# C8051F060/1/2/3/4/5/6/7

## 8K ISP FLASH MCU Family

### Analog Peripherals

- **Two 16-Bit SAR ADCs**
  - 16-bit resolution
  - ±0.75 LSB INL, guaranteed no missing codes
  - Programmable throughput up to 1 Msps
  - Operate as two single-ended or one differential converter
  - Direct memory access; data stored in RAM without software overhead
  - Data-dependent windowed interrupt generator
- **10-bit SAR ADC (C8051F060/1/2/3)**
  - Programmable throughput up to 200 ksps
  - 8 external inputs, single-ended or differential
  - Built-in temperature sensor
- **Two 12-bit DACs (C8051F060/1/2/3)**
  - Can synchronize outputs to timers for jitter-free waveform generation
- **Three Analog Comparators**
  - Programmable hysteresis/response time
- **Voltage Reference**
- **Precision VDD 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

### High Speed 8051 µC Core

- Pipelined instruction architecture; executes 70% of instruction set in 1 or 2 system clocks
- Up to 25 MIPS throughput with 25 MHz clock
- Flexible Interrupt sources

### Memory

- 4352 Bytes internal data RAM (4 k + 256)
- 64 kB (C8051F060/1/2/3/4/5), 32 kB (C8051F066/7) Flash; In-system programmable in 512-byte sectors
- External 64 kB data memory interface with multiplexed and non-multiplexed modes (C8051F060/2/4/6)

### Digital Peripherals

- 59 general purpose I/O pins (C8051F060/2/4/6)
- 24 general purpose I/O pins (C8051F061/3/5/7)
- Bosch Controller Area Network (CAN 2.0B - C8051F060/1/2/3)
- 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

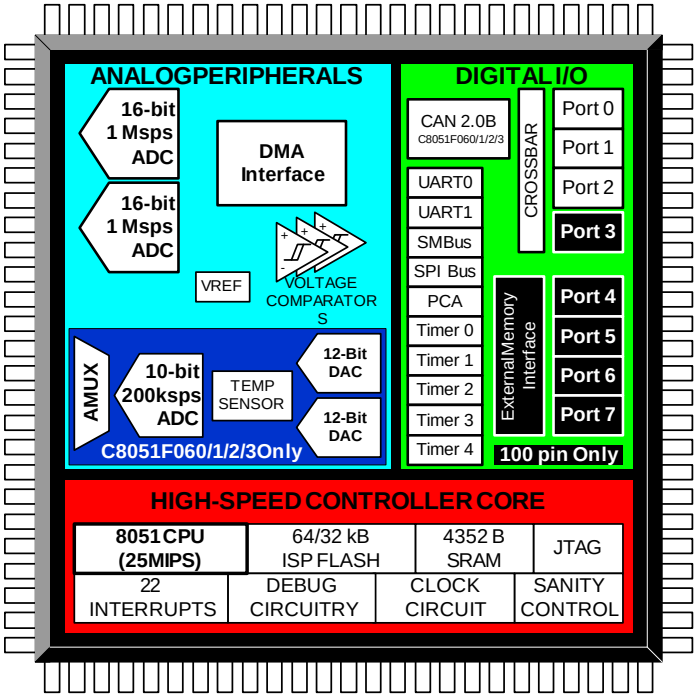
- Internal calibrated precision oscillator: 24.5 MHz
- External oscillator: Crystal, RC, C, or clock

**Supply Voltage ..... 2.7 to 3.6 V**

- Multiple power saving sleep and shutdown modes

**100-Pin and 64-Pin TQFP Packages Available**

**Temperature Range: -40 to +85 °C**





Copyright © 2003 by Silicon Laboratories

## Table of Contents


**1. System Overview ..... 19**
1.1. CIP-51™ Microcontroller Core ..... 25
1.1.1. Fully 8051 Compatible ..... 25
1.1.2. Improved Throughput ..... 25
1.1.3. Additional Features ..... 26
1.2. On-Chip Memory ..... 27
1.3. JTAG Debug and Boundary Scan ..... 28
1.4. Programmable Digital I/O and Crossbar ..... 29
1.5. Programmable Counter Array ..... 30
1.6. Controller Area Network ..... 31
1.7. Serial Ports ..... 32
1.8. 16-Bit Analog to Digital Converters ..... 33
1.9. 10-Bit Analog to Digital Converter ..... 34
1.10.12-bit Digital to Analog Converters ..... 35
1.11.Analog Comparators ..... 36
**2. Absolute Maximum Ratings ..... 37**
**3. Global DC Electrical Characteristics ..... 38**
**4. Pinout and Package Definitions ..... 39**
**5. 16-Bit ADCs (ADC0 and ADC1) ..... 51**
5.1. Single-Ended or Differential Operation ..... 52
5.1.1. Pseudo-Differential Inputs ..... 52
5.2. Voltage Reference ..... 53
5.3. ADC Modes of Operation ..... 54
5.3.1. Starting a Conversion ..... 54
5.3.2. Tracking Modes ..... 54
5.3.3. Settling Time Requirements ..... 56
5.4. Calibration ..... 66
5.5. ADC0 Programmable Window Detector ..... 69
**6. Direct Memory Access Interface (DMA0) ..... 75**
6.1. Writing to the Instruction Buffer ..... 75
6.2. DMA0 Instruction Format ..... 76
6.3. XRAM Addressing and Setup ..... 76
6.4. Instruction Execution in Mode 0 ..... 77
6.5. Instruction Execution in Mode 1 ..... 78
6.6. Interrupt Sources ..... 79
6.7. Data Buffer Overflow Warnings and Errors ..... 79
**7. 10-Bit ADC (ADC2, C8051F060/1/2/3) ..... 87**
7.1. Analog Multiplexer ..... 88
7.2. Modes of Operation ..... 89
7.2.1. Starting a Conversion ..... 89
7.2.2. Tracking Modes ..... 90
7.2.3. Settling Time Requirements ..... 91

7.3. Programmable Window Detector ..... 97
7.3.1. Window Detector In Single-Ended Mode ..... 99
7.3.2. Window Detector In Differential Mode ..... 100
**8. DACs, 12-Bit Voltage Mode (DAC0 and DAC1, C8051F060/1/2/3) ..... 103**
8.1. DAC Output Scheduling ..... 104
8.1.1. Update Output On-Demand ..... 104
8.1.2. Update Output Based on Timer Overflow ..... 104
8.2. DAC Output Scaling/Justification ..... 104
**9. Voltage Reference 2 (C8051F060/2) ..... 111**
**10. Voltage Reference 2 (C8051F061/3) ..... 113**
**11. Voltage Reference 2 (C8051F064/5/6/7) ..... 115**
**12. Comparators ..... 117**
12.1.Comparator Inputs ..... 119
**13. CIP-51 Microcontroller ..... 123**
13.1.Instruction Set ..... 125
13.1.1.Instruction and CPU Timing ..... 125
13.1.2.MOVX Instruction and Program Memory ..... 125
13.2.Memory Organization ..... 130
13.2.1.Program Memory ..... 130
13.2.2.Data Memory ..... 131
13.2.3.General Purpose Registers ..... 131
13.2.4.Bit Addressable Locations ..... 131
13.2.5.Stack ..... 131
13.2.6.Special Function Registers ..... 132
13.2.6.1.SFR Paging ..... 132
13.2.6.2.Interrupts and SFR Paging ..... 132
13.2.6.3.SFR Page Stack Example ..... 134
13.2.7.Register Descriptions ..... 148
13.3.Interrupt Handler ..... 151
13.3.1.MCU Interrupt Sources and Vectors ..... 151
13.3.2.External Interrupts ..... 151
13.3.3.Interrupt Priorities ..... 153
13.3.4.Interrupt Latency ..... 153
13.3.5.Interrupt Register Descriptions ..... 154
13.4.Power Management Modes ..... 160
13.4.1.Idle Mode ..... 160
13.4.2.Stop Mode ..... 161
**14. Reset Sources ..... 163**
14.1.Power-on Reset ..... 164
14.2.Power-fail Reset ..... 164
14.3.External Reset ..... 164
14.4.Missing Clock Detector Reset ..... 165
14.5.Comparator0 Reset ..... 165
14.6.External CNVSTR2 Pin Reset ..... 165
14.7.Watchdog Timer Reset ..... 165

14.7.1.Enable/Reset WDT ..... 166
14.7.2.Disable WDT ..... 166
14.7.3.Disable WDT Lockout ..... 166
14.7.4.Setting WDT Interval ..... 166
**15. Oscillators ..... 171**
15.1.Programmable Internal Oscillator ..... 171
15.2.External Oscillator Drive Circuit ..... 173
15.3.System Clock Selection ..... 173
15.4.External Crystal Example ..... 175
15.5.External RC Example ..... 175
15.6.External Capacitor Example ..... 175
**16. Flash Memory ..... 177**
16.1.Programming The Flash Memory ..... 177
16.2.Non-volatile Data Storage ..... 178
16.3.Security Options ..... 179
16.3.1.Summary of Flash Security Options ..... 183
**17. External Data Memory Interface and On-Chip XRAM ..... 187**
17.1.Accessing XRAM ..... 187
17.1.1.16-Bit MOVX Example ..... 187
17.1.2.8-Bit MOVX Example ..... 187
17.2.Configuring the External Memory Interface ..... 188
17.3.Port Selection and Configuration ..... 188
17.4.Multiplexed and Non-multiplexed Selection ..... 190
17.4.1.Multiplexed Configuration ..... 190
17.4.2.Non-multiplexed Configuration ..... 191
17.5.Memory Mode Selection ..... 192
17.5.1.Internal XRAM Only ..... 192
17.5.2.Split Mode without Bank Select ..... 192
17.5.3.Split Mode with Bank Select ..... 193
17.5.4.External Only ..... 193
17.6.Timing ..... 194
17.6.1.Non-multiplexed Mode ..... 196
17.6.1.1.16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’. ..... 196
17.6.1.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘101’ or ‘111’. ..... 197
17.6.1.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘110’. ..... 198
17.6.2.Multiplexed Mode ..... 199
17.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’. ..... 199
17.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’. ..... 200
17.6.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘010’. ..... 201
**18. Port Input/Output ..... 203**
18.1.Ports 0 through 3 and the Priority Crossbar Decoder ..... 205
18.1.1.Crossbar Pin Assignment and Allocation ..... 205
18.1.2.Configuring the Output Modes of the Port Pins ..... 206
18.1.3.Configuring Port Pins as Digital Inputs ..... 207
18.1.4.Weak Pull-ups ..... 207

18.1.5.Configuring Port 1 and 2 pins as Analog Inputs ..... 207
18.1.6.Crossbar Pin Assignment Example ..... 208
18.2.Ports 4 through 7 (C8051F060/2/4/6 only) ..... 219
18.2.1.Configuring Ports which are not Pinned Out ..... 219
18.2.2.Configuring the Output Modes of the Port Pins ..... 219
18.2.3.Configuring Port Pins as Digital Inputs ..... 219
18.2.4.Weak Pull-ups ..... 219
18.2.5.External Memory Interface ..... 220
**19. Controller Area Network (CAN0, C8051F060/1/2/3) ..... 225**
19.1.Bosch CAN Controller Operation ..... 227
19.2.CAN Registers ..... 228
19.2.1.CAN Controller Protocol Registers ..... 228
19.2.2.Message Object Interface Registers ..... 228
19.2.3.Message Handler Registers ..... 228
19.2.4.CIP-51 MCU Special Function Registers ..... 229
19.2.5.Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers 229
19.2.6.CAN0ADR Autoincrement Feature ..... 229
**20. System Management BUS / I2C BUS (SMBUS0) ..... 235**
20.1.Supporting Documents ..... 236
20.2.SMBus Protocol ..... 236
20.2.1.Arbitration ..... 237
20.2.2.Clock Low Extension ..... 237
20.2.3.SCL Low Timeout ..... 237
20.2.4.SCL High (SMBus Free) Timeout ..... 237
20.3.SMBus Transfer Modes ..... 238
20.3.1.Master Transmitter Mode ..... 238
20.3.2.Master Receiver Mode ..... 238
20.3.3.Slave Transmitter Mode ..... 239
20.3.4.Slave Receiver Mode ..... 239
20.4.SMBus Special Function Registers ..... 241
20.4.1.Control Register ..... 241
20.4.2.Clock Rate Register ..... 244
20.4.3.Data Register ..... 245
20.4.4.Address Register ..... 245
20.4.5.Status Register ..... 246
**21. Enhanced Serial Peripheral Interface (SPI0) ..... 251**
21.1.Signal Descriptions ..... 252
21.1.1.Master Out, Slave In (MOSI) ..... 252
21.1.2.Master In, Slave Out (MISO) ..... 252
21.1.3.Serial Clock (SCK) ..... 252
21.1.4.Slave Select (NSS) ..... 252
21.2.SPI0 Master Mode Operation ..... 253
21.3.SPI0 Slave Mode Operation ..... 255
21.4.SPI0 Interrupt Sources ..... 255

21.5.Serial Clock Timing ..... 256
21.6.SPI Special Function Registers ..... 258
**22. UART0 ..... 265**
22.1.UART0 Operational Modes ..... 266
22.1.1.Mode 0: Synchronous Mode ..... 266
22.1.2.Mode 1: 8-Bit UART, Variable Baud Rate ..... 267
22.1.3.Mode 2: 9-Bit UART, Fixed Baud Rate ..... 269
22.1.4.Mode 3: 9-Bit UART, Variable Baud Rate ..... 270
22.2.Multiprocessor Communications ..... 271
22.2.1.Configuration of a Masked Address ..... 271
22.2.2.Broadcast Addressing ..... 271
22.3.Frame and Transmission Error Detection ..... 272
**23. UART1 ..... 277**
23.1.Enhanced Baud Rate Generation ..... 278
23.2.Operational Modes ..... 279
23.2.1.8-Bit UART ..... 279
23.2.2.9-Bit UART ..... 280
23.3.Multiprocessor Communications ..... 281
**24. Timers ..... 287**
24.1.Timer 0 and Timer 1 ..... 287
24.1.1.Mode 0: 13-bit Counter/Timer ..... 287
24.1.2.Mode 1: 16-bit Counter/Timer ..... 289
24.1.3.Mode 2: 8-bit Counter/Timer with Auto-Reload ..... 289
24.1.4.Mode 3: Two 8-bit Counter/Timers (Timer 0 Only) ..... 290
24.2.Timer 2, Timer 3, and Timer 4 ..... 295
24.2.1.Configuring Timer 2, 3, and 4 to Count Down ..... 295
24.2.2.Capture Mode ..... 296
24.2.3.Auto-Reload Mode ..... 297
24.2.4.Toggle Output Mode ..... 298
**25. Programmable Counter Array ..... 303**
25.1.PCA Counter/Timer ..... 304
25.2.Capture/Compare Modules ..... 305
25.2.1.Edge-triggered Capture Mode ..... 306
25.2.2.Software Timer (Compare) Mode ..... 307
25.2.3.High Speed Output Mode ..... 308
25.2.4.Frequency Output Mode ..... 309
25.2.5.8-Bit Pulse Width Modulator Mode ..... 310
25.2.6.16-Bit Pulse Width Modulator Mode ..... 311
25.3.Register Descriptions for PCA0 ..... 312
**26. JTAG (IEEE 1149.1) ..... 317**
26.1.Boundary Scan ..... 318
26.1.1.EXTEST Instruction ..... 321
26.1.2.SAMPLE Instruction ..... 321
26.1.3.BYPASS Instruction ..... 321
26.1.4.IDCODE Instruction ..... 321

26.2.Flash Programming Commands ..... 322
26.3.Debug Support ..... 325
**27. Document Change List ..... 327**
27.1.Revision 1.1 to Revision 1.2 ..... 327

## List of Figures

**1. System Overview ..... 19**
Figure 1.1. C8051F060 / C8051F062 Block Diagram ..... 21
Figure 1.2. C8051F061 / C8051F063 Block Diagram ..... 22
Figure 1.3. C8051F064 / C8051F066 Block Diagram ..... 23
Figure 1.4. C8051F065 / C8051F067 Block Diagram ..... 24
Figure 1.5. Comparison of Peak MCU Execution Speeds ..... 25
Figure 1.6. On-Board Clock and Reset ..... 26
Figure 1.7. On-Chip Memory Map ..... 27
Figure 1.8. Development/In-System Debug Diagram ..... 28
Figure 1.9. Digital Crossbar Diagram ..... 29
Figure 1.10. PCA Block Diagram ..... 30
Figure 1.11. CAN Controller Overview ..... 31
Figure 1.12. 16-Bit ADC Block Diagram ..... 33
Figure 1.13. 10-Bit ADC Diagram ..... 34
Figure 1.14. DAC System Block Diagram ..... 35
Figure 1.15. Comparator Block Diagram ..... 36
**2. Absolute Maximum Ratings ..... 37**
**3. Global DC Electrical Characteristics ..... 38**
**4. Pinout and Package Definitions ..... 39**
Figure 4.1. C8051F060 / C8051F062 Pinout Diagram (TQFP-100) ..... 45
Figure 4.2. C8051F064 / C8051F066 Pinout Diagram (TQFP-100) ..... 46
Figure 4.3. TQFP-100 Package Drawing ..... 47
Figure 4.4. C8051F061 / C8051F063 Pinout Diagram (TQFP-64) ..... 48
Figure 4.5. C8051F065 / C8051F067 Pinout Diagram (TQFP-64) ..... 49
Figure 4.6. TQFP-64 Package Drawing ..... 50
**5. 16-Bit ADCs (ADC0 and ADC1) ..... 51**
Figure 5.1. 16-Bit ADC0 and ADC1 Control Path Diagram ..... 51
Figure 5.2. 16-bit ADC0 and ADC1 Data Path Diagram ..... 52
Figure 5.3. Voltage Reference Block Diagram ..... 53
Figure 5.4. ADC Track and Conversion Example Timing ..... 55
Figure 5.5. ADC0 and ADC1 Equivalent Input Circuits ..... 56
Figure 5.6. AMX0SL: AMUX Configuration Register ..... 57
Figure 5.7. ADC0CF: ADC0 Configuration Register ..... 58
Figure 5.8. ADC1CF: ADC1 Configuration Register ..... 59
Figure 5.9. ADC0CN: ADC0 Control Register ..... 60
Figure 5.10. ADC1CN: ADC1 Control Register ..... 61
Figure 5.11. REF0CN: Reference Control Register 0 ..... 62
Figure 5.12. REF1CN: Reference Control Register 1 ..... 62
Figure 5.13. ADC0H: ADC0 Data Word MSB Register ..... 63
Figure 5.14. ADC0L: ADC0 Data Word LSB Register ..... 63
Figure 5.15. ADC0 Data Word Example ..... 64
Figure 5.16. ADC1H: ADC1 Data Word MSB Register ..... 65

Figure 5.17. ADC1L: ADC1 Data Word LSB Register ..... 65
Figure 5.18. ADC1 Data Word Example ..... 65
Figure 5.19. Calibration Coefficient Locations ..... 66
Figure 5.20. Offset and Gain Register Mapping ..... 67
Figure 5.21. Offset and Gain Calibration Block Diagram ..... 67
Figure 5.22. ADC0CPT: ADC Calibration Pointer Register ..... 68
Figure 5.23. ADC0CCF: ADC Calibration Coefficient Register ..... 68
Figure 5.24. ADC0GTH: ADC0 Greater-Than Data High Byte Register ..... 69
Figure 5.25. ADC0GTL: ADC0 Greater-Than Data Low Byte Register ..... 69
Figure 5.26. ADC0LTH: ADC0 Less-Than Data High Byte Register ..... 70
Figure 5.27. ADC0LTL: ADC0 Less-Than Data Low Byte Register ..... 70
Figure 5.28. 16-Bit ADC0 Window Interrupt Example: Single-Ended Data ..... 71
Figure 5.29. 16-Bit ADC0 Window Interrupt Example: Differential Data ..... 72
**6. Direct Memory Access Interface (DMA0) ..... 75**
Figure 6.1. DMA0 Block Diagram ..... 75
Figure 6.2. DMA Mode 0 Operation ..... 77
Figure 6.3. DMA Mode 1 Operation ..... 78
Figure 6.4. DMA0CN: DMA0 Control Register ..... 80
Figure 6.5. DMA0CF: DMA0 Configuration Register ..... 81
Figure 6.6. DMA0IPT: DMA0 Instruction Write Address Register ..... 82
Figure 6.7. DMA0IDT: DMA0 Instruction Write Data Register ..... 82
Figure 6.8. DMA0BND: DMA0 Instruction Boundary Register ..... 83
Figure 6.9. DMA0ISW: DMA0 Instruction Status Register ..... 83
Figure 6.10. DMA0DAH: DMA0 Data Address Beginning MSB Register ..... 84
Figure 6.11. DMA0DAL: DMA0 Data Address Beginning LSB Register ..... 84
Figure 6.12. DMA0DSH: DMA0 Data Address Pointer MSB Register ..... 84
Figure 6.13. DMA0DSL: DMA0 Data Address Pointer LSB Register ..... 84
Figure 6.14. DMA0CTH: DMA0 Repeat Counter Limit MSB Register ..... 85
Figure 6.15. DMA0CTL: DMA0 Repeat Counter Limit LSB Register ..... 85
Figure 6.16. DMA0CSH: DMA0 Repeat Counter MSB Register ..... 85
Figure 6.17. DMA0CSL: DMA0 Repeat Counter LSB Register ..... 85
**7. 10-Bit ADC (ADC2, C8051F060/1/2/3) ..... 87**
Figure 7.1. ADC2 Functional Block Diagram ..... 87
Figure 7.2. Temperature Sensor Transfer Function ..... 89
Figure 7.3. 10-Bit ADC Track and Conversion Example Timing ..... 90
Figure 7.4. ADC2 Equivalent Input Circuits ..... 91
Figure 7.5. AMX2CF: AMUX2 Configuration Register ..... 92
Figure 7.6. AMX2SL: AMUX2 Channel Select Register ..... 93
Figure 7.7. ADC2CF: ADC2 Configuration Register ..... 94
Figure 7.8. ADC2H: ADC2 Data Word MSB Register ..... 95
Figure 7.9. ADC2L: ADC2 Data Word LSB Register ..... 95
Figure 7.10. ADC2CN: ADC2 Control Register ..... 96
Figure 7.11. ADC2GTH: ADC2 Greater-Than Data High Byte Register ..... 97
Figure 7.12. ADC2GTL: ADC2 Greater-Than Data Low Byte Register ..... 97
Figure 7.13. ADC2LTH: ADC2 Less-Than Data High Byte Register ..... 98

Figure 7.14. ADC2LTL: ADC2 Less-Than Data Low Byte Register ..... 98
Figure 7.15. ADC Window Compare Example: Right-Justified Single-Ended Data . 99
Figure 7.16. ADC Window Compare Example: Left-Justified Single-Ended Data.... 99
Figure 7.17. ADC Window Compare Example: Right-Justified Differential Data.... 100
Figure 7.18. ADC Window Compare Example: Left-Justified Differential Data..... 100
**8. DACs, 12-Bit Voltage Mode (DAC0 and DAC1, C8051F060/1/2/3) ..... 103**
Figure 8.1. DAC Functional Block Diagram..... 103
Figure 8.2. DAC0H: DAC0 High Byte Register ..... 105
Figure 8.3. DAC0L: DAC0 Low Byte Register..... 105
Figure 8.4. DAC0CN: DAC0 Control Register..... 106
Figure 8.5. DAC1H: DAC1 High Byte Register ..... 107
Figure 8.6. DAC1L: DAC1 Low Byte Register..... 107
Figure 8.7. DAC1CN: DAC1 Control Register..... 108
**9. Voltage Reference 2 (C8051F060/2) ..... 111**
Figure 9.1. Voltage Reference Functional Block Diagram ..... 111
Figure 9.2. REF2CN: Reference Control Register 2 ..... 112
**10. Voltage Reference 2 (C8051F061/3) ..... 113**
Figure 10.1. Voltage Reference Functional Block Diagram..... 113
Figure 10.2. REF2CN: Reference Control Register 2 ..... 114
**11. Voltage Reference 2 (C8051F064/5/6/7) ..... 115**
Figure 11.1. Voltage Reference Functional Block Diagram..... 115
Figure 11.2. REF2CN: Reference Control Register 2 ..... 116
**12. Comparators ..... 117**
Figure 12.1. Comparator Functional Block Diagram ..... 117
Figure 12.2. Comparator Hysteresis Plot ..... 118
Figure 12.3. CPTnCN: Comparator 0, 1, and 2 Control Register..... 120
Figure 12.4. CPTnMD: Comparator Mode Selection Register ..... 121
**13. CIP-51 Microcontroller ..... 123**
Figure 13.1. CIP-51 Block Diagram..... 124
Figure 13.2. Memory Map ..... 130
Figure 13.3. SFR Page Stack..... 133
Figure 13.4. SFR Page Stack While Using SFR Page 0x0F To Access Port 5..... 134
Figure 13.5. SFR Page Stack After ADC2 Window Comparator Interrupt Occurs . 135
Figure 13.6. SFR Page Stack Upon PCA Interrupt Occurring During an ADC2 ISR.... 136
Figure 13.7. SFR Page Stack Upon Return From PCA Interrupt ..... 137
Figure 13.8. SFR Page Stack Upon Return From ADC2 Window Interrupt ..... 138
Figure 13.9. SFRPGCN: SFR Page Control Register ..... 139
Figure 13.10. SFRPAGE: SFR Page Register ..... 139
Figure 13.11. SFRNEXT: SFR Next Register..... 140
Figure 13.12. SFRLAST: SFR Last Register..... 140
Figure 13.13. SP: Stack Pointer ..... 148
Figure 13.14. DPL: Data Pointer Low Byte..... 148
Figure 13.15. DPH: Data Pointer High Byte ..... 148
Figure 13.16. PSW: Program Status Word..... 149

Figure 13.17. ACC: Accumulator ..... 150
Figure 13.18. B: B Register ..... 150
Figure 13.19. IE: Interrupt Enable ..... 154
Figure 13.20. IP: Interrupt Priority ..... 155
Figure 13.21. EIE1: Extended Interrupt Enable 1 ..... 156
Figure 13.22. EIE2: Extended Interrupt Enable 2 ..... 157
Figure 13.23. EIP1: Extended Interrupt Priority 1 ..... 158
Figure 13.24. EIP2: Extended Interrupt Priority 2 ..... 159
Figure 13.25. PCON: Power Control ..... 161
**14. Reset Sources ..... 163**
Figure 14.1. Reset Sources ..... 163
Figure 14.2. Reset Timing ..... 164
Figure 14.3. WDTCN: Watchdog Timer Control Register ..... 167
Figure 14.4. RSTSRC: Reset Source Register ..... 168
**15. Oscillators ..... 171**
Figure 15.1. Oscillator Diagram ..... 171
Figure 15.2. OSCICL: Internal Oscillator Calibration Register ..... 172
Figure 15.3. OSCICN: Internal Oscillator Control Register ..... 172
Figure 15.4. CLKSEL: Oscillator Clock Selection Register ..... 173
Figure 15.5. OSCXCN: External Oscillator Control Register ..... 174
**16. Flash Memory ..... 177**
Figure 16.1. C8051F060/1/2/3/4/5 Flash Program Memory Map and Security Bytes .. 180
Figure 16.2. C8051F066/7 Flash Program Memory Map and Security Bytes ..... 181
Figure 16.3. FLACL: Flash Access Limit ..... 182
Figure 16.4. FLSCL: Flash Memory Control ..... 184
Figure 16.5. PSCTL: Program Store Read/Write Control ..... 185
**17. External Data Memory Interface and On-Chip XRAM ..... 187**
Figure 17.1. EMI0CN: External Memory Interface Control ..... 189
Figure 17.2. EMI0CF: External Memory Configuration ..... 189
Figure 17.3. Multiplexed Configuration Example ..... 190
Figure 17.4. Non-multiplexed Configuration Example ..... 191
Figure 17.5. EMIF Operating Modes ..... 192
Figure 17.6. EMI0TC: External Memory Timing Control ..... 194
Figure 17.7. Non-multiplexed 16-bit MOVX Timing ..... 196
Figure 17.8. Non-multiplexed 8-bit MOVX without Bank Select Timing ..... 197
Figure 17.9. Non-multiplexed 8-bit MOVX with Bank Select Timing ..... 198
Figure 17.10. Multiplexed 16-bit MOVX Timing ..... 199
Figure 17.11. Multiplexed 8-bit MOVX without Bank Select Timing ..... 200
Figure 17.12. Multiplexed 8-bit MOVX with Bank Select Timing ..... 201
**18. Port Input/Output ..... 203**
Figure 18.1. Port I/O Cell Block Diagram ..... 203
Figure 18.2. Port I/O Functional Block Diagram ..... 204
Figure 18.3. Priority Crossbar Decode Table ..... 205
Figure 18.4. Crossbar Example: ..... 209

Figure 18.5. XBR0: Port I/O Crossbar Register 0 ..... 210
Figure 18.6. XBR1: Port I/O Crossbar Register 1 ..... 211
Figure 18.7. XBR2: Port I/O Crossbar Register 2 ..... 212
Figure 18.8. XBR3: Port I/O Crossbar Register 3 ..... 213
Figure 18.9. P0: Port0 Data Register ..... 214
Figure 18.10. P0MDOUT: Port0 Output Mode Register ..... 214
Figure 18.11. P1: Port1 Data Register ..... 215
Figure 18.12. P1MDIN: Port1 Input Mode Register ..... 215
Figure 18.13. P1MDOUT: Port1 Output Mode Register ..... 216
Figure 18.14. P2: Port2 Data Register ..... 216
Figure 18.15. P2MDIN: Port2 Input Mode Register ..... 217
Figure 18.16. P2MDOUT: Port2 Output Mode Register ..... 217
Figure 18.17. P3: Port3 Data Register ..... 218
Figure 18.18. P3MDOUT: Port3 Output Mode Register ..... 218
Figure 18.19. P4: Port4 Data Register ..... 221
Figure 18.20. P4MDOUT: Port4 Output Mode Register ..... 221
Figure 18.21. P5: Port5 Data Register ..... 222
Figure 18.22. P5MDOUT: Port5 Output Mode Register ..... 222
Figure 18.23. P6: Port6 Data Register ..... 223
Figure 18.24. P6MDOUT: Port6 Output Mode Register ..... 223
Figure 18.25. P7: Port7 Data Register ..... 224
Figure 18.26. P7MDOUT: Port7 Output Mode Register ..... 224
**19. Controller Area Network (CAN0, C8051F060/1/2/3) ..... 225**
Figure 19.1. CAN Controller Diagram ..... 226
Figure 19.2. Typical CAN Bus Configuration ..... 226
Figure 19.3. CAN0DATH: CAN Data Access Register High Byte ..... 231
Figure 19.4. CAN0DATL: CAN Data Access Register Low Byte ..... 231
Figure 19.5. CAN0ADR: CAN Address Index Register ..... 232
Figure 19.6. CAN0CN: CAN Control Register ..... 232
Figure 19.7. CAN0TST: CAN Test Register ..... 233
Figure 19.8. CAN0STA: CAN Status Register ..... 233
**20. System Management BUS / I2C BUS (SMBUS0) ..... 235**
Figure 20.1. SMBus0 Block Diagram ..... 235
Figure 20.2. Typical SMBus Configuration ..... 236
Figure 20.3. SMBus Transaction ..... 237
Figure 20.4. Typical Master Transmitter Sequence ..... 238
Figure 20.5. Typical Master Receiver Sequence ..... 238
Figure 20.6. Typical Slave Transmitter Sequence ..... 239
Figure 20.7. Typical Slave Receiver Sequence ..... 240
Figure 20.8. SMB0CN: SMBus0 Control Register ..... 243
Figure 20.9. SMB0CR: SMBus0 Clock Rate Register ..... 244
Figure 20.10. SMB0DAT: SMBus0 Data Register ..... 245
Figure 20.11. SMB0ADR: SMBus0 Address Register ..... 246
Figure 20.12. SMB0STA: SMBus0 Status Register ..... 247
**21. Enhanced Serial Peripheral Interface (SPI0) ..... 251**

Figure 21.1. SPI Block Diagram ..... 251
Figure 21.2. Multiple-Master Mode Connection Diagram ..... 254
Figure 21.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram 254
Figure 21.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram 254
Figure 21.5. Master Mode Data/Clock Timing ..... 256
Figure 21.6. Slave Mode Data/Clock Timing (CKPHA = 0) ..... 257
Figure 21.7. Slave Mode Data/Clock Timing (CKPHA = 1) ..... 257
Figure 21.8. SPI0CFG: SPI0 Configuration Register ..... 258
Figure 21.9. SPI0CN: SPI0 Control Register ..... 259
Figure 21.10. SPI0CKR: SPI0 Clock Rate Register ..... 260
Figure 21.11. SPI0DAT: SPI0 Data Register ..... 261
Figure 21.12. SPI Master Timing (CKPHA = 0) ..... 262
Figure 21.13. SPI Master Timing (CKPHA = 1) ..... 262
Figure 21.14. SPI Slave Timing (CKPHA = 0) ..... 263
Figure 21.15. SPI Slave Timing (CKPHA = 1) ..... 263
**22. UART0 ..... 265**
Figure 22.1. UART0 Block Diagram ..... 265
Figure 22.2. UART0 Mode 0 Timing Diagram ..... 267
Figure 22.3. UART0 Mode 0 Interconnect ..... 267
Figure 22.4. UART0 Mode 1 Timing Diagram ..... 267
Figure 22.5. UART0 Modes 2 and 3 Timing Diagram ..... 269
Figure 22.6. UART0 Modes 1, 2, and 3 Interconnect Diagram ..... 270
Figure 22.7. UART Multi-Processor Mode Interconnect Diagram ..... 272
Figure 22.8. SCON0: UART0 Control Register ..... 274
Figure 22.9. SSTA0: UART0 Status and Clock Selection Register ..... 275
Figure 22.10. SBUF0: UART0 Data Buffer Register ..... 276
Figure 22.11. SADDR0: UART0 Slave Address Register ..... 276
Figure 22.12. SADEN0: UART0 Slave Address Enable Register ..... 276
**23. UART1 ..... 277**
Figure 23.1. UART1 Block Diagram ..... 277
Figure 23.2. UART1 Baud Rate Logic ..... 278
Figure 23.3. UART Interconnect Diagram ..... 279
Figure 23.4. 8-Bit UART Timing Diagram ..... 279
Figure 23.5. 9-Bit UART Timing Diagram ..... 280
Figure 23.6. UART Multi-Processor Mode Interconnect Diagram ..... 281
Figure 23.7. SCON1: Serial Port 1 Control Register ..... 282
Figure 23.8. SBUF1: Serial (UART1) Port Data Buffer Register ..... 283
**24. Timers ..... 287**
Figure 24.1. T0 Mode 0 Block Diagram ..... 288
Figure 24.2. T0 Mode 2 Block Diagram ..... 289
Figure 24.3. T0 Mode 3 Block Diagram ..... 290
Figure 24.4. TCON: Timer Control Register ..... 291
Figure 24.5. TMOD: Timer Mode Register ..... 292

