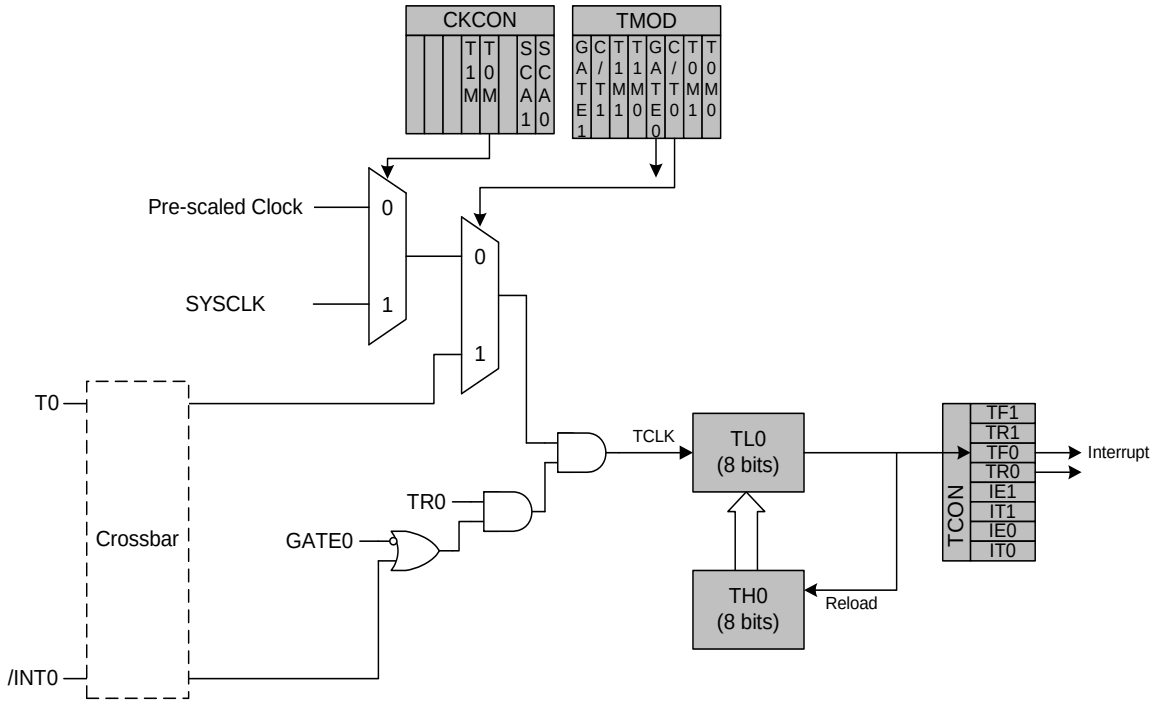



**Figure 23.2. T0 Mode 2 Block Diagram**

### 23.1.4. 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.

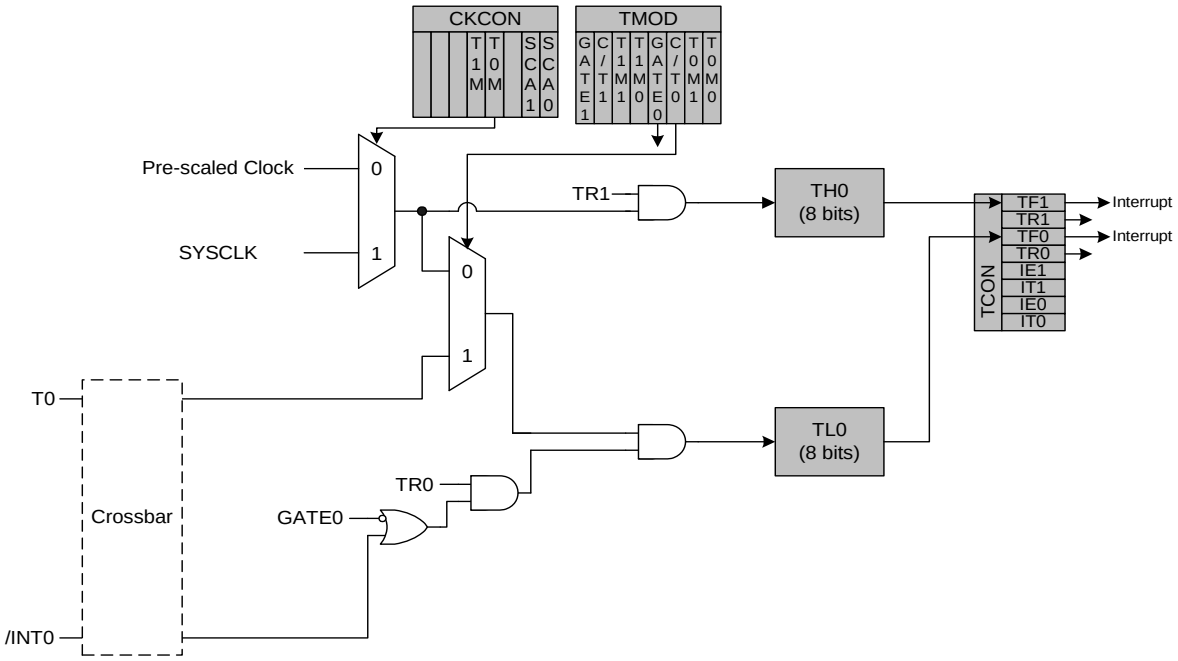



**Figure 23.3. T0 Mode 3 Block Diagram**

## SFR Definition 23.1. TCON: Timer Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0x88
SFR Page: 0

Bit7: TF1: Timer 1 Overflow Flag.
Set 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.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

Bit6: TR1: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit5: TF0: Timer 0 Overflow Flag.
Set 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.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

Bit4: TR0: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit3: 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 if IT1 = 1. This flag is the inverse of the /INT1 signal.

Bit2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured /INT1 interrupt will be falling-edge sensitive or active-low.
0: /INT1 is level triggered, active-low.
1: /INT1 is edge triggered, falling-edge.

Bit1: IE0: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. This flag is the inverse of the /INT0 signal.

Bit0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured /INT0 interrupt will be falling-edge sensitive or active-low.
0: /INT0 is level triggered, active logic-low.
1: /INT0 is edge triggered, falling-edge.

## SFR Definition 23.2. TMOD: Timer Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| GATE1 | C/T1 | T1M1 | T1M0 | GATE0 | C/T0 | T0M1 | T0M0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x89
SFR Page: 0

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 = logic 1.

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

| T1M1 | T1M0 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: 13-bit counter/timer |
| 0 | 1 | Mode 1: 16-bit counter/timer |
| 1 | 0 | Mode 2: 8-bit counter/timer with auto-reload |
| 1 | 1 | Mode 3: Timer 1 inactive |

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 = logic 1.

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

| T0M1 | T0M0 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: 13-bit counter/timer |
| 0 | 1 | Mode 1: 16-bit counter/timer |
| 1 | 0 | Mode 2: 8-bit counter/timer with auto-reload |
| 1 | 1 | Mode 3: Two 8-bit counter/timers |

## SFR Definition 23.3. CKCON: Clock Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | T1M | T0M | - | SCA1 | SCA0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8E
SFR Page: 0

Bits7–5: UNUSED. Read = 000b, Write = don't care.
Bit4: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic 1.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.
Bit3: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic 1.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1-SCA0.
1: Counter/Timer 0 uses the system clock.
Bit2: UNUSED. Read = 0b, Write = don't care.
Bits1–0: SCA1–SCA0: Timer 0/1 Prescale Bits
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

| SCA1 | SCA0 | Prescaled Clock |
|---|---|---|
| 0 | 0 | System clock divided by 12 |
| 0 | 1 | System clock divided by 4 |
| 1 | 0 | System clock divided by 48 |
| 1 | 1 | External clock divided by 8* |

***Note:** External clock divided by 8 is synchronized with the system clock, and external clock must be less than or equal to the system clock frequency to operate the timer in this mode.

## SFR Definition 23.4. TL0: Timer 0 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8A
SFR Page: 0

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

### SFR Definition 23.5. TL1: Timer 1 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8B
SFR Page: 0

Bits 7–0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

### SFR Definition 23.6. TH0: Timer 0 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8C
SFR Page: 0

Bits 7–0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

### SFR Definition 23.7. TH1: Timer 1 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0x8D
SFR Page: 0

Bits 7–0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

## 23.2. Timer 2, Timer 3, and Timer 4

Timers 2, 3, and 4 are 16-bit counter/timers, each formed by two 8-bit SFR's: TMRnL (low byte) and TMRnH (high byte) where n = 2, 3, and 4 for timers 2, 3, and 4 respectively. Timers 2 and 4 feature auto-reload, capture, and toggle output modes with the ability to count up or down. Timer 3 features auto-reload and capture modes, with the ability to count up or down. Capture Mode and Auto-reload mode are selected using bits in the Timer 2, 3, and 4 Control registers (TMRnCN). Toggle output mode is selected using the Timer 2 or 4 Configuration registers (TMRnCF). These timers may also be used to generate a square-wave at an external pin. As with Timers 0 and 1, Timers 2, 3, and 4 can use either the system clock (divided by one, two, or twelve), external clock (divided by eight) or transitions on an external input pin as its clock source. Timer 2 and 3 can be used to start an ADC Data Conversion and Timers 2, 3, and 4 can schedule DAC outputs. Timers 1, 2, 3, or 4 may be used to generate baud rates for UART 0. Only Timer 1 can be used to generate baud rates for UART 1.

The Counter/Timer Select bit C/Tn bit (TMRnCN.1) configures the peripheral as a counter or timer. Clearing C/Tn configures the Timer to be in a timer mode (i.e., the system clock or transitions on an external pin as the input for the timer). When C/Tn is set to 1, the timer is configured as a counter (i.e., high-to-low transitions at the Tn input pin increment (or decrement) the counter/timer register. Timer 3 and Timer 2 share the T2 input pin. Refer to **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238** for information on selecting and configuring external I/O pins for digital peripherals, such as the Tn pin.

Timer 2, 3, and 4 can use either SYSCLK, SYSCLK divided by 2, SYSCLK divided by 12, an external clock divided by 8, or high-to-low transitions on the Tn input pin as its clock source when operating in Counter/Timer with Capture mode. Clearing the C/Tn bit (TMRnCN.1) selects the system clock/external clock as the input for the timer. The Timer Clock Select bits TnM0 and TnM1 in TMRnCF can be used to select the system clock undivided, system clock divided by two, system clock divided by 12, or an external clock provided at the XTAL1/XTAL2 pins divided by 8 (see SFR Definition 23.13). When C/Tn is set to logic 1, a high-to-low transition at the Tn input pin increments the counter/timer register (i.e., configured as a counter).

### 23.2.1. Configuring Timer 2, 3, and 4 to Count Down

Timers 2, 3, and 4 have the ability to count down. When the timer's Decrement Enable Bit (DCENn) in the Timer Configuration Register (See SFR Definition 23.13) is set to '1', the timer can then count *up* or *down*. When DCENn = 1, the direction of the timer's count is controlled by the TnEX pin's logic level (Timer 3 shares the T2EX pin with Timer 2). When TnEX = 1, the counter/timer will count up; when TnEX = 0, the counter/timer will count down. To use this feature, TnEX must be enabled in the digital crossbar and configured as a digital input.

**Note: When DCENn = 1, other functions of the TnEX input (i.e., capture and auto-reload) are not available. TnEX will only control the direction of the timer when DCENn = 1.**

### 23.2.2. Capture Mode

In Capture Mode, Timer 2, 3, and 4 will operate as a 16-bit counter/timer with capture facility. When the Timer External Enable bit (found in the TMRnCN register) is set to ‘1’, a high-to-low transition on the TnEX input pin (Timer 3 shares the T2EX pin with Timer 2) causes the 16-bit value in the associated timer (THn, TLn) to be loaded into the capture registers (RCAPnH, RCAPnL). If a capture is triggered in the counter/timer, the Timer External Flag (TMRnCN.6) will be set to ‘1’ and an interrupt will occur if the interrupt is enabled. See **Section “11.3. Interrupt Handler” on page 154** for further information concerning the configuration of interrupt sources.

As the 16-bit timer register increments and overflows TMRnH:TMRnL, the TFn Timer Overflow/Underflow Flag (TMRnCN.7) is set to ‘1’ and an interrupt will occur if the interrupt is enabled. The timer can be configured to count down by setting the Decrement Enable Bit (TMRnCF.0) to ‘1’. This will cause the timer to decrement with every timer clock/count event and underflow when the timer transitions from 0x0000 to 0xFFFF. Just as in overflows, the Overflow/Underflow Flag (TFn) will be set to ‘1’, and an interrupt will occur if enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RLn (TMRnCN.0) and the Timer 2, 3, and 4 Run Control bit TRn (TMRnCN.2) to logic 1. The Timer 2, 3, and 4 respective External Enable EXENn (TMRnCN.3) must also be set to logic 1 to enable captures. If EXENn is cleared, transitions on TnEX will be ignored.

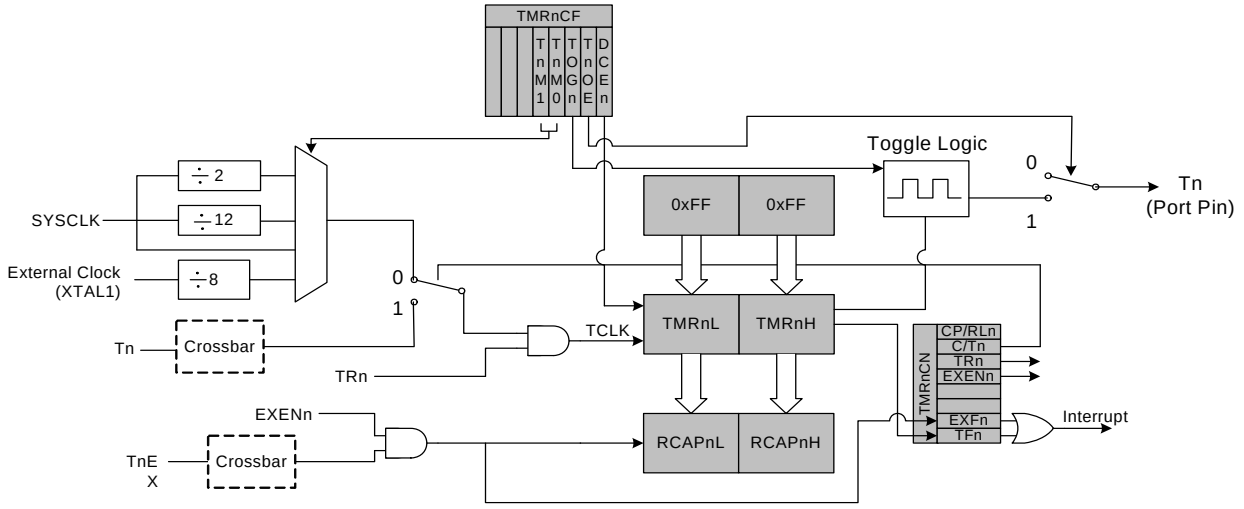



**Figure 23.4. T2, 3, and 4 Capture Mode Block Diagram**

### 23.2.3. Auto-Reload Mode

In Auto-Reload Mode, the counter/timer can be configured to count up or down and cause an interrupt/flag to occur upon an overflow/underflow event. When counting up, the counter/timer will set its overflow/underflow flag (TFn) and cause an interrupt (if enabled) upon overflow/underflow, and the values in the Reload/Capture Registers (RCAPnH and RCAPnL) are loaded into the timer and the timer is restarted. When the Timer External Enable Bit (EXENn) bit is set to '1' and the Decrement Enable Bit (DCENn) is '0', a falling edge ('1'-to-'0' transition) on the TnEX pin will cause a timer reload. Note that timer overflows will also cause auto-reloads. When DCENn is set to '1', the state of the TnEX pin controls whether the counter/timer counts *up* (increments) or *down* (decrements), and will not cause an auto-reload or interrupt event (Timer 3 shares the T2EX pin with Timer 2). See **Section 23.2.1** for information concerning configuration of a timer to count down.

When counting down, the counter/timer will set its overflow/underflow flag (TFn) and cause an interrupt (if enabled) when the value in the TMRnH and TMRnL registers matches the 16-bit value in the Reload/Capture Registers (RCAPnH and RCAPnL). This is considered an underflow event, and will cause the timer to load the value 0xFFFF. The timer is automatically restarted when an underflow occurs.

Counter/Timer with Auto-Reload mode is selected by clearing the CP/RLn bit. Setting TRn to logic 1 enables and starts the timer.

In Auto-Reload Mode, the External Flag (EXFn) toggles upon every overflow or underflow and does not cause an interrupt. The EXFn flag can be used as the most significant bit (MSB) of a 17-bit counter.

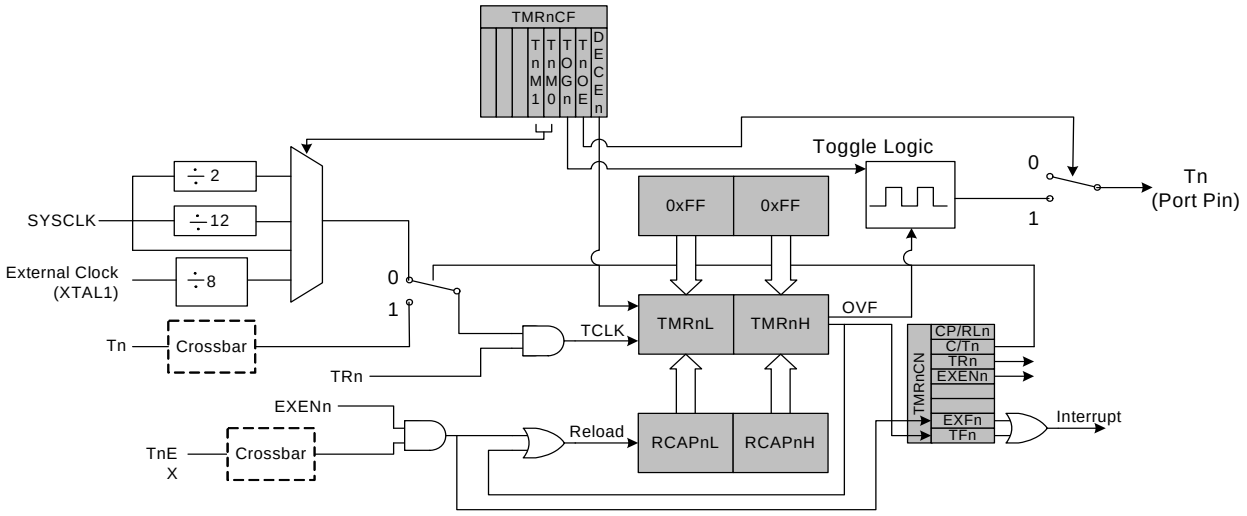



**Figure 23.5. Tn Auto-reload (T2,3,4) and Toggle Mode (T2,4) Block Diagram**

### 23.2.4. Toggle Output Mode (Timer 2 and Timer 4 Only)

Timers 2 and 4 have the capability to toggle the state of their respective output port pins (T2 or T4) to produce a 50% duty cycle waveform output. The port pin state will change upon the overflow or underflow of the respective timer (depending on whether the timer is counting *up* or *down*). The toggle frequency is determined by the clock source of the timer and the values loaded into RCAPnH and RCAPnL. When counting DOWN, the auto-reload value for the timer is 0xFFFF, and underflow will occur when the value in the timer matches the value stored in RCAPnH:RCAPnL. When counting UP, the auto-reload value for the timer is RCAPnH:RCAPnL, and overflow will occur when the value in the timer transitions from 0xFFFF to the reload value.