Figure 24.6. CKCON: Clock Control Register ..... 293
Figure 24.7. TL0: Timer 0 Low Byte ..... 294
Figure 24.8. TL1: Timer 1 Low Byte ..... 294
Figure 24.9. TH0: Timer 0 High Byte ..... 294
Figure 24.10. TH1: Timer 1 High Byte ..... 294
Figure 24.11. T2, 3, and 4 Capture Mode Block Diagram ..... 296
Figure 24.12. T2, 3, and 4 Auto-reload Mode Block Diagram ..... 297
Figure 24.13. TMRnCN: Timer 2, 3, and 4 Control Registers ..... 299
Figure 24.14. TMRnCF: Timer 2, 3, and 4 Configuration Registers ..... 300
Figure 24.15. RCAPnL: Timer 2, 3, and 4 Capture Register Low Byte ..... 301
Figure 24.16. RCAPnH: Timer 2, 3, and 4 Capture Register High Byte ..... 301
Figure 24.17. TMRnL: Timer 2, 3, and 4 Low Byte ..... 301
Figure 24.18. TMRnH: Timer 2, 3, and 4 High Byte ..... 302
**25. Programmable Counter Array ..... 303**
Figure 25.1. PCA Block Diagram ..... 303
Figure 25.2. PCA Counter/Timer Block Diagram ..... 304
Figure 25.3. PCA Interrupt Block Diagram ..... 305
Figure 25.4. PCA Capture Mode Diagram ..... 306
Figure 25.5. PCA Software Timer Mode Diagram ..... 307
Figure 25.6. PCA High Speed Output Mode Diagram ..... 308
Figure 25.7. PCA Frequency Output Mode ..... 309
Figure 25.8. PCA 8-Bit PWM Mode Diagram ..... 310
Figure 25.9. PCA 16-Bit PWM Mode ..... 311
Figure 25.10. PCA0CN: PCA Control Register ..... 312
Figure 25.11. PCA0MD: PCA0 Mode Register ..... 313
Figure 25.12. PCA0CPMn: PCA0 Capture/Compare Mode Registers ..... 314
Figure 25.13. PCA0L: PCA0 Counter/Timer Low Byte ..... 315
Figure 25.14. PCA0H: PCA0 Counter/Timer High Byte ..... 315
Figure 25.15. PCA0CPLn: PCA0 Capture Module Low Byte ..... 316
Figure 25.16. PCA0CPHn: PCA0 Capture Module High Byte ..... 316
**26. JTAG (IEEE 1149.1) ..... 317**
Figure 26.1. IR: JTAG Instruction Register ..... 317
Figure 26.2. DEVICEID: JTAG Device ID Register ..... 321
Figure 26.3. FLASHCON: JTAG Flash Control Register ..... 323
Figure 26.4. FLASHDAT: JTAG Flash Data Register ..... 324
Figure 26.5. FLASHADR: JTAG Flash Address Register ..... 324
**27. Document Change List ..... 327**

## List of Tables

**1. System Overview ..... 19**
Table 1.1. Product Selection Guide ..... 20
**2. Absolute Maximum Ratings ..... 37**
Table 2.1. Absolute Maximum Ratings* ..... 37
**3. Global DC Electrical Characteristics ..... 38**
Table 3.1. Global DC Electrical Characteristics ..... 38
**4. Pinout and Package Definitions ..... 39**
Table 4.1. Pin Definitions ..... 39
**5. 16-Bit ADCs (ADC0 and ADC1) ..... 51**
Table 5.1. Conversion Timing (tConv) ..... 55
Table 5.2. 16-Bit ADC0 and ADC1 Electrical Characteristics ..... 73
Table 5.3. Voltage Reference 0 and 1 Electrical Characteristics ..... 74
**6. Direct Memory Access Interface (DMA0) ..... 75**
Table 6.1. DMA0 Instruction Set ..... 76
**7. 10-Bit ADC (ADC2, C8051F060/1/2/3) ..... 87**
Table 7.1. ADC2 Electrical Characteristics ..... 101
**8. DACs, 12-Bit Voltage Mode (DAC0 and DAC1, C8051F060/1/2/3) ..... 103**
Table 8.1. DAC Electrical Characteristics ..... 109
**9. Voltage Reference 2 (C8051F060/2) ..... 111**
Table 9.1. Voltage Reference Electrical Characteristics ..... 112
**10. Voltage Reference 2 (C8051F061/3) ..... 113**
Table 10.1. Voltage Reference Electrical Characteristics ..... 114
**11. Voltage Reference 2 (C8051F064/5/6/7) ..... 115**
Table 11.1. Voltage Reference Electrical Characteristics ..... 116
**12. Comparators ..... 117**
Table 12.1. Comparator Electrical Characteristics ..... 122
**13. CIP-51 Microcontroller ..... 123**
Table 13.1. CIP-51 Instruction Set Summary ..... 126
Table 13.2. Special Function Register (SFR) Memory Map ..... 141
Table 13.3. Special Function Registers ..... 143
Table 13.4. Interrupt Summary ..... 152
**14. Reset Sources ..... 163**
Table 14.1. Reset Electrical Characteristics ..... 169
**15. Oscillators ..... 171**
Table 15.1. Internal Oscillator Electrical Characteristics ..... 173
**16. Flash Memory ..... 177**
Table 16.1. Flash Electrical Characteristics ..... 178
**17. External Data Memory Interface and On-Chip XRAM ..... 187**
Table 17.1. AC Parameters for External Memory Interface ..... 202
**18. Port Input/Output ..... 203**
Table 18.1. Port I/O DC Electrical Characteristics ..... 203
**19. Controller Area Network (CAN0, C8051F060/1/2/3) ..... 225**

Table 19.1.CAN Register Index and Reset Values ..... 229
**20. System Management BUS / I2C BUS (SMBUS0) ..... 235**
Table 20.1.SMB0STA Status Codes and States ..... 248
**21. Enhanced Serial Peripheral Interface (SPI0) ..... 251**
Table 21.1.SPI Slave Timing Parameters ..... 264
**22. UART0 ..... 265**
Table 22.1.UART0 Modes ..... 266
Table 22.2.Oscillator Frequencies for Standard Baud Rates ..... 273
**23. UART1 ..... 277**
Table 23.1.Timer Settings for Standard Baud Rates Using the Internal Oscillator 284
Table 23.2.Timer Settings for Standard Baud Rates Using an External Oscillator 284
Table 23.3.Timer Settings for Standard Baud Rates Using an External Oscillator 285
Table 23.4.Timer Settings for Standard Baud Rates Using an External Oscillator 285
Table 23.5.Timer Settings for Standard Baud Rates Using an External Oscillator 286
Table 23.6.Timer Settings for Standard Baud Rates Using an External Oscillator 286
**24. Timers ..... 287**
**25. Programmable Counter Array ..... 303**
Table 25.1.PCA Timebase Input Options ..... 304
Table 25.2.PCA0CPM Register Settings for PCA Capture/Compare Modules ..... 305
**26. JTAG (IEEE 1149.1) ..... 317**
Table 26.1.Boundary Data Register Bit Definitions (C8051F060/2/4/6) ..... 318
Table 26.2.Boundary Data Register Bit Definitions (C8051F061/3/5/7) ..... 320
**27. Document Change List ..... 327**

# 1. System Overview

The C8051F06x family of devices are fully integrated mixed-signal System-on-a-Chip MCUs with 59 digital I/O pins (C8051F060/2/4/6) or 24 digital I/O pins (C8051F061/3/5/7), and two integrated 16-bit 1 Msps ADCs. Highlighted features are listed below; refer to Table 1.1 for specific product feature selection.

- High-Speed pipelined 8051-compatible CIP-51 microcontroller core (up to 25 MIPS)
- Two 16-bit 1 Msps ADCs with a Direct Memory Access controller
- Controller Area Network (CAN 2.0B) Controller with 32 message objects, each with its own indentifier mask (C8051F060/1/2/3)
- In-system, full-speed, non-intrusive debug interface on-chip
- 10-bit 200 ksps ADC with PGA and 8-channel analog multiplexer (C8051F060/1/2/3)
- Two 12-bit DACs with programmable update scheduling (C8051F060/1/2/3)
- 64 kB (C8051F060/1/2/3/4/5) or 32 kB (C8051F066/7) of in-system programmable Flash memory
- 4352 (4096 + 256) bytes of on-chip RAM
- External Data Memory Interface with 64 kB direct address space (C8051F060/2/4/6)
- SPI, SMBus/I2C, and (2) UART serial interfaces implemented in hardware
- Five general purpose 16-bit Timers
- Programmable Counter/Timer Array with six capture/compare modules
- On-chip Watchdog Timer, VDD Monitor, and Temperature Sensor

With on-chip VDD monitor, Watchdog Timer, and clock oscillator, the C8051F06x family of 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 2.7 to 3.6 V operation over the industrial temperature range (-45 to +85 °C). The C8051F060/2/4/6 are available in a 100-pin TQFP package and the C8051F061/3/5/7 are available in a 64-pin TQFP package (see block diagrams in Figure 1.1, Figure 1.2, Figure 1.3 and Figure 1.4).

**Table 1.1. Product Selection Guide**

| | MIPS (Peak) | Flash Memory | RAM | External Memory Interface | SMBus/I2C and SPI | CAN | UARTS | Timers (16-bit) | Programmable Counter Array | Digital Port I/O's | 16-bit 1 Msps ADC Typical INL (LSBs) | 10-bit 200 ksps ADC Inputs | Voltage Reference | Temperature Sensor | DAC Resolution (bits) | DAC Outputs | Analog Comparators | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C8051F060 | 25 | 64 k | 4352 | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 59 | ±0.75 | 8 | ✓ | ✓ | 12 | 2 | 3 | 100 TQFP |
| C8051F061 | 25 | 64 k | 4352 | - | ✓ | ✓ | 2 | 5 | ✓ | 24 | ±0.75 | 8 | ✓ | ✓ | 12 | 2 | 3 | 64 TQFP |
| C8051F062 | 25 | 64 k | 4352 | ✓ | ✓ | ✓ | 2 | 5 | ✓ | 59 | ±1.5 | 8 | ✓ | ✓ | 12 | 2 | 3 | 100 TQFP |
| C8051F063 | 25 | 64 k | 4352 | - | ✓ | ✓ | 2 | 5 | ✓ | 24 | ±1.5 | 8 | ✓ | ✓ | 12 | 2 | 3 | 64 TQFP |
| C8051F064 | 25 | 64 k | 4352 | ✓ | ✓ | - | 2 | 5 | ✓ | 59 | ±0.75 | - | ✓ | - | - | - | 3 | 100 TQFP |
| C8051F065 | 25 | 64 k | 4352 | - | ✓ | - | 2 | 5 | ✓ | 24 | ±0.75 | - | ✓ | - | - | - | 3 | 64 TQFP |
| C8051F066 | 25 | 32 k | 4352 | ✓ | ✓ | - | 2 | 5 | ✓ | 59 | ±0.75 | - | ✓ | - | - | - | 3 | 100 TQFP |
| C8051F067 | 25 | 32 k | 4352 | - | ✓ | - | 2 | 5 | ✓ | 24 | ±0.75 | - | ✓ | - | - | - | 3 | 64 TQFP |

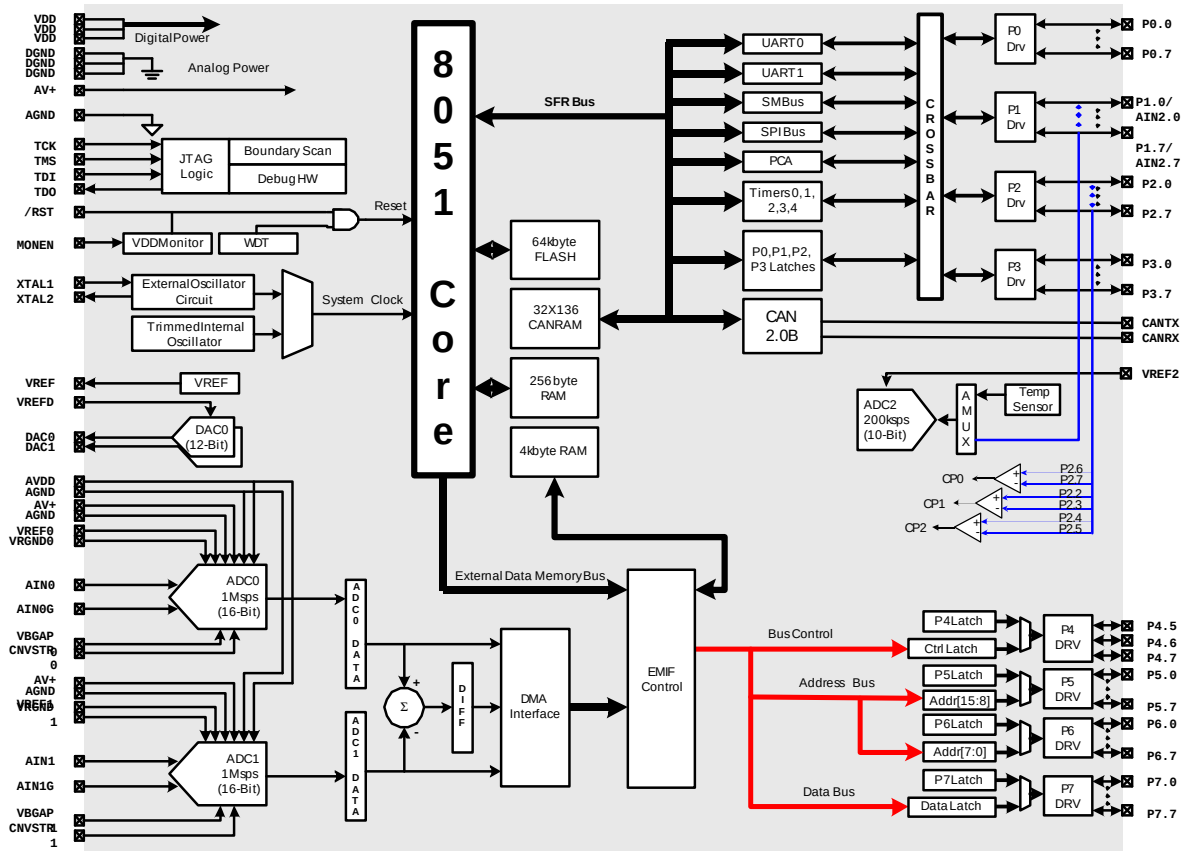



**Figure 1.1. C8051F060 / C8051F062 Block Diagram**

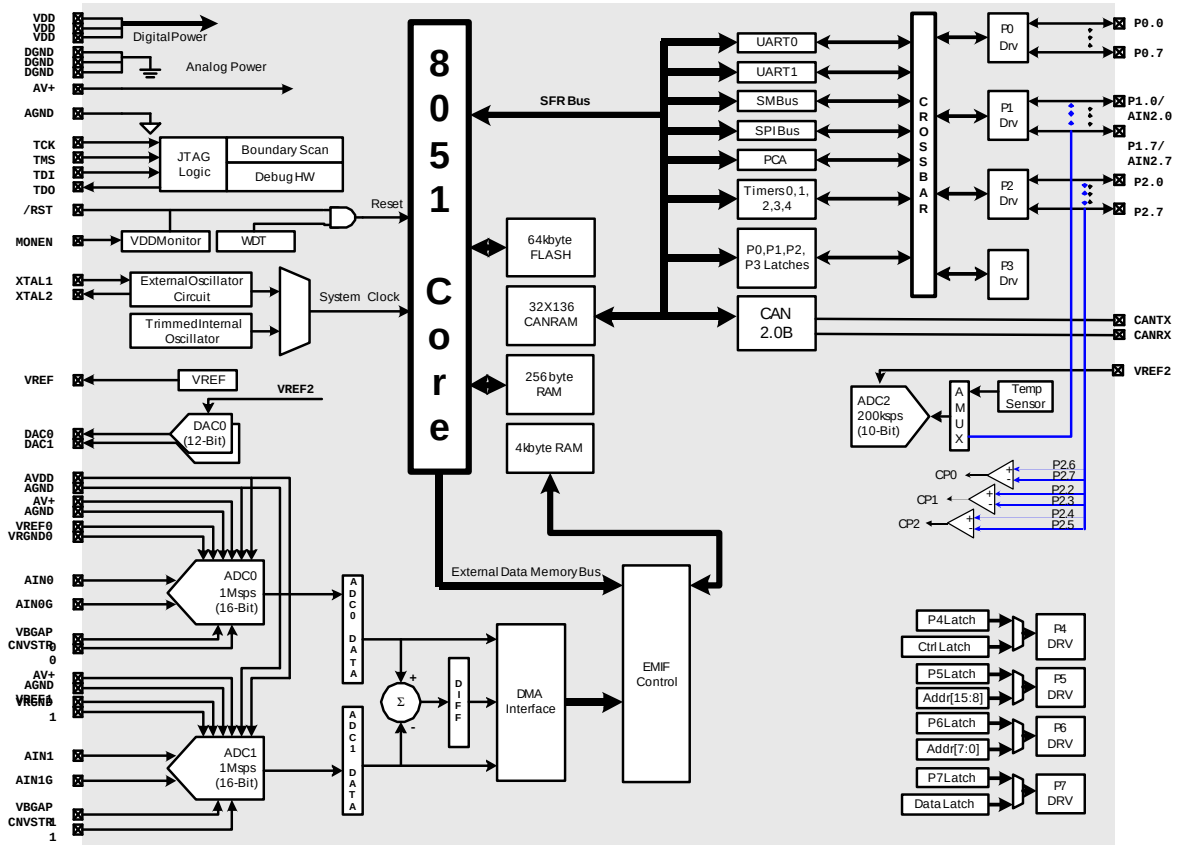



**Figure 1.2. C8051F061 / C8051F063 Block Diagram**

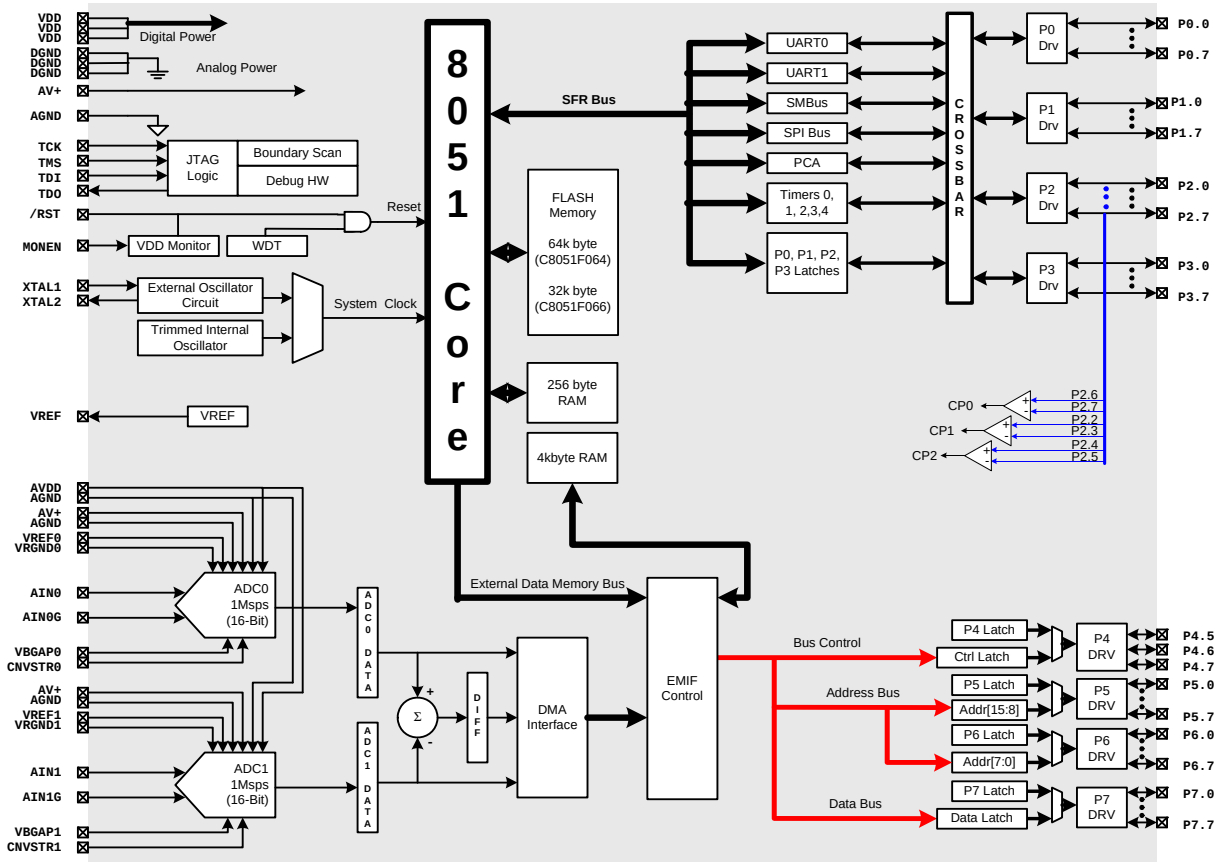



**Figure 1.3. C8051F064 / C8051F066 Block Diagram**

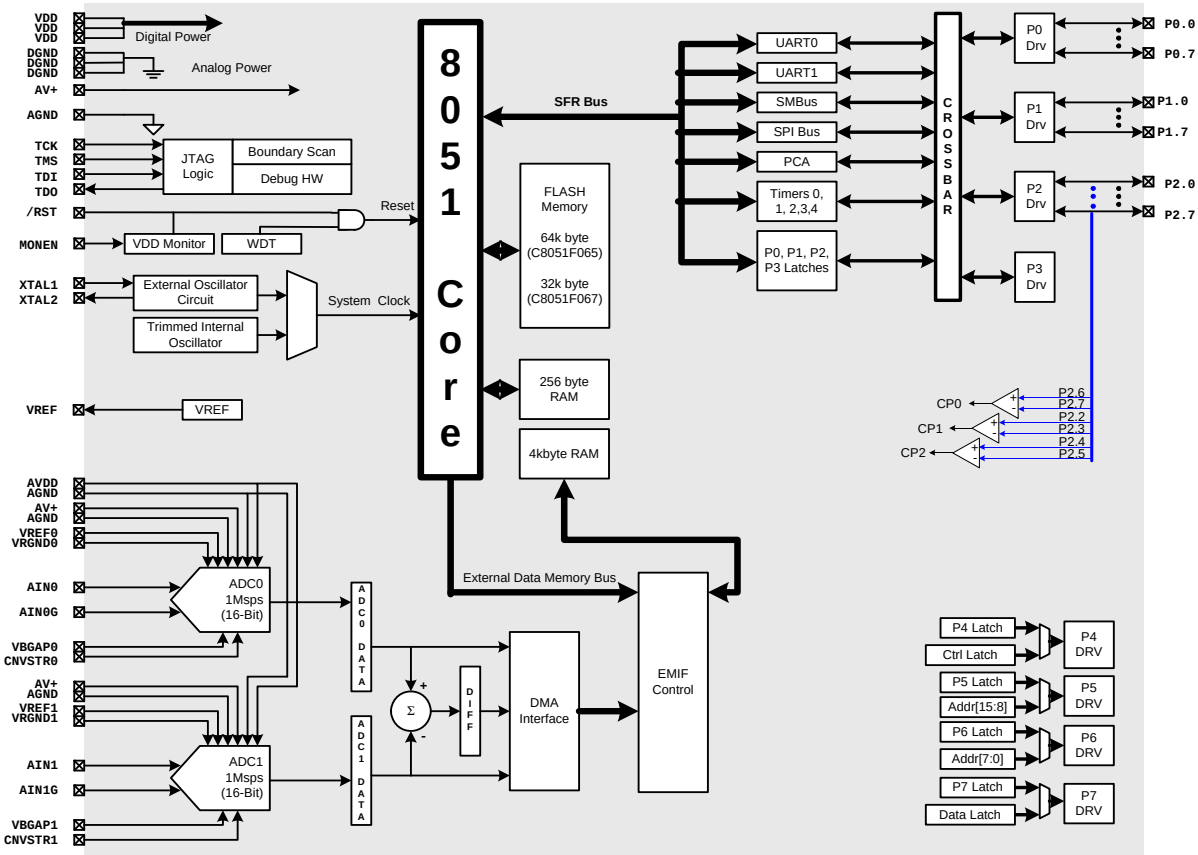



**Figure 1.4. C8051F065 / C8051F067 Block Diagram**

## 1.1. CIP-51™ Microcontroller Core

### 1.1.1. Fully 8051 Compatible

The C8051F06x family of devices utilizes Silicon Labs' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The 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 bit-addressable 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 25 MHz, it has a peak throughput of 25 MIPS. Figure 1.5 shows a comparison of peak throughputs of various 8-bit microcontroller cores with their maximum system clocks.

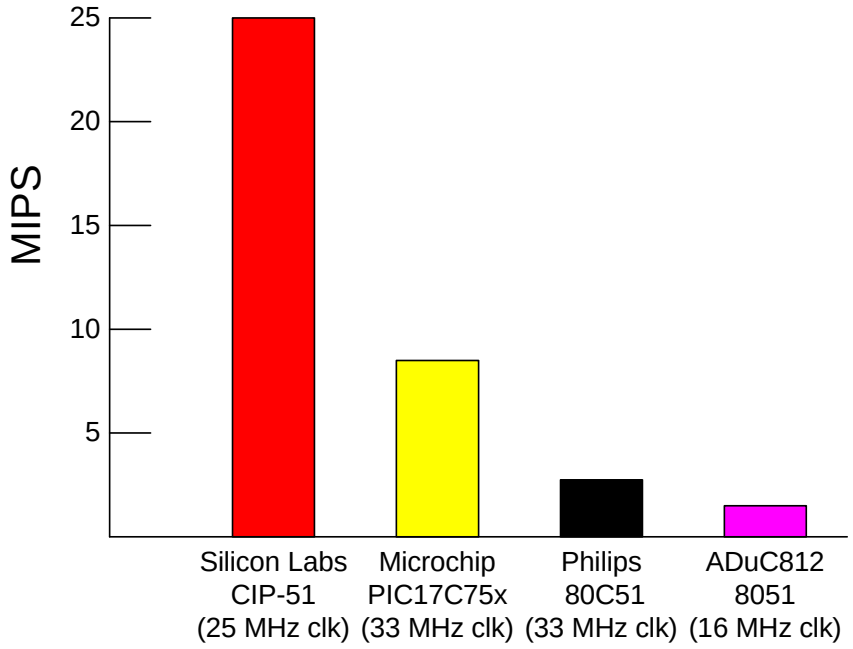



**Figure 1.5. Comparison of Peak MCU Execution Speeds**

### 1.1.3. Additional Features

The C8051F06x MCU family includes several key enhancements to the CIP-51 core and peripherals to improve overall performance and ease of use in end applications.

The extended interrupt handler provides 22 interrupt sources into the CIP-51, 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 VDD monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator0, a forced software reset, the CNVSTR2 input pin, and the /RST pin. The /RST pin is bi-directional, accommodating an external reset, or allowing the internally generated POR to be output on the /RST pin. Each reset source except for the VDD monitor and Reset Input pin may be disabled by the user in software; the VDD 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 fast (up to 25 MHz) internal oscillator as needed.

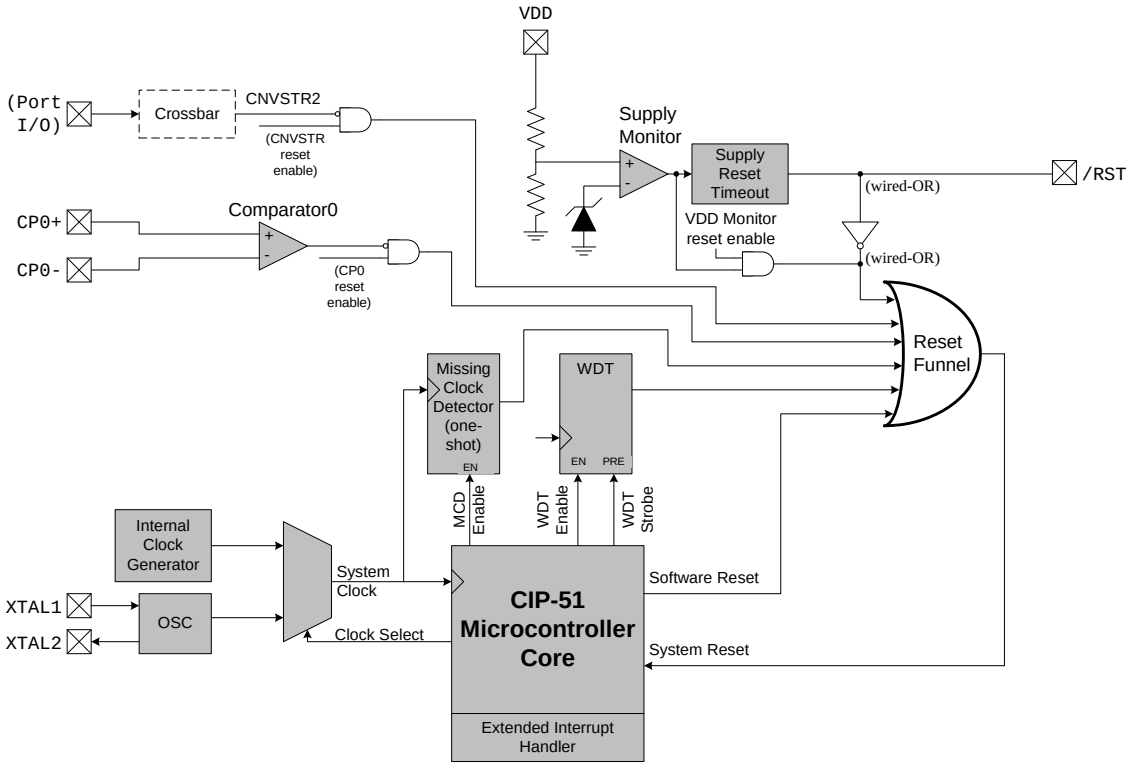



**Figure 1.6. On-Board Clock and Reset**

## 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 CIP-51 SFR address space contains up to 256 *SFR Pages*. In this way, the CIP-51 MCU can accommodate the many SFRs required to control and configure the various peripherals featured on the device. 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 CIP-51 in the C8051F060/1/2/3/4/5/6/7 MCUs additionally has an on-chip 4 kB RAM block. The on-chip 4 kB block can be addressed over the entire 64 k external data memory address range (overlapping 4 k boundaries). The C8051F060/2/4/6 also have an external memory interface (EMIF) for accessing off-chip data memory or memory-mapped peripherals. 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 4 k directed to on-chip, above 4 k directed to EMIF). The EMIF is also configurable for multiplexed or non-multiplexed address/data lines.

The MCU's program memory consists of 64 k (C8051F060/1/2/3/4/5) or 32 k (C8051F066/7) of Flash. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. On the C8051F060/1/2/3/4/5, the 1024 bytes from addresses 0xFC00 to 0xFFFF are reserved. There is also a single 128 byte Scratchpad Memory sector on all devices which may be used by firmware for non-volatile data storage. See Figure 1.7 for the MCU system memory map.

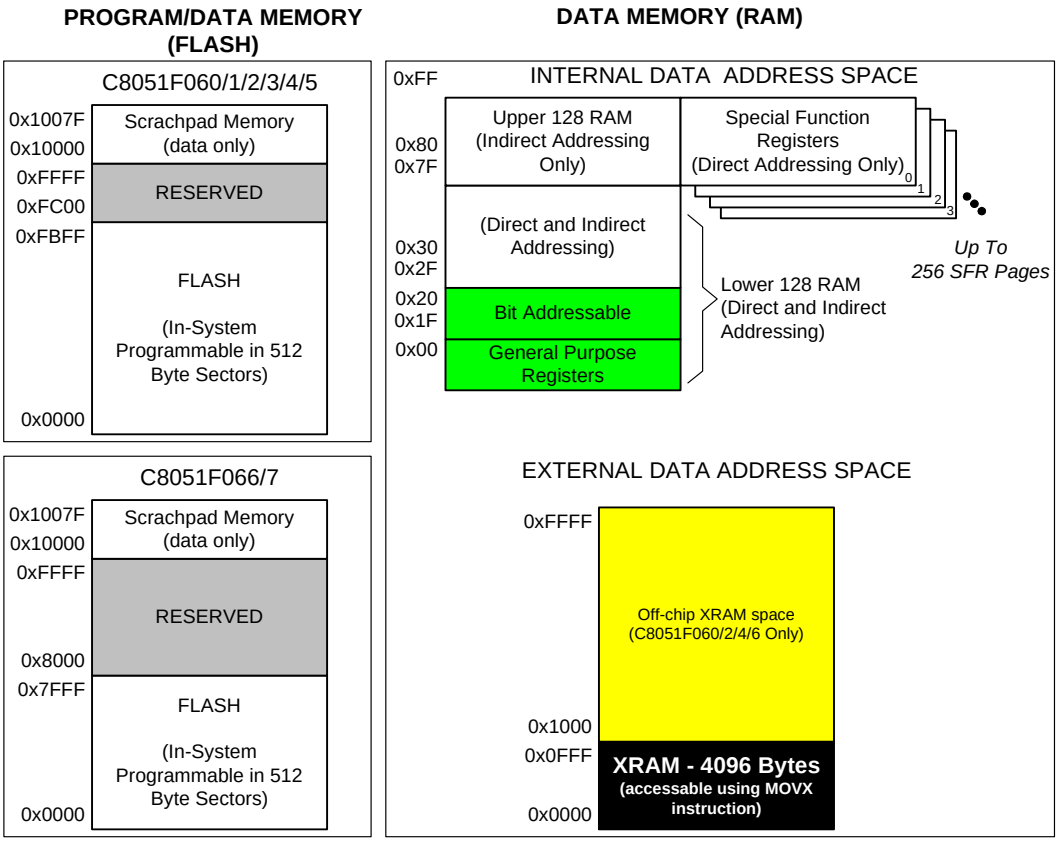



**Figure 1.7. On-Chip Memory Map**

## 1.3. JTAG Debug and Boundary Scan

The C8051F06x family has on-chip JTAG boundary scan and debug circuitry that 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 Laboratories' 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 ADCs and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized with instruction execution.

The C8051F060DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F06x 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 C8051F060 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.