To output a square wave, the timer is placed in reload mode (the Capture/Reload Select Bit in TMRnCN and the Timer/Counter Select Bit in TMRnCN are cleared to ‘0’). The timer output is enabled by setting the Timer Output Enable Bit in TMRnCF to ‘1’. The timer should be configured via the timer clock source and reload/underflow values such that the timer overflow/underflows at 1/2 the desired output frequency. The port pin assigned by the crossbar as the timer’s output pin should be configured as a digital output (see **Section “18. Port Input/Output” on page 235**). Setting the timer’s Run Bit (TRn) to ‘1’ will start the toggle of the pin. A Read/Write of the Timer’s Toggle Output State Bit (TMRnCF.2) is used to read the state of the toggle output, or to force a value of the output. This is useful when it is desired to start the toggle of a pin in a known state, or to force the pin into a desired state when the toggle mode is halted.

**Equation 23.1. Square Wave Frequency (Timer 2 and Timer 4 Only)**

$$F_{sq} = \frac{F_{TCLK}}{2 \times (65536 - RCAPn)}$$

## SFR Definition 23.8. TMRnCN: Timer 2, 3, and 4 Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| TFn | EXFn | - | - | EXENn | TRn | C/Tn | CP/RLn | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: TMR2CN:0xC8;TMR3CN:0xC8;TMR4CN:0xC8
SFR Page: TMR2CN: page 0;TMR3CN: page 1;TMR4CN: page 2

Bit7: TFn: Timer 2, 3, and 4 Overflow/Underflow Flag.
Set by hardware when either the Timer overflows from 0xFFFF to 0x0000, underflows from the value placed in RCAPnH:RCAPnL to 0xFFFF (in Auto-reload Mode), or underflows from 0x0000 to 0xFFFF (in Capture Mode). When the Timer interrupt is enabled, setting this bit causes the CPU to vector to the Timer interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: EXFn: Timer 2, 3, or 4 External Flag.
Set by hardware when either a capture or reload is caused by a high-to-low transition on the TnEX input pin and EXENn is logic 1. When the Timer interrupt is enabled, setting this bit causes the CPU to vector to the Timer Interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit5–4: Reserved.

Bit3: EXENn: Timer 2, 3, and 4 External Enable.
Enables high-to-low transitions on TnEX to trigger captures, reloads, and control the direction of the timer/counter (up or down count). If DCENn = 1, TnEX will determine if the timer counts up or down when in Auto-reload Mode. If EXENn = 1, TnEX should be configured as a digital input.
0: Transitions on the TnEX pin are ignored.
1: Transitions on the TnEX pin cause capture, reload, or control the direction of timer count (up or down) as follows:
Capture Mode: ‘1’-to-’0’ Transition on TnEX pin causes RCAPnH:RCAPnL to capture timer value.
Auto-Reload Mode:
DCENn = 0: ‘1’-to-’0’ transition causes reload of timer and sets the EXFn Flag.
DCENn = 1: TnEX logic level controls direction of timer (up or down).

Bit2: TRn: Timer 2, 3, and 4 Run Control.
This bit enables/disables the respective Timer.
0: Timer disabled.
1: Timer enabled and running/counting.

Bit1: C/Tn: Counter/Timer Select.
0: Timer Function: Timer incremented by clock defined by TnM1:TnM0 (TMRnCF.4:TMRnCF.3).
1: Counter Function: Timer incremented by high-to-low transitions on external input pin.

Bit0: CP/RLn: Capture/Reload Select.
This bit selects whether the Timer functions in capture or auto-reload mode.
0: Timer is in Auto-Reload Mode.
1: Timer is in Capture Mode.

**Note:** Timer 3 and Timer 2 share the T2 and T2EX pins.

## SFR Definition 23.9. TMRnCF: Timer 2, 3, and 4 Configuration

| | | | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | TnM1 | TnM0 | TOGn | TnOE | DCENn | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: TMR2CF:0xC9;TMR3CF:0xC9;TMR4CF:0xC9
SFR Page TMR2CF: page 0;TMR3CF: page 1;TMR4CF: Page 2

Bit7–5: Reserved.

Bit4–3: TnM1 and TnM0: Timer Clock Mode Select Bits.
Bits used to select the Timer clock source. The sources can be the System Clock (SYSCLK), SYSCLK divided by 2 or 12, or the external clock divided by 8. Clock source is selected as follows:
00: SYSCLK/12
01: SYSCLK
10: EXTERNAL CLOCK/8 (Synchronized to the System Clock)
11: SYSCLK/2

Bit2: TOGn: Toggle output state bit.
When timer is used to toggle a port pin, this bit can be used to read the state of the output, or can be written to in order to force the state of the output (Timer 2 and Timer 4 Only).

Bit1: TnOE: Timer output enable bit.
This bit enables the timer to output a 50% duty cycle output to the timer's assigned external port pin.
*NOTE: A timer is configured for Square Wave Output as follows:*
*CP/RLn = 0*
*C/Tn = 0*
*TnOE = 1*
*Load RCAPnH:RCAPnL (See "Square Wave Frequency (Timer 2 and Timer 4 Only)" on page 320.)*
*Configure Port Pin to output squarewave (See* **Section "18. Port Input/Output" on page 235***)*
0: Output of toggle mode not available at Timers's assigned port pin.
1: Output of toggle mode available at Timers's assigned port pin.

Bit0: DCENn: Decrement Enable Bit.
This bit enables the timer to count up or down as determined by the state of TnEX.
0: Timer will count up, regardless of the state of TnEX.
1: Timer will count up or down depending on the state of TnEX as follows:
if TnEX = 0, the timer counts DOWN.
if TnEX = 1, the timer counts UP.

**Note:** Timer 3 and Timer 2 share the T2 and T2EX pins.

## SFR Definition 23.10. RCAPnL: Timer 2, 3, and 4 Capture Register Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: RCAP2L: 0xCA; RCAP3L: 0xCA; RCAP4L: 0xCA

SFR Page: RCAP2L: page 0; RCAP3L: page 1; RCAP4L: page 2

Bits 7–0: RCAP2, 3, and 4L: Timer 2, 3, and 4 Capture Register Low Byte.
The RCAP2, 3, and 4L register captures the low byte of Timer 2, 3, and 4 when Timer 2, 3, and 4 is configured in capture mode. When Timer 2, 3, and 4 is configured in auto-reload mode, it holds the low byte of the reload value.

## SFR Definition 23.11. RCAPnH: Timer 2, 3, and 4 Capture Register High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: RCAP2H: 0xCB; RCAP3H: 0xCB; RCAP4H: 0xCB

SFR Page: RCAP2H: page 0; RCAP3H: page 1; RCAP4H: page 2

Bits 7–0: RCAP2, 3, and 4H: Timer 2, 3, and 4 Capture Register High Byte.
The RCAP2, 3, and 4H register captures the high byte of Timer 2, 3, and 4 when Timer 2, 3, and 4 is configured in capture mode. When Timer 2, 3, and 4 is configured in auto-reload mode, it holds the high byte of the reload value.

## SFR Definition 23.12. TMRnL: Timer 2, 3, and 4 Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: TMR2L: 0xCC; TMR3L: 0xCC; TMR4L: 0xCC

SFR Page: TMR2L: page 0; TMR3L: page 1; TMR4L: page 2

Bits 7–0: TL2, 3, and 4: Timer 2, 3, and 4 Low Byte.
The TL2, 3, and 4 register contains the low byte of the 16-bit Timer 2, 3, and 4

## SFR Definition 23.13. TMRnH Timer 2, 3, and 4 High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: TMR2H: 0xCD; TMR3H: 0xCD; TMR4H: 0xCD

SFR Page: TMR2H: page 0; TMR3H: page 1; TMR4H: page 2

Bits 7–0: TH2, 3, and 4: Timer 2, 3, and 4 High Byte.
The TH2, 3, and 4 register contains the high byte of the 16-bit Timer 2, 3, and 4

## 24. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. PCA0 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 (See **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238**). The counter/timer is driven by a programmable timebase that can select between six inputs as its source: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI line. 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-Bit PWM, or 16-Bit PWM (each is described in **Section 24.2**). The PCA is configured and controlled through the system controller's Special Function Registers. The basic PCA block diagram is shown in Figure 24.1.