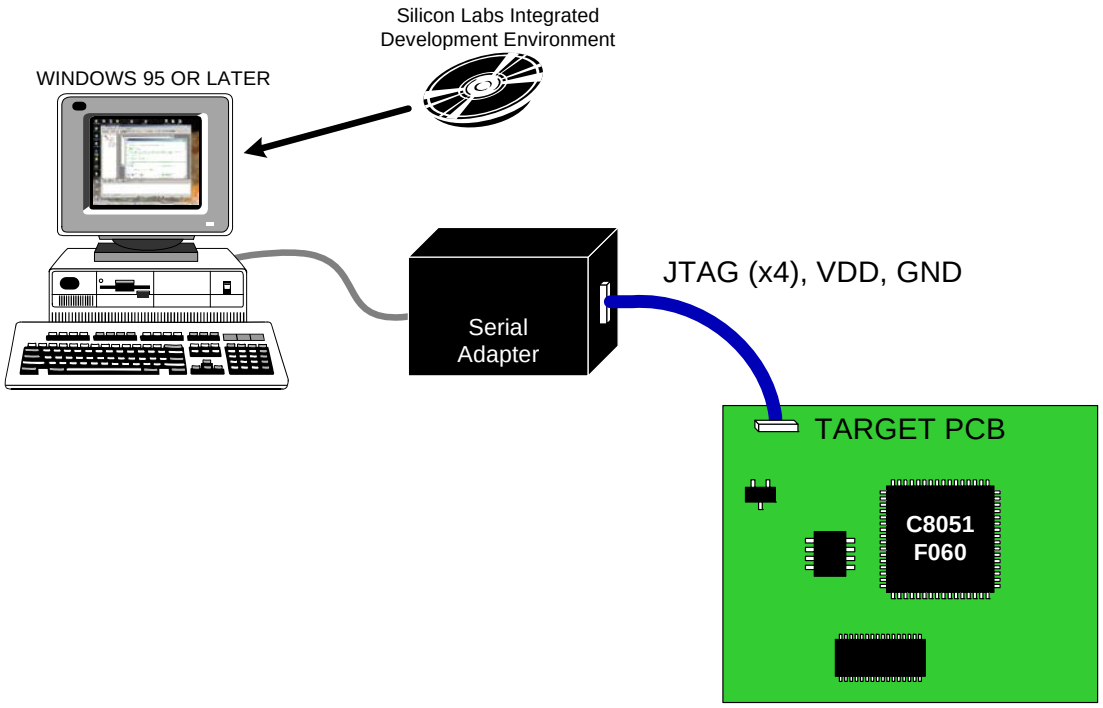



**Figure 1.8. Development/In-System Debug Diagram**

## 1.4. Programmable Digital I/O and Crossbar

Three standard 8051 Ports (0, 1, and 2) are available on the MCUs. The C8051F060/2/4/6 have 4 additional 8-bit ports (3, 5, 6, and 7), and a 3-bit port (port 4) for a total of 59 general-purpose I/O Pins. The Ports behave like the standard 8051 with a few enhancements.

Each port pin can be configured as either a push-pull or open-drain output. Also, the "weak pull-ups" 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 1.9) Unlike microcontrollers with standard multiplexed digital I/O ports, all combinations of functions are supported with all package options offered.

The on-chip counter/timers, serial buses, HW interrupts, 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.

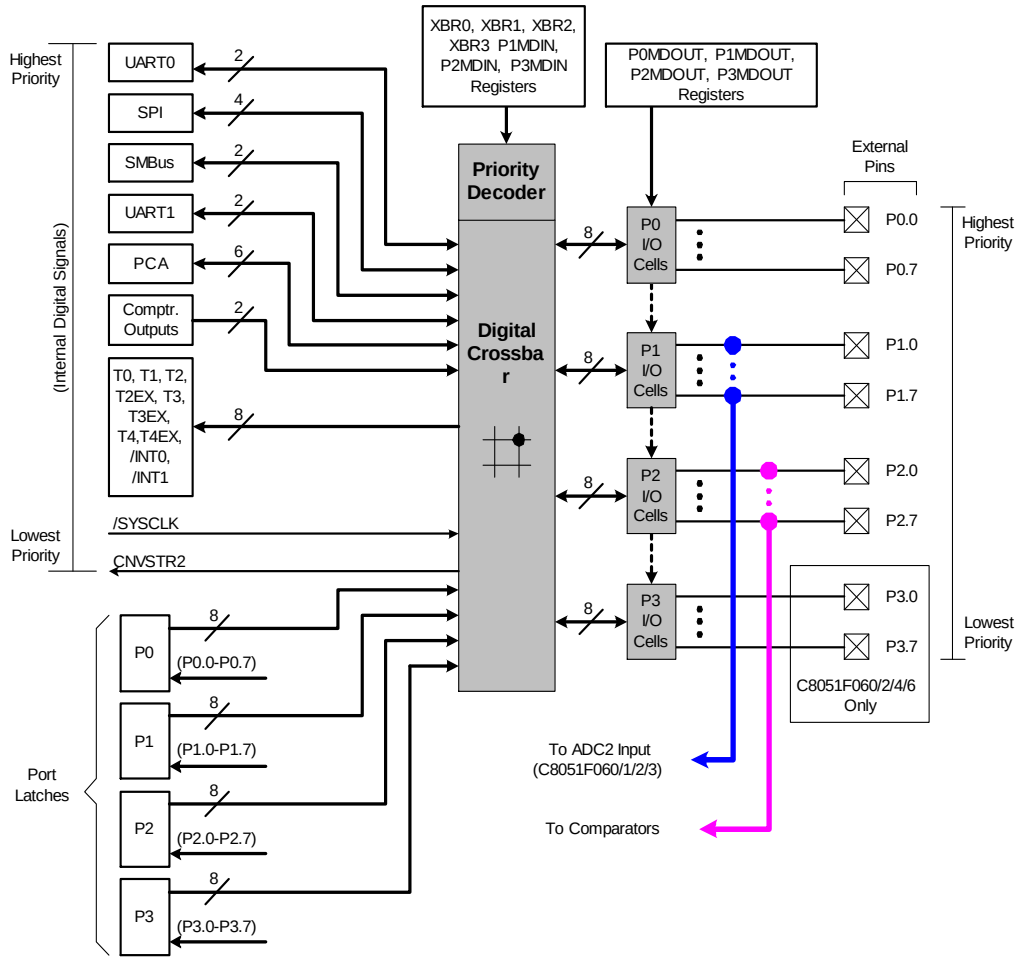



**Figure 1.9. Digital Crossbar Diagram**

## 1.5. Programmable Counter Array

The C8051F06x MCU family includes an on-board Programmable Counter/Timer Array (PCA) 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.

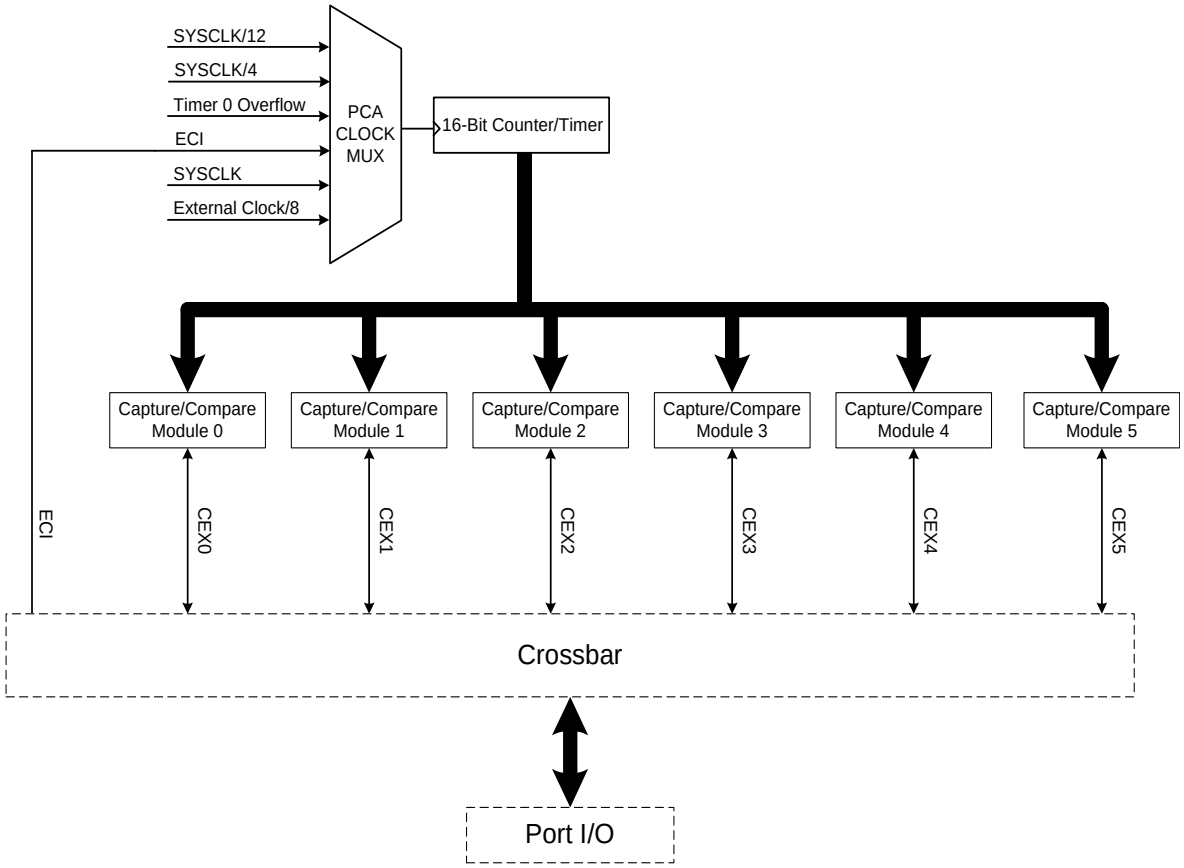



**Figure 1.10. PCA Block Diagram**

## 1.6. Controller Area Network

The C8051F060/1/2/3 devices feature a Controller Area Network (CAN) controller that implements serial communication using the CAN protocol. The CAN controller facilitates communication on a CAN network in accordance with the Bosch specification 2.0A (basic CAN) and 2.0B (full CAN). The CAN controller consists of a CAN Core, Message RAM (separate from the C8051 RAM), a message handler state machine, and control registers.

The CAN controller can operate at bit rates up to 1 Mbit/second. Silicon Labs CAN has 32 message objects each having its own identifier mask used for acceptance filtering of received messages. Incoming data, message objects and identifier masks are stored in the CAN message RAM. All protocol functions for transmission of data and acceptance filtering is performed by the CAN controller and not by the C8051 MCU. In this way, minimal CPU bandwidth is used for CAN communication. The C8051 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFR) in the C8051.

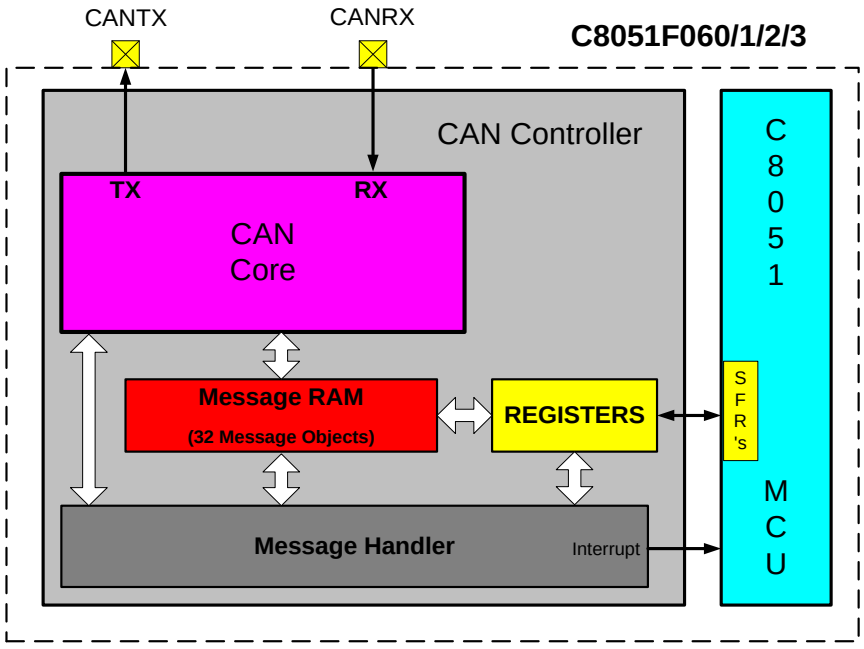



**Figure 1.11. CAN Controller Overview**

## 1.7. Serial Ports

The C8051F06x MCU Family includes two Enhanced Full-Duplex UARTs, an enhanced 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. 16-Bit Analog to Digital Converters

The C8051F060/1/2/3/4/5/6/7 devices have two on-chip 16-bit SAR ADCs (ADC0 and ADC1), which can be used independently in single-ended mode, or together in differential mode. ADC0 and ADC1 can directly access on-chip or external RAM, using the DMA interface. With a maximum throughput of 1 Msps, the ADCs offer 16 bit performance with two available linearity grades. ADC0 and ADC1 each have the capability to use dedicated, on-chip voltage reference circuitry or an external voltage reference source.

The ADCs are under full control of the CIP-51 microcontroller via the associated Special Function Registers. The system controller can also put the ADCs into shutdown mode to save power.

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. The two ADCs can operate independently, or be synchronized to perform conversions at the same time. Conversion completions are indicated by status bits, and can generate interrupts. The resulting 16-bit data words are latched into SFRs upon completion of a conversion. A DMA interface is also provided, which can gather conversions from the ADCs, and directly store them to on-chip or external RAM.

ADC0 also contains Window Compare registers, which can be configured to interrupt the controller when ADC0 data is within or outside of a specified range. ADC0 can monitor a key voltage continuously in background mode, and not interrupt the controller unless the converted data is within the specified window.

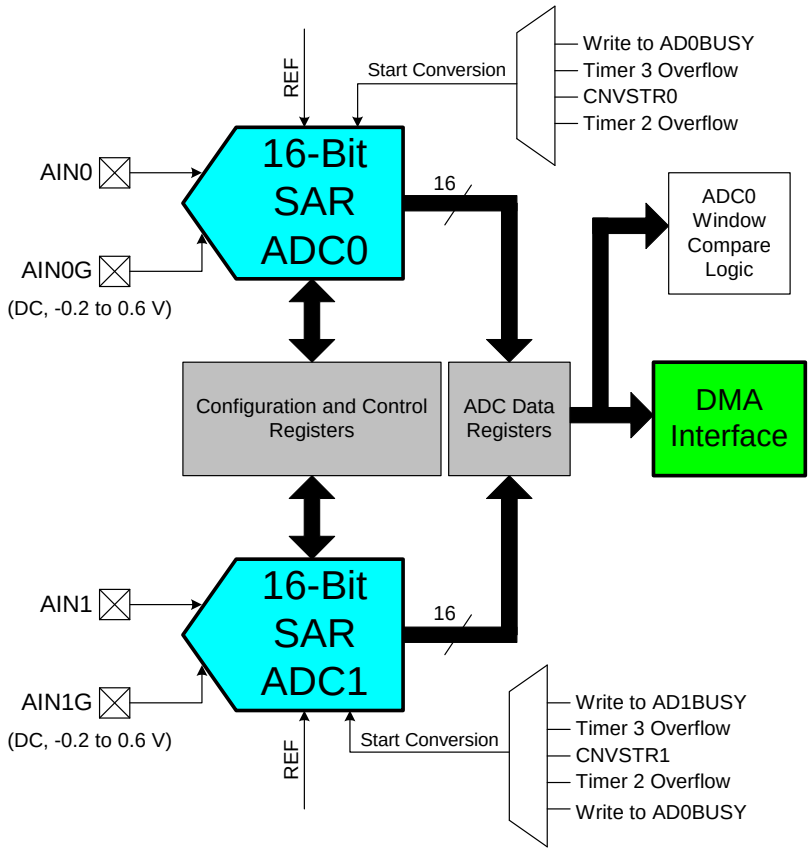



**Figure 1.12. 16-Bit ADC Block Diagram**

## 1.9. 10-Bit Analog to Digital Converter

The C8051F060/1/2/3 devices have an on-board 10-bit SAR ADC (ADC2) with a 9-channel input multiplexer and programmable gain amplifier. This ADC features a 200 ksps maximum throughput and true 10-bit performance with an INL of ±1LSB. Eight input pins are available for measurement and can be programmed as single-ended or differential inputs. Additionally, the on-chip temperature sensor can be used as an input to the ADC. 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 the external VREF2 pin. User software may put ADC2 into shutdown mode to save power.

A flexible conversion scheduling system allows ADC2 conversions to be initiated by software commands, timer overflows, or an external input signal. Conversion completions are indicated by a status bit and an interrupt (if enabled), and the resulting 10-bit data word is latched into two SFR locations upon completion.

ADC2 also contains Window Compare registers, which can be configured to interrupt the controller when ADC2 data is within or outside of a specified range. ADC2 can monitor a key voltage continuously in background mode, and not interrupt the controller unless the converted data is within the specified window.

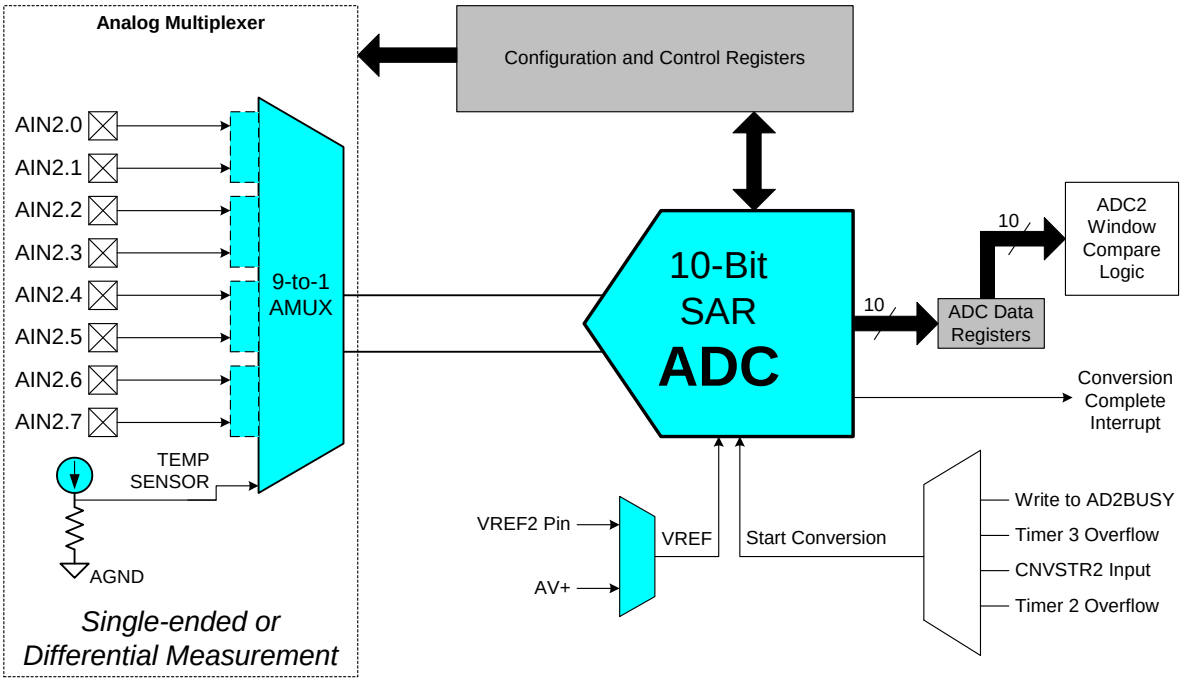



**Figure 1.13. 10-Bit ADC Diagram**

## 1.10. 12-bit Digital to Analog Converters

The C8051F060/1/2/3 MCUs 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 C8051F060/2 devices or via the VREF2 pin on C8051F061/3 devices, which is shared with ADC2. The DACs are especially useful as references for the comparators or offsets for the differential inputs of the ADCs.

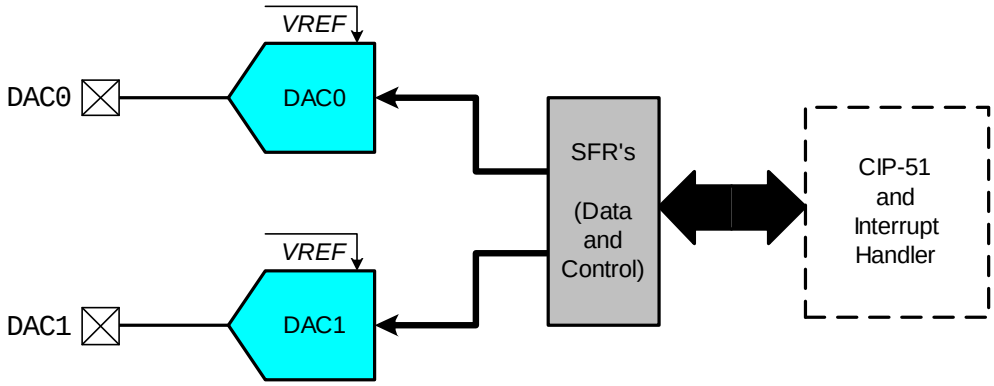



**Figure 1.14. DAC System Block Diagram**

## 1.11. Analog Comparators

The C8051F060/1/2/3/4/5/6/7 MCUs include three analog comparators on-chip. The comparators have software programmable hysteresis and response time. Each comparator can generate an interrupt on its 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. Outputs from the comparator can be routed through the crossbar. The comparators can be programmed to a low power shutdown mode when not in use.

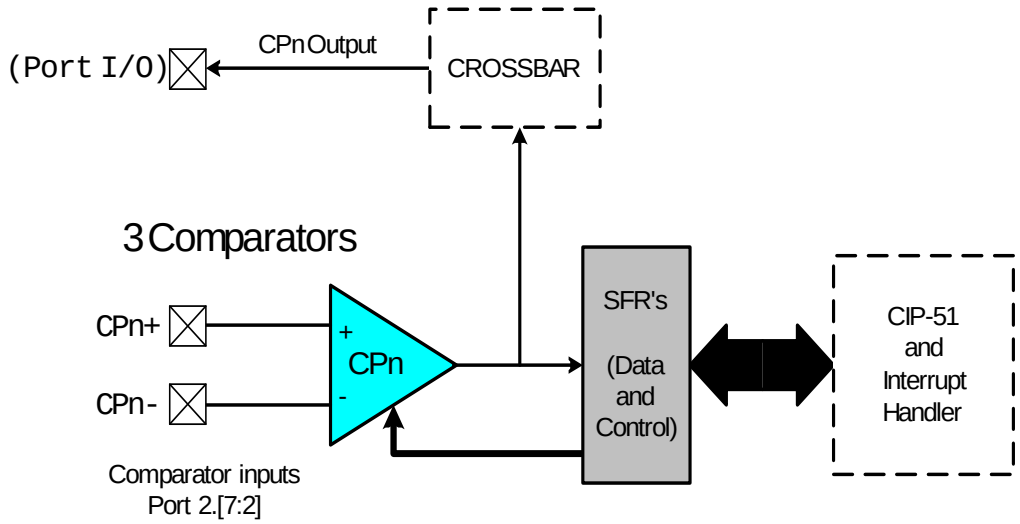



**Figure 1.15. Comparator Block Diagram**

## 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 VDD, AV+, AVDD, and Port 0) with respect to DGND | | -0.3 | | VDD + 0.3 | V |
| Voltage on any Port 0 Pin with respect to DGND. | | -0.3 | | 5.8 | V |
| Voltage on VDD, AV+, or AVDD with respect to DGND | | -0.3 | | 4.2 | V |
| Maximum Total current through VDD, AV+, AVDD, 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 |

* 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**

**-40 to +85 °C, 25 MHz System Clock unless otherwise specified.**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Analog Supply Voltage (AV+, AVDD) | (Note 1) | 2.7 | 3.0 | 3.6 | V |
| Digital Supply Voltage (VDD) | | 2.7 | 3.0 | 3.6 | V |
| Analog-to-Digital Supply Delta (\|VDD - AV+\| or \|VDD - AVDD\|) | | | | 0.5 | V |
| Supply Current from Analog Peripherals (active) | Internal REF, ADC, DAC, Comparators all enabled. (Note 2) | | 14 | | mA |
| Supply Current from Analog Peripherals (inactive) | Internal REF, ADC, DAC, Comparators all disabled, oscillator disabled. | | 0.2 | | µA |
| Supply Current from CPU and Digital Peripherals (CPU active) (Note 3) | VDD=2.7 V, Clock=25 MHz<br>VDD=2.7 V, Clock=1 MHz<br>VDD=2.7 V, Clock=32 kHz<br>VDD=3.0 V, Clock=25 MHz<br>VDD=3.0 V, Clock=1 MHz<br>VDD=3.0 V, Clock=32 kHz | | 18<br>0.7<br>30<br>20<br>1.0<br>35 | | mA<br>mA<br>µA<br>mA<br>mA<br>µA |
| Supply Current from CPU and Digital Peripherals (CPU inactive, not accessing Flash) (Note 3) | VDD=2.7 V, Clock=25 MHz<br>VDD=2.7 V, Clock=1 MHz<br>VDD=2.7 V, Clock=32 kHz<br>VDD=3.0 V, Clock=25 MHz<br>VDD=3.0 V, Clock=1 MHz<br>VDD=3.0 V, Clock=32 kHz | | 13<br>0.5<br>20<br>16<br>0.8<br>23 | | mA<br>mA<br>µA<br>mA<br>mA<br>µA |
| Supply Current with all systems shut down | Oscillator not running | | 0.2 | | µA |
| VDD Supply RAM Data Retention Voltage | | | 1.5 | | V |
| SYSCLK (System Clock) | (Note 4) | 0 | | 25 | MHz |
| Specified Operating Temperature Range | | -40 | | +85 | °C |

Note 1: Analog Supply AV+ must be greater than 1 V for VDD monitor to operate.
Note 2: Internal Oscillator and VDD Monitor current not included. Individual supply current contributions for each peripheral are listed in the chapter.
Note 3: Current increases linearly with supply Voltage.
Note 4: 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 |
|---|---|---|---|---|---|---|
| | F060 | F061 | F064 | F065 | | |
| | F062 | F063 | F066 | F067 | | |
| VDD | 37, 64, 90 | 26, 40, 55 | 37, 64, 90 | 26, 40, 55 | | Digital Supply Voltage. Must be tied to +2.7 to +3.6 V. |
| DGND | 38, 63, 89 | 27, 39, 54 | 38, 63, 89 | 27, 39, 54 | | Digital Ground. Must be tied to Ground. |
| AV+ | 11, 16, 24 | 7, 10, 18 | 11, 16, 24 | 7, 10, 18 | | Analog Supply Voltage. Must be tied to +2.7 to +3.6 V. |
| AVDD | 13 | 23 | 13 | 23 | | Analog Supply Voltage. Must be tied to +2.7 to +3.6 V. |
| AGND | 10, 14, 17, 23 | 6, 11, 19, 22 | 10, 14, 17, 23 | 6, 11, 19, 22 | | Analog Ground. Must be tied to Ground. |
| TMS | 96 | 52 | 96 | 52 | D In | JTAG Test Mode Select with internal pull-up. |
| TCK | 97 | 53 | 97 | 53 | D In | JTAG Test Clock with internal pull-up. |
| TDI | 98 | 56 | 98 | 56 | D In | JTAG Test Data Input with internal pull-up. TDI is latched on the rising edge of TCK. |
| TDO | 99 | 57 | 99 | 57 | D Out | JTAG Test Data Output with internal pull-up. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver. |
| /RST | 100 | 58 | 100 | 58 | D I/O | Device Reset. Open-drain output of internal VDD monitor. Is driven low when VDD is <2.7 V and MONEN is high. An external source can initiate a system reset by driving this pin low. |
| XTAL1 | 26 | 20 | 26 | 20 | 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 over-driven by an external CMOS clock, this becomes the system clock. |
| XTAL2 | 27 | 21 | 27 | 21 | A Out | Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator. |
| MONEN | 28 | 63 | 28 | 63 | D In | VDD Monitor Enable. When tied high, this pin enables the internal VDD monitor, which forces a system reset when VDD is < 2.7 V. When tied low, the internal VDD monitor is disabled. Recommended configuration is to connect directly to VDD. |
| VREF | 4 | 61 | 4 | 61 | A Out | Bandgap Voltage Reference Output |
| VREF0 | 21 | 15 | 21 | 15 | A I/O | Bandgap Voltage Reference Output for ADC0. ADC0 Voltage Reference Input. |

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

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | F060 | F061 | F064 | F065 | | |
| | F062 | F063 | F066 | F067 | | |
| VRGND0 | 20 | 14 | 20 | 14 | A In | ADC0 Voltage Reference Ground. This pin should be grounded if using the ADC. |
| VBGAP0 | 22 | 16 | 22 | 16 | A Out | ADC0 Bandgap Bypass Pin. |
| VREF1 | 6 | 2 | 6 | 2 | A I/O | Bandgap Voltage Reference Output for ADC1. ADC1 Voltage Reference Input. |
| VRGND1 | 7 | 3 | 7 | 3 | A In | ADC1 Voltage Reference Ground. This pin should be grounded if using the ADC. |
| VBGAP1 | 5 | 1 | 5 | 1 | A Out | ADC1 Bandgap Bypass Pin. |
| VREF2 | 2 | | | | A In | ADC2 Voltage Reference Input. |
| | | 62 | | | A In | ADC2, DAC0, and DAC1 Voltage Reference Input. |
| VREFD | 3 | | | | A In | DAC0 and DAC1 Voltage Reference Input. |
| AIN0 | 18 | 12 | 18 | 12 | A In | ADC0 Signal Input (See ADC0 Specification for complete description). |
| AIN0G | 19 | 13 | 19 | 13 | A In | ADC0 DC Bias Input (See ADC0 Specification for complete description). |
| AIN1 | 9 | 5 | 9 | 5 | A In | ADC1 Signal Input (See ADC1 Specification for complete description). |
| AIN1G | 8 | 4 | 8 | 4 | A In | ADC1 DC Bias Input (See ADC1 Specification for complete description). |
| CNVSTR0 | 15 | 9 | 15 | 9 | D In | External Conversion Start Source for ADC0 |
| CNVSTR1 | 12 | 8 | 12 | 8 | D In | External Conversion Start Source for ADC1 |
| CANTX | 94 | 59 | | | D Out | Controller Area Network Transmit Output. |
| CANRX | 95 | 60 | | | D In | Controller Area Network Receive Input. |
| DAC0 | 25 | 17 | | | A Out | Digital to Analog Converter 0 Voltage Output. (See DAC Specification for complete description). |
| DAC1 | 1 | 64 | | | A Out | Digital to Analog Converter 1 Voltage Output. (See DAC Specification for complete description). |
| P0.0 | 62 | 51 | 62 | 51 | D I/O | Port 0.0. See Port Input/Output section for complete description. |
| P0.1 | 61 | 50 | 61 | 50 | D I/O | Port 0.1. See Port Input/Output section for complete description. |
| P0.2 | 60 | 49 | 60 | 49 | D I/O | Port 0.2. See Port Input/Output section for complete description. |
| P0.3 | 59 | 48 | 59 | 48 | D I/O | Port 0.3. See Port Input/Output section for complete description. |
| P0.4 | 58 | 47 | 58 | 47 | D I/O | Port 0.4. See Port Input/Output section for complete description. |

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

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | F060 | F061 | F064 | F065 | | |
| | F062 | F063 | F066 | F067 | | |
| P0.5 | 57 | 46 | 57 | 46 | D I/O | Port 0.5. See Port Input/Output section for complete description. |
| P0.6 | 56 | 45 | 56 | 45 | D I/O | Port 0.6. See Port Input/Output section for complete description. |
| P0.7 | 55 | 44 | 55 | 44 | D I/O | Port 0.7. See Port Input/Output section for complete description. |
| P1.0/AIN2.0 | 36 | 33 | 36 | 33 | D I/O<br>A In | Port 1.0. See Port Input/Output section for complete description.<br>ADC2 Input Channel 0 (C8051F060/1/2/3 Only). |
| P1.1/AIN2.1 | 35 | 32 | 35 | 32 | D I/O<br>A In | Port 1.1. See Port Input/Output section for complete description.<br>ADC2 Input Channel 1 (C8051F060/1/2/3 Only). |
| P1.2/AIN2.2 | 34 | 31 | 34 | 31 | D I/O<br>A In | Port 1.2. See Port Input/Output section for complete description.<br>ADC2 Input Channel 2 (C8051F060/1/2/3 Only). |
| P1.3/AIN2.3 | 33 | 30 | 33 | 30 | D I/O<br>A In | Port 1.3. See Port Input/Output section for complete description.<br>ADC2 Input Channel 3 (C8051F060/1/2/3 Only). |
| P1.4/AIN2.4 | 32 | 29 | 32 | 29 | D I/O<br>A In | Port 1.4. See Port Input/Output section for complete description.<br>ADC2 Input Channel 4 (C8051F060/1/2/3 Only). |
| P1.5/AIN2.5 | 31 | 28 | 31 | 28 | D I/O<br>A In | Port 1.5. See Port Input/Output section for complete description.<br>ADC2 Input Channel 5 (C8051F060/1/2/3 Only). |
| P1.6/AIN2.6 | 30 | 25 | 30 | 25 | D I/O<br>A In | Port 1.6. See Port Input/Output section for complete description.<br>ADC2 Input Channel 6 (C8051F060/1/2/3 Only). |
| P1.7/AIN2.7 | 29 | 24 | 29 | 24 | D I/O<br>A In | Port 1.7. See Port Input/Output section for complete description.<br>ADC2 Input Channel 7 (C8051F060/1/2/3 Only). |
| P2.0 | 46 | 43 | 46 | 43 | D I/O | Port 2.0. See Port Input/Output section for complete description. |
| P2.1 | 45 | 42 | 45 | 42 | D I/O | Port 2.1. See Port Input/Output section for complete description. |
| P2.2 | 44 | 41 | 44 | 41 | D I/O | Port 2.2. See Port Input/Output section for complete description. |
| P2.3 | 43 | 38 | 43 | 38 | D I/O | Port 2.3. See Port Input/Output section for complete description. |
| P2.4 | 42 | 37 | 42 | 37 | D I/O | Port 2.4. See Port Input/Output section for complete description. |

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

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | F060 | F061 | F064 | F065 | | |
| | F062 | F063 | F066 | F067 | | |
| P2.5 | 41 | 36 | 41 | 36 | D I/O | Port 2.5. See Port Input/Output section for complete description. |
| P2.6 | 40 | 35 | 40 | 35 | D I/O | Port 2.6. See Port Input/Output section for complete description. |
| P2.7 | 39 | 34 | 39 | 34 | D I/O | Port 2.7. See Port Input/Output section for complete description. |
| P3.0 | 54 | | 54 | | D I/O | Port 3.0. See Port Input/Output section for complete description. |
| P3.1 | 53 | | 53 | | D I/O | Port 3.1. See Port Input/Output section for complete description. |
| P3.2 | 52 | | 52 | | D I/O | Port 3.2. See Port Input/Output section for complete description. |
| P3.3 | 51 | | 51 | | D I/O | Port 3.3. See Port Input/Output section for complete description. |
| P3.4 | 50 | | 50 | | D I/O | Port 3.4. See Port Input/Output section for complete description. |
| P3.5 | 49 | | 49 | | D I/O | Port 3.5. See Port Input/Output section for complete description. |
| P3.6 | 48 | | 48 | | D I/O | Port 3.6. See Port Input/Output section for complete description. |
| P3.7 | 47 | | 47 | | D I/O | Port 3.7. See Port Input/Output section for complete description. |
| P4.5/ALE | 93 | | 93 | | D I/O | Port 4.5. See Port Input/Output section for complete description.<br>ALE Strobe for External Memory Address Bus (Multiplexed mode). |
| P4.6/$\overline{RD}$ | 92 | | 92 | | D I/O | Port 4.6. See Port Input/Output section for complete description.<br>/RD Strobe for External Memory Address Bus. |
| P4.7/$\overline{WR}$ | 91 | | 91 | | D I/O | Port 4.7. See Port Input/Output section for complete description.<br>/WR Strobe for External Memory Address Bus. |
| P5.0/A8 | 88 | | 88 | | D I/O | Port 5.0. See Port Input/Output section for complete description.<br>Bit 8 External Memory Address Bus (Non-multiplexed mode). |
| P5.1/A9 | 87 | | 87 | | D I/O | Port 5.1. See Port Input/Output section for complete description. |

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

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | F060 | F061 | F064 | F065 | | |
| | F062 | F063 | F066 | F067 | | |
| P5.2/A10 | 86 | | 86 | | D I/O | Port 5.2. See Port Input/Output section for complete description. |
| P5.3/A11 | 85 | | 85 | | D I/O | Port 5.3. See Port Input/Output section for complete description. |
| P5.4/A12 | 84 | | 84 | | D I/O | Port 5.4. See Port Input/Output section for complete description. |
| P5.5/A13 | 83 | | 83 | | D I/O | Port 5.5. See Port Input/Output section for complete description. |
| P5.6/A14 | 82 | | 82 | | D I/O | Port 5.6. See Port Input/Output section for complete description. |
| P5.7/A15 | 81 | | 81 | | D I/O | Port 5.7. See Port Input/Output section for complete description. |
| P6.0/A8m/A0 | 80 | | 80 | | D I/O | Port 6.0. See Port Input/Output section for complete description.<br>Bit 8 External Memory Address Bus (Multiplexed mode).<br>Bit 0 External Memory Address Bus (Non-multiplexed mode). |
| P6.1/A9m/A1 | 79 | | 79 | | D I/O | Port 6.1. See Port Input/Output section for complete description. |
| P6.2/A10m/A2 | 78 | | 78 | | D I/O | Port 6.2. See Port Input/Output section for complete description. |
| P6.3/A11m/A3 | 77 | | 77 | | D I/O | Port 6.3. See Port Input/Output section for complete description. |
| P6.4/A12m/A4 | 76 | | 76 | | D I/O | Port 6.4. See Port Input/Output section for complete description. |
| P6.5/A13m/A5 | 75 | | 75 | | D I/O | Port 6.5. See Port Input/Output section for complete description. |
| P6.6/A14m/A6 | 74 | | 74 | | D I/O | Port 6.6. See Port Input/Output section for complete description. |
| P6.7/A15m/A7 | 73 | | 73 | | D I/O | Port 6.7. See Port Input/Output section for complete description. |
| P7.0/AD0m/D0 | 72 | | 72 | | D I/O | Port 7.0. See Port Input/Output section for complete description.<br>Bit 0 External Memory Address/Data Bus (Multiplexed mode).<br>Bit 0 External Memory Data Bus (Non-multiplexed mode). |
| P7.1/AD1m/D1 | 71 | | 71 | | D I/O | Port 7.1. See Port Input/Output section for complete description. |

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

| Name | Pin Numbers | | | | Type | Description |
|---|---|---|---|---|---|---|
| | F060 | F061 | F064 | F065 | | |
| | F062 | F063 | F066 | F067 | | |
| P7.2/AD2m/D2 | 70 | | 70 | | D I/O | Port 7.2. See Port Input/Output section for complete description. |
| P7.3/AD3m/D3 | 69 | | 69 | | D I/O | Port 7.3. See Port Input/Output section for complete description. |
| P7.4/AD4m/D4 | 68 | | 68 | | D I/O | Port 7.4. See Port Input/Output section for complete description. |
| P7.5/AD5m/D5 | 67 | | 67 | | D I/O | Port 7.5. See Port Input/Output section for complete description. |
| P7.6/AD6m/D6 | 66 | | 66 | | D I/O | Port 7.6. See Port Input/Output section for complete description. |
| P7.7/AD7m/D7 | 65 | | 65 | | D I/O | Port 7.7. See Port Input/Output section for complete description. |
| NC | | | 1, 2, 3, 25, 94, 95 | 17, 59, 60, 62, 64 | | No Connection. |

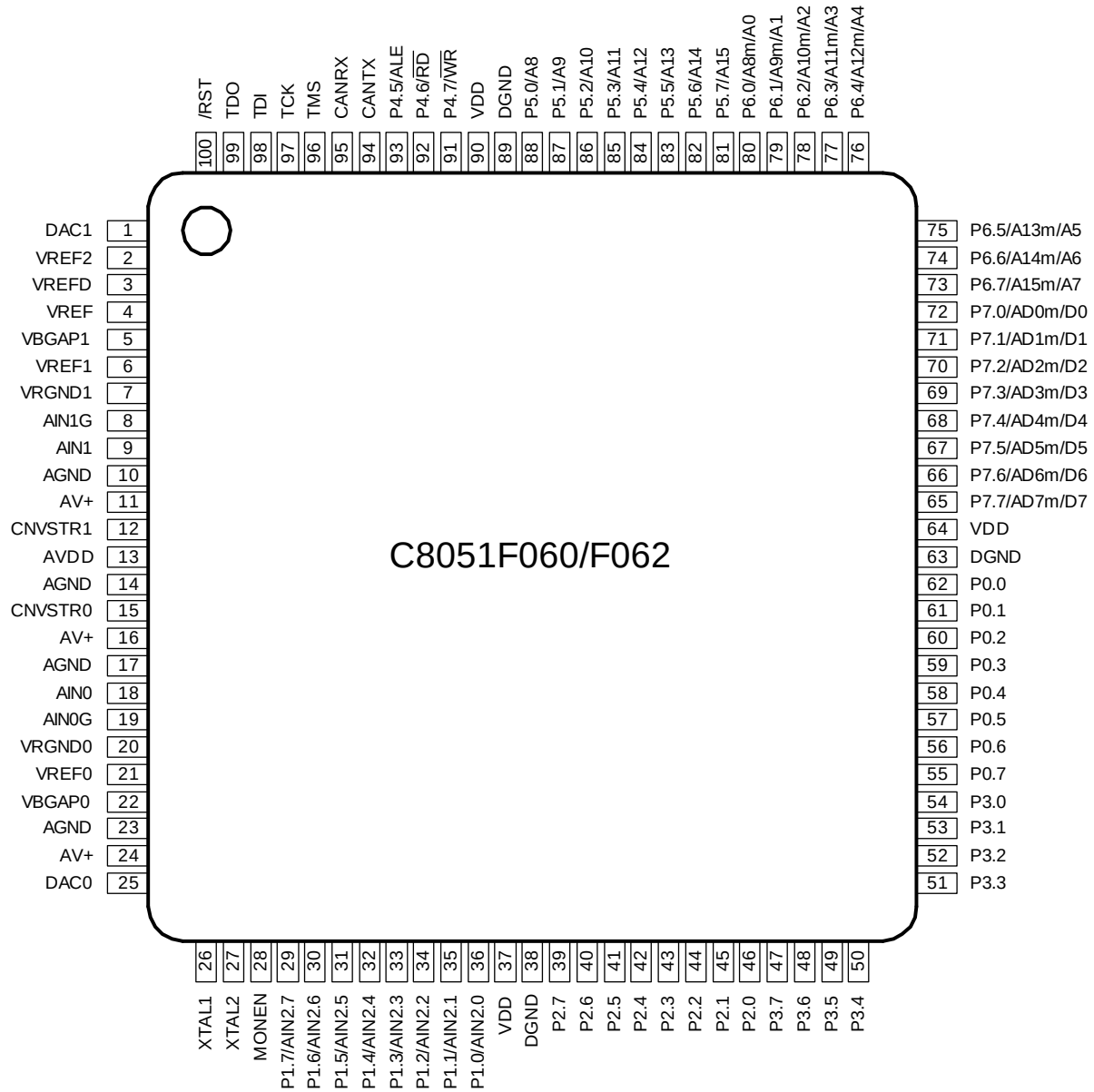



**Figure 4.1. C8051F060 / C8051F062 Pinout Diagram (TQFP-100)**

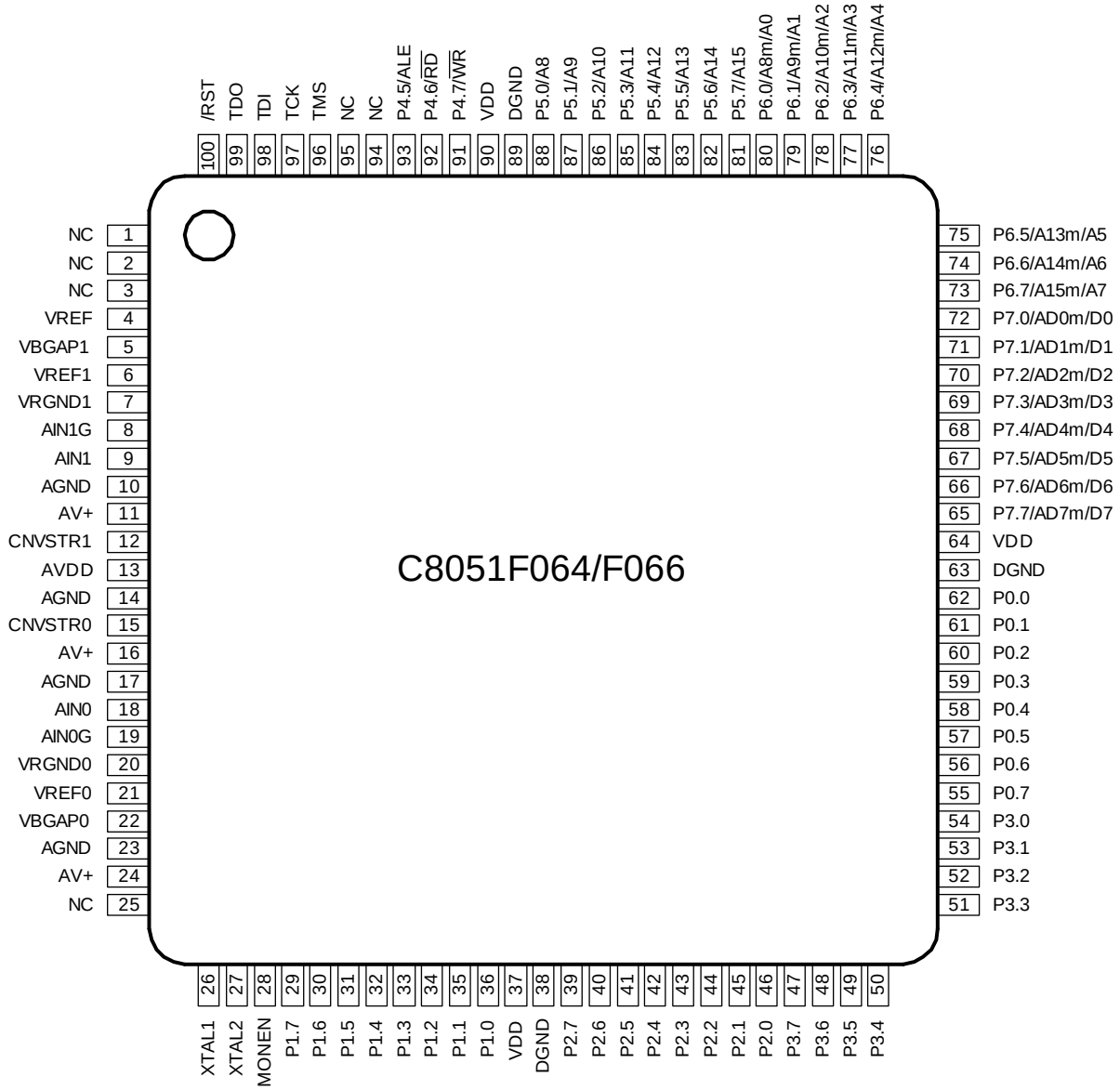
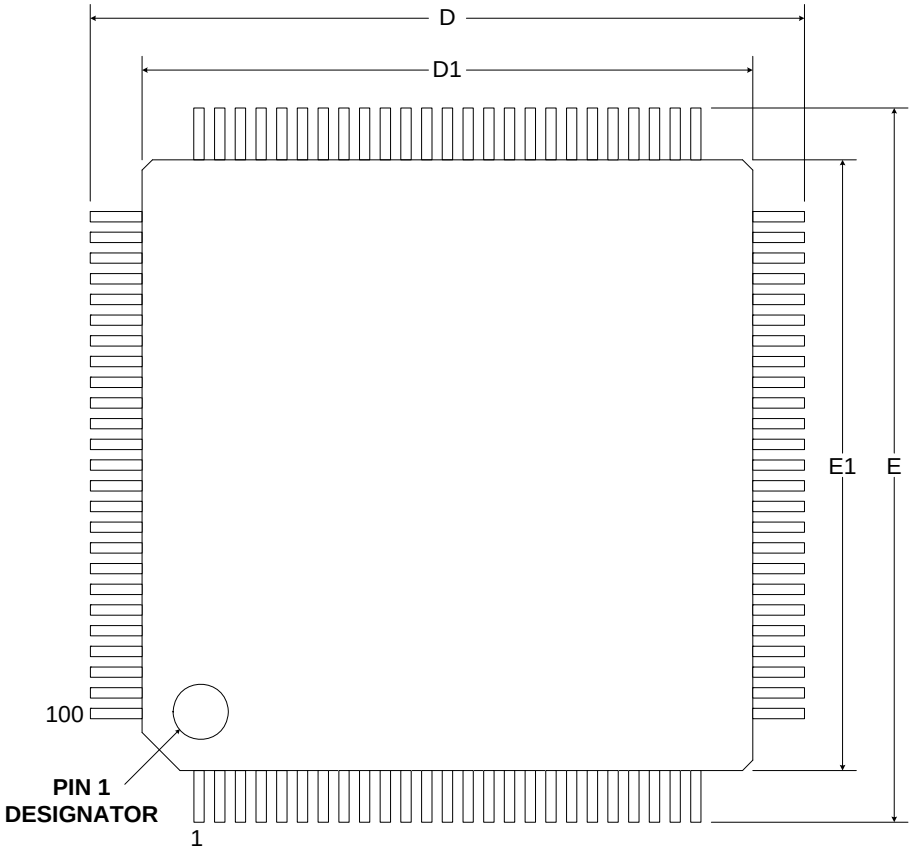



**Figure 4.2. C8051F064 / C8051F066 Pinout Diagram (TQFP-100)**




| | 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**

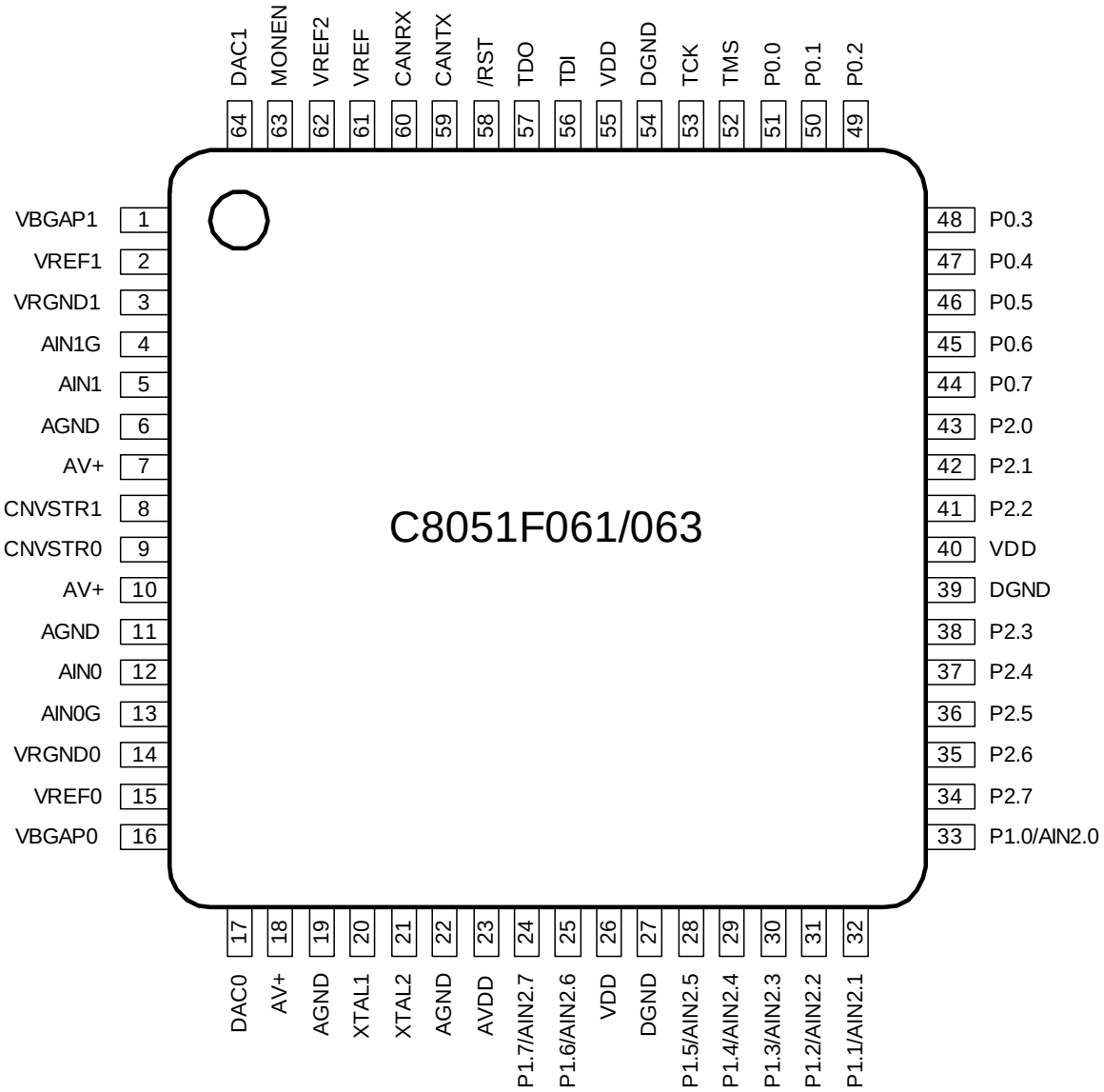



**Figure 4.4. C8051F061 / C8051F063 Pinout Diagram (TQFP-64)**

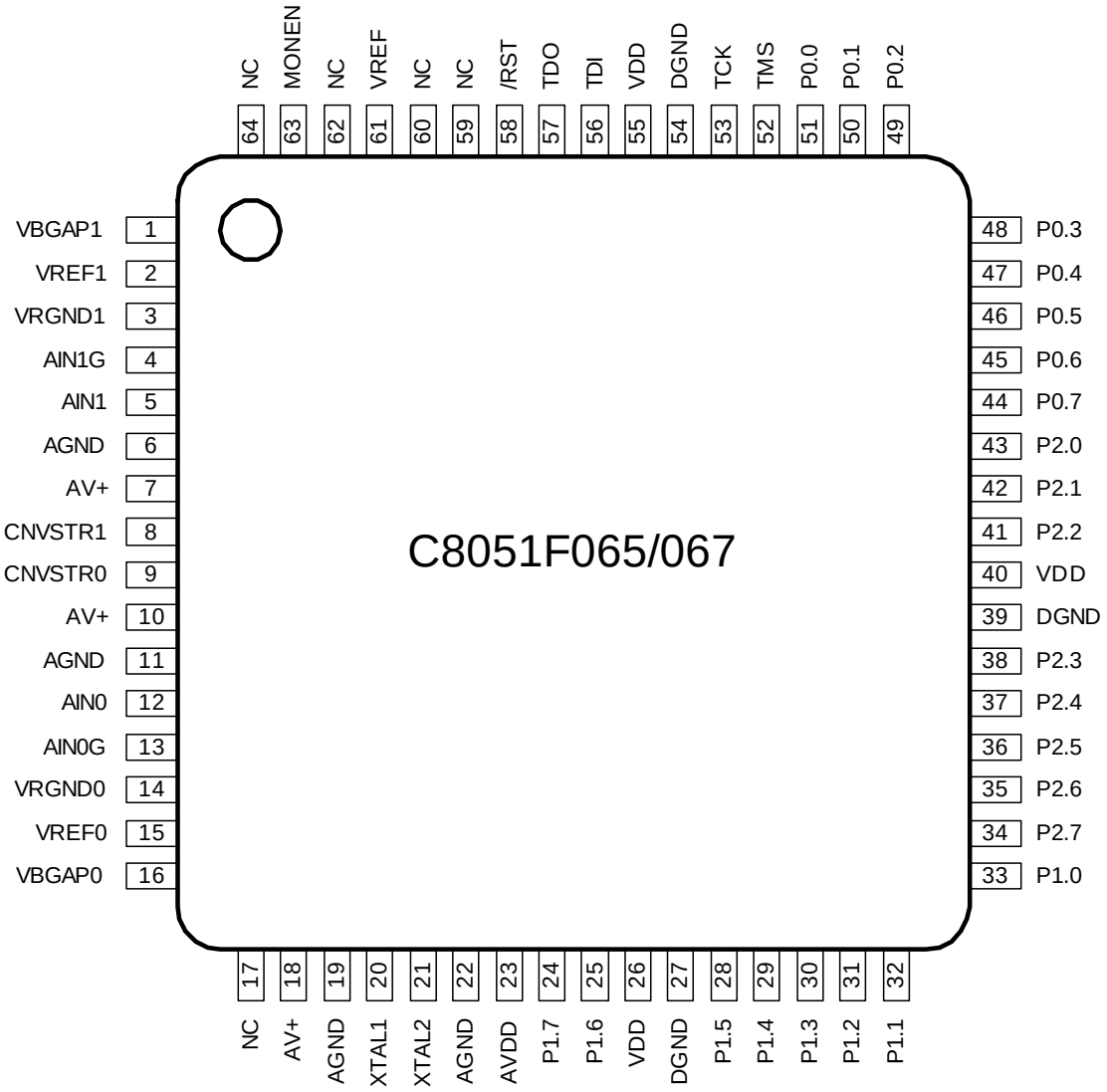
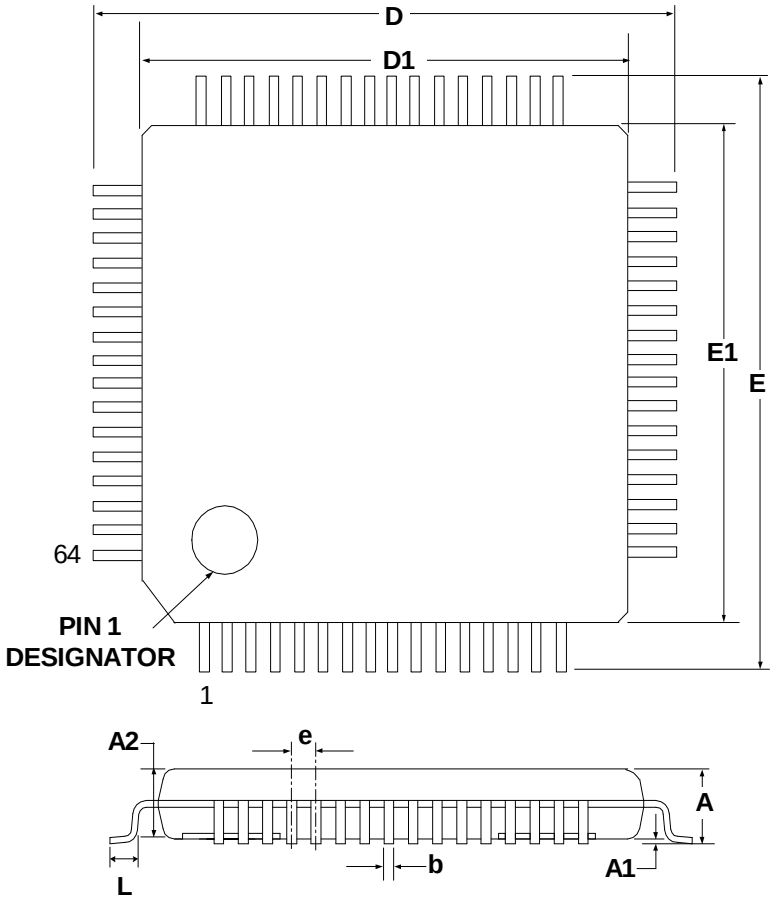



**Figure 4.5. C8051F065 / C8051F067 Pinout Diagram (TQFP-64)**




| | MIN (mm) | NOM (mm) | MAX (mm) |
|---|---|---|---|
| A | - | - | 1.20 |
| A1 | 0.05 | - | 0.15 |
| A2 | 0.95 | - | 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. 16-Bit ADCs (ADC0 and ADC1)

The ADC subsystem for the C8051F060/1/2/3/4/5/6/7 consists of two 1 Msps, 16-bit successive-approximation-register ADCs with integrated track-and-hold, a Programmable Window Detector, and a DMA interface (see block diagrams in Figure 5.1 and Figure 5.2). The ADCs can be configured as two separate, single-ended ADCs, or as a differential pair. The Data Conversion Modes, Window Detector, and DMA interface are all configurable under software control via the Special Function Registers shown in Figure 5.1 and Figure 5.2. The voltage references used by ADC0 and ADC1 are selected as described in Section 5.2. The ADCs and their respective track-and-hold circuitry can be independently enabled or disabled with the Special Function Registers. Either ADC can be enabled by setting the ADnEN bit in the ADC's Control register (ADCnCN) to logic 1. The ADCs are in low power shutdown when these bits are logic 0.

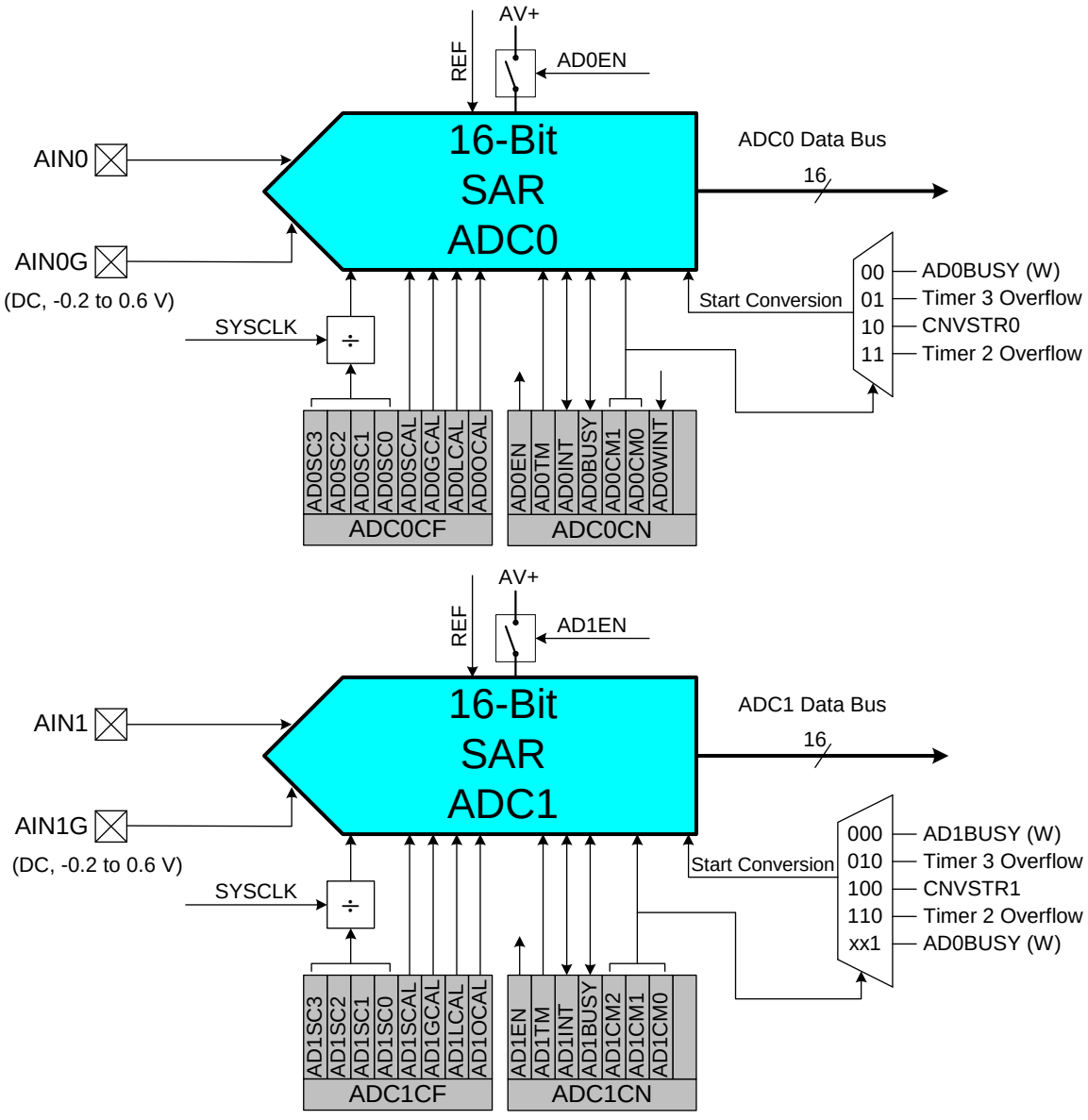



**Figure 5.1. 16-Bit ADC0 and ADC1 Control Path Diagram**

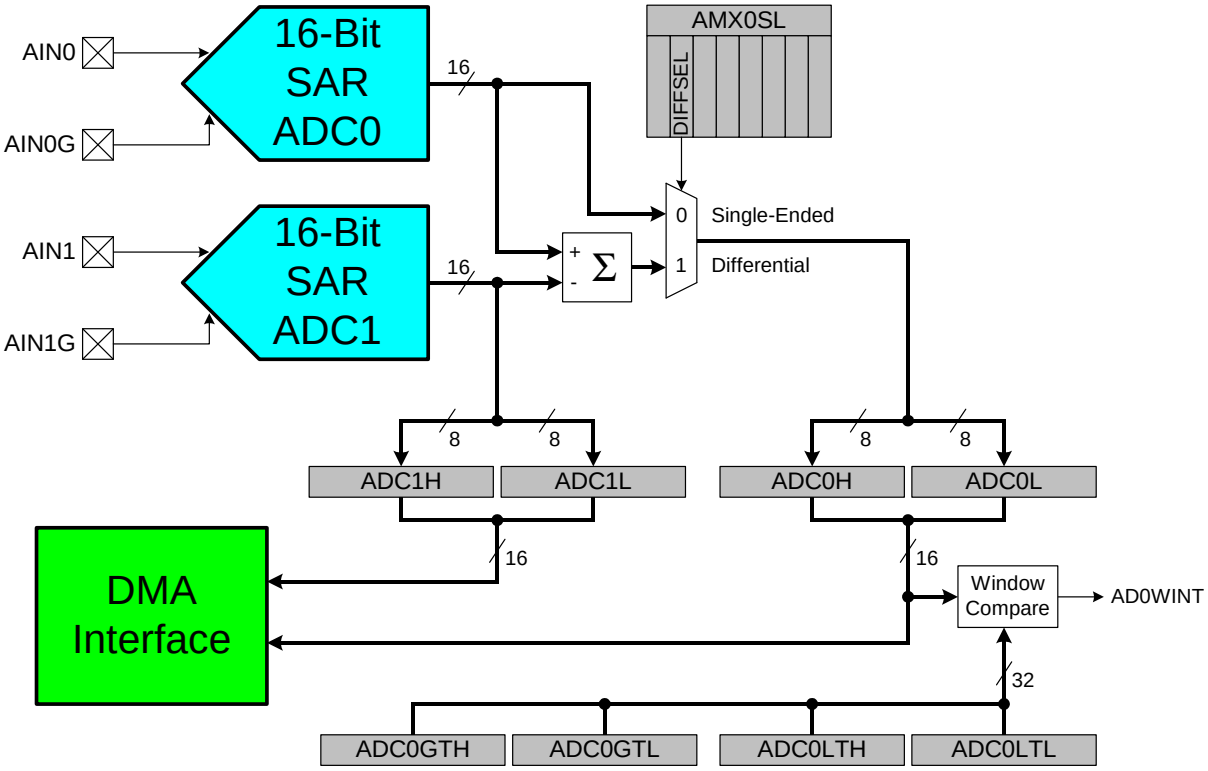



**Figure 5.2. 16-bit ADC0 and ADC1 Data Path Diagram**

## 5.1. Single-Ended or Differential Operation

ADC0 and ADC1 can be programmed to operate independently as single-ended ADCs, or together to accept a differential input. In single-ended mode, the ADCs can be configured to sample simultaneously, or to use different conversion speeds. In differential mode, ADC1 is a slave to ADC0, and its configuration is based on ADC0 settings, except during offset or gain calibrations. The DIFFSEL bit in the Channel Select Register AMX0SL (Figure 5.6) selects between single-ended and differential mode.

### 5.1.1. Pseudo-Differential Inputs

The inputs to the ADCs are pseudo-differential. The actual voltage measured by each ADC is equal to the voltage between the AINn pin and the AINnG pin. AINnG must be a DC signal between -0.2 and 0.6 V. In most systems, AINnG will be connected to AGND. If not tied to AGND, the AINnG signal can be used to negate a limited amount of fixed offset, but it is recommended that the internal offset calibration features of the device be used for this purpose. When operating in differential mode, AIN0G and AIN1G should be tied together. AINn must remain above AINnG in both modes for accurate conversion results.

## 5.2. Voltage Reference

The voltage reference circuitries for ADC0 and ADC1 allow for many different voltage reference configurations. Each ADC has the capability to use its own dedicated, on-chip voltage reference, or an off-chip reference circuit. A block diagram of the reference circuitry for one ADC is shown in Figure 5.3.

The internal voltage reference circuit for each ADC consists of an independent, temperature stable 1.2 V bandgap voltage reference generator, with an output buffer amplifier which multiplies the bandgap reference by 2. The maximum load seen by the VREFn (VREF0 or VREF1) pin must be less than 100 µA to AGND. Bypass capacitors of 0.1 µF and 47 µF are recommended from the VREFn pin to VRGNDn.

The voltage reference circuitry for each ADC is controlled in the Reference Control Registers. REF0CN (defined in Figure 5.11) is the Reference Control Register for ADC0, and REF1CN (defined in Figure 5.12) is the Reference Control Register for ADC1. The REFnCN registers are used to enable/disable the internal reference and bias generator circuitry for each ADC independently. The BIASEn bits enable the on-board bias generators for each ADC, while the REFBEn bits enable the 2x buffer amplifiers which drive the VREFn pins. 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 (approximately 25 k Ohms). If the internal voltage reference for an ADC is used, the BIASEn and REFBEn bits for that ADC must both be set to logic 1. If an external reference is used, the REFBEn bit should be set to logic 0. Note that the BIASEn bit for an ADC must be set to logic 1 to enable that ADC, regardless of the voltage reference that is used. If an ADC is not being used, the BIASEn bit can be set to logic 0 to conserve power. The electrical specifications for the Voltage References are given in Table 5.3.