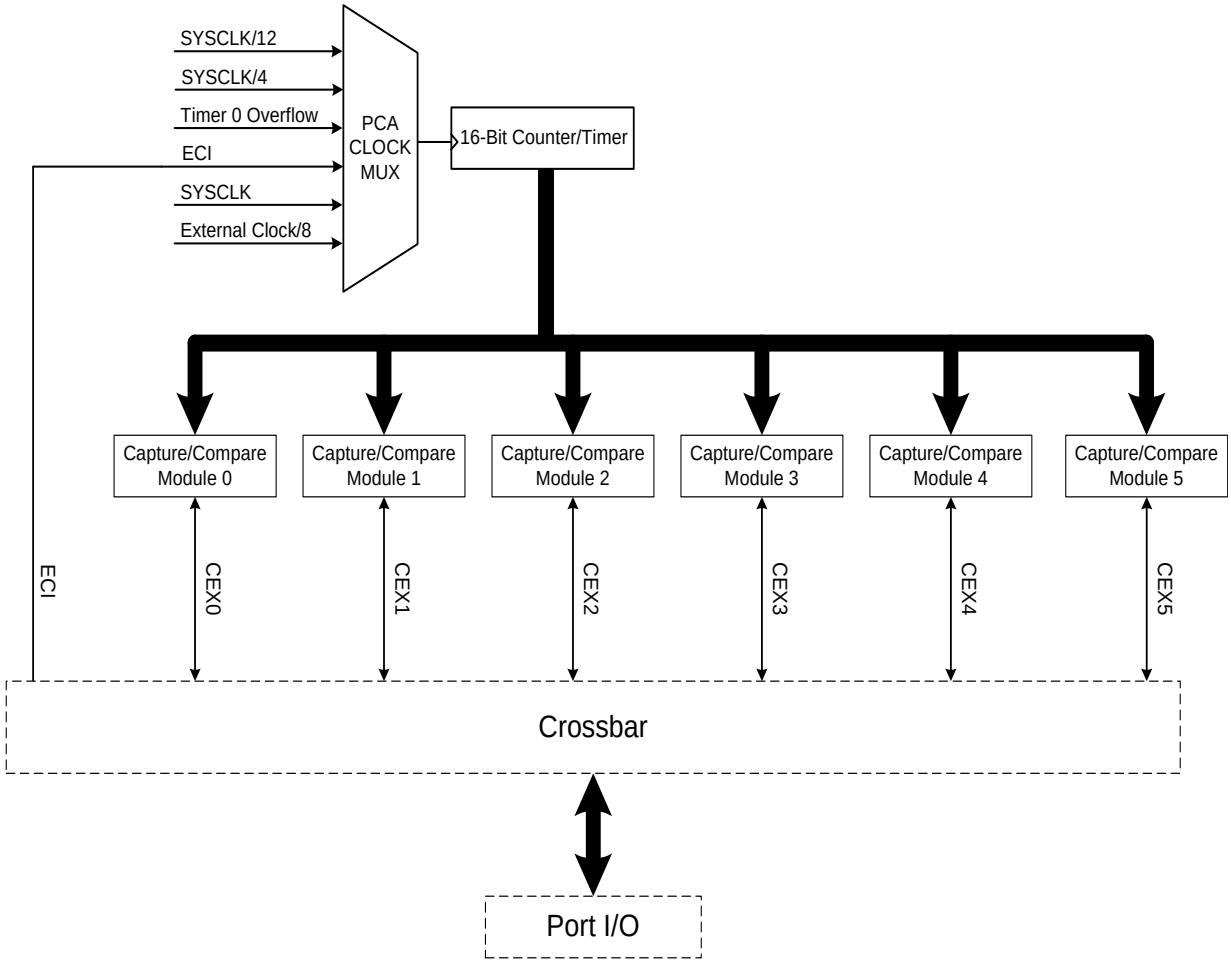



**Figure 24.1. PCA Block Diagram**

## 24.1. 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 24.1.

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 (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic 1). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

**Table 24.1. 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* |
| ***Note:** External clock divided by 8 is synchronized with the system clock. | | | |

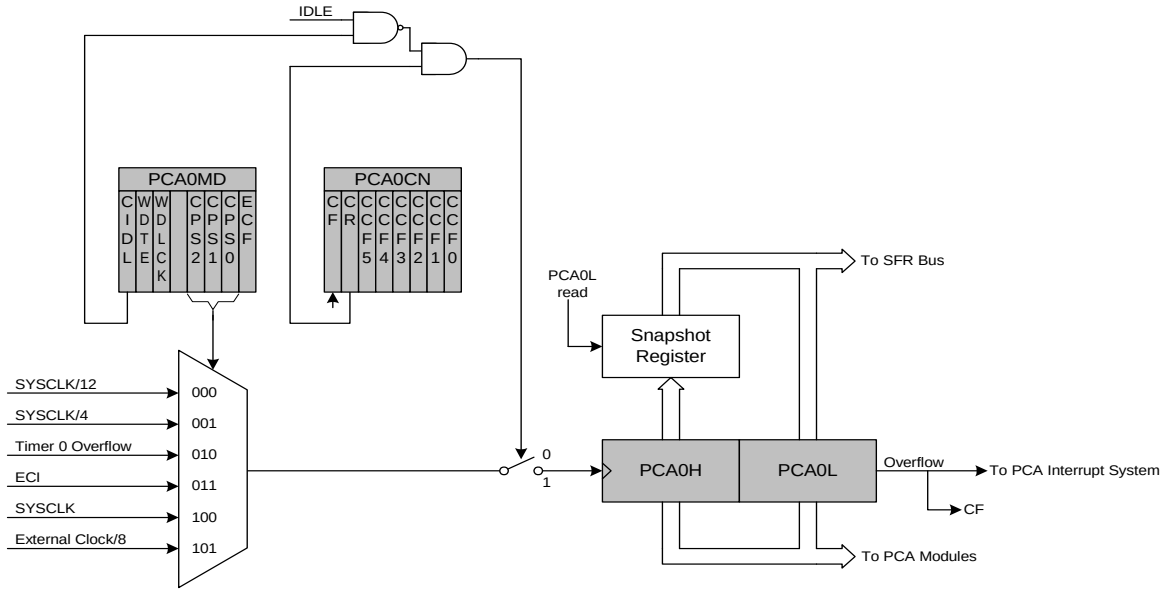



**Figure 24.2. PCA Counter/Timer Block Diagram**

**Important Note About the PCA0CN Register:** If the main PCA counter (PCA0H : PCA0L) overflows during the execution phase of a read-modify-write instruction (bit-wise SETB or CLR, ANL, ORL, XRL) that targets the PCA0CN register, the CF (Counter Overflow) bit will not be set. If the CF flag is used by software to keep track of counter overflows, the following steps should be taken when performing a bit-wise operation on the PCA0CN register:

Step 1. Disable global interrupts.
Step 2. Read PCA0L. This will latch the value of PCA0H.
Step 3. Read PCA0H, saving the value.
Step 4. Execute the bit-wise operation on CCFn (for example, CLR CCF0, or CCF0 = 0;).
Step 5. Read PCA0L.
Step 6. Read PCA0H, saving the value.
Step 7. If the value of PCA0H read in Step 3 is 0xFF and the value for PCA0H read in Step 6 is 0x00, then manually set the CF bit in software (for example, SETB CF, or CF = 1;).
Step 8. Re-enable interrupts.

## 24.2. 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-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 24.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA0 capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic 1. See Figure 24.3 for details on the PCA interrupt configuration.

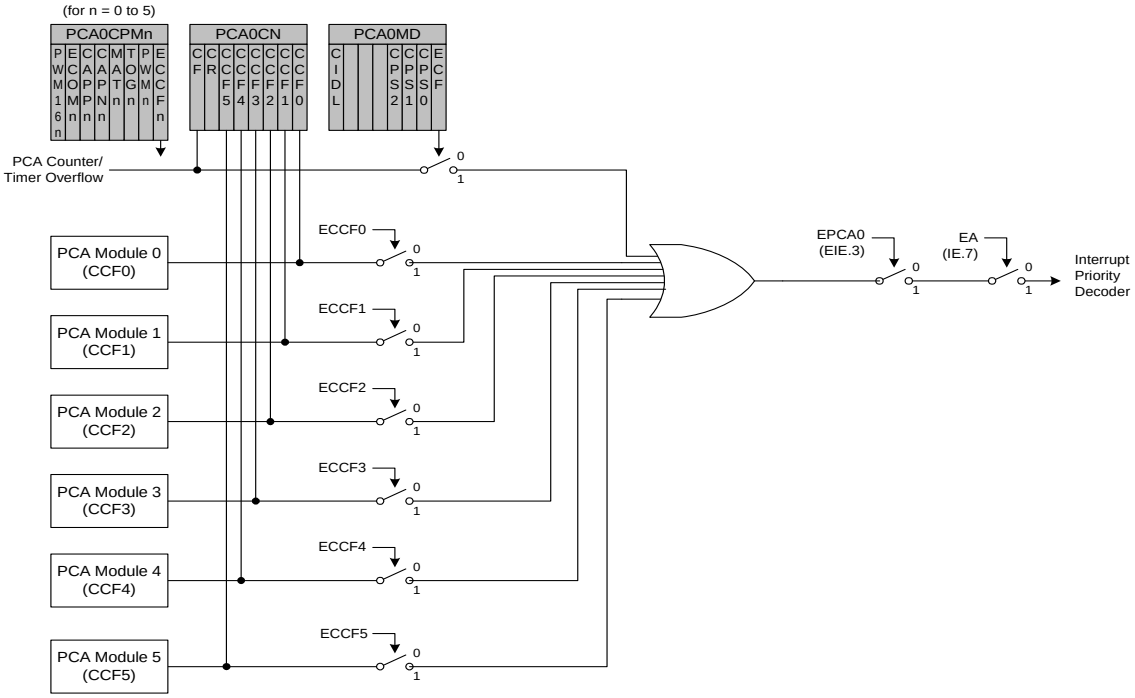



**Figure 24.3. PCA Interrupt Block Diagram**

**Table 24.2. PCA0CPM Register Settings for PCA Capture/Compare Modules**

| PWM16 | ECOM | CAPP | CAPN | MAT | TOG | PWM | ECCF | Operation Mode |
|---|---|---|---|---|---|---|---|---|
| X | X | 1 | 0 | 0 | 0 | 0 | X | Capture triggered by positive edge on CEXn |
| X | X | 0 | 1 | 0 | 0 | 0 | X | Capture triggered by negative edge on CEXn |
| X | X | 1 | 1 | 0 | 0 | 0 | X | Capture triggered by transition on CEXn |
| X | 1 | 0 | 0 | 1 | 0 | 0 | X | Software Timer |
| X | 1 | 0 | 0 | 1 | 1 | 0 | X | High Speed Output |
| X | 1 | 0 | 0 | 0 | 1 | 1 | X | Frequency Output |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 8-Bit Pulse Width Modulator |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 16-Bit Pulse Width Modulator |

X = Don't Care

### 24.2.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes PCA0 to capture the value of the PCA0 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 and an interrupt request is generated if CCF interrupts are 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.

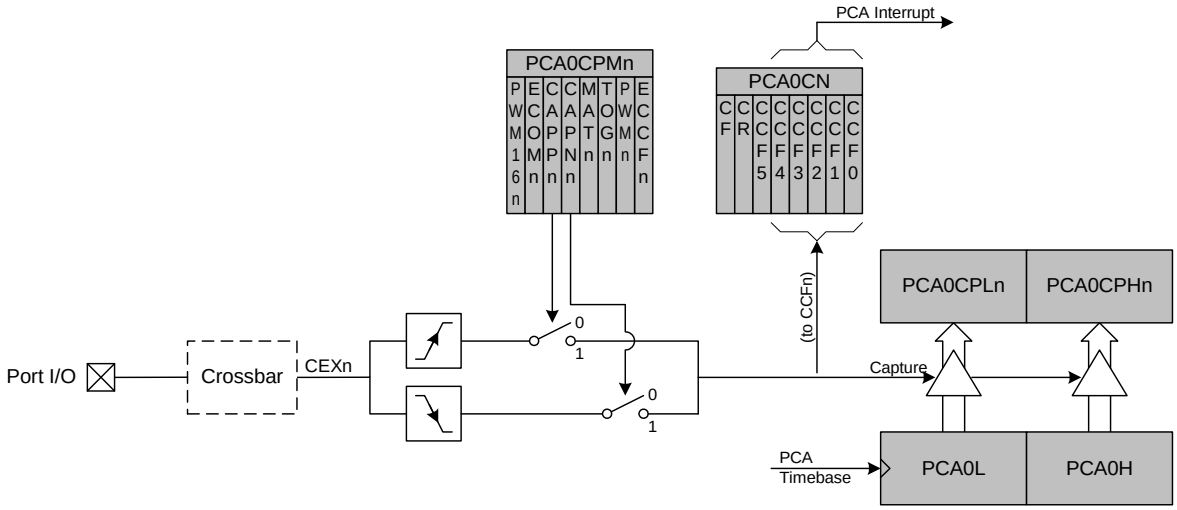



**Figure 24.4. PCA Capture Mode Diagram**

**Note:** The signal at CEXn must be high or low for at least 2 system clock cycles in order to be valid.

### 24.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA0 counter/timer 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 and an interrupt request is generated if CCF interrupts are 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'.

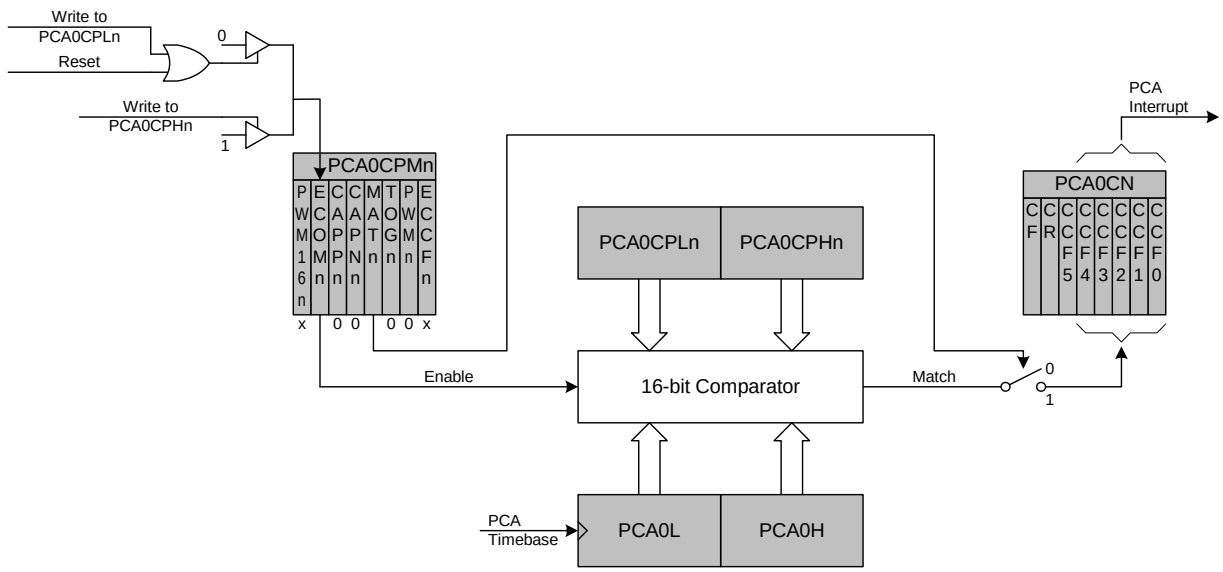



**Figure 24.5. PCA Software Timer Mode Diagram**

### 24.2.3. 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) Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output 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'.

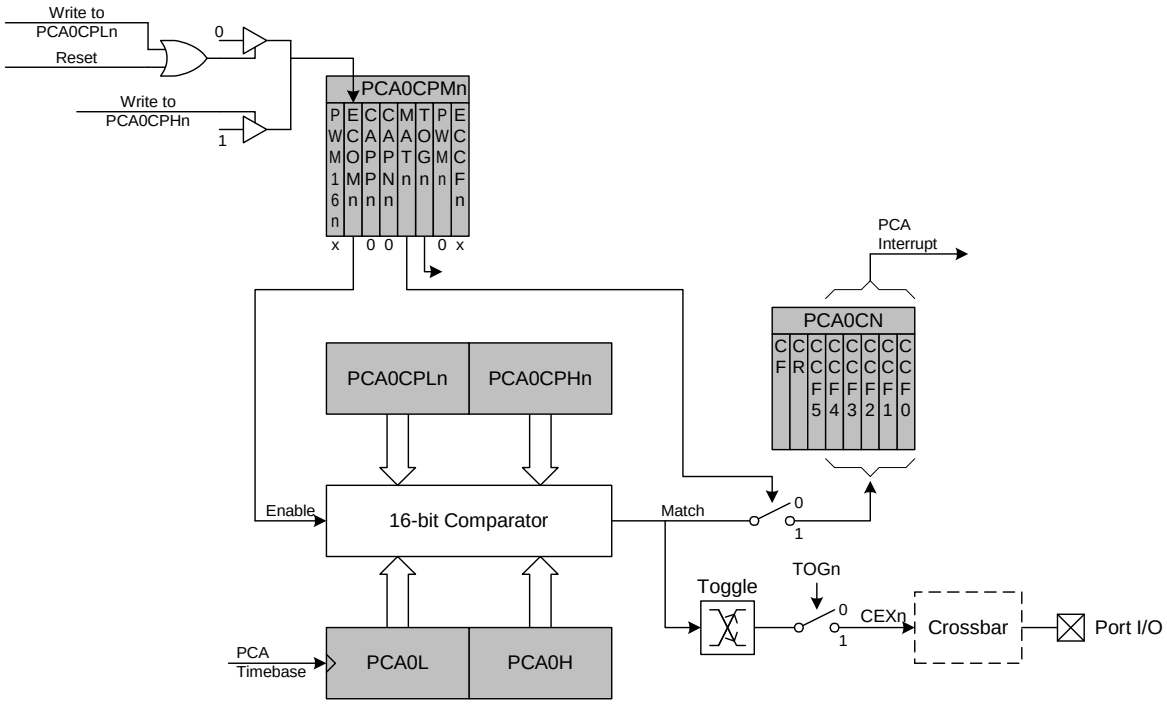



**Figure 24.6. PCA High Speed Output Mode Diagram**

### 24.2.4. 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 24.1.

**Equation 24.1. Square Wave Frequency Output**

$$F_{sqr} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

**Note:** A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

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 PCA0 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.

**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'.

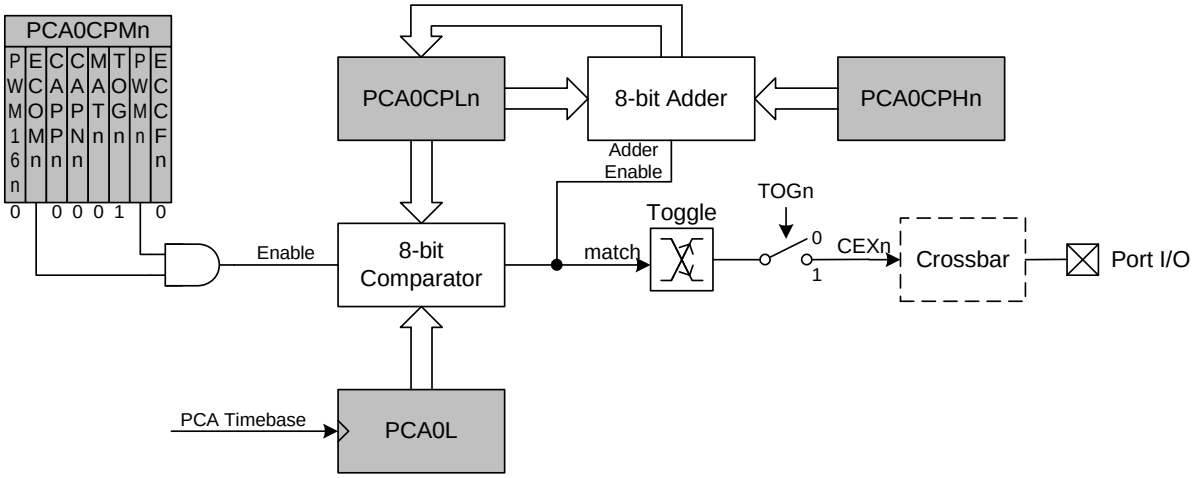



**Figure 24.7. PCA Frequency Output Mode**

### 24.2.5. 8-Bit Pulse Width Modulator Mode

Each module can be used independently to generate pulse width modulated (PWM) outputs on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA0 counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA0 counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be high. When the count value in PCA0L overflows, the CEXn output will be low (see Figure 24.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the counter/timer's high byte (PCA0H) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register enables 8-Bit Pulse Width Modulator mode. The duty cycle for 8-Bit PWM Mode is given by Equation 24.2.