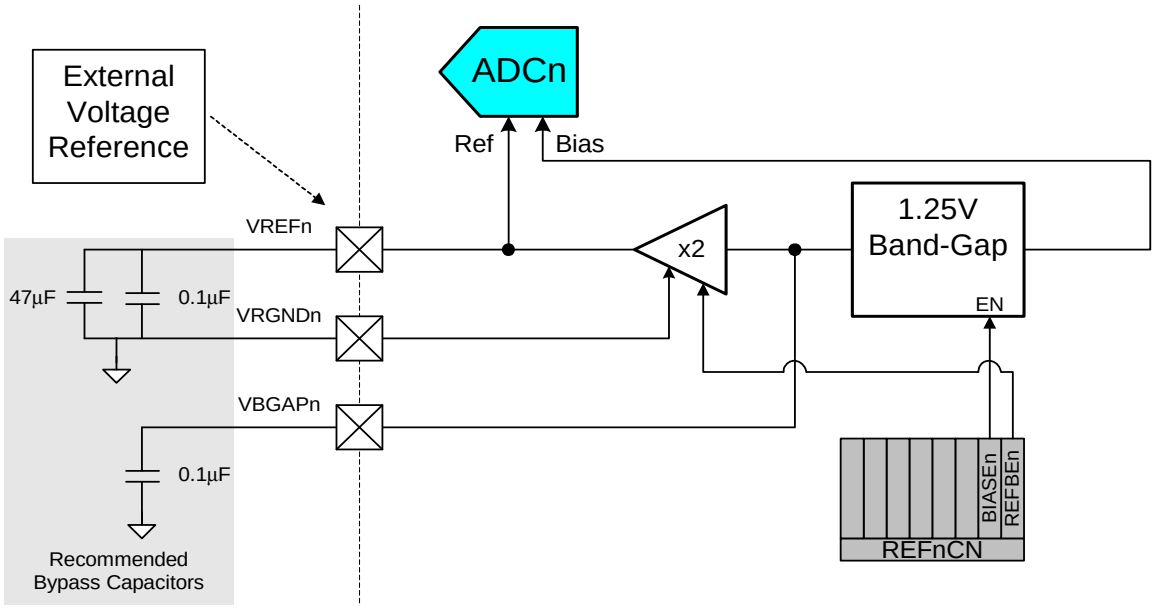



**Figure 5.3. Voltage Reference Block Diagram**

## 5.3. ADC Modes of Operation

ADC0 and ADC1 have a maximum conversion speed of 1 Msps. The conversion clocks for the ADCs are derived from the system clock. The ADCnSC bits in the ADCnCF register determine how many system clocks (from 1 to 16) are used for each conversion clock.

### 5.3.1. Starting a Conversion

For ADC0, conversions 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. For ADC0, 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).

ADC1 conversions can be initiated in five different ways, according to the ADC1 Start of Conversion Mode bits (AD1CM2-AD1CM0) in ADC1CN. For ADC1, conversions may be initiated by:

1. Writing a '1' to the AD1BUSY bit of ADC1CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR1;
4. A Timer 2 overflow (i.e. timed continuous conversions);
5. Writing a '1' to the AD0BUSY bit of ADC0CN.

The ADnBUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of ADnBUSY triggers an interrupt (when enabled) and sets the ADnINT interrupt flag (ADCnCN.5). In single-ended mode, the converted data for ADCn is available in the ADCn data word MSB and LSB registers, ADCnH, ADCnL. In differential mode, the converted data (combined from ADC0 and ADC1) is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L.

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

Step 1. Write a '0' to ADnINT;
Step 2. Write a '1' to ADnBUSY;
Step 3. Poll ADnINT for '1';
Step 4. Process ADCn data.

When an external start-of-conversion source is required in differential mode the two pins (CNVSTR0 and CNVSTR1) should be tied together.

### 5.3.2. Tracking Modes

The ADnTM bit in register ADCnCN controls the ADCn track-and-hold mode. When the ADC is enabled, the ADC input is continuously tracked when a conversion is not in progress. When the ADnTM bit is logic 1, each conversion is preceded by a tracking period (after the start-of-conversion signal). When the CNVSTRn signal is used to initiate conversions, the ADC will track until a rising edge occurs on the CNVSTRn pin (see Figure 5.4 and Table 5.1 for conversion timing parameters). Setting ADnTM to 1 can be useful to ensure that settling time requirements are met when an external multiplexer is used on the analog input (see Section "5.3.3. Settling Time Requirements" on page 56).

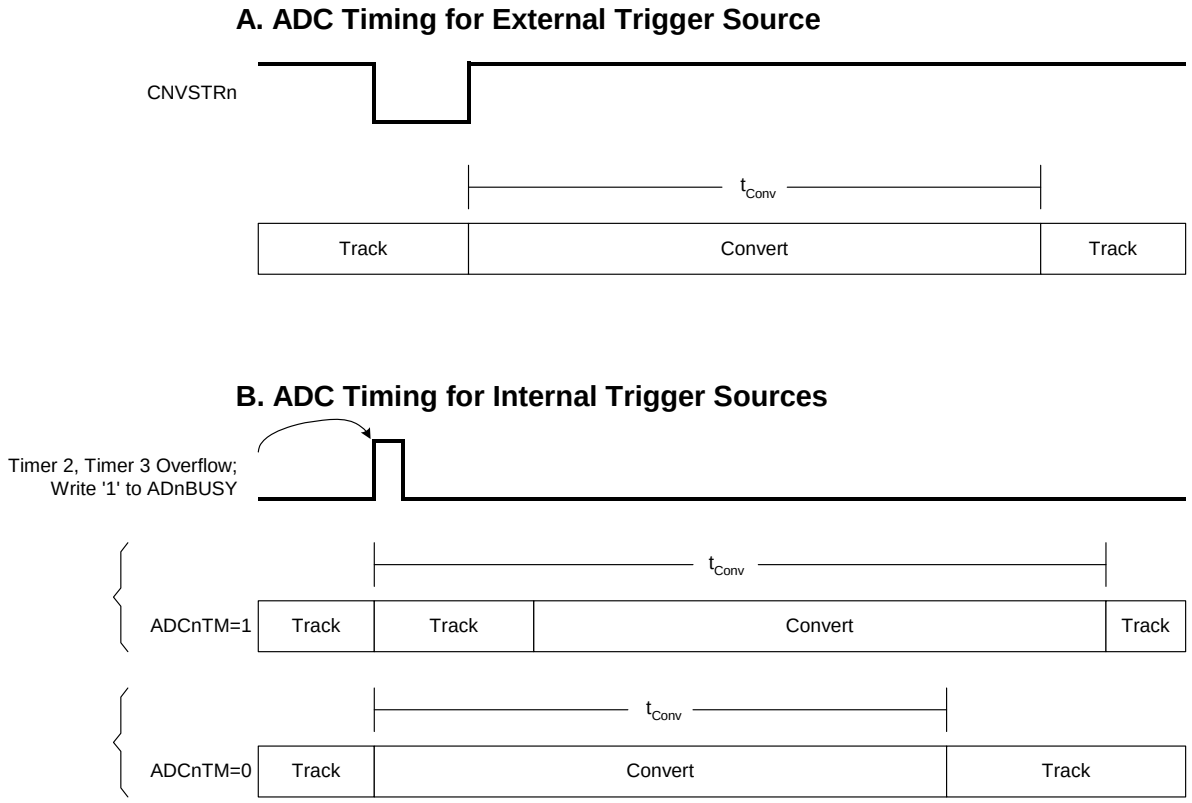



**Figure 5.4. ADC Track and Conversion Example Timing**

**Table 5.1. Conversion Timing ($t_{Conv}$)**

| ADnSC3-0 | ADCnTM = 0 | ADCnTM = 1 | ADnSC3-0 | ADCnTM = 0 | ADCnTM = 1 |
|---|---|---|---|---|---|
| 0000 | 21*$t_{SYSCLK}$ | 38*$t_{SYSCLK}$ | 1000 | 171*$t_{SYSCLK}$ | 315*$t_{SYSCLK}$ |
| 0001 | 40*$t_{SYSCLK}$ | 72*$t_{SYSCLK}$ | 1001 | 189*$t_{SYSCLK}$ | 349*$t_{SYSCLK}$ |
| 0010 | 58*$t_{SYSCLK}$ | 106*$t_{SYSCLK}$ | 1010 | 208*$t_{SYSCLK}$ | 384*$t_{SYSCLK}$ |
| 0011 | 78*$t_{SYSCLK}$ | 142*$t_{SYSCLK}$ | 1011 | 226*$t_{SYSCLK}$ | 418*$t_{SYSCLK}$ |
| 0100 | 97*$t_{SYSCLK}$ | 177*$t_{SYSCLK}$ | 1100 | 245*$t_{SYSCLK}$ | 453*$t_{SYSCLK}$ |
| 0101 | 115*$t_{SYSCLK}$ | 211*$t_{SYSCLK}$ | 1101 | 263*$t_{SYSCLK}$ | 487*$t_{SYSCLK}$ |
| 0110 | 134*$t_{SYSCLK}$ | 246*$t_{SYSCLK}$ | 1110 | 282*$t_{SYSCLK}$ | 522*$t_{SYSCLK}$ |
| 0111 | 152*$t_{SYSCLK}$ | 280*$t_{SYSCLK}$ | 1111 | 300*$t_{SYSCLK}$ | 556*$t_{SYSCLK}$ |

### 5.3.3. Settling Time Requirements

The ADC requires a minimum tracking time before an accurate conversion can be performed. This tracking time is determined by the ADC input resistance, the ADC sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 5.5 shows the equivalent ADC 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. An absolute minimum tracking time of 280 ns is required prior to the start of a conversion.

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

**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 ADC input resistance and any external source resistance.
*n* is the ADC resolution in bits (16).

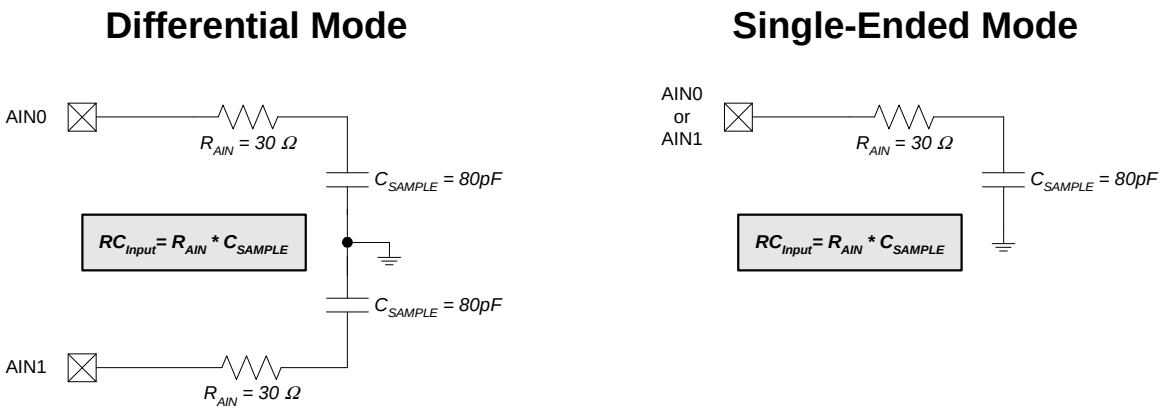



**Figure 5.5. ADC0 and ADC1 Equivalent Input Circuits**

**Figure 5.6. AMX0SL: AMUX Configuration Register**

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

SFR Address: 0xBB
SFR Page: 0

Bit 7: RESERVED. Write to 0b.
Bit 6: DIFFSEL: Fully Differential Conversion Mode Select Bit.
0: Operate In Single-Ended Mode.
1: Operate In Differential Mode.
Bit 5-0: RESERVED. Write to 000000b.

NOTE: For single-ended mode, the ADC0 Data Word is stored in ADC0H and ADC0L, while the ADC1 Data Word is stored in ADC1H and ADC1L.
In differential mode, the combined ADC Data Word is stored in ADC0H and ADC0L, and is a 2’s complement number. ADC1’s Data Word (single-ended) is also stored in ADC1H and ADC1L.

**Figure 5.7. ADC0CF: ADC0 Configuration Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD0SC3 | AD0SC2 | AD0SC1 | AD0SC0 | AD0SCAL | AD0GCAL | AD0LCAL | AD0OCAL | 11110000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xBC
SFR Page: 0

Bits 7-4: AD0SC3-0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is divided down from the system clock according to the AD0SC bits (AD0SC3-0). The number of system clocks used for each SAR conversion clock is equal to AD0SC + 1. (Note: the ADC0 SAR Conversion Clock should be less than or equal to 25 MHz). See Table 5.1 for conversion timing details.

Bit 3: AD0SCAL: System Calibration Enable.
0: Internal ground and reference voltage are used during offset and gain calibration.
1: External voltages can be used during offset and gain calibration.

Bit 2: AD0GCAL: Gain Calibration.
Read:
0: Gain Calibration is completed or not yet started.
1: Gain Calibration is in progress.
Write:
0: No Effect.
1: Initiates a gain calibration if ADC0 is idle.

Bit 1: AD0LCAL: Linearity Calibration
Read
0: Linearity Calibration is completed or not yet started
1: Linearity Calibration is in progress
Write
0: No Effect
1: Initiates a linearity calibration if ADC0 is idle

Bit 0: AD0OCAL: Offset Calibration.
Read:
0: Offset Calibration is completed or not yet started.
1: Offset Calibration is in progress.
Write:
0: No Effect.
1: Initiates an offset calibration if ADC0 is idle.

**Figure 5.8. ADC1CF: ADC1 Configuration Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD1SC3 | AD1SC2 | AD1SC1 | AD1SC0 | AD1SCAL | AD1GCAL | AD1LCAL | AD1OCAL | 11110000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xBC
SFR Page: 1

Bits 7-4: AD1SC3-0: ADC1 SAR Conversion Clock Period Bits.
SAR Conversion clock is divided down from the system clock according to the AD1SC bits (AD1SC3-0). The number of system clocks used for each SAR conversion clock is equal to AD1SC + 1. (Note: the ADC1 SAR Conversion Clock should be less than or equal to 25 MHz). See Table 5.1 for conversion timing details.

Bit 3: AD1SCAL: System Calibration Enable.
0: Internal ground and reference voltage are used for offset and gain calibration.
1: External voltages can be used for offset and gain calibration.

Bit 2: AD1GCAL: Gain Calibration.
Read:
0: Gain Calibration is completed or not yet started.
1: Gain Calibration is in progress.
Write:
0: No Effect.
1: Initiates a gain calibration if ADC1 is idle.

Bit 1: AD1LCAL: Linearity Calibration
Read
0: Linearity Calibration is completed or not yet started
1: Linearity Calibration is in progress
Write
0: No Effect
1: Initiates a linearity calibration if ADC1 is idle

Bit 0: AD1OCAL: Offset Calibration.
Read:
0: Offset Calibration is completed or not yet started.
1: Offset Calibration is in progress.
Write:
0: No Effect.
1: Initiates an offset calibration if ADC1 is idle.

**Figure 5.9. ADC0CN: ADC0 Control Register**

| 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 | - | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xE8
SFR Page: 0

Bit 7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions or calibrations.

Bit 6: AD0TM: ADC Track Mode Bit.
0: When the ADC is enabled, tracking is continuous unless a conversion is in process.
1: Tracking Defined by AD0CM1-0 bits.

Bit 5: 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.

Bit 4: 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.

Bits 3-2: 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 is followed by the conversion.
01: Tracking started by the overflow of Timer 3 and is followed by the conversion.
10: ADC0 conversion starts on rising CNVSTR0 edge.
11: Tracking started by the overflow of Timer 2 and is followed by the conversion.
**See Figure 5.4 and Table 5.1 for conversion timing parameters.**

Bit 1: 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.

Bit 0: RESERVED: Write to 0b.

**Figure 5.10. ADC1CN: ADC1 Control Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| AD1EN | AD1TM | AD1INT | AD1BUSY | AD1CM2 | AD1CM1 | AD1CM0 | - | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xE8
SFR Page: 1

Bit 7: AD1EN: ADC1 Enable Bit.
0: ADC1 Disabled. ADC1 is in low-power shutdown.
1: ADC1 Enabled. ADC1 is active and ready for data conversions or calibrations.

Bit 6: AD1TM: ADC Track Mode Bit.
0: When the ADC is enabled, tracking is continuous unless a conversion is in process.
1: Tracking Defined by AD1CM2-0 bits.

Bit 5: AD1INT: ADC1 Conversion Complete Interrupt Flag.
This flag must be cleared by software.
0: ADC1 has not completed a data conversion since the last time this flag was cleared.
1: ADC1 has completed a data conversion.

Bit 4: AD1BUSY: ADC1 Busy Bit.
Read:
0: ADC1 Conversion is complete or a conversion is not currently in progress. AD1INT is set to logic 1 on the falling edge of AD1BUSY.
1: ADC1 Conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC1 Conversion if AD1CM2-0 = 000b.

Bits 3-1: AD1CM2-0: ADC1 Start of Conversion Mode Select.
If AD1TM = 0:
000: ADC1 conversion initiated on every write of ‘1’ to AD1BUSY.
010: ADC1 conversion initiated on overflow of Timer 3.
100: ADC1 conversion initiated on rising edge of external CNVSTR1.
110: ADC1 conversion initiated on overflow of Timer 2.
xx1: ADC1 conversion initiated on every write of ‘1’ to AD0BUSY in ADC0CN
If AD1TM = 1:
000: Tracking starts with the write of ‘1’ to AD1BUSY and is followed by the conversion.
010: Tracking started by the overflow of Timer 3 and is followed by the conversion.
100: ADC1 conversion starts on rising CNVSTR1 edge.
110: Tracking started by the overflow of Timer 2 and is followed by the conversion.
xx1: Tracking starts with the write of ‘1’ to AD0BUSY and is followed by the conversion.
**See Figure 5.4 and Table 5.1 for conversion timing parameters.**

Bit 0: RESERVED: Write to 0b.

**Figure 5.11. REF0CN: Reference Control Register 0**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | BIASE0 | REFBE0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD1
SFR Page: 0

Bits7-2: RESERVED. Read = 000000b; Write = 000000b.
Bit1: BIASE0: ADC0 Bias Generator Enable Bit. (Must be '1' if using ADC0).
0: ADC0 Internal Bias Generator Off.
1: ADC0 Internal Bias Generator On.
Bit0: REFBE0: Internal Reference Buffer for ADC0 Enable Bit.
0: Internal Reference Buffer for ADC0 Off. External voltage reference can be used.
1: Internal Reference Buffer for ADC0 On. Internal voltage reference is driven on the VREF0 pin.

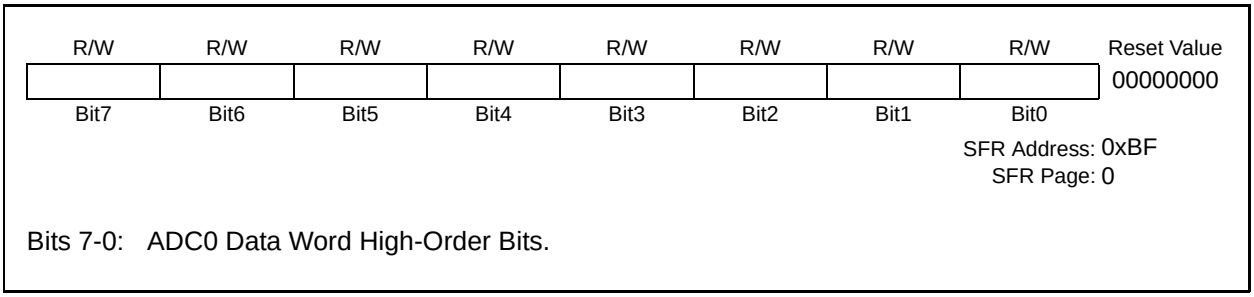
**Figure 5.12. REF1CN: Reference Control Register 1**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | BIASE1 | REFBE1 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD1
SFR Page: 1

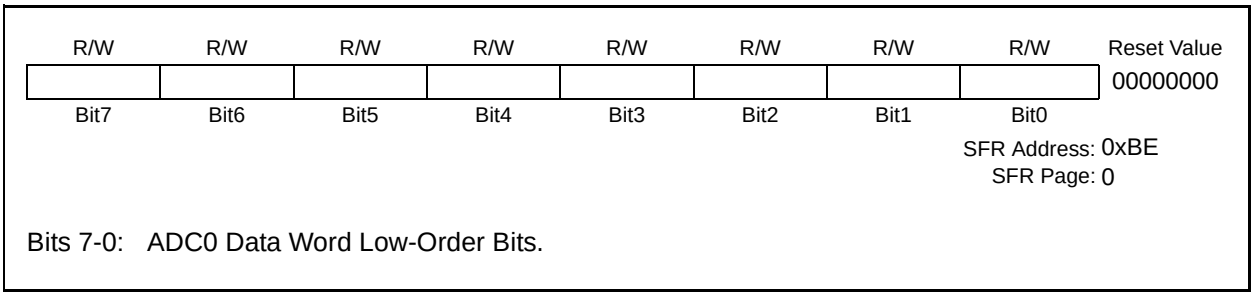
Bits7-2: RESERVED. Read = 000000b; Write = 000000b.
Bit1: BIASE1: ADC1 Bias Generator Enable Bit. (Must be '1' if using ADC1).
0: ADC1 Internal Bias Generator Off.
1: ADC1 Internal Bias Generator On.
Bit0: REFBE1: Internal Reference Buffer for ADC1 Enable Bit.
0: Internal Reference Buffer for ADC1 Off. External voltage reference can be used.
1: Internal Reference Buffer for ADC1 On. Internal voltage reference is driven on the VREF1 pin.

**Figure 5.13. ADC0H: ADC0 Data Word MSB Register**




Bits 7-0: ADC0 Data Word High-Order Bits.

**Figure 5.14. ADC0L: ADC0 Data Word LSB Register**




Bits 7-0: ADC0 Data Word Low-Order Bits.

**Figure 5.15. ADC0 Data Word Example**

**16-bit ADC0 Data Word appears in the ADC0 Data Word Registers as follows:**

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

| AIN0-AIN0G (Volts) | **ADC0H:ADC0L** |
|---|---|
| VREF * (65535/65536) | 0xFFFF |
| VREF / 2 | 0x8000 |
| VREF * (32767/65536) | 0x7FFF |
| 0 | 0x0000 |

Example: ADC0 Data Word Conversion Map, AIN0-AIN1 Differential Input Pair (AMX0SL = 0x40)

| AIN0-AIN1 (Volts) | **ADC0H:ADC0L** |
|---|---|
| VREF * (32767/32768) | 0x7FFF |
| VREF / 2 | 0x4000 |
| VREF * (1/32768) | 0x0001 |
| 0 | 0x0000 |
| -VREF * (1/32768) | 0xFFFF |
| -VREF / 2 | 0xC000 |
| -VREF | 0x8000 |

$$Code = Vin \times \frac{Gain}{VREF} \times 2^n$$ ; 'n' = 16 for Single-Ended; 'n'=15 for Differential.

**Figure 5.16. ADC1H: ADC1 Data Word MSB 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: 0xBF
SFR Page: 1

Bits 7-0: ADC1 Data Word High-Order Bits.

**Figure 5.17. ADC1L: ADC1 Data Word LSB 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: 0xBE
SFR Page: 1

Bits 7-0: ADC1 Data Word Low-Order Bits.

**Figure 5.18. ADC1 Data Word Example**

**16-bit ADC1 Data Word appears in the ADC1 Data Word Registers as follows:**

Example: ADC1 Data Word Conversion Map, AIN1 Input in Single-Ended Mode (AMX1SL = 0x00)

| AIN1-AIN1G (Volts) | **ADC1H:ADC1L** |
|---|---|
| VREF * (65535/65536) | 0xFFFF |
| VREF / 2 | 0x8000 |
| VREF * (32767/65536) | 0x7FFF |
| 0 | 0x0000 |

$$Code = Vin \times \frac{Gain}{VREF} \times 2^n \quad \text{; 'n' = 16}$$

For differential mode, the differential data word appears in ADC0H and ADC0L. The single-ended ADC1 results are always present in ADC1H and ADC1L, regardless of the operating mode.

## 5.4. Calibration

The ADCs are calibrated for linearity, offset, and gain in production. ADC0 and ADC1 can also be independently calibrated for each of these parameters in-system. Calibrations are initiated using bits in the ADC0 or ADC1 Configuration Register. The calibration coefficients can be accessed using the ADC Calibration Pointer Register (ADC0CPT, Figure 5.22) and the ADC Calibration Coefficient Register (ADC0CCF, Figure 5.23). The CPTR bits in ADC0CPT allow the ADC0CCF register to read and write specific calibration coefficients. Figure 5.19 shows the Calibration Coefficient locations.

**Figure 5.19. Calibration Coefficient Locations**

| | ADC0CCF | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ADC0CPT Bits 5-0 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| 0x00 | Linearity Calibration Coefficients (locations 0x00 through 0x12) | | | | | | | |
| . . | | | | | | | | |
| 0x12 | | | | | | | | |
| 0x13 | Offset7 | Offset6 | Offset5 | Offset4 | Offset3 | Offset2 | Offset1 | Offset0 |
| 0x14 | | | Offset13 | Offset12 | Offset11 | Offset10 | Offset9 | Offset8 |
| 0x15 | Gain7 | Gain6 | Gain5 | Gain4 | Gain3 | Gain2 | Gain1 | Gain0 |
| 0x16 | | | | Gain12 | Gain11 | Gain10 | Gain9 | Gain8 |

The ADCs are calibrated for linearity in production. Under normal circumstances, no additional linearity calibration is necessary. If linearity calibrations are desired, they can be initiated by setting the ADCnLCAL bit to ‘1’. When the calibration is finished, the ADCnLCAL bit will be set to ‘0’ by the hardware. Linearity Calibration Coefficients are stored in the locations shown in Figure 5.19.

Offset and gain calibrations can be performed using either internal or external voltages as calibration sources. The ADCnSCAL bit determines whether the internal or external voltages are used in the calibration process. To ensure accuracy, offset calibration should be done prior to a gain calibration. The offset and gain calibration coefficients are decoded in Figure 5.20. Offset calibration is initiated by setting the ADCnOCAL bit to ‘1’. When the calibration is finished, the ADCnOCAL bit will be set to ‘0’ by the hardware. Offset calibration can compensate for offset errors of approximately ±3.125% of full scale. The offset value is added to the AINnG input prior to digitization by the ADC. Gain calibration is initiated by setting the ADCnGCAL bit to ‘1’. When the calibration is finished, the ADCnGCAL bit will be set to ‘0’ by the hardware. Gain calibration can compensate for slope errors of approximately ±3.125%. The gain value is added to the ADC’s VREF path to change the slope of the converter’s transfer function. Figure 5.21 shows how the offset and gain values affect the analog signals used by the ADC.

**Figure 5.20. Offset and Gain Register Mapping**

**The offset register value affects the offset at the analog input as follows:**

| Offset Register (14 Bits) | Approximate Offset Change (V) |
|---|---|
| 0x3FFF | -3.125% * VREF |
| 0x2000 | 0 |
| 0x0000 | +3.125% * VREF |

$$\text{Offset Change} \cong \frac{0x2000 - \text{Offset Register}}{8192} \times 3.125\% \times VREF$$

**The gain register value affects the slope of the ADC transfer function as follows:**

| Gain Register (13 Bits) | Approximate Slope Change |
|---|---|
| 0x1FFF | +3.125% |
| 0x1000 | 0 |
| 0x0000 | -3.125% |

$$\text{Slope Change} \cong \frac{\text{Gain Register} - 0x1000}{4096} \times 3.125\%$$

**Figure 5.21. Offset and Gain Calibration Block Diagram**

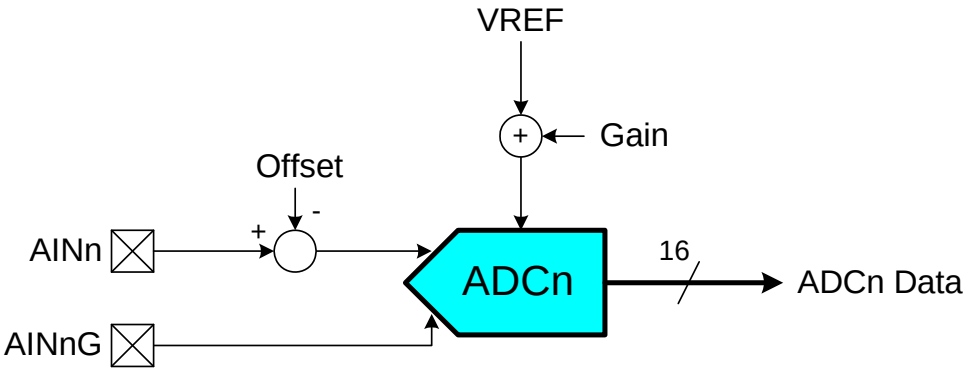


**Figure 5.22. ADC0CPT: ADC Calibration Pointer Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| INCR | ADCSEL | CPTR5 | CPTR4 | CPTR3 | CPTR2 | CPTR1 | CPTR0 | 11010111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xBA
SFR Page: F

Bit 7: INCR: Pointer Address Automatic Increment.
0: Disable Auto-Increment.
1: Enable Auto-Increment. CPTR5-0 will automatically be incremented after each read or write to ADC0CCF.

Bit 6: ADCSEL: ADC Calibration Coefficient Select.
0: Reads and Writes of ADC0CCF will access ADC0 Calibration Coefficients.
1: Reads and Writes of ADC0CCF will access ADC1 Calibration Coefficients.

Bits 5-0: CPTR5-0: Calibration Coefficent Pointer.
Select which Calibration Coefficient location will be accessed when ADC0CCF is read or written.

**Figure 5.23. ADC0CCF: ADC Calibration Coefficient Register**

| 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: 0xBB
SFR Page: F

Bits 7-0: Calibration Coefficients at the location specified in ADC0CPT. See Table 5.19.

## 5.5. 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). The Window Detector can be used in single-ended or differential mode. In signle-ended mode, the Window Detector compares the ADC0GTx and ADC0LTx registers to the output of ADC0. In differential mode, the combined output of ADC0 and ADC1 (contained in the ADC0 data registers) is used for the comparison. Reference comparisons are shown starting on page 71. 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.

**Figure 5.24. ADC0GTH: ADC0 Greater-Than Data High Byte Register**

| 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: 0xC5
SFR Page: 0

Bits 7-0: High byte of ADC0 Greater-Than Data Word.

**Figure 5.25. ADC0GTL: ADC0 Greater-Than Data Low Byte Register**

| 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: 0xC4
SFR Page: 0

Bits 7-0: Low byte of ADC0 Greater-Than Data Word.

**Figure 5.26. ADC0LTH: ADC0 Less-Than Data High Byte 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: 0xC7
SFR Page: 0

Bits 7-0: High byte of ADC0 Less-Than Data Word.

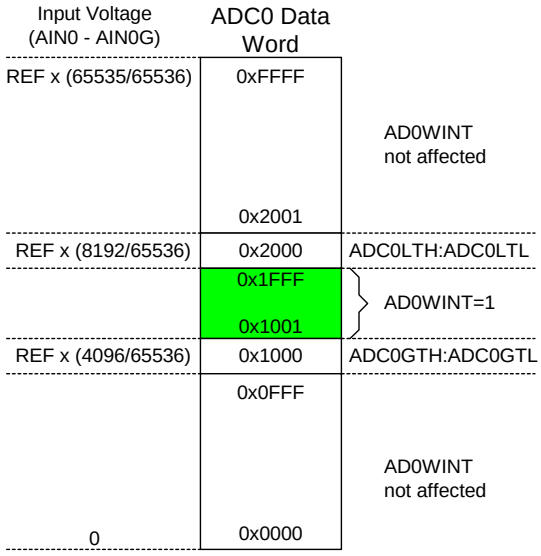
**Figure 5.27. ADC0LTL: ADC0 Less-Than Data Low Byte Register**

| 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 | |

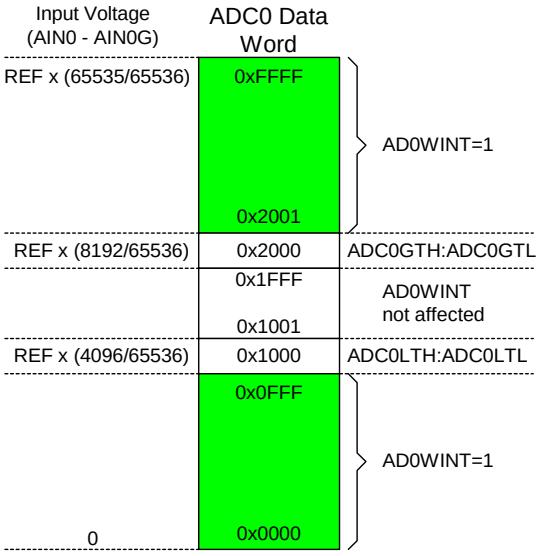
SFR Address: 0xC6
SFR Page: 0

Bits 7-0: Low byte of ADC0 Less-Than Data Word.

**Figure 5.28. 16-Bit ADC0 Window Interrupt Example: Single-Ended Data**

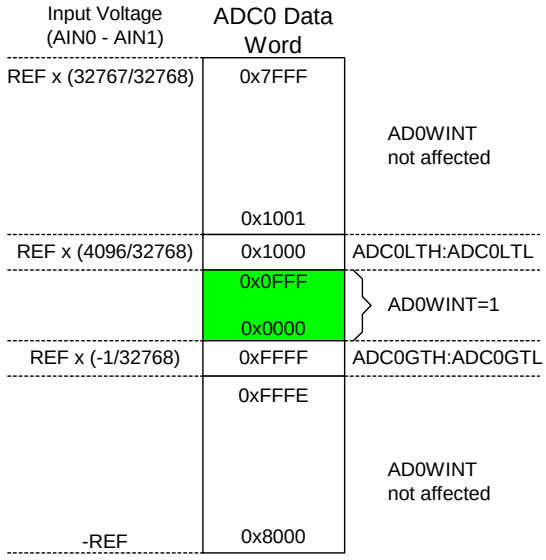



Given:
AMX0SL = 0x00,
ADC0LTH:ADC0LTL = 0x2000,
ADC0GTH:ADC0GTL = 0x1000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = ‘1’) if the resulting ADC0 Data Word is < 0x2000 and > 0x1000.

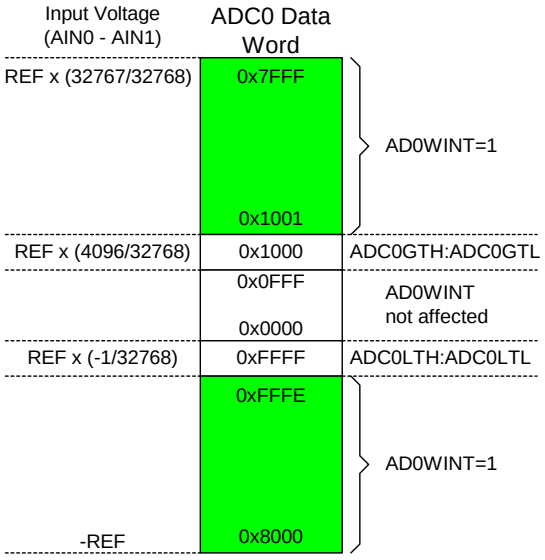



Given:
AMX0SL = 0x00,
ADC0LTH:ADC0LTL = 0x1000,
ADC0GTH:ADC0GTL = 0x2000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = ‘1’) if the resulting ADC0 Data Word is > 0x2000 or < 0x1000.

**Figure 5.29. 16-Bit ADC0 Window Interrupt Example: Differential Data**




Given:
AMX0SL = 0x40,
ADC0LTH:ADC0LTL = 0x1000,
ADC0GTH:ADC0GTL = 0xFFFF.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = ‘1’) if the resulting ADC0 Data Word is < 0x1000 and > 0xFFFF. (In two’s-complement math, 0xFFFF = -1.)