**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'.

**Equation 24.2. 8-Bit PWM Duty Cycle**

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

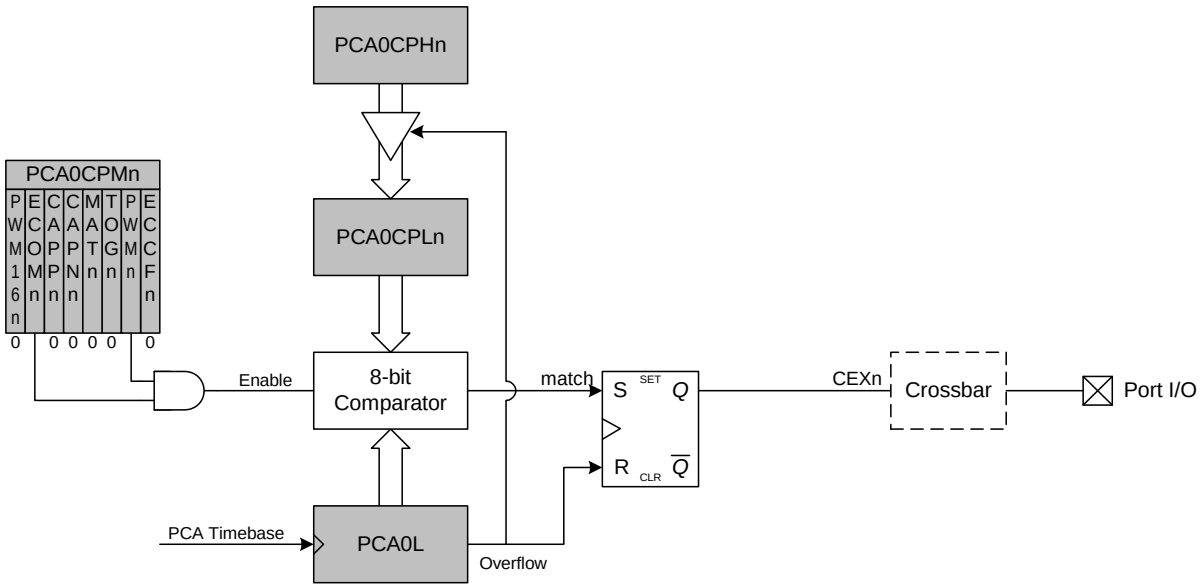



**Figure 24.8. PCA 8-Bit PWM Mode Diagram**

### 24.2.6. 16-Bit Pulse Width Modulator Mode

Each PCA0 module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA0 clocks for the low time of the PWM signal. When the PCA0 counter matches the module contents, the output on CEXn is asserted high; when the counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA0 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, CCFn should also be set to logic 1 to enable match interrupts. The duty cycle for 16-Bit PWM Mode is given by Equation 24.3.

**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’.

**Equation 24.3. 16-Bit PWM Duty Cycle**

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

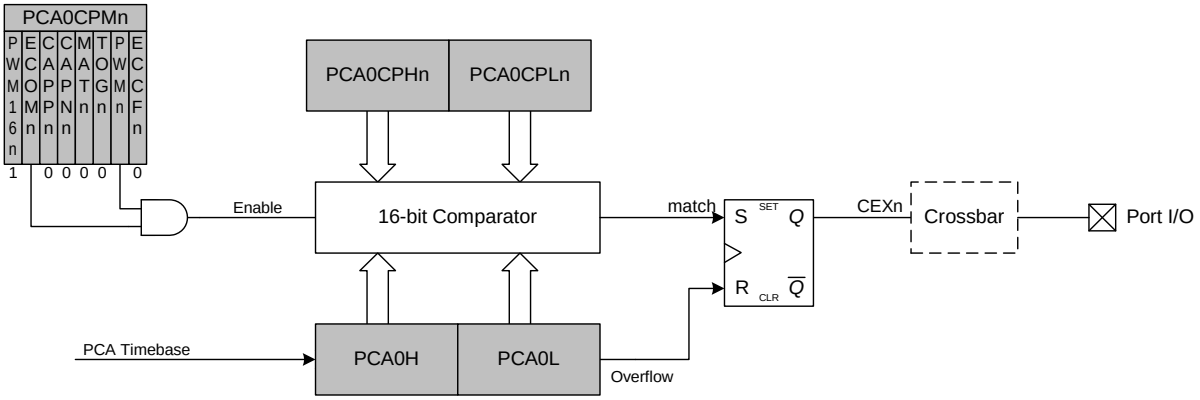



**Figure 24.9. PCA 16-Bit PWM Mode**

## 24.3. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of PCA0.

### SFR Definition 24.1. PCA0CN: PCA Control

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CF | CR | CCF5 | CCF4 | CCF3 | CCF2 | CCF1 | CCF0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD8
SFR Page: 0

Bit7: CF: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA0 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 CF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: CR: PCA0 Counter/Timer Run Control.
This bit enables/disables the PCA0 Counter/Timer.
0: PCA0 Counter/Timer disabled.
1: PCA0 Counter/Timer enabled.

Bit5: CCF5: PCA0 Module 5 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit4: CCF4: PCA0 Module 4 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit3: CCF3: PCA0 Module 3 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: CCF2: PCA0 Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit1: CCF1: PCA0 Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit0: CCF0: PCA0 Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

## SFR Definition 24.2. PCA0MD: PCA0 Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CIDL | - | - | - | CPS2 | CPS1 | CPS0 | ECF | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD9
SFR Page: 0

Bit7: CIDL: PCA0 Counter/Timer Idle Control.
Specifies PCA0 behavior when CPU is in Idle Mode.
0: PCA0 continues to function normally while the system controller is in Idle Mode.
1: PCA0 operation is suspended while the system controller is in Idle Mode.

Bits6–4: UNUSED. Read = 000b, Write = don't care.

Bits3–1: CPS2-CPS0: PCA0 Counter/Timer Pulse Select.
These bits select the timebase source for the PCA0 counter

| 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 clock divided by 8 (synchronized with system clock) |
| 1 | 1 | 0 | Reserved |
| 1 | 1 | 1 | Reserved |

Bit0: ECF: PCA Counter/Timer Overflow Interrupt Enable.
This bit sets the masking of the PCA0 Counter/Timer Overflow (CF) interrupt.
0: Disable the CF interrupt.
1: Enable a PCA0 Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

## SFR Definition 24.3. PCA0CPMn: PCA0 Capture/Compare Mode

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PWM16n | ECOMn | CAPPn | CAPNn | MATn | TOGn | PWMn | ECCFn | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: PCA0CPM0: 0xDA, PCA0CPM1: 0xDB, PCA0CPM2: 0xDC, PCA0CPM3: 0xDD, PCA0CPM4: 0xDE, PCA0CPM5: 0xDF

SFR Page: PCA0CPM0: page 0, PCA0CPM1: page 0, PCA0CPM2: page 0, PCA0CPM3: 0, PCA0CPM4: page 0, PCA0CPM5: page 0

Bit7: PWM16n: 16-bit Pulse Width Modulation Enable
This bit selects 16-bit mode when Pulse Width Modulation mode is enabled (PWMn = 1).
0: 8-bit PWM selected.
1: 16-bit PWM selected.

Bit6: ECOMn: Comparator Function Enable.
This bit enables/disables the comparator function for PCA0 module n.
0: Disabled.
1: Enabled.

Bit5: CAPPn: Capture Positive Function Enable.
This bit enables/disables the positive edge capture for PCA0 module n.
0: Disabled.
1: Enabled.

Bit4: CAPNn: Capture Negative Function Enable.
This bit enables/disables the negative edge capture for PCA0 module n.
0: Disabled.
1: Enabled.

Bit3: MATn: Match Function Enable.
This bit enables/disables the match function for PCA0 module n. When enabled, matches of the PCA0 counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
0: Disabled.
1: Enabled.

Bit2: TOGn: Toggle Function Enable.
This bit enables/disables the toggle function for PCA0 module n. When enabled, matches of the PCA0 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.
0: Disabled.
1: Enabled.

Bit1: PWMn: Pulse Width Modulation Mode Enable.
This bit enables/disables the PWM function for PCA0 module n. When enabled, a pulse width modulated signal is output on the CEXn pin. 8-bit PWM is used if PWM16n is logic 0; 16-bit mode is used if PWM16n logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit0: 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.

### SFR Definition 24.4. PCA0L: PCA0 Counter/Timer Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xF9
SFR Page: 0

Bits 7–0: PCA0L: PCA0 Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA0 Counter/Timer.

### SFR Definition 24.5. PCA0H: PCA0 Counter/Timer High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xFA
SFR Page: 0

Bits 7–0: PCA0H: PCA0 Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA0 Counter/Timer.

### SFR Definition 24.6. PCA0CPLn: PCA0 Capture Module Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: PCA0CPL0: 0xFB, PCA0CPL1: 0xFD, PCA0CPL2: 0xE9, PCA0CPL3: 0xEB, PCA0CPL4: 0xED, PCA0CPL5: 0xE1

SFR Page: PCA0CPL0: page 0, PCA0CPL1: page 0, PCA0CPL2: page 0, PCA0CPL3: page 0, PCA0CPL4: page 0, PCA0CPL5: page 0

Bits7–0: PCA0CPLn: PCA0 Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

## SFR Definition 24.7. PCA0CPHn: PCA0 Capture Module High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: PCA0CPH0: 0xFC, PCA0CPH1: 0xFD, PCA0CPH2: 0xEA, PCA0CPH3: 0xEC, PCA0CPH4: 0xEE, PCA0CPH5: 0xE2

SFR Page: PCA0CPH0: page 0, PCA0CPH1: page 0, PCA0CPH2: page 0, PCA0CPH3: page 0, PCA0CPH4: page 0, PCA0CPH5: page 0

Bits7–0: PCA0CPHn: PCA0 Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

## NOTES:

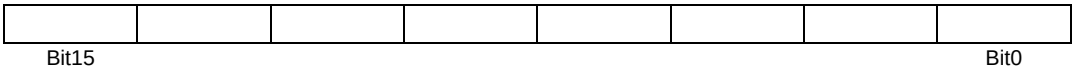
## 25. JTAG (IEEE 1149.1)

Each MCU has an on-chip JTAG interface and logic to support boundary scan for production and in-system testing, Flash read/write operations, and non-intrusive in-circuit debug. The JTAG interface is fully compliant with the IEEE 1149.1 specification. Refer to this specification for detailed descriptions of the Test Interface and Boundary-Scan Architecture. Access of the JTAG Instruction Register (IR) and Data Registers (DR) are as described in the Test Access Port and Operation of the IEEE 1149.1 specification.

The JTAG interface is accessed via four dedicated pins on the MCU: TCK, TMS, TDI, and TDO.

Through the 16-bit JTAG Instruction Register (IR), any of the eight instructions shown in Figure 25.1 can be commanded. There are three DR's associated with JTAG Boundary-Scan, and four associated with Flash read/write operations on the MCU.

### JTAG Register Definition 25.1. IR: JTAG Instruction Register




| IR Value | Instruction | Description |
|---|---|---|
| 0x0000 | EXTEST | Selects the Boundary Data Register for control and observability of all device pins |
| 0x0002 | SAMPLE/ PRELOAD | Selects the Boundary Data Register for observability and presetting the scan-path latches |
| 0x0004 | IDCODE | Selects device ID Register |
| 0xFFFF | BYPASS | Selects Bypass Data Register |
| 0x0082 | Flash Control | Selects FLASHCON Register to control how the interface logic responds to reads and writes to the FLASHDAT Register |
| 0x0083 | Flash Data | Selects FLASHDAT Register for reads and writes to the Flash memory |
| 0x0084 | Flash Address | Selects FLASHADR Register which holds the address of all Flash read, write, and erase operations |
| 0x0085 | Flash Scale | Selects FLASHSCL Register which controls the Flash one-shot timer and read-always enable |

## 25.1. Boundary Scan

The DR in the Boundary Scan path is an 134-bit shift register. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature via the EXTEST and SAMPLE commands.

**Table 25.1. Boundary Data Register Bit Definitions**

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

| Bit | Action | Target |
|---|---|---|
| 0 | Capture | Reset Enable from MCU (64-pin TQFP devices) |
| | Update | Reset Enable to $\overline{\text{RST}}$ pin (64-pin TQFP devices) |
| 1 | Capture | Reset input from $\overline{\text{RST}}$ pin (64-pin TQFP devices) |
| | Update | Reset output to $\overline{\text{RST}}$ pin (64-pin TQFP devices) |
| 2 | Capture | Reset Enable from MCU (100-pin TQFP devices) |
| | Update | Reset Enable to $\overline{\text{RST}}$ pin (100-pin TQFP devices) |
| 3 | Capture | Reset input from $\overline{\text{RST}}$ pin (100-pin TQFP devices) |
| | Update | Reset output to $\overline{\text{RST}}$ pin (100-pin TQFP devices) |
| 4 | Capture | External Clock from XTAL1 pin |
| | Update | Not used |
| 5 | Capture | Weak pullup enable from MCU |
| | Update | Weak pullup enable to Port Pins |
| 6, 8, 10, 12, 14, 16, 18, 20 | Capture | P0.n output enable from MCU (e.g. Bit6=P0.0, Bit8=P0.1, etc.) |
| | Update | P0.n output enable to pin (e.g. Bit6=P0.0oe, Bit8=P0.1oe, etc.) |
| 7, 9, 11, 13, 15, 17, 19, 21 | Capture | P0.n input from pin (e.g. Bit7=P0.0, Bit9=P0.1, etc.) |
| | Update | P0.n output to pin (e.g. Bit7=P0.0, Bit9=P0.1, etc.) |
| 22, 24, 26, 28, 30, 32, 34, 36 | Capture | P1.n output enable from MCU |
| | Update | P1.n output enable to pin |
| 23, 25, 27, 29, 31, 33, 35, 37 | Capture | P1.n input from pin |
| | Update | P1.n output to pin |
| 38, 40, 42, 44, 46, 48, 50, 52 | Capture | P2.n output enable from MCU |
| | Update | P2.n output enable to pin |
| 39, 41, 43, 45, 47, 49, 51, 53 | Capture | P2.n input from pin |
| | Update | P2.n output to pin |
| 54, 56, 58, 60, 62, 64, 66, 68 | Capture | P3.n output enable from MCU |
| | Update | P3.n output enable to pin |
| 55, 57, 59, 61, 63, 65, 67, 69 | Capture | P3.n input from pin |
| | Update | P3.n output to pin |
| 70, 72, 74, 76, 78, 80, 82, 84 | Capture | P4.n output enable from MCU |
| | Update | P4.n output enable to pin |
| 71, 73, 75, 77, 79, 81, 83, 85 | Capture | P4.n input from pin |
| | Update | P4.n output to pin |
| 86, 88, 90, 92, 94, 96, 98, 100 | Capture | P5.n output enable from MCU |
| | Update | P5.n output enable to pin |
| 87, 89, 91, 93, 95, 97, 99, 101 | Capture | P5.n input from pin |
| | Update | P5.n output to pin |
| 102, 104, 106, 108, 110, 112, 114, 116 | Capture | P6.n output enable from MCU |
| | Update | P6.n output enable to pin |

**Table 25.1. Boundary Data Register Bit Definitions (Continued)**

| Bit | Action | Target |
|---|---|---|
| 103, 105, 107, 109, 111, 113, 115, 117 | Capture | P6.n input from pin |
| | Update | P6.n output to pin |
| 118, 120, 122, 124, 126, 128, 130, 132 | Capture | P7.n output enable from MCU |
| | Update | P7.n output enable to pin |
| 119, 121, 123, 125, 127, 129, 131, 133 | Capture | P7.n input from pin |
| | Update | P7.n output to pin |

### 25.1.1. EXTEST Instruction

The EXTEST instruction is accessed via the IR. The Boundary DR provides control and observability of all the device pins as well as the Weak Pullup feature. All inputs to on-chip logic are set to logic 1.

### 25.1.2. SAMPLE Instruction

The SAMPLE instruction is accessed via the IR. The Boundary DR provides observability and presetting of the scan-path latches.

### 25.1.3. BYPASS Instruction

The BYPASS instruction is accessed via the IR. It provides access to the standard JTAG Bypass data register.

### 25.1.4. IDCODE Instruction

The IDCODE instruction is accessed via the IR. It provides access to the 32-bit Device ID register.

**JTAG Register Definition 25.2. DEVICEID: JTAG Device ID**

| Version | Part Number | Manufacturer ID | 1 | Reset Value |
|---|---|---|---|---|
| Bit31 Bit28 | Bit27 Bit12 | Bit11 Bit1 | Bit0 | 0xn0003243 |

Version = 0000b

Part Number = 0000 0000 0000 0111b (C8051F120/1/2/3/4/5/6/7 or C8051F130/1/2/3)

Manufacturer ID = 0010 0100 001b (Silicon Labs)

## 25.2. Flash Programming Commands

The Flash memory can be programmed directly over the JTAG interface using the Flash Control, Flash Data, Flash Address, and Flash Scale registers. These Indirect Data Registers are accessed via the JTAG Instruction Register. Read and write operations on indirect data registers are performed by first setting the appropriate DR address in the IR register. Each read or write is then initiated by writing the appropriate Indirect Operation Code (IndOpCode) to the selected data register. Incoming commands to this register have the following format:

| 19:18 | 17:0 |
|---|---|
| IndOpCode | WriteData |

IndOpCode: These bit set the operation to perform according to the following table:

| IndOpCode | Operation |
|---|---|
| 0x | Poll |
| 10 | Read |
| 11 | Write |

The Poll operation is used to check the Busy bit as described below. Although a Capture-DR is performed, no Update-DR is allowed for the Poll operation. Since updates are disabled, polling can be accomplished by shifting in/out a single bit.

The Read operation initiates a read from the register addressed by the DRAddress. Reads can be initiated by shifting only 2 bits into the indirect register. After the read operation is initiated, polling of the Busy bit must be performed to determine when the operation is complete.

The write operation initiates a write of WriteData to the register addressed by DRAddress. Registers of any width up to 18 bits can be written. If the register to be written contains fewer than 18 bits, the data in WriteData should be left-justified, i.e. its MSB should occupy bit 17 above. This allows shorter registers to be written in fewer JTAG clock cycles. For example, an 8-bit register could be written by shifting only 10 bits. After a Write is initiated, the Busy bit should be polled to determine when the next operation can be initiated. The contents of the Instruction Register should not be altered while either a read or write operation is busy.

Outgoing data from the indirect Data Register has the following format:

| 19 | 18:1 | 0 |
|---|---|---|
| 0 | ReadData | Busy |

The Busy bit indicates that the current operation is not complete. It goes high when an operation is initiated and returns low when complete. Read and Write commands are ignored while Busy is high. In fact, if polling for Busy to be low will be followed by another read or write operation, JTAG writes of the next operation can be made while checking for Busy to be low. They will be ignored until Busy is read low, at which time the new operation will initiate. This bit is placed ate bit 0 to allow polling by single-bit shifts. When waiting for a Read to complete and Busy is 0, the following 18 bits can be shifted out to obtain the resulting data. ReadData is always right-justified. This allows registers shorter than 18 bits to be read using a reduced number of shifts. For example, the results from a byte-read requires 9 bit shifts (Busy + 8 bits).