Given:
AMX0SL = 0x40,
ADC0LTH:ADC0LTL = 0xFFFF,
ADC0GTH:ADC0GTL = 0x1000.
An ADC0 End of Conversion will cause an ADC0 Window Compare Interrupt (AD0WINT = ‘1’) if the resulting ADC0 Data Word is < 0xFFFF or > 0x1000. (In two’s-complement math, 0xFFFF = -1.)

**Table 5.2. 16-Bit ADC0 and ADC1 Electrical Characteristics**

**VDD = 3.0 V, AV+ = 3.0 V, AVDD = 3.0 V, VREF = 2.50 V (REFBE=0), -40 to +85 °C unless otherwise specified**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 16 | | bits |
| Integral Nonlinearity<br>(C8051F060/1/4/5/6/7) | Single-Ended<br>Differential | | ±0.75<br>±0.5 | ±2<br>±1 | LSB |
| Integral Nonlinearity<br>(C8051F062/3) | Single-Ended<br>Differential | | ±1.5<br>±1 | ±4<br>±2 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | | ±0.5 | | LSB |
| Offset Error | | | 0.1 | | mV |
| Full Scale Error | | | 0.008 | | %F.S. |
| Gain Temperature Coefficient | | | 0.5 | | ppm/°C |
| **Dynamic Performance (Sampling Rate = 1 Msps, AVDD, AV+ = 3.3V)** | | | | | |
| Signal-to-Noise Plus Distortion | Fin = 10 kHz, Single-Ended<br>Fin = 100 kHz, Single-Ended<br>Fin = 10 kHz, Differential<br>Fin = 100 kHz, Differential | | 86<br>84<br>89<br>88 | | dB<br>dB<br>dB<br>dB |
| Total Harmonic Distortion | Fin = 10 kHz, Single-Ended<br>Fin = 100 kHz, Single-Ended<br>Fin = 10 kHz, Differential<br>Fin = 100 kHz, Differential | | 96<br>84<br>103<br>93 | | dB<br>dB<br>dB<br>dB |
| Spurious-Free Dynamic Range | Fin = 10 kHz, Single-Ended<br>Fin = 100 kHz, Single-Ended<br>Fin = 10 kHz, Differential<br>Fin = 100 kHz, Differential | | 97<br>88<br>104<br>99 | | dB<br>dB<br>dB<br>dB |
| CMRR | Fin = 10 kHz | | 86 | | dB |
| Channel Isolation | | | 100 | | dB |
| **Timing** | | | | | |
| SAR Clock Frequency | | | | 25 | MHz |
| Conversion Time in SAR Clocks | | 18 | | | clocks |
| Track/Hold Acquisition Time | | 280 | | | ns |
| Throughput Rate | | | | 1 | Msps |
| Aperture Delay | External CNVST Signal | | 1.5 | | ns |
| RMS Aperture Jitter | External CNVST Signal | | 5 | | ps |
| **Analog Inputs** | | | | | |
| Input Voltage Range | Single-Ended (AINn - AINnG)<br>Differential (AIN0 - AIN1) | 0<br>-VREF | | VREF<br>VREF | V<br>V |
| Input Capacitance | | | 80 | | pF |

**Table 5.2. 16-Bit ADC0 and ADC1 Electrical Characteristics (Continued)**

**VDD = 3.0 V, AV+ = 3.0 V, AVDD = 3.0 V, VREF = 2.50 V (REFBE=0), -40 to +85 °C unless otherwise specified**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Operating Input Range | AIN0 or AIN1<br>AIN0G or AIN1G (DC Only) | -0.2<br>-0.2 | | AV+<br>0.6 | V<br>V |
| **Power Specifications** | | | | | |
| Power Supply Current (each ADC) | Operating Mode, 1 Msps<br>AV+<br>AVDD<br>Shutdown Mode | | <br>4.0<br>2.0<br><1 | | <br>mA<br>mA<br>µA |
| Power Supply Rejection | VDD ± 5% | | ±0.5 | | LSB |

**Table 5.3. Voltage Reference 0 and 1 Electrical Characteristics**

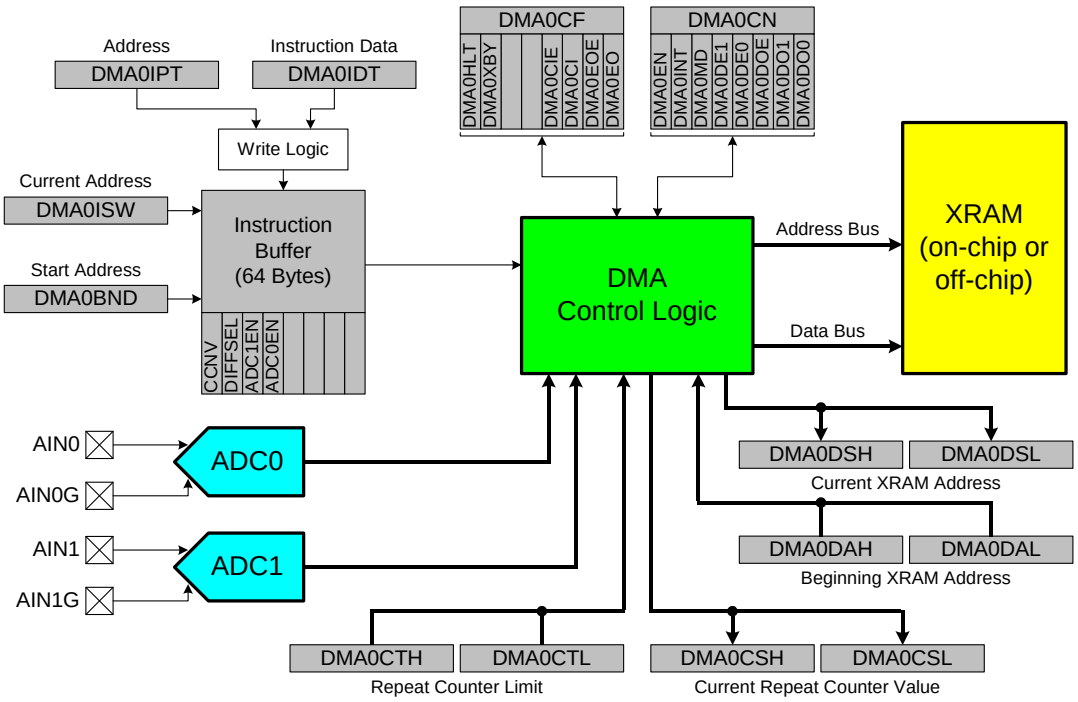
**VDD = 3.0 V, AV+ = 3.0 V, AVDD = 3.0 V, -40 to +85 °C unless otherwise specified**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Internal Reference** | | | | | |
| Output Voltage | 25 °C ambient | 2.36 | 2.43 | 2.48 | V |
| VREF Temperature Coefficient | | | 15 | | ppm/°C |
| Power Supply Current (each Voltage Reference) | AV+ | | 1.5 | | mA |
| **External Reference** | | | | | |
| Input Voltage Range | | 2.0 | | AV+ | V |
| Input Current | ADC throughput = 1 Msps | | 450 | | µA |

# 6. Direct Memory Access Interface (DMA0)

The DMA interface works in conjunction with ADC0 and ADC1 to write ADC outputs directly to a specified region of XRAM. The DMA interface is configured by software using the Special Function Registers shown in Figure 6.1. Up to 64 instructions can be programmed into the Instruction Buffer to designate a sequence of DMA operations. The Instruction Buffer is accessed by the DMA Control Logic, which gathers the appropriate data from the ADCs and controls writes to XRAM. The DMA instructions tell the DMA Control Logic which ADC(s) to expect results from, but do not initiate the actual conversions. It is important to configure the ADCs for the desired start-of-conversion source, voltage reference, and SAR clock frequency prior to starting the DMA interface. For information on setting up the ADCs, refer to Section “5. 16-Bit ADCs (ADC0 and ADC1)” on page 51.

**Figure 6.1. DMA0 Block Diagram**




## 6.1. Writing to the Instruction Buffer

The Instruction Buffer has 64 8-bit locations that can be programmed with a sequence of DMA instructions. Filling the Instruction Buffer is done with the Special Function Registers DMA0IPT (DMA Instruction Write Address Register, Figure 6.6) and DMA0IDT (DMA Instruction Write Data Register, Figure 6.7). Instructions are written to the Instruction Buffer at address DMA0IPT when the instruction word is written to DMA0IDT. Reading the register DMA0IDT will return the instruction word at location DMA0IPT. After a write or read operation on DMA0IDT, the DMA0IPT register is automatically incremented to the next Instruction Buffer location.

## 6.2. DMA0 Instruction Format

DMA instructions can request single-ended data from both ADC0 and ADC1, as well as the differential combination of the two ADC inputs. The instruction format is identical to the DMA0IDT register, shown in Figure 6.7. Depending on which bits are set to ‘1’ in the instruction word, either 2 or 4 bytes of data will be written to XRAM for each DMA instruction cycle (excluding End-Of-Operation instructions). Table 6.1 details all of the valid DMA instructions. Instructions not listed in the table are not valid DMA instructions, and should not be used. Note that the ADCs can be independently controlled by the microcontroller when their outputs are not requested by the DMA.

**Table 6.1. DMA0 Instruction Set**

| Instruction Word | Description | First Data Written to XRAM (2 bytes) | Second Data Written to XRAM (2 bytes) |
|---|---|---|---|
| 00000000b | End-Of-Operation | none | none |
| 10000000b | End-Of-Operation with Continuous Conversion | none | none |
| x0010000b | Retrieve ADC0 Data | ADC0H:ADC0L | none |
| x0100000b | Retrieve ADC1 Data | ADC1H:ADC1L | none |
| x0110000b | Retrieve ADC0 and ADC1 Data | ADC0H:ADC0L | ADC1H:ADC1L |
| x10x0000b | Retrieve Differential Data | ADC0H:ADC0L (differential result from both ADCs) | none |
| x11x0000b | Retrieve Differential and ADC1 Data | ADC0H:ADC0L (differential result from both ADCs) | ADC1H:ADC1L |

## 6.3. XRAM Addressing and Setup

The DMA Interface can be configured to access either on-chip or off-chip XRAM. Any writes to on-chip XRAM by the DMA Control Logic occur when the processor core is not accessing the on-chip XRAM. This ensures that the DMA will not interfere with processor instruction timing.

Off-chip XRAM access (only available on the C8051F060/2/4/6) is controlled by the DMA0HLT bit in DMA0CF (DMA Configuration Register, Figure 6.5). The DMA will have full access to off-chip XRAM when this bit is ‘0’, and the processor core will have full access to off-chip XRAM when this bit is ‘1’. The DMA0HLT bit should be controlled in software when both the processor core and the DMA Interface require access to off-chip XRAM data space. Before setting DMA0HLT to ‘1’, the software should check the DMA0XBY bit to ensure that the DMA is not currently accessing off-chip XRAM. The processor core cannot access off-chip XRAM while DMA0HLT is ‘0’. The processor will continue as though it was able to perform the desired memory access, but the data will not be written to or read from off-chip XRAM. When the processor core is finished accessing off-chip XRAM, DMA0HLT should be set back to ‘0’in software to return control to the DMA Interface. The DMA Control Logic will wait until DMA0HLT is ‘0’ before writing data to off-chip XRAM. If new data becomes available to the DMA Interface before the previous data has been written, an overflow condition will occur, and the new data word may be lost.

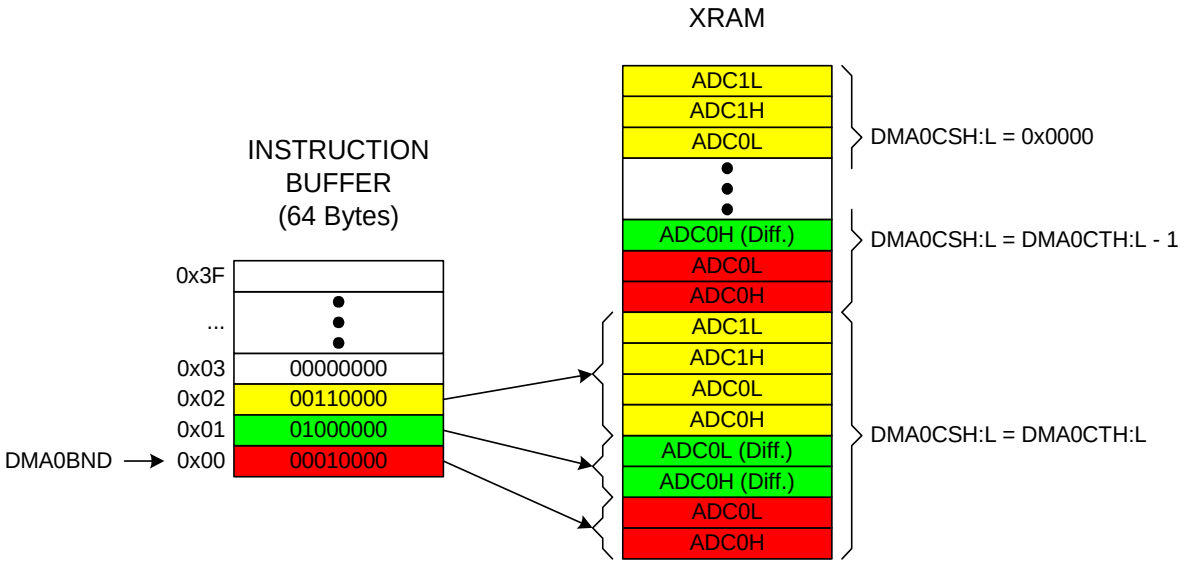
The Data Address Pointer Registers (DMA0DSH and DMA0DSL) contain the 16-bit XRAM address location where the DMA interface will write data. When the DMA is initially enabled, the DMA Data Address

Pointer Registers are initialized to the values contained in the DMA Data Address Beginning Registers (DMA0DAH and DMA0DAL). The Data Address Pointer Registers are automatically incremented by 2 or 4 after each data write by the DMA interface.

## 6.4. Instruction Execution in Mode 0

When the DMA interface begins an operation cycle, the DMA Instruction Status Register (DMA0ISW, Figure 6.9) is loaded with the address contained in the DMA Instruction Boundary Register (DMA0BND, Figure 6.8). The instruction is fetched from the Instruction Buffer, and the DMA Control Logic waits for data from the appropriate ADC(s). The DMA will execute each instruction once, and then increment DMA0ISW to the next instruction address. When the current DMA instruction is an End of Operation instruction, the Instruction Status Register is reset to the Instruction Boundary Register. If the Continuous Conversion bit (bit 7, CCNV) in the End of Operation instruction word is set to '1', the Repeat Counter is ignored, and the DMA will continue to execute instructions indefinitely. When CCNV is set to '0', the Repeat Counter (registers DMA0CSH and DMA0CSL) is decremented, and the DMA will continue to execute instructions until the Repeat Counter reaches 0x0000. The Repeat Counter is initialized with the Repeat Counter Limit value (registers DMA0CTH and DMA0CTL) at the beginning of the DMA operation. An example of Mode 0 operation is shown in Figure 6.2.

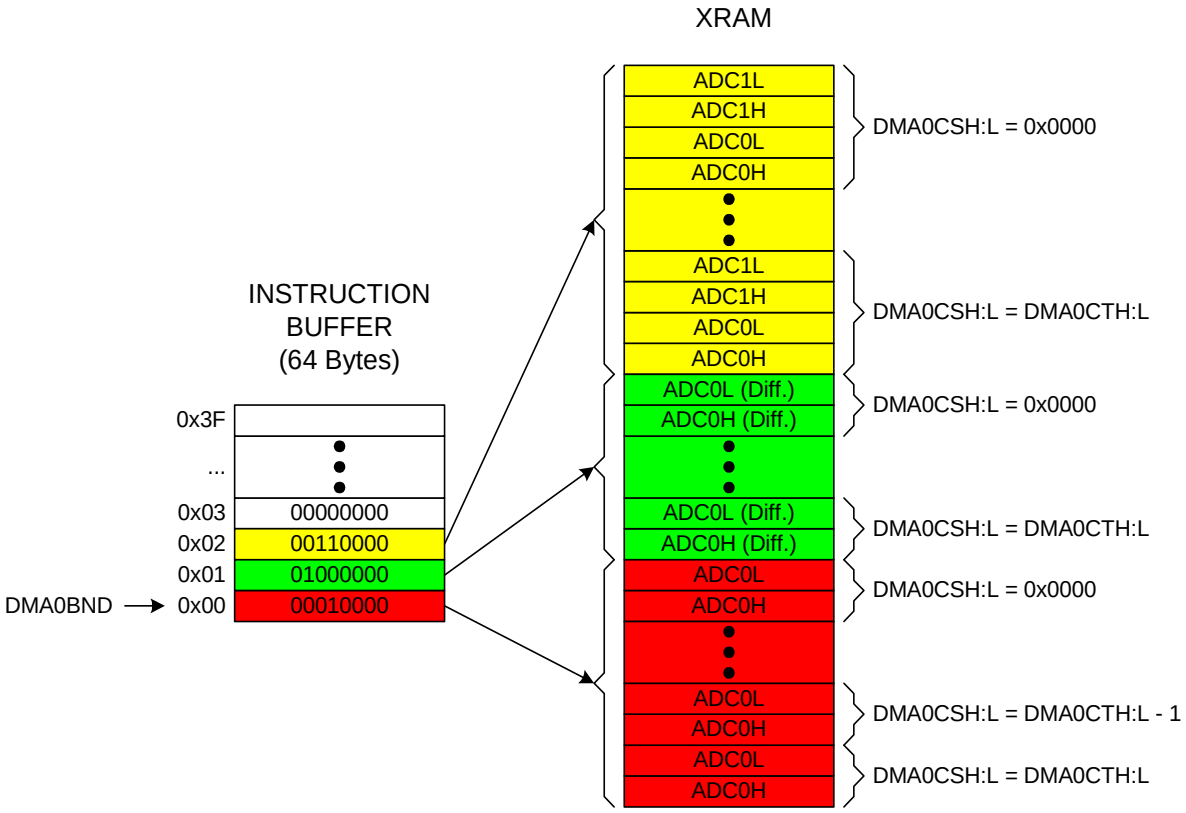
**Figure 6.2. DMA Mode 0 Operation**



## 6.5. Instruction Execution in Mode 1

When the DMA interface begins an operation cycle, the DMA Instruction Status Register (DMA0ISW, Figure 6.9) is loaded with the address contained within the DMA Instruction Boundary Register (DMA0BND, Figure 6.8). The instruction is fetched from the Instruction Buffer, and the DMA Control Logic waits for data from the appropriate ADC(s). At the end of an instruction, the Repeat Counter (Registers DMA0CSH and DMA0CSL) is decremented, and the instruction will be repeated until the Repeat Counter reaches 0x0000. The Repeat Counter is then reset to the Repeat Counter Limit value (Registers DMA0CTH and DMA0CTL), and the DMA will increment DMA0ISW to the next instruction address. When the current DMA instruction is an End of Operation instruction, the Instruction Status Register is reset to the Instruction Boundary Register. If the Continuous Conversion bit (bit 7, CCNV) in the End of Operation instruction word is set to '1', the DMA will continue to execute instructions. When CCNV is set to '0', the DMA will stop executing instructions at this point. An example of Mode 1 operation is shown in Figure 6.3.

**Figure 6.3. DMA Mode 1 Operation**



## 6.6. Interrupt Sources

The DMA contains multiple interrupt sources. Some of these can be individually enabled to generate interrupts as necessary. The DMA Control Register (DMA0CN, Figure 6.4) and DMA Configuration Register (DMA0CF, Figure 6.5) contain the enable bits and flags for the DMA interrupt sources. When an interrupt is enabled and the interrupt condition occurs, a DMA interrupt will be generated (EIE2.7 is set to '1').

The DMA flags that can generate a DMA0 interrupt are:

1. DMA Operations Complete (DMA0CN.6, DMA0INT) occurs when all DMA operations have been completed, and the DMA interface is idle.
2. ADC1 Data Overflow Error (DMA0CN.4, DMA0DE1) occurs when the DMA interface cannot access XRAM for two conversion cycles of ADC1. This flag indicates that at least one conversion result from ADC1 has been discarded.
3. ADC0 Data Overflow Error (DMA0CN.3, DMA0DE0) occurs when the DMA interface cannot access XRAM for two conversion cycles of ADC0. This flag indicates that at least one conversion result from ADC0 has been discarded.
4. ADC1 Data Overflow Warning (DMA0CN.1, DMA0DO1) occurs when data from ADC0 becomes available and the DMA has not yet written the previous results to XRAM. This interrupt source can be enabled and disabled with the Data Overflow Warning Enable bit (DMA0CN.2, DMA0DOE).
5. ADC0 Data Overflow Warning (DMA0CN.0, DMA0DO0) occurs when data from ADC1 becomes available and the DMA has not yet written the previous results to XRAM. This interrupt source can be enabled and disabled with the Data Overflow Warning Enable bit (DMA0CN.2, DMA0DOE).
6. Repeat Counter Overflow (DMA0CF.2, DMA0CI) occurs when the Repeat Counter reaches the Repeat Counter Limit. This interrupt source can be enabled and disabled with the Repeat Counter Overflow Interrupt Enable bit (DMA0CF.3, DMA0CIE).
7. End Of Operation (DMA0CF.0, DMA0EO) occurs when an End Of Operation instruction is reached in the Instruction Buffer. This interrupt source can be enabled and disabled with the End Of Operation Interrupt Enable bit (DMA0CF.1, DMA0EOE).

## 6.7. Data Buffer Overflow Warnings and Errors

The data paths from the ADCs to XRAM are double-buffered when using the DMA interface. When a conversion is completed by the ADC, it first enters the ADCs data register. If the DMA's data buffer is empty, the conversion results will immediately be written into the DMA's internal data buffer for that ADC. Data in the DMA's internal data buffer is written to XRAM at the first available opportunity (see Section "6.3. XRAM Addressing and Setup" on page 76). Conversion results from the ADC's data registers are not copied into the DMA's data buffer until data in the buffer has been written to XRAM. When a conversion is completed and the DMA's data buffer is not empty, an overflow warning flag is generated. If a second conversion data word becomes available before the DMA's data buffer is written to XRAM, the data in the ADC's data registers is over-written with the new data word, and a data overflow error flag is generated.

**Figure 6.4. DMA0CN: DMA0 Control Register**

SFR Page: 3
SFR Address: 0xD8 (bit addressable)

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| DMA0EN | DMA0INT | DMA0MD | DMA0DE1 | DMA0DE0 | DMA0DOE | DMA0DO1 | DMA0DO0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit 7: DMA0EN: DMA0 Enable.
Write:
0: Stop DMA0 Operations.
1: Begin DMA0 Operations.
Read:
0: DMA0 is Idle.
1: DMA0 Operation is in Progress.

Bit 6: DMA0INT: DMA0 Operations Complete Flag.
0: DMA0 has not completed all operations.
1: DMA0 operations are complete. This bit must be cleared by software.

Bit 5: DMA0MD: DMA0 Mode Select.
0: DMA0 will operate in Mode 0.
1: DMA0 will operate in Mode 1.

Bit 4: DMA0DE1: ADC1 Data Overflow Error Flag.
0: ADC1 Data Overflow has not occured.
1: ADC1 Data Overflow has occured, and data from ADC1 has been lost. This bit must be cleared by software.

Bit 3: DMA0DE0: ADC0 Data Overflow Error Flag.
0: ADC0 Data Overflow has not occured.
1: ADC0 Data Overflow has occured, and data from ADC0 has been lost. This bit must be cleared by software.

Bit 2: DMA0DOE: Data Overflow Warning Interrupt Enable.
0: Disable Data Overflow Warning interrupts.
1: Enable Data Overflow Warning interrups.

Bit 1: DMA0DO1: ADC1 Data Overflow Warning Flag.
0: No ADC1 Data Buffer Warnings have been issued.
1: ADC1 Data Buffer is full, and the DMA has not written previous data to XRAM. This bit must be cleared by software.

Bit 0: DMA0DO0: ADC0 Data Overflow Warning Flag.
0: No ADC0 Data Buffer Warnings have been issued.
1: ADC0 Data Buffer is full, and the DMA has not written previous data to XRAM. This bit must be cleared by software.

**Figure 6.5. DMA0CF: DMA0 Configuration Register**

SFR Page: 3
SFR Address: 0xF8 (bit addressable)

| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| DMA0HLT | DMA0XBY | - | - | DMA0CIE | DMA0CI | DMA0EOE | DMA0EO | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit 7: DMA0HLT: Halt DMA0 Off-Chip XRAM Access (C8051F060/2/4/6 Only).
0: DMA0 has complete access to off-chip XRAM.
1: Processor core has complete access to off-chip XRAM. DMA0 will wait until this bit is ‘0’ before writing to off-chip XRAM locations.

Bit 6: DMA0XBY: Off-chip XRAM Busy Flag (C8051F060/2/4/6 Only).
0: DMA0 is not accessing off-chip XRAM.
1: DMA0 is accessing off-chip XRAM.

Bits 5-4: RESERVED. Write to 00b.

Bit 3: DMA0CIE: Repeat Counter Overflow Interrupt Enable.
0: Disable Repeat Counter Overflow Interrupt.
1: Enable Repeat Counter Overflow Interrupt.

Bit 2: DMA0CI: Repeat Counter Overflow Flag.
0: Repeat Counter Overflow has not occured.
1: Repeat Counter Overflow has occured. This bit must be cleared by software.

Bit 1: DMA0EOE: End-Of-Operation Interrupt Enable.
0: Disable End-Of-Operation Interrupt.
1: Enable End-Of-Operation Interrupt.

Bit 0: DMA0EO: End-Of-Operation Flag.
0: End-Of-Operation Instruction has not been received.
1: End-Of-Operation Instruction has been received. This bit must be cleared by software.

**Figure 6.6. DMA0IPT: DMA0 Instruction Write Address Register**

SFR Page: 3
SFR Address: 0xDD

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

Bits 7-6: Unused.
Bits 5-0: DMA0 instruction address to write (or read). When DMA0IDT is written or read, this register will be incremented to point to the next instruction address.

**Figure 6.7. DMA0IDT: DMA0 Instruction Write Data Register**

SFR Page: 3
SFR Address: 0xDE

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value† |
|---|---|---|---|---|---|---|---|---|
| CCNV | DIFFSEL | ADC1EN | ADC0EN | - | - | - | - | xxxxxxxx |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit 7: CCNV: Continuous Conversion.
0: Disable Continuous Conversion.
1: Enable Continuous Conversion. Repeat Counter value is ignored, and conversions will continue.
Bit 6: DIFFSEL: Wait for data in differential mode.
0: Differential Data will not be collected.
1: Wait for differential data, and store to XRAM.
Bit 5: ADC1EN: Wait for data from ADC1.
0: ADC1 Data will not be collected.
1: Wait for ADC1 data, and store to XRAM.
Bit 4: ADC0EN: Wait for data from ADC0.
0: ADC0 Data will not be collected.
1: Wait for ADC0 data, and store to XRAM. If DIFFSEL is also ‘1’, only the differential data will be stored.
Bits 3-0: RESERVED. Write to 0000b.

For more details on DMA instruction words, see Section “6.2. DMA0 Instruction Format” on page 76.

† This register points to a dedicated RAM location and its reset value is indeterminate.

**Figure 6.8. DMA0BND: DMA0 Instruction Boundary Register**

SFR Page: 3
SFR Address: 0xFD

| 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 | |

Bits 7-6: Unused.
Bits 5-0: DMA0 instruction address to begin with when executing DMA instructions.

**Figure 6.9. DMA0ISW: DMA0 Instruction Status Register**

SFR Page: 3
SFR Address: 0xFE

| 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 | |

Bits 7-6: Unused.
Bits 5-0: Contains the address of the current DMA0 Instruction to be executed.

**Figure 6.10. DMA0DAH: DMA0 Data Address Beginning MSB Register**

SFR Page: 3
SFR Address: 0xDA

| 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 | |

Bits 7-0: DMA0 Address Beginning High-Order Bits.

**Figure 6.11. DMA0DAL: DMA0 Data Address Beginning LSB Register**

SFR Page: 3
SFR Address: 0xD9

| 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 | |

Bits 7-0: DMA0 Address Beginning Low-Order Bits.

**Figure 6.12. DMA0DSH: DMA0 Data Address Pointer MSB Register**

SFR Page: 3
SFR Address: 0xDC

| 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 | |

Bits 7-0: DMA0 Address Pointer High-Order Bits.

**Figure 6.13. DMA0DSL: DMA0 Data Address Pointer LSB Register**

SFR Page: 3
SFR Address: 0xDB

| 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 | |

Bits 7-0: DMA0 Address Pointer Low-Order Bits.

**Figure 6.14. DMA0CTH: DMA0 Repeat Counter Limit MSB Register**

SFR Page: 3
SFR Address: 0xFA

| 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 | |

Bits 7-0: DMA0 Repeat Counter Limit High-Order Bits.

**Figure 6.15. DMA0CTL: DMA0 Repeat Counter Limit LSB Register**

SFR Page: 3
SFR Address: 0xF9

| 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 | |

Bits 7-0: DMA0 Repeat Counter Limit Low-Order Bits.

**Figure 6.16. DMA0CSH: DMA0 Repeat Counter MSB Register**

SFR Page: 3
SFR Address: 0xFC

| 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 | |

Bits 7-0: DMA0 Repeat Counter High-Order Bits.

**Figure 6.17. DMA0CSL: DMA0 Repeat Counter LSB Register**

SFR Page: 3
SFR Address: 0xFB

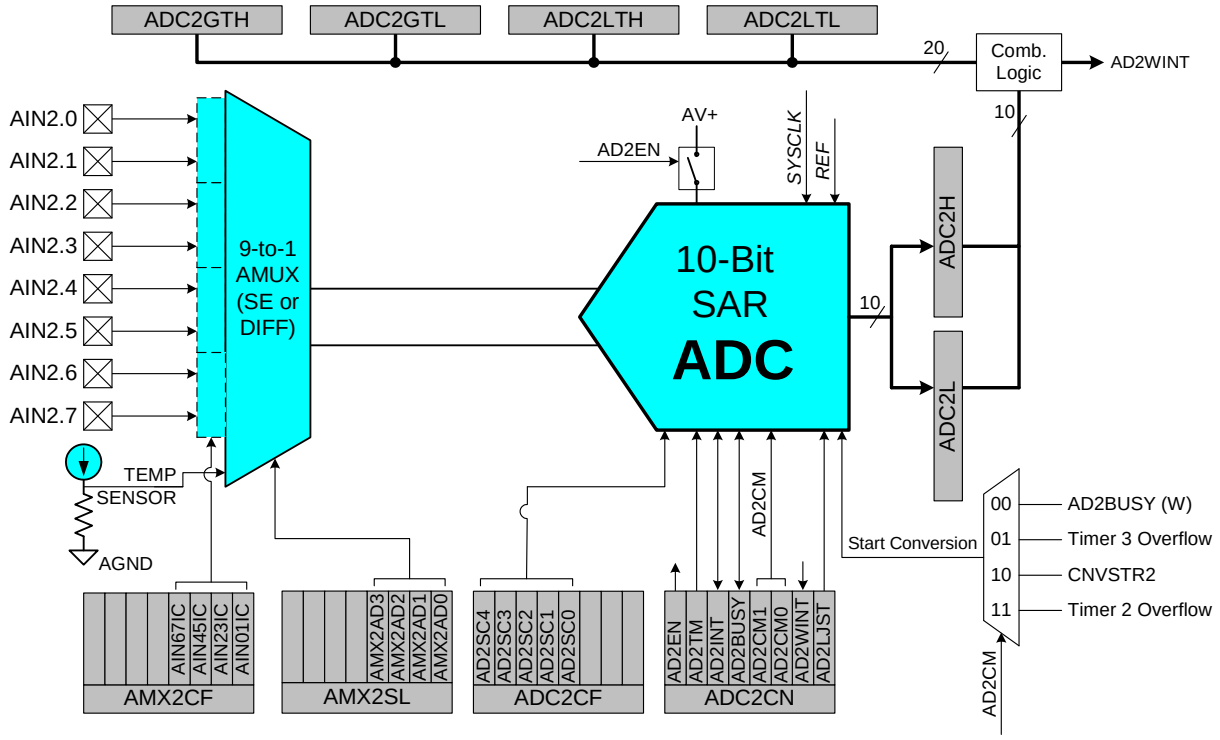
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7-0: DMA0 Repeat Counter Low-Order Bits.

# 7. 10-Bit ADC (ADC2, C8051F060/1/2/3)

The ADC2 subsystem for the C8051F060/1/2/3 consists of an analog multiplexer (referred to as AMUX2), and a 200 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector (see block diagram in Figure 7.1). The AMUX2, data conversion modes, and window detector can all be configured from within software via the Special Function Registers shown in Figure 7.1. ADC2 operates in both Single-ended and Differential modes, and may be configured to measure any of the pins on Port 1, or the Temperature Sensor output. The ADC2 subsystem 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.

**Figure 7.1. ADC2 Functional Block Diagram**



## 7.1. Analog Multiplexer

The analog multiplexer (AMUX2) selects the inputs to the ADC, allowing any of the pins on Port 1 to be measured in single-ended mode, or as a differential pair. Additionally, the on-chip temperature sensor may be selected as a single-ended input. The ADC2 input channels are configured and selected in the AMX-2CF and AMX2SL registers as described in Figure 7.5 and Figure 7.6, respectively. In Single-ended Mode, the selected pin is measured with respect to AGND. In Differential Mode, the selected differential pair is measured with respect to one another. The polarity of the differential measurement depends on the setting of the AMX2AD3-0 bits in the AMX2SL register. For example, if pins AIN2.0 and AIN2.1 are configured for differential measurement (AIN01IC = 1), and AMX2AD3-0 = 0000b, the ADC will measure the voltage (AIN2.0 - AIN2.1). If AMX2AD3-0 is changed to 0001b, the ADC will measure the same voltage, with opposite polarity (AIN2.1 - AIN2.0).

The conversion code format differs between Single-ended and Differential modes. The registers ADC2H and ADC2L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD2LJST bit (ADC2CN.0). When in Single-ended Mode, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from ‘0’ to VREF * 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC2H and ADC2L registers are set to ‘0’.

| Input Voltage | Right-Justified ADC2H:ADC2L (AD2LJST = 0) | Left-Justified ADC2H:ADC2L (AD2LJST = 1) |
|---|---|---|
| VREF * 1023/1024 | 0x03FF | 0xFFC0 |
| VREF * 512/1024 | 0x0200 | 0x8000 |
| VREF * 256/1024 | 0x0100 | 0x4000 |
| 0 | 0x0000 | 0x0000 |

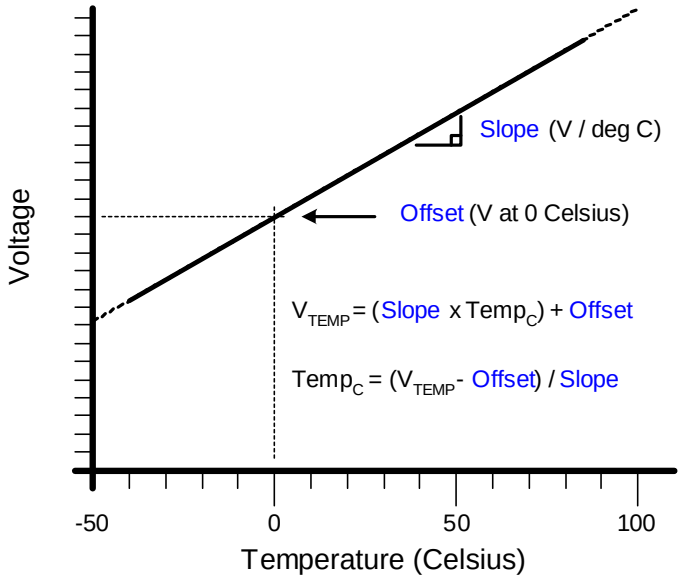
When in Differential Mode, conversion codes are represented as 10-bit signed 2’s complement numbers. Inputs are measured from -VREF to VREF * 511/512. Example codes are shown below for both right-justified and left-justified data. For right-justified data, the unused MSBs of ADC2H are a sign-extension of the data word. For left-justified data, the unused LSBs in the ADC2L register are set to ‘0’.

| Input Voltage | Right-Justified ADC2H:ADC2L (AD2LJST = 0) | Left-Justified ADC2H:ADC2L (AD2LJST = 1) |
|---|---|---|
| VREF * 511/512 | 0x01FF | 0x7FC0 |
| VREF * 256/512 | 0x0100 | 0x4000 |
| 0 | 0x0000 | 0x0000 |
| -VREF * 256/512 | 0xFF00 | 0xC000 |
| - VREF | 0xFE00 | 0x8000 |

**Important Note About ADC2 Input Configuration:** Port 1 pins selected as ADC2 inputs should be configured as analog inputs. To configure a Port 1 pin for analog input, set to ‘1’ the corresponding bit in register P1MDIN. Port 1 pins used as ADC2 inputs will be skipped by the crossbar for peripheral assignments. See Section “18. Port Input/Output” on page 203 for more Port I/O configuration details.

The Temperature Sensor transfer function is shown in Figure 7.2 on Page 89. The output voltage ($V_{TEMP}$) is a single-ended input to ADC2 when the Temperature Sensor is selected by bits AMX2AD3-0 in register AMX2SL. Typical values for the Slope and Offset parameters can be found in Table 7.1.

**Figure 7.2. Temperature Sensor Transfer Function**




## 7.2. Modes of Operation

ADC2 has a maximum conversion speed of 200 ksps. The ADC2 conversion clock is a divided version of the system clock, determined by the AD2SC bits in the ADC2CF register (system clock divided by (AD2SC + 1) for 0 ≤ AD2SC ≤ 31). The ADC2 conversion clock should be no more than 3 MHz.

### 7.2.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC2 Start of Conversion Mode bits (AD2CM1-0) in register ADC2CN. Conversions may be initiated by one of the following:

1. Writing a '1' to the AD2BUSY bit of register ADC2CN
2. A Timer 3 overflow (i.e. timed continuous conversions)
3. A rising edge on the CNVSTR2 input signal (Assigned by the crossbar)
4. A Timer 2 overflow

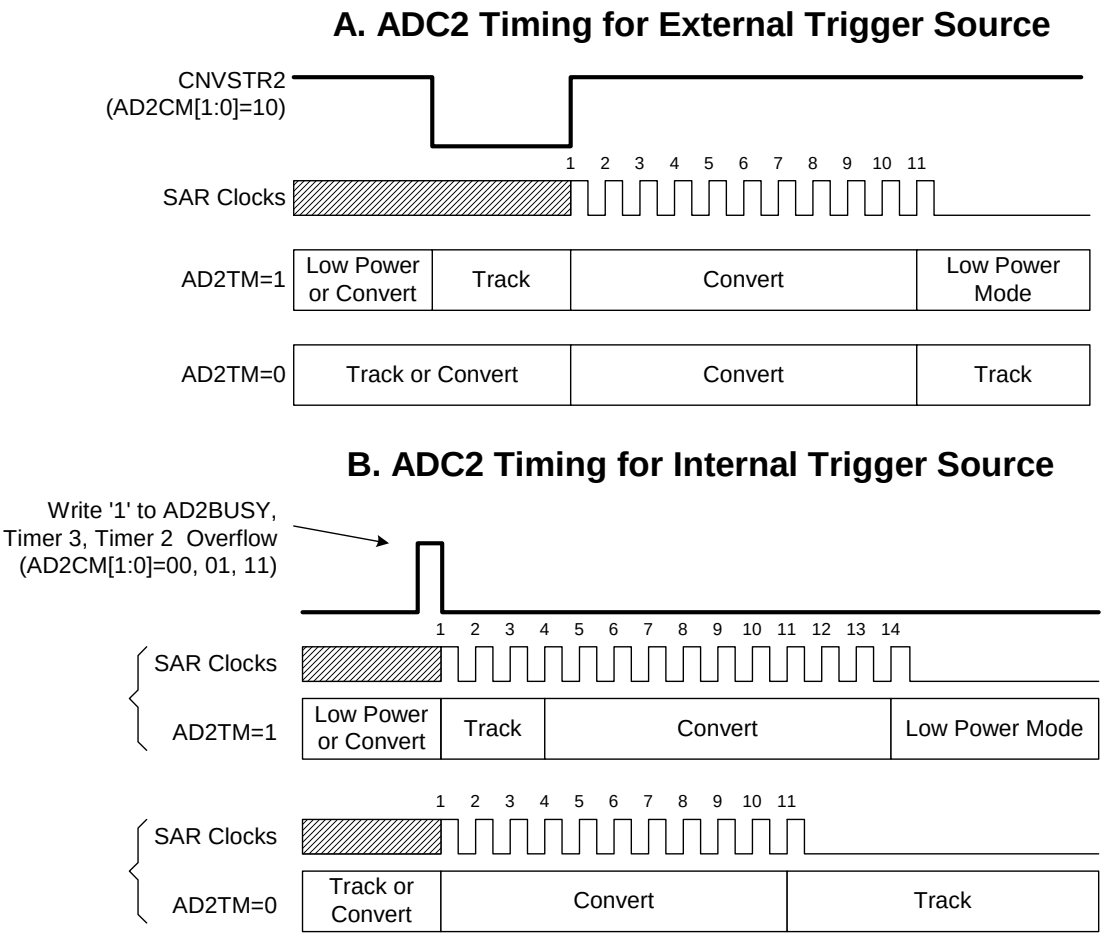
When CNVSTR2 is used as a conversion start source, it must be enabled in the crossbar, and the corresponding pin must be set to open-drain, high-impedance mode (see Section "18. Port Input/Output" on page 203 for more details on Port I/O configuration).

Writing a '1' to AD2BUSY provides software control of ADC2 whereby conversions are performed "on-demand". During conversion, the AD2BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD2BUSY triggers an interrupt (when enabled) and sets the ADC2 interrupt flag (AD2INT). Note: When polling for ADC conversion completions, the ADC2 interrupt flag (AD2INT) should be used. Converted data is available in the ADC2 data registers, ADC2H and ADC2L, when bit AD2INT is logic 1. Note that when Timer 2 or Timer 3 overflows are used as the conversion source, low byte overflows are used if the timer is in 8-bit mode; and high byte overflows are used if the timer is in 16-bit mode. See Section "24. Timers" on page 287 for timer 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 7.3). Tracking can also be disabled (shutdown) when the device is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX settings are frequently changed, due to the settling time requirements described in Section "7.2.3. Settling Time Requirements" on page 91.

**Figure 7.3. 10-Bit ADC 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 AMUX2 resistance, the ADC2 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the minimum tracking time requirements.

Figure 7.4 shows the equivalent ADC2 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required ADC2 settling time for a given settling accuracy (SA) may be approximated by Equation 7.1. When measuring the Temperature Sensor output, $R_{TOTAL}$ reduces to $R_{MUX}$. See Table 7.1 for ADC2 minimum settling time requirements.

**Equation 7.1. ADC2 Settling Time Requirements**

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

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 AMUX2 resistance and any external source resistance.
*n* is the ADC resolution in bits (10).

**Figure 7.4. ADC2 Equivalent Input Circuits**

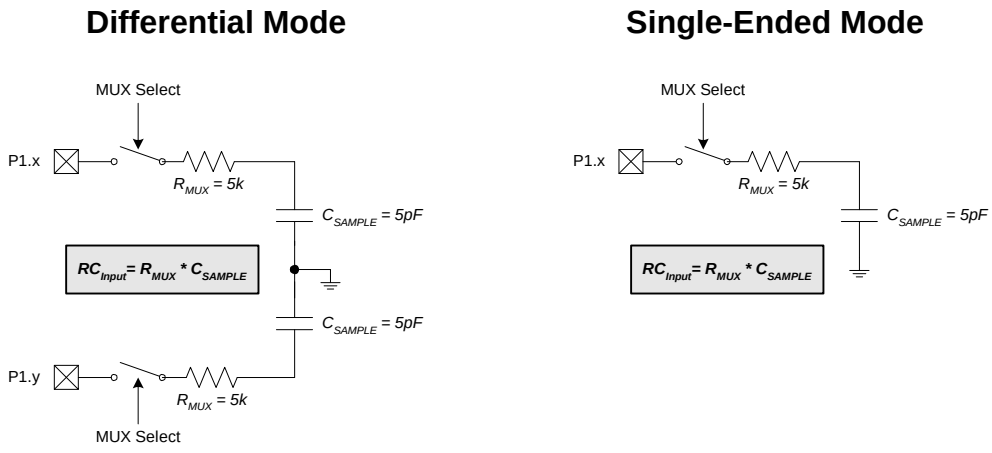


**Figure 7.5. AMX2CF: AMUX2 Configuration Register**

SFR Page: 2
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 | |

Bits 7-4: UNUSED. Read = 0000b; Write = don't care.
Bit 3: AIN67IC: 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 a differential input pair.
Bit 2: AIN45IC: 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 a differential input pair.
Bit 1: AIN23IC: 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 a differential input pair.
Bit 0: AIN01IC: 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 a differential input pair.

NOTE: The ADC2 Data Word is in the 2's complement format for channels configured as differential. The polarity of a differential measurement is determined by the AMX2SL setting. See Figure 7.5 for more details on multiplexer channel selection.

**Figure 7.6. AMX2SL: AMUX2 Channel Select Register**

SFR Page: 2
SFR Address: 0xBB

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | AMX2AD3 | AMX2AD2 | AMX2AD1 | AMX2AD0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bits 7-4: UNUSED. Read = 0000b; Write = don't care.
Bits 3-0: AMX2AD3-0: AMX2 Address Bits.
0000-1111b: ADC input multiplexer channel selected per chart below.

| AMX2AD3-0 | Single-Ended Measurement | | AMX2AD3-0 | Differential Measurement | |
|---|---|---|---|---|---|
| 0000 | AIN2.0 | AIN01IC = 0 | 0000 | +(AIN2.0) -(AIN2.1) | AIN01IC = 1 |
| 0001 | AIN2.1 | | 0001 | +(AIN2.1) -(AIN2.0) | |
| 0010 | AIN2.2 | AIN23IC = 0 | 0010 | +(AIN2.2) -(AIN2.3) | AIN23IC = 1 |
| 0011 | AIN2.3 | | 0011 | +(AIN2.3) -(AIN2.2) | |
| 0100 | AIN2.4 | AIN45IC = 0 | 0100 | +(AIN2.4) -(AIN2.5) | AIN45IC = 1 |
| 0101 | AIN2.5 | | 0101 | +(AIN2.5) -(AIN2.4) | |
| 0110 | AIN2.6 | AIN67IC = 0 | 0110 | +(AIN2.6) -(AIN2.7) | AIN67IC = 1 |
| 0111 | AIN2.7 | | 0111 | +(AIN2.7) -(AIN2.6) | |
| 1xxx | Temperature Sensor | | 1xxx | - | |

**Figure 7.7. ADC2CF: ADC2 Configuration Register**

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 | - | - | - | 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 *ADSC* refers to the 5-bit value held in bits AD2SC4-AD2SC0. SAR Conversion clock requirements are given in Table 7.1.

$$ADSC = \frac{SYSCLK}{CLK_{SAR}} - 1$$

Bits2-0: UNUSED. Read = 000b; Write = don't care.

**Figure 7.8. ADC2H: ADC2 Data Word MSB Register**

SFR Page: 2
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: ADC2 Data Word High-Order Bits.
For AD2LJST = 0: Bits 7-2 are the sign extension of Bit 1. Bits 1-0 are the upper 2 bits of the 10-bit ADC2 Data Word.
For AD2LJST = 1: Bits 7-0 are the most-significant bits of the 10-bit ADC2 Data Word.

**Figure 7.9. ADC2L: ADC2 Data Word LSB Register**

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 Low-Order Bits.
For AD2LJST = 0: Bits 7-0 are the lower 8 bits of the 10-bit Data Word.
For AD2LJST = 1: Bits 7-6 are the lower 2 bits of the 10-bit Data Word. Bits 5-0 will always read ‘0’.

**Figure 7.10. ADC2CN: ADC2 Control Register**

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 | AD2CM1 | AD2CM0 | AD2WINT | AD2LJST | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

Bit 7: 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 progress.
1: Low-power Track Mode: Tracking Defined by AD2CM2-0 bits (see below).

Bit5: AD2INT: ADC2 Conversion Complete Interrupt Flag.
0: ADC2 has not completed a data conversion since the last time AD2INT was cleared.
1: ADC2 has completed a data conversion.

Bit 4: 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

Bits 3-2: AD2CM1-0: ADC2 Start of Conversion Mode Select.
When AD2TM = 0:
00: ADC2 conversion initiated on every write of ‘1’ to AD2BUSY.
01: ADC2 conversion initiated on overflow of Timer 3.
10: ADC2 conversion initiated on rising edge of external CNVSTR2 pin.
11: ADC2 conversion initiated on overflow of Timer 2.
When AD2TM = 1:
00: Tracking initiated on write of ‘1’ to AD2BUSY and lasts 3 SAR clocks, followed by conversion.
01: Tracking initiated on overflow of Timer 3 and lasts 3 SAR clocks, followed by conversion.
10: ADC2 tracks only when CNVSTR2 input is logic low; conversion starts on rising CNVSTR2 edge.
11: Tracking initiated on overflow of Timer 2 and lasts 3 SAR clocks, followed by conversion.

Bit 1: AD2WINT: ADC2 Window Compare Interrupt Flag.
0: ADC2 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC2 Window Comparison Data match has occurred.

Bit 0: AD2LJST: ADC2 Left Justify Select.
0: Data in ADC2H:ADC2L registers are right-justified.
1: Data in ADC2H:ADC2L registers are left-justified.

## 7.3. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC2 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD2WINT in register ADC2CN) can also be used in polled mode. The ADC2 Greater-Than (ADC2GTH, ADC2GTL) and Less-Than (ADC2LTH, ADC2LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC2 Less-Than and ADC2 Greater-Than registers.

**Figure 7.11. ADC2GTH: ADC2 Greater-Than Data High Byte Register**

SFR Page: 2
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 ADC2 Greater-Than Data Word.

**Figure 7.12. ADC2GTL: ADC2 Greater-Than Data Low Byte Register**

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: Low byte of ADC2 Greater-Than Data Word.

**Figure 7.13. ADC2LTH: ADC2 Less-Than Data High Byte Register**

SFR Page: 2
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 ADC2 Less-Than Data Word.

**Figure 7.14. ADC2LTL: ADC2 Less-Than Data Low Byte Register**

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: Low byte of ADC2 Less-Than Data Word.

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

Figure 7.15 shows two example window comparisons for right-justified, single-ended data, with ADC2LTH:ADC2LTL = 0x0080 (128d) and ADC2GTH:ADC2GTL = 0x0040 (64d). In single-ended mode, the input voltage can range from '0' to VREF * (1023/1024) with respect to AGND, and is represented by a 10-bit unsigned integer value. In the left example, an AD2WINT interrupt will be generated if the ADC2 conversion word (ADC2H:ADC2L) is within the range defined by ADC2GTH:ADC2GTL and ADC2LTH:ADC2LTL (if 0x0040 < ADC2H:ADC2L < 0x0080). In the right example, and AD2WINT interrupt will be generated if the ADC2 conversion word is outside of the range defined by the ADC2GT and ADC2LT registers (if ADC2H:ADC2L < 0x0040 or ADC2H:ADC2L > 0x0080). Figure 7.16 shows an example using left-justified data with the same comparison values.

**Figure 7.15. ADC Window Compare Example: Right-Justified Single-Ended Data**

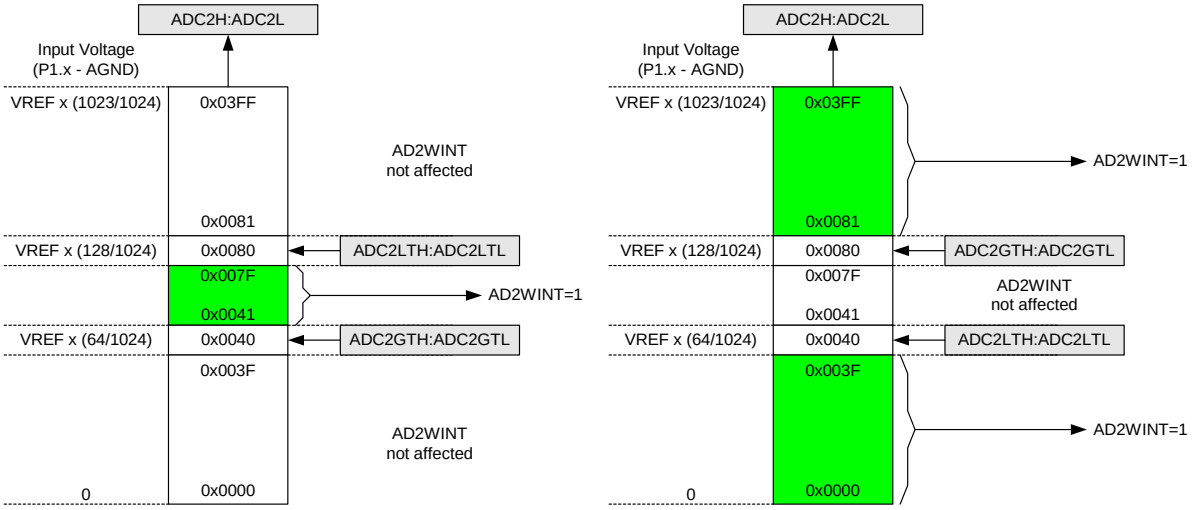
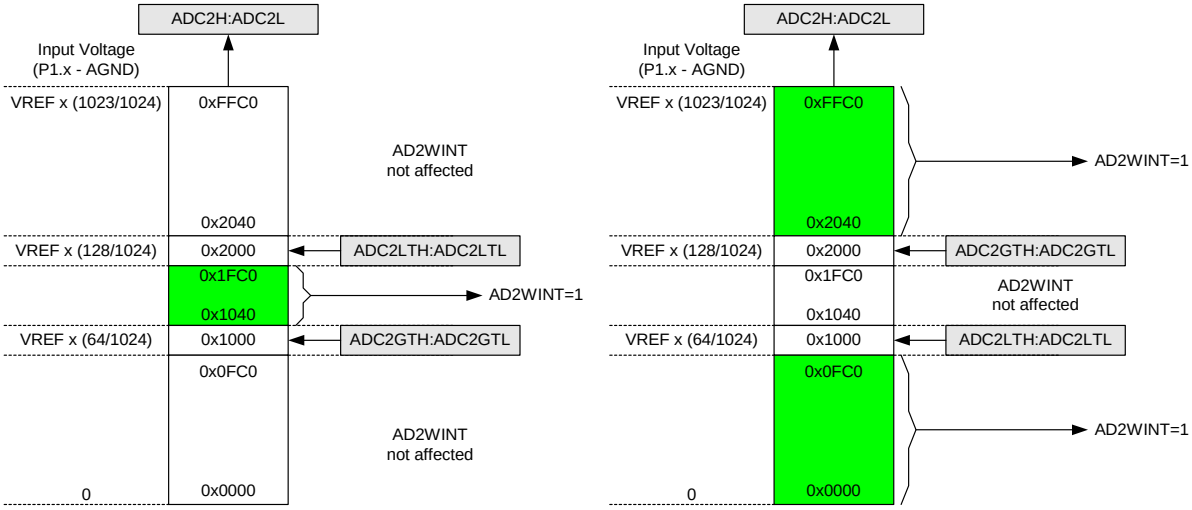



**Figure 7.16. ADC Window Compare Example: Left-Justified Single-Ended Data**



### 7.3.2. Window Detector In Differential Mode

Figure 7.17 shows two example window comparisons for right-justified, differential data, with ADC2LTH:ADC2LTL = 0x0040 (+64d) and ADC2GTH:ADC2GTH = 0xFFFF (-1d). In differential mode, the measurable voltage between the input pins is between -VREF and VREF*(511/512). Output codes are represented as 10-bit 2's complement signed integers. In the left example, an AD2WINT interrupt will be generated if the ADC2 conversion word (ADC2H:ADC2L) is within the range defined by ADC2GTH:ADC2GTL and ADC2LTH:ADC2LTL (if 0xFFFF (-1d) < ADC2H:ADC2L < 0x0040 (64d)). In the right example, an AD2WINT interrupt will be generated if the ADC2 conversion word is outside of the range defined by the ADC2GT and ADC2LT registers (if ADC2H:ADC2L < 0xFFFF (-1d) or ADC2H:ADC2L > 0x0040 (+64d)). Figure 7.18 shows an example using left-justified data with the same comparison values.

**Figure 7.17. ADC Window Compare Example: Right-Justified Differential Data**

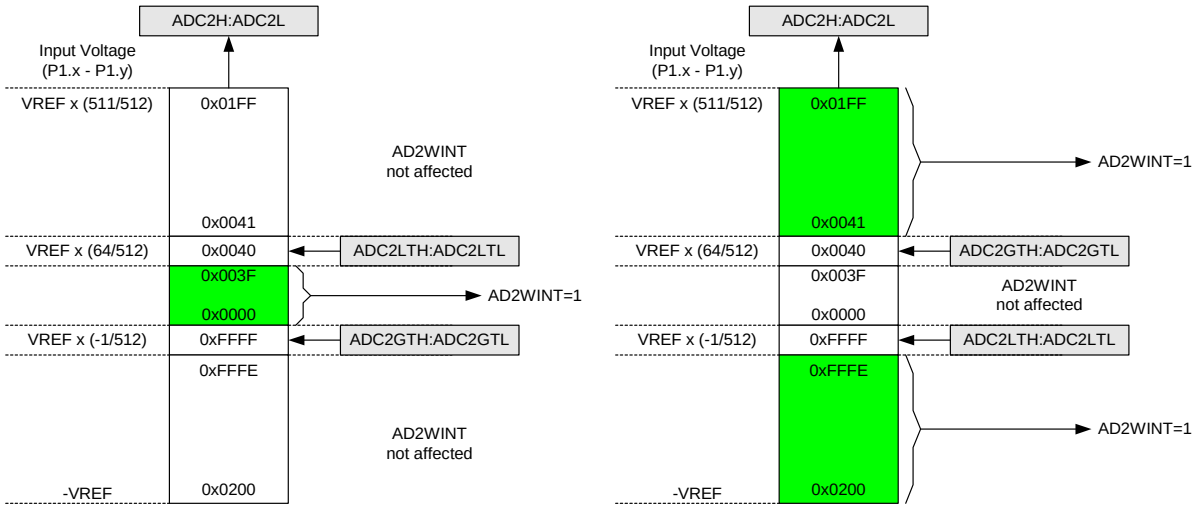



**Figure 7.18. ADC Window Compare Example: Left-Justified Differential Data**

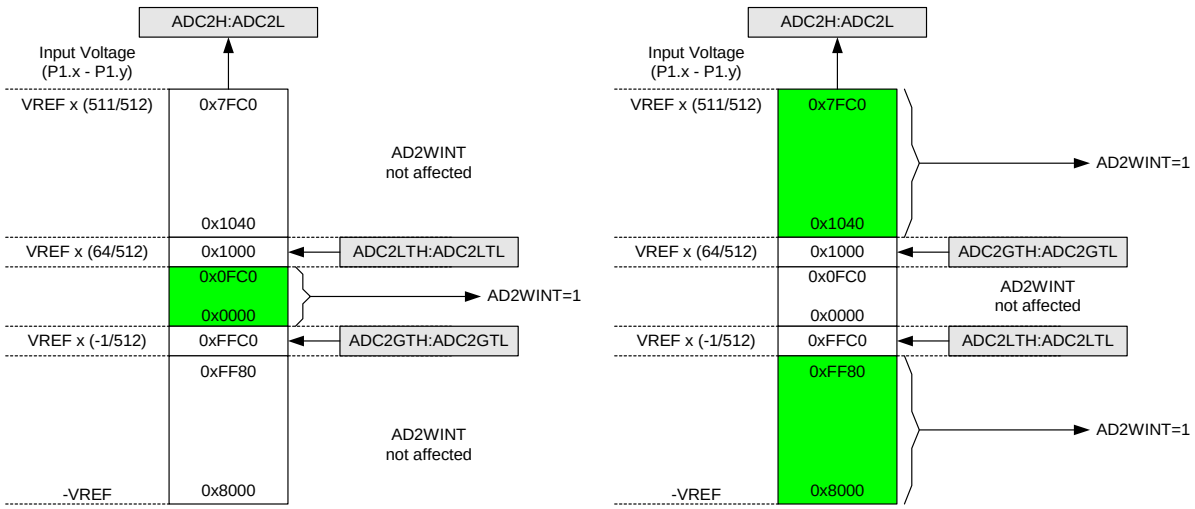


**Table 7.1. ADC2 Electrical Characteristics**

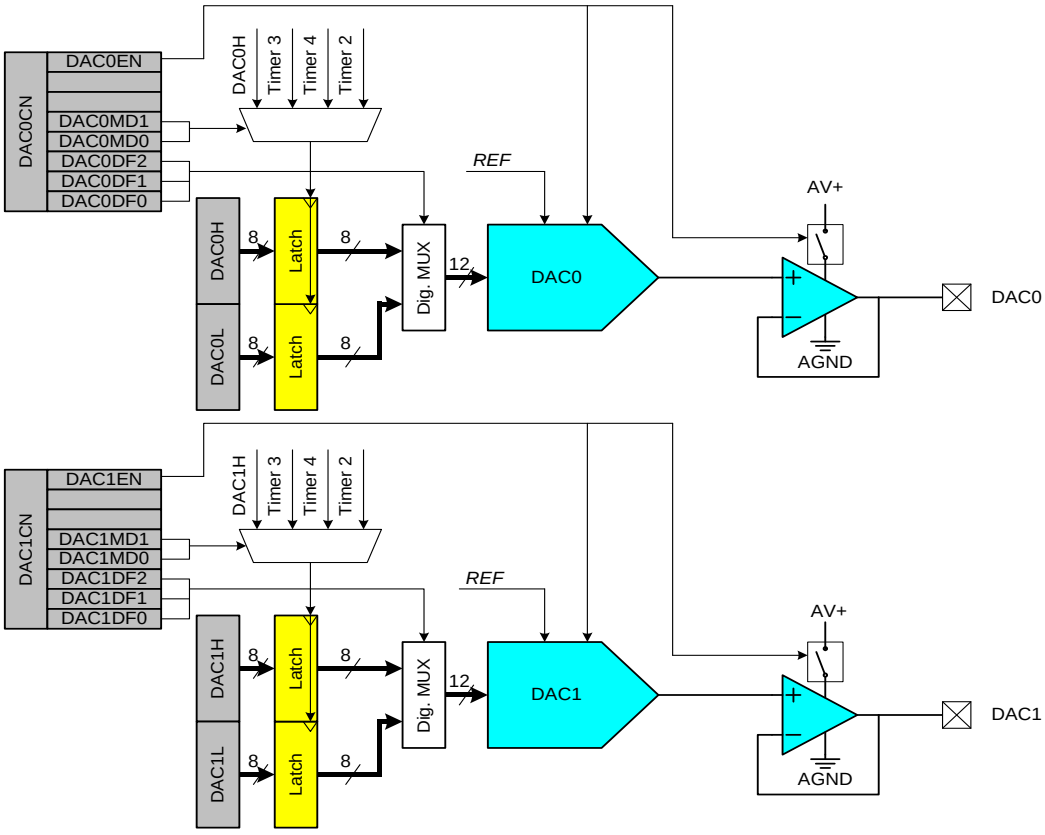
VDD = 3.0 V, VREF = 2.40 V (REFSL=0), PGA Gain = 1, -40°C to +85°C unless otherwise specified

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **DC Accuracy** | | | | | |
| Resolution | | | 10 | | bits |
| Integral Nonlinearity | | | ±0.5 | ±1 | LSB |
| Differential Nonlinearity | Guaranteed Monotonic | | ±0.5 | ±1 | LSB |
| Offset Error | | -12 | 1 | 12 | LSB |
| Full Scale Error | Differential mode | -15 | -5 | 5 | LSB |
| Offset Temperature Coefficient | | | 3.6 | | ppm/°C |
| **DYNAMIC PERFORMANCE (10 kHz sine-wave Differential input, 1 dB below Full Scale, 200 ksps)** | | | | | |
| Signal-to-Noise Plus Distortion | | 53 | 55.5 | | dB |
| Total Harmonic Distortion | Up to the 5$^{th}$ harmonic | | -67 | | dB |
| Spurious-Free Dynamic Range | | | 78 | | dB |
| **Conversion Rate** | | | | | |
| SAR Conversion Clock | | | | 3 | MHz |
| Conversion Time in SAR Clocks | | 10 | | | clocks |
| Track/Hold Acquisition Time | | 300 | | | ns |
| Throughput Rate | | | | 200 | ksps |
| **Analog Inputs** | | | | | |
| ADC Input Voltage Range | Single Ended (AIN+ - AGND)<br>Differential (AIN+ - AIN-) | 0<br>-VREF | | VREF<br>VREF | V<br>V |
| Absolute Pin Voltage with respect to AGND | Single Ended or Differential | 0 | | AV+ | V |
| Input Capacitance | | | 5 | | pF |
| **Temperature Sensor** | | | | | |
| Linearity | | | ±0.2 | | °C |
| Offset | Temp = 0 °C | | 776 | | mV |
| Offset Error (Note 1) | Temp = 0 °C | | ±8.9 | | mV |
| Slope | | | 2.89 | | mV/°C |
| Slope Error (Note 1) | | | ±63 | | µV/°C |
| **Power Specifications** | | | | | |
| Power Supply Current (VDD supplied to ADC2) | Operating Mode, 200 ksps | | 400 | 900 | µA |
| Power Supply Rejection | | | ±0.3 | | mV/V |
| Note 1: Represents one standard deviation from the mean value. | | | | | |

# 8. DACs, 12-Bit Voltage Mode (DAC0 and DAC1, C8051F060/1/2/3)

The C8051F060/1/2/3 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 (C8051F060/2 devices) or the VREF2 pin (C8051F061/3 devices). See Section “9. Voltage Reference 2 (C8051F060/2)” on page 111 or Section “10. Voltage Reference 2 (C8051F061/3)” on page 113 for more information on configuring the voltage reference for the DACs. Note that the BIASE bit described in the voltage reference sections must be set to ‘1’ to use the DACs.

**Figure 8.1. DAC Functional Block Diagram**



## 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.

### 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

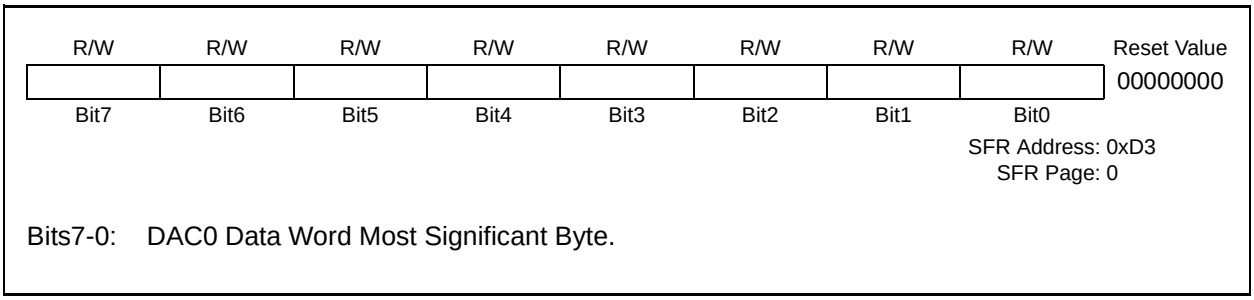
Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the DAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the DAC is used to generate a waveform of a defined sampling 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.

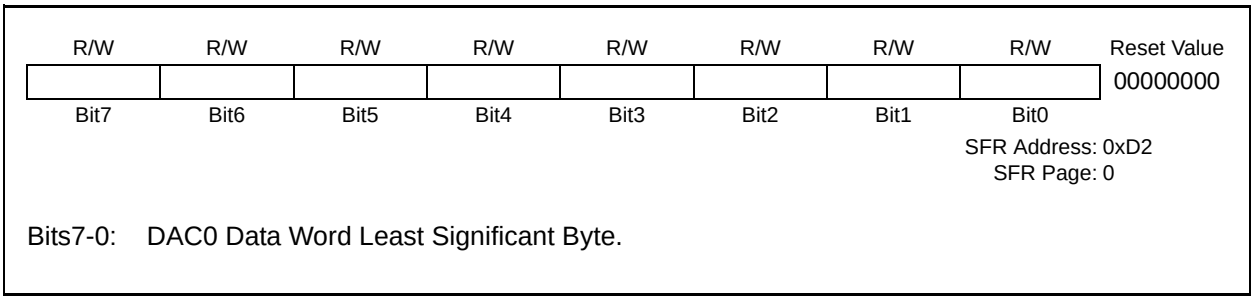
DAC1 is functionally the same as DAC0 described above. The electrical specifications for both DAC0 and DAC1 are given in Table 8.1.

**Figure 8.2. DAC0H: DAC0 High Byte Register**




Bits7-0: DAC0 Data Word Most Significant Byte.

**Figure 8.3. DAC0L: DAC0 Low Byte Register**




Bits7-0: DAC0 Data Word Least Significant Byte.

**Figure 8.4. DAC0CN: DAC0 Control Register**

| 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 | | |

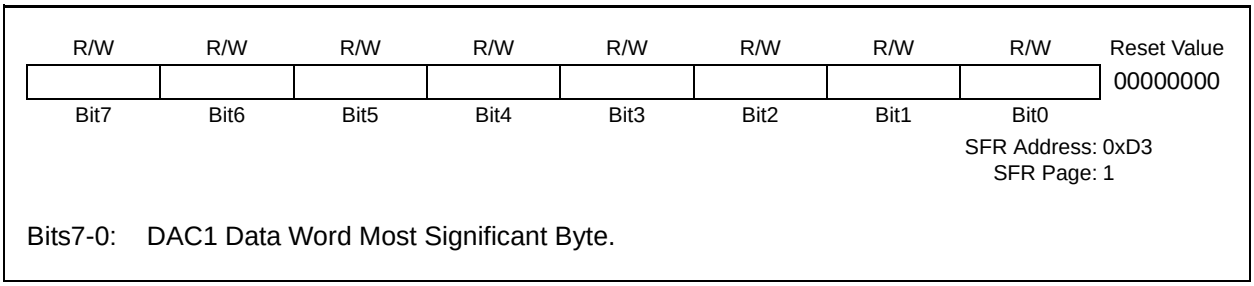
011: The most significant 7-bits of the DAC0 Data Word is in DAC0H[6:0], while the least significant 5-bits are in DAC0L[7:3].

| 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].