## JTAG Register Definition 25.3. FLASHCON: JTAG Flash Control

Reset Value

| SFLE | WRMD2 | WRMD1 | WRMD0 | RDMD3 | RDMD2 | RDMD1 | RDMD0 | 00000000 |
|---|---|---|---|---|---|---|---|---|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

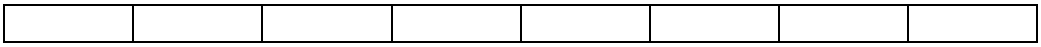
Bit7: SFLE: Scratchpad Flash Memory Access Enable
When this bit is set, Flash reads and writes are directed to the two 128-byte Scratchpad Flash sectors. When SFLE is set to logic 1, Flash accesses out of the address range 0x00-0xFF should not be attempted (with the exception of address 0x400, which can be used to simultaneously erase both Scratchpad areas). Reads/Writes out of this range will yield undefined results.
0: Flash access directed to the Program/Data Flash sector.
1: Flash access directed to the two 128 byte Scratchpad sectors.

Bits6–4: WRMD2–0: Write Mode Select Bits.
The Write Mode Select Bits control how the interface logic responds to writes to the FLASHDAT Register per the following values:
000: A FLASHDAT write replaces the data in the FLASHDAT register, but is otherwise ignored.
001: A FLASHDAT write initiates a write of FLASHDAT into the memory address by the FLASHADR register. FLASHADR is incremented by one when complete.
010: A FLASHDAT write initiates an erasure (sets all bytes to 0xFF) of the Flash page containing the address in FLASHADR. The data written must be 0xA5 for the erase to occur. FLASHADR is not affected. If FLASHADR = 0x1FBFE – 0x1FBFF, the entire user space will be erased (i.e. entire Flash memory except for Reserved area 0x1FC00 – 0x1FFFF).
(All other values for WRMD2-0 are reserved.)

Bits3–0: RDMD3–0: Read Mode Select Bits.
The Read Mode Select Bits control how the interface logic responds to reads from the FLASHDAT Register per the following values:
0000: A FLASHDAT read provides the data in the FLASHDAT register, but is otherwise ignored.
0001: A FLASHDAT read initiates a read of the byte addressed by the FLASHADR register if no operation is currently active. This mode is used for block reads.
0010: A FLASHDAT read initiates a read of the byte addressed by FLASHADR only if no operation is active and any data from a previous read has already been read from FLASHDAT. This mode allows single bytes to be read (or the last byte of a block) without initiating an extra read.
(All other values for RDMD3–0 are reserved.)

### JTAG Register Definition 25.4. FLASHDAT: JTAG Flash Data

Reset Value
0000000000

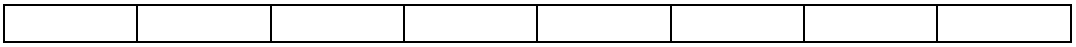



This register is used to read or write data to the Flash memory across the JTAG interface.

Bits9–2: DATA7–0: Flash Data Byte.
Bit1: FAIL: Flash Fail Bit.
0: Previous Flash memory operation was successful.
1: Previous Flash memory operation failed. Usually indicates the associated memory location was locked.
Bit0: BUSY: Flash Busy Bit.
0: Flash interface logic is not busy.
1: Flash interface logic is processing a request. Reads or writes while BUSY = 1 will not initiate another operation.

### JTAG Register Definition 25.5. FLASHADR: JTAG Flash Address

Reset Value
0x00000




This register holds the address for all JTAG Flash read, write, and erase operations. This register autoincrements after each read or write, regardless of whether the operation succeeded or failed.

Bits15–0: Flash Operation 17-bit Address.

## 25.3. Debug Support

Each MCU has on-chip JTAG and debug logic that provides non-intrusive, full speed, in-circuit debug support using the production part installed in the end application, via the four pin JTAG I/F. Silicon Labs' debug system supports inspection and modification of memory and registers, breakpoints, and single stepping. No additional target RAM, program memory, or communications channels are required. All the digital and analog peripherals are functional and work correctly (remain synchronized) while debugging. The Watchdog Timer (WDT) is disabled when the MCU is halted during single stepping or at a breakpoint.

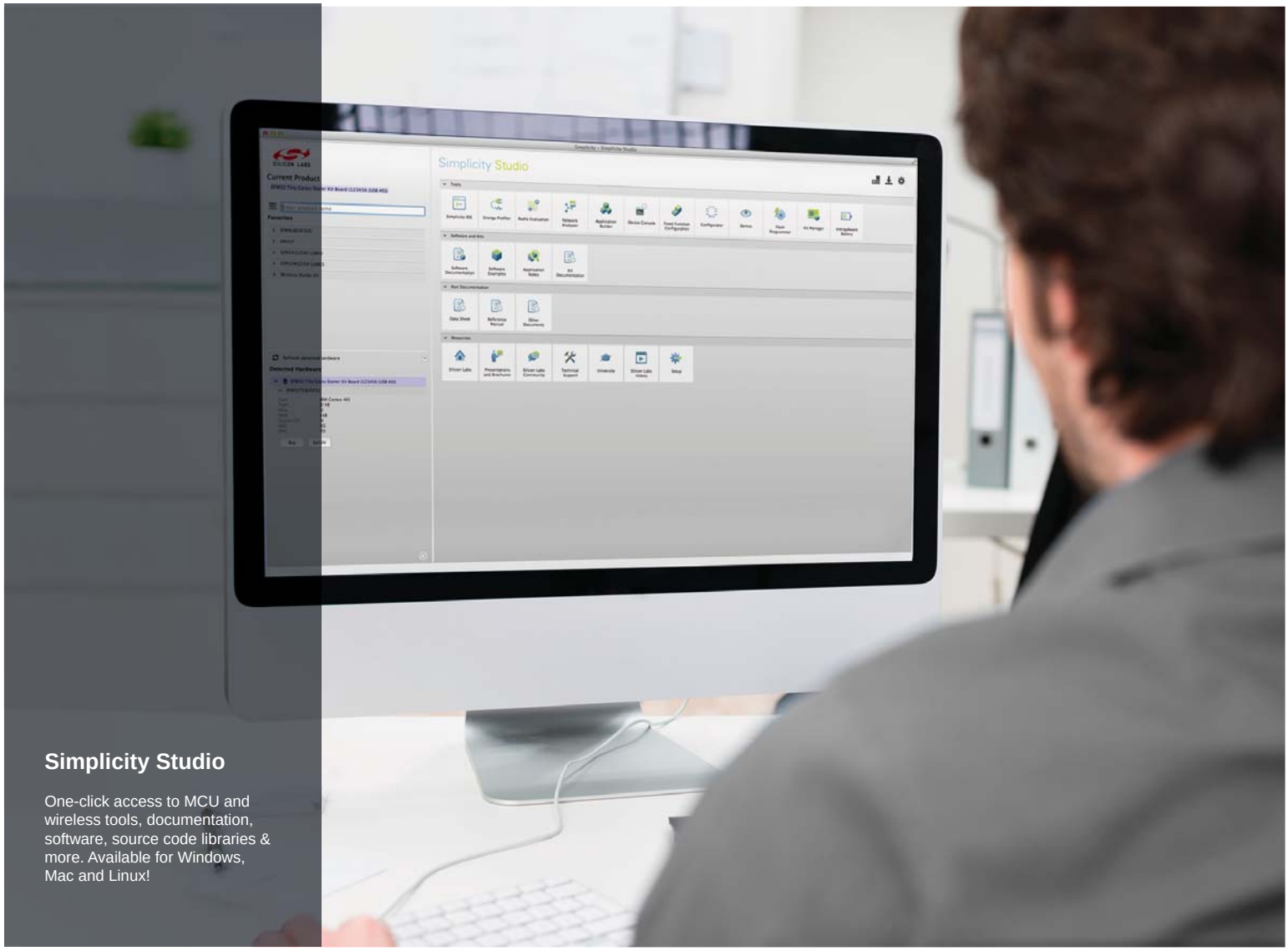
The C8051F120DK is a development kit with all the hardware and software necessary to develop application code and perform in-circuit debug with each MCU in the C8051F12x and C8051F13x device families. Each kit includes development software for the PC, a Serial Adapter (for connection to JTAG) and a target application board with a C8051F120 installed. Serial cables and wall-mount power supply are also included.

## NOTES:

# Document Change List

## Revision 1.3 to Revision 1.4

- Added new paragraph tags: SFR Definition and JTAG Register Definition.
- Product Selection Guide Table 1.1: Added RoHS-compliant ordering information.
- Overview Chapter, Figure 1.8, “On-Chip Memory Map”: Corrected on-chip XRAM size to “8192 Bytes”.
- SAR8 Chapter: Table 7.1, “ADC2 Electrical Characteristics”: Track/Hold minimum spec corrected to “300 ns”.
- SAR8 Chapter: Table 7.1, “ADC2 Electrical Characteristics”: Total Harmonic Distortion typical spec corrected to “-51 dB”.
- Oscillators Chapter, Figure 14.1, “Oscillator Diagram”: Corrected location of IOSCEN arrow.
- CIP51 Chapter, **Section 11.3**: Added note describing EA change behavior when followed by single-cycle instruction.
- CIP51 Chapter, Interrupt Summary Table: Added “SFRPAGE” column and SFRPAGE value for each interrupt source.
- CIP-51 Chapter, Figure 11.2, “Memory Map”: Corrected on-chip XRAM size to “8192 Bytes”.
- Port I/O Chapter, Crossbar Priority Figures: Character formatting problem corrected.
- Port I/O Chapter, P7MDOUT Register Description: Removed references to UART and SMBus peripherals.
- Port I/O Chapter, P3MDOUT Register Description: Corrected text to read “P3MDOUT.[7:0]”.
- Timers Chapter: References to “TnCON” corrected to read “TMRnCN”.
- PCA0 Chapter, Section 24.1: Added note about PCA0CN Register and effects of read-modify-write instructions on the CF bit.




## Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

**IoT Portfolio**
*www.silabs.com/IoT*

**SW/HW**
*www.silabs.com/simplicity*

**Quality**
*www.silabs.com/quality*

**Support and Community**
*community.silabs.com*

**Disclaimer**
Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

**Trademark Information**
Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem ®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

**Silicon Laboratories Inc.**
**400 West Cesar Chavez**
**Austin, TX 78701**
**USA**

**http://www.silabs.com**