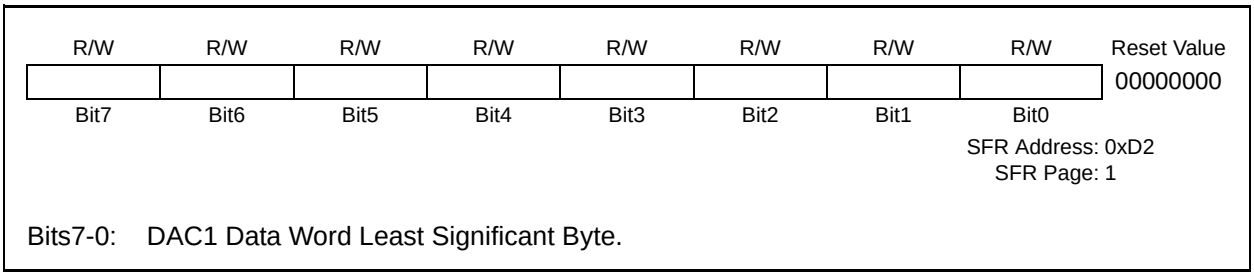
| DAC0H | | | | | | | | DAC0L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSB | | | | | | | | | | | LSB | | | | |

**Figure 8.5. DAC1H: DAC1 High Byte Register**




Bits7-0: DAC1 Data Word Most Significant Byte.

**Figure 8.6. DAC1L: DAC1 Low Byte Register**




Bits7-0: DAC1 Data Word Least Significant Byte.

**Figure 8.7. DAC1CN: DAC1 Control Register**

| 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**

VDD = 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 |
| VDD 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 = 40pF | | 0.44 | | V/µs |
| Output Settling Time to 1/2 LSB | Load = 40pF, Output swing from code 0xFFF to 0x014 | | 10 | | µs |
| Output Voltage Swing | | 0 | | VREF-1LSB | V |
| Startup Time | | | 10 | | µs |
| **Analog Outputs** | | | | | |
| Load Regulation | $I_L$ = 0.01mA to 0.3mA at code 0xFFF | | 60 | | ppm |
| **Power Consumption (each DAC)** | | | | | |
| Power Supply Current (AV+ supplied to DAC) | Data Word = 0x7FF | | 300 | 500 | µA |

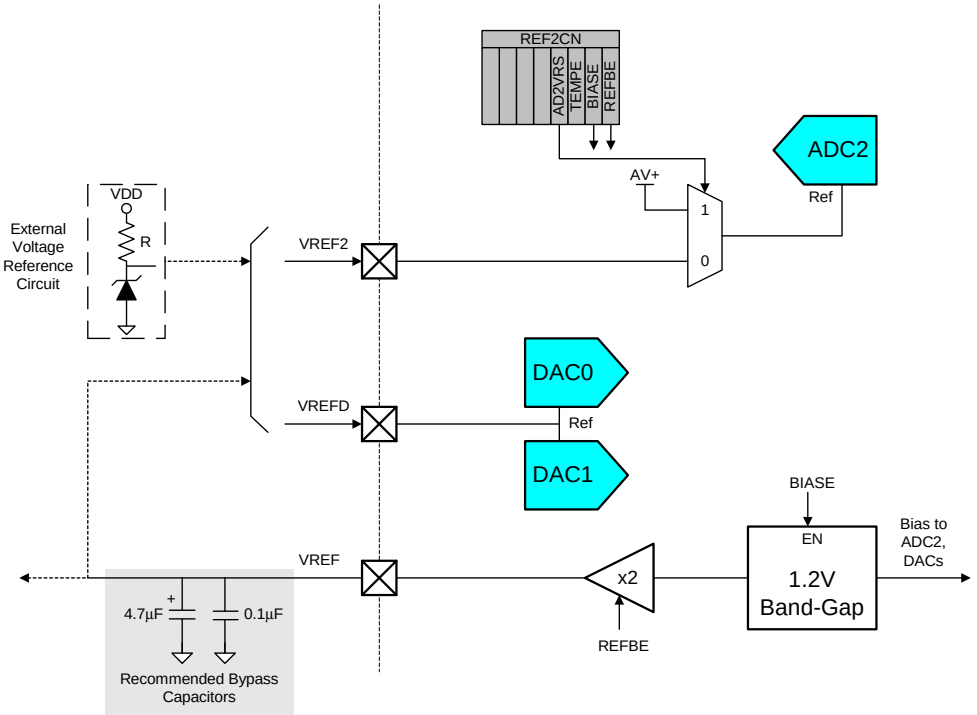
# 9. Voltage Reference 2 (C8051F060/2)

The voltage reference circuitry offers full flexibility in operating the ADC2 and DAC modules. Two voltage reference input pins allow ADC2 and the two DACs to reference an external voltage reference or the on-chip voltage reference output. ADC2 may also reference the analog power supply voltage, via the VREF multiplexer shown in Figure 9.1.

The internal voltage reference circuit consists of a 1.2 V, temperature stable 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 shown in Figure 9.1. 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, as shown in Figure 9.1.

The Reference Control Register 2, REF2CN (defined in Figure 9.2) enables/disables the internal reference generator and selects the reference input for ADC2. The BIASE bit in REF2CN 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 ADC2 or either DAC is used, regardless of the voltage reference used. If neither ADC2 nor the DACs are being used, both of these bits can be set to logic 0 to conserve power. Bit AD2VRS selects between VREF2 and AV+ for the ADC2 voltage reference source. The electrical specifications for the Voltage Reference are given in Table 9.1.

**Figure 9.1. Voltage Reference Functional Block Diagram**



The temperature sensor connects to the highest order input of the ADC2 input multiplexer (see Section “7. 10-Bit ADC (ADC2, C8051F060/1/2/3)” on page 87). The TEMPE bit within REF2CN enables and disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state, and any A/D measurements performed on the sensor while disabled result in meaningless data.

**Figure 9.2. REF2CN: Reference Control Register 2**

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

SFR Address: 0xD1
SFR Page: 2

Bits7-4: UNUSED. Read = 0000b; Write = don’t care.
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 ADC2 or DACs).
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**

**VDD = 3.0 V, AV+ = 3.0 V, -40 to +85 °C unless otherwise specified**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Internal Reference (REFBE = 1)** | | | | | |
| Output Voltage | 25 °C ambient | 2.36 | 2.43 | 2.48 | V |
| VREF Power Supply Current | | | 50 | | µA |
| 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 |
| **External Reference (REFBE = 0)** | | | | | |
| Input Voltage Range | | 1.00 | | (AV+) - 0.3 | V |
| Input Current | | | 0 | 1 | µA |

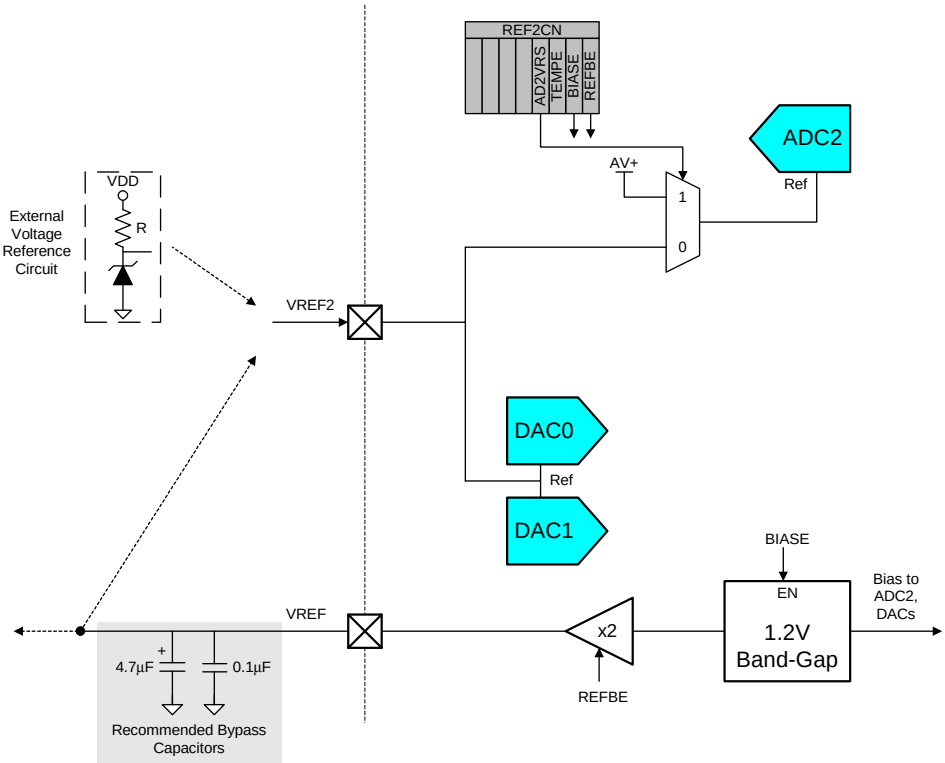
# 10. Voltage Reference 2 (C8051F061/3)

The internal voltage reference circuit consists of a 1.2 V, temperature stable 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 VREF2 input pin shown in Figure 10.1. 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, as shown in Figure 10.1.

The VREF2 pin provides a voltage reference input for ADC2 and the DACs. ADC2 may also reference the analog power supply voltage, via the VREF multiplexers shown in Figure 10.1.

The Reference Control Register 2, REF2CN (defined in Figure 10.2) enables/disables the internal reference generator and selects the reference input for ADC2. The BIASE bit in REF2CN 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 ADC2 or either DAC is used, regardless of the voltage reference used. If neither ADC2 nor the DACs are being used, both of these bits can be set to logic 0 to conserve power. Bit AD2VRS selects between VREF2 and AV+ for the ADC2 voltage reference source. The electrical specifications for the Voltage Reference are given in Table 10.1.

**Figure 10.1. Voltage Reference Functional Block Diagram**



The temperature sensor connects to the highest order input of the ADC2 input multiplexer (see Section “7. 10-Bit ADC (ADC2, C8051F060/1/2/3)” on page 87). The TEMPE bit within REF2CN enables and disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state, and any A/D measurements performed on the sensor while disabled result in meaningless data.

**Figure 10.2. REF2CN: Reference Control Register 2**

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

SFR Address: 0xD1
SFR Page: 2

Bits7-4: UNUSED. Read = 0000b; Write = don’t care.
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 ADC2 or DACs).
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 10.1. Voltage Reference Electrical Characteristics**

**VDD = 3.0 V, AV+ = 3.0 V, -40 to +85 °C unless otherwise specified**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Internal Reference (REFBE = 1)** | | | | | |
| Output Voltage | 25 °C ambient | 2.36 | 2.43 | 2.48 | V |
| VREF Power Supply Current | | | 50 | | µA |
| 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 |
| **External Reference (REFBE = 0)** | | | | | |
| Input Voltage Range | | 1.00 | | (AV+) - 0.3 | V |
| Input Current | | | 0 | 1 | µA |

# 11. Voltage Reference 2 (C8051F064/5/6/7)

The internal voltage reference circuit consists of a 1.2 V, temperature stable bandgap voltage reference generator and a gain-of-two output buffer amplifier. The internal reference may be routed to the VREF pin as shown in Figure 11.1. 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, as shown in Figure 11.1.

The Reference Control Register 2, REF2CN (defined in Figure 11.2) enables/disables the internal reference generator. The BIASE bit in REF2CN 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. The electrical specifications for the Voltage Reference are given in Table 11.1.

**Figure 11.1. Voltage Reference Functional Block Diagram**

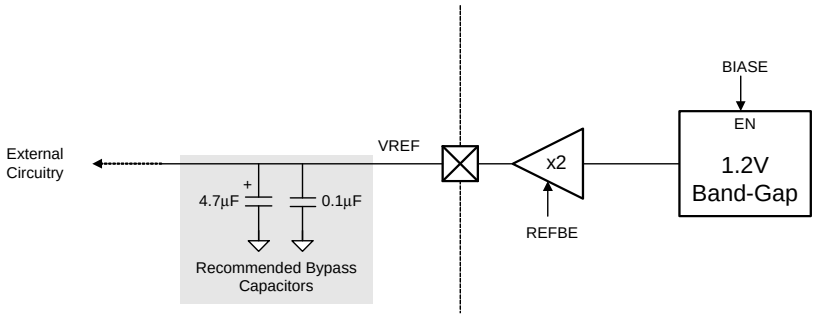


**Figure 11.2. REF2CN: Reference Control Register 2**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | - | - | 0 | 0 | BIASE | REFBE | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xD1
SFR Page: 2

Bits7-4: UNUSED. Read = 0000b; Write = don't care.
Bits2-3: RESERVED. Must Write to 00b.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC2 or DACs).
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 11.1. Voltage Reference Electrical Characteristics**

**VDD = 3.0 V, AV+ = 3.0 V, -40 to +85 °C unless otherwise specified**

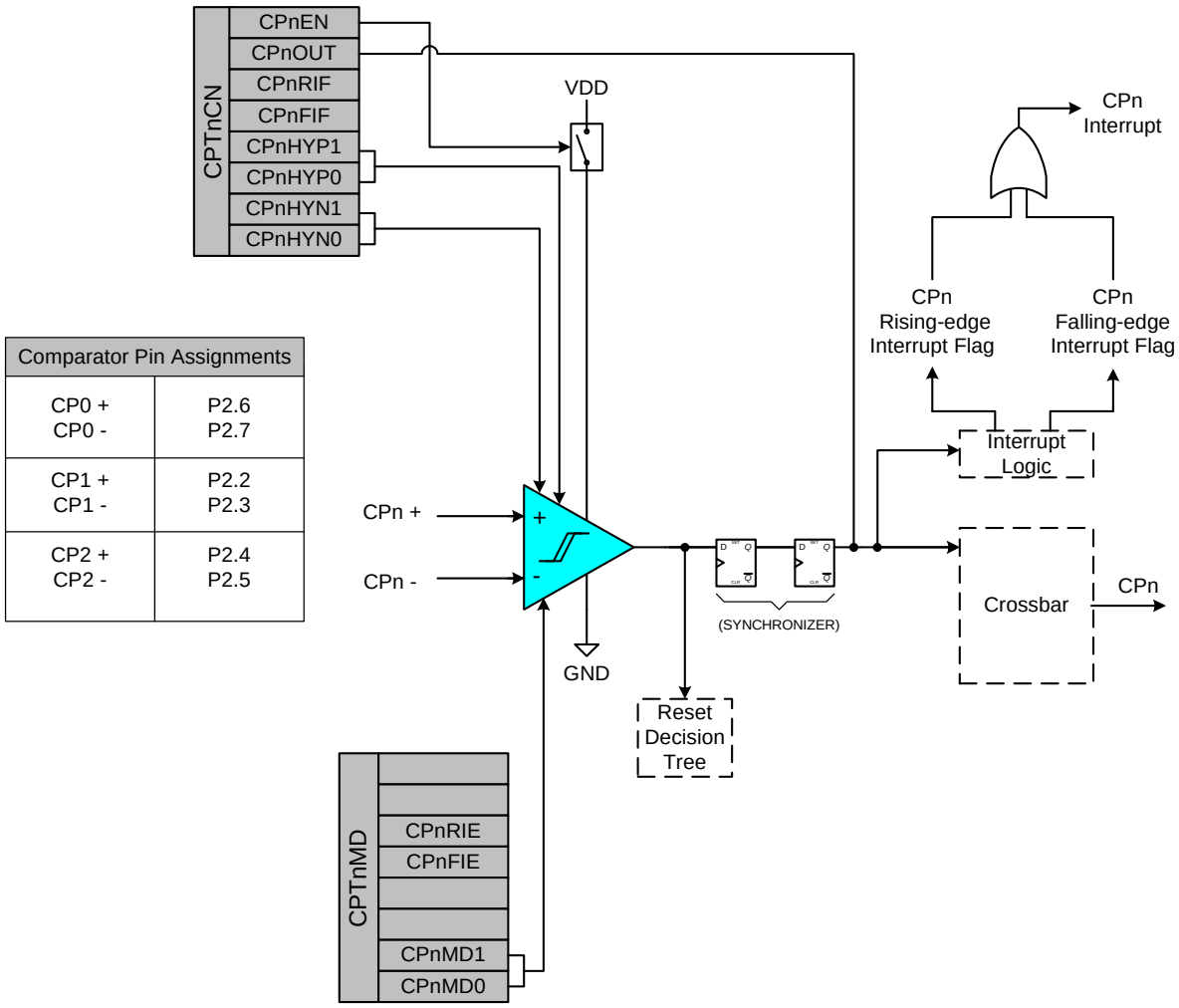
| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Internal Reference (REFBE = 1)** | | | | | |
| Output Voltage | 25 °C ambient | 2.36 | 2.43 | 2.48 | V |
| VREF Power Supply Current | | | 50 | | µA |
| VREF Short-Circuit Current | | | | 30 | mA |
| VREF Temperature Coefficient | | | 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 |

# 12. Comparators

C8051F06x family of devices include three on-chip programmable voltage comparators, shown in Figure 12.1. Each comparator offers programmable response time and hysteresis. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull, and Comparator inputs should be configured as analog inputs (see Section “18.1.5. Configuring Port 1 and 2 pins as Analog Inputs” on page 207). The Comparator may also be used as a reset source (see Section “14.5. Comparator0 Reset” on page 165).

The output of a Comparator can be polled by software, used as an interrupt source, used as a reset source, and/or routed to a Port pin. Each comparator can be individually enabled and disabled (shutdown). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to less than 1 µA. See Section “18.1.1. Crossbar Pin Assignment and Allocation” on page 205 for details on configuring the Comparator output via the digital Crossbar. The Comparator inputs can be externally driven from -0.25 V to (VDD) + 0.25 V without damage or upset. The

**Figure 12.1. Comparator Functional Block Diagram**

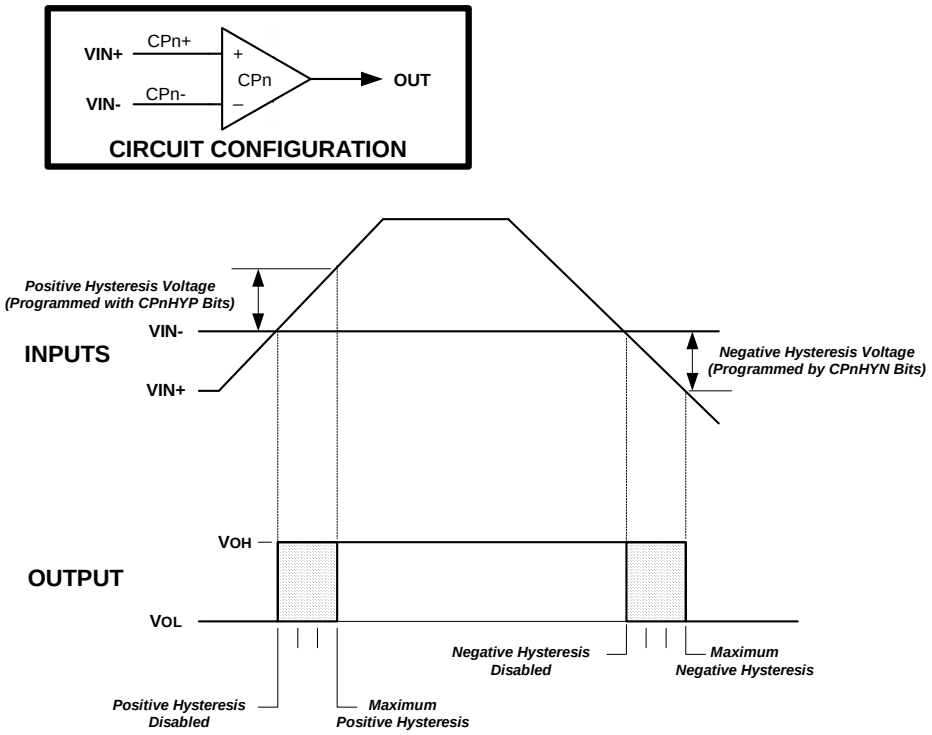



| Comparator Pin Assignments | |
|---|---|
| CP0 +<br>CP0 - | P2.6<br>P2.7 |
| CP1 +<br>CP1 - | P2.2<br>P2.3 |
| CP2 +<br>CP2 - | P2.4<br>P2.5 |

complete electrical specifications for the Comparator are given in Table 12.1.

The Comparator response time may be configured in software using the CPnMD1-0 bits in register CPTnMD (see Figure 12.4). Selecting a longer response time reduces the amount of power consumed by the comparator. See Table 12.1 for complete timing and current consumption specifications.

**Figure 12.2. Comparator Hysteresis Plot**




The hysteresis of the Comparator is software-programmable via its Comparator Control register (CPTnCN). 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 Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in Figure 12.3). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Figure 12.2, 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 CPnHYP bits.

Comparator interrupts can be generated on either rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “13.3. Interrupt Handler” on page 151). The rising and/or falling -edge interrupts are enabled using the comparator’s Rising/Falling Edge Interrupt Enable Bits (CPnRIE and CPnFIE) in their respective Comparator Mode Selection Register (CPTnMD), shown in Figure 12.4. These bits allow the user to control which edge (or both) will cause a comparator interrupt. However, the comparator interrupt must also be enabled in the Extended Interrupt Enable Register (EIE1). The CPnFIF flag is set to logic 1 upon a Comparator falling-edge interrupt, and the CPnRIF flag is set to logic 1 upon the Comparator rising-edge interrupt. Once set, these bits remain set until cleared by software. The output state of a Comparator can be obtained at any time by reading the CPnOUT bit. A Comparator is enabled by setting its respective CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.Upon enabling a comparator, the output of the comparator is not immediately valid. Before using a comparator as an interrupt or reset source, software should wait for a minimum of the specified “Power-up time” as specified in Table 12.1, “Comparator Electrical Characteristics,” on page 122.

## 12.1. Comparator Inputs

The Port pins selected as comparator inputs should be configured as analog inputs in the Port 2 Input Configuration Register (for details on Port configuration, see Section “18.1.3. Configuring Port Pins as Digital Inputs” on page 207). The inputs for Comparator are on Port 2 as follows:

| Comparator Input | Port PIN |
|---|---|
| CP0 + | P2.6 |
| CP0 - | P2.7 |
| CP1 + | P2.2 |
| CP1 - | P2.3 |
| CP2 + | P2.4 |
| CP2 - | P2.5 |

**Figure 12.3. CPTnCN: Comparator 0, 1, and 2 Control Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CPnEN | CPnOUT | CPnRIF | CPnFIF | CPnHYP1 | CPnHYP0 | CPnHYN1 | CPnHYN0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: CPT0CN: 0x88; CPT1CN: 0x88; CPT2CN: 0x88
SFR Pages: CPT0CN: page 1; CPT1CN: page 2; CPT2CN: page 3

Bit7: CPnEN: Comparator Enable Bit. **(Please see note below.)**
0: Comparator Disabled.
1: Comparator Enabled.

Bit6: CPnOUT: Comparator Output State Flag.
0: Voltage on CPn+ < CPn-.
1: Voltage on CPn+ > CPn-.

Bit5: CPnRIF: Comparator Rising-Edge Interrupt Flag.
0: No Comparator Rising Edge Interrupt has occurred since this flag was last cleared.
1: Comparator Rising Edge Interrupt has occurred. Must be cleared by software.

Bit4: CPnFIF: Comparator Falling-Edge Interrupt Flag.
0: No Comparator Falling-Edge Interrupt has occurred since this flag was last cleared.
1: Comparator Falling-Edge Interrupt has occurred. Must be cleared by software.

Bits3-2: CPnHYP1-0: Comparator Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 20 mV.

Bits1-0: CPnHYN1-0: Comparator Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 20 mV.

NOTE: Upon enabling a comparator, the output of the comparator is not immediately valid. Before using a comparator as an interrupt or reset source, software should wait for a minimum of the specified “Power-up time” as specified in Table 12.1, “Comparator Electrical Characteristics,” on page 122.

**Figure 12.4. CPTnMD: Comparator Mode Selection Register**

| R/W | R/W | R/W | R/W | R | R | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| - | - | CPnRIE | CPnFIE | - | - | CPnMD1 | CPnMD0 | 00000010 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: CPT0MD: 0x89; CPT1MD: 0x89; CPT2MD: 0x89

SFR Page: CPT0MD: page 1; CPT1MD: page 2; CPT2MD: page 3

Bits7-6: UNUSED. Read = 00b, Write = don't care.

Bit 5: CPnRIE: Comparator Rising-Edge Interrupt Enable Bit.
0: Comparator rising-edge interrupt disabled.
1: Comparator rising-edge interrupt enabled.

Bit 4: CPnFIE: Comparator Falling-Edge Interrupt Enable Bit.
0: Comparator falling-edge interrupt disabled.
1: Comparator falling-edge interrupt enabled.

Bits3-2: UNUSED. Read = 00b, Write = don't care.

Bits1-0: CPnMD1-CPnMD0: Comparator Mode Select
These bits select the response time for the Comparator.

| Mode | CPnMD1 | CPnMD0 | Notes |
|---|---|---|---|
| 0 | 0 | 0 | Fastest Response Time |
| 1 | 0 | 1 | - |
| 2 | 1 | 0 | - |
| 3 | 1 | 1 | Lowest Power Consumption |

**Table 12.1. Comparator Electrical Characteristics**

**VDD = 3.0 V, -40 to +85 °C unless otherwise specified.**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Response Time, Mode 0 | CPn+ - CPn- = 100 mV | | 100 | | ns |
| | CPn+ - CPn- = 10 mV | | 250 | | ns |
| Response Time, Mode 1 | CPn+ - CPn- = 100 mV | | 175 | | ns |
| | CPn+ - CPn- = 10 mV | | 500 | | ns |
| Response Time, Mode 2 | CPn+ - CPn- = 100 mV | | 320 | | ns |
| | CPn+ - CPn- = 10 mV | | 1100 | | ns |
| Response Time, Mode 3 | CPn+ - CPn- = 100 mV | | 1050 | | ns |
| | CPn+ - CPn- = 10 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 | 3 | 5 | 7 | mV |
| Positive Hysteresis 3 | CPnHYP1-0 = 10 | 7 | 10 | 15 | mV |
| Positive Hysteresis 4 | CPnHYP1-0 = 11 | 15 | 20 | 25 | mV |
| Negative Hysteresis 1 | CPnHYN1-0 = 00 | | 0 | 1 | mV |
| Negative Hysteresis 2 | CPnHYN1-0 = 01 | 3 | 5 | 7 | mV |
| Negative Hysteresis 3 | CPnHYN1-0 = 10 | 7 | 10 | 15 | mV |
| Negative Hysteresis 4 | CPnHYN1-0 = 11 | 15 | 20 | 25 | mV |
| Inverting or Non-Inverting Input Voltage Range | | -0.25 | | VDD + 0.25 | V |
| Input Capacitance | | | 7 | | pF |
| Input Bias Current | | -5 | 0.001 | +5 | nA |
| Input Offset Voltage | | -5 | | +5 | mV |
| **Power Supply** | | | | | |
| Power Supply Rejection | | | 0.1 | 1 | mV/V |
| Power-up Time | | | 10 | | µs |
| Supply Current at DC | Mode 0 | | 7.6 | | µA |
| | Mode 1 | | 3.2 | | µA |
| | Mode 2 | | 1.3 | | µA |
| | Mode 3 | | 0.4 | | µA |

# 13. 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 24), two full-duplex UARTs (see description in Section 22 and Section 23), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (see Section 13.2.6), and 59/24 General-Purpose I/O Pins (see description in Section 18). The CIP-51 also includes on-chip debug hardware (see description in Section 26), 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.

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- 256 Bytes of Internal RAM
- 59/24 General-Purpose I/O Pins
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

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 13.1 for a block diagram). The CIP-51 includes the following features:

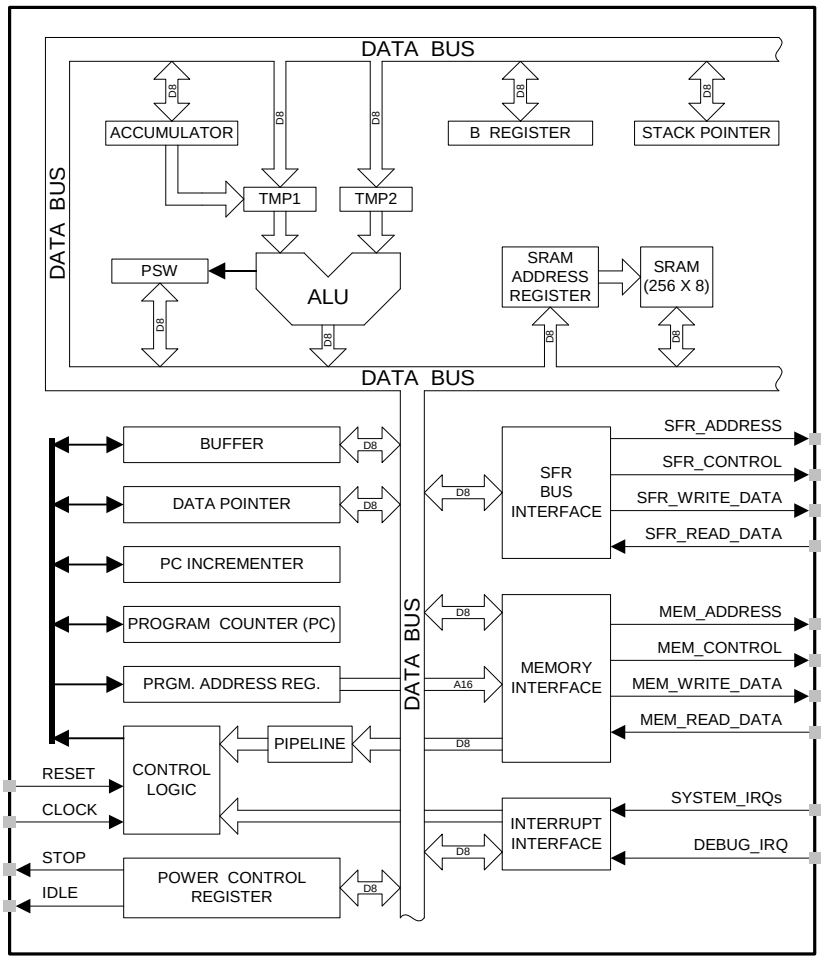
### Performance

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

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

**Figure 13.1. 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 a single byte at a time 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) which interfaces to the CIP-51 via its JTAG port to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

## 13.1. 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.

### 13.1.1. 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 13.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

### 13.1.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip XRAM, and writing to 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 “16. Flash Memory” on page 177). 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 187 for details.

**Table 13.1. CIP-51 Instruction Set Summary**

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

**Table 13.1. CIP-51 Instruction Set Summary (Continued)**

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

**Table 13.1. CIP-51 Instruction Set Summary (Continued)**

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

**Notes on Registers, Operands and Addressing Modes:**

**Rn** - Register R0-R7 of the currently selected register bank.

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

**rel** - 8-bit, signed (two's 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.

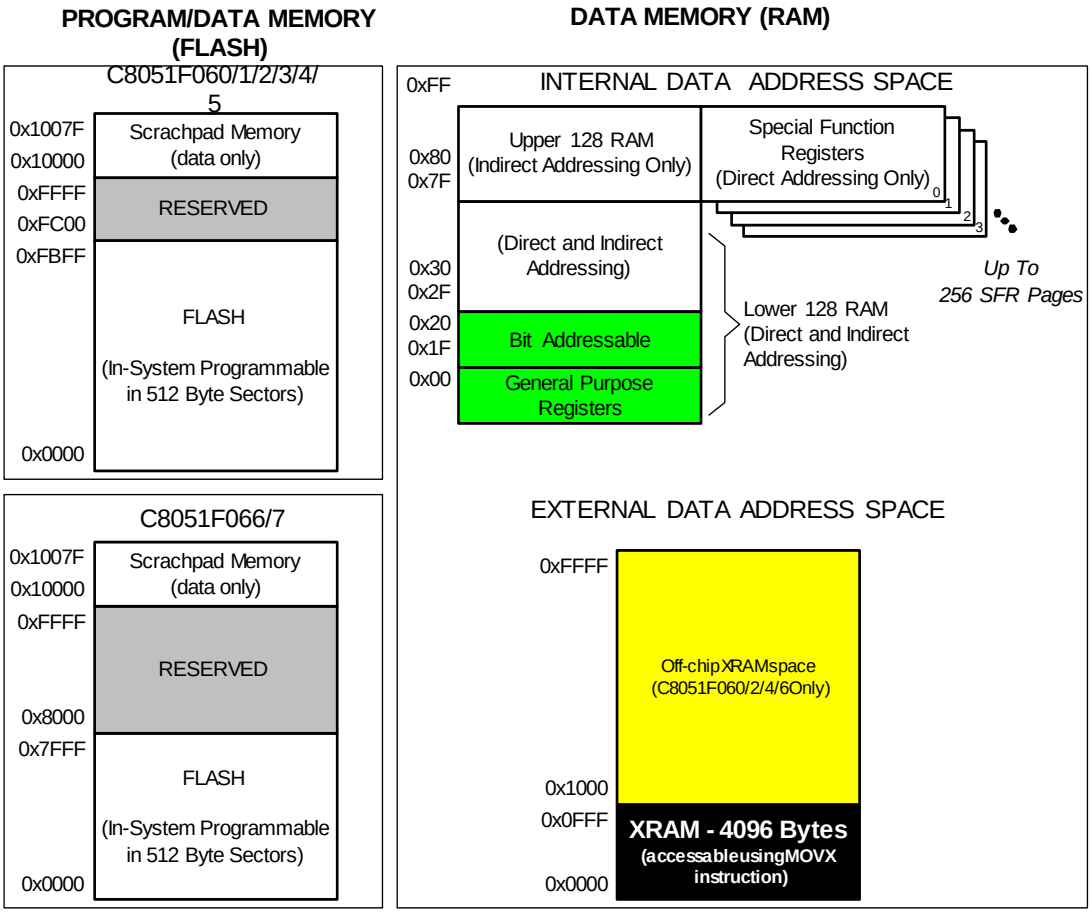
**addr16** - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space.

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

## 13.2. 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 64 k bytes (C8051F060/1/2/3/4/5) or 32 k bytes (C8051F066/7) of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 13.2.

**Figure 13.2. Memory Map**




### 13.2.1. Program Memory

The CIP-51 has a 64 k byte program memory space. The C8051F060/1/2/3/4/5 devices implement 64 k bytes of this program memory space as in-system re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0xFFFF. Note: 1024 bytes (0xFC00 to 0xFFFF) of this memory are reserved, and are not available for user program storage. The C8051F066/7 implement 32 k bytes of this program memory space as in-system re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x7FFF.

Program memory is normally assumed to be read-only (using the MOVC instruction). However, the CIP-51 can write to program memory by enabling Flash writes, 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 “16. Flash Memory” on page 177 for further details.

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

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFRs) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing above 0x7F will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 13.2 illustrates the data memory organization of the CIP-51.

### 13.2.3. 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 Figure 13.16). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

### 13.2.4. 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 (a bit source or destination operand 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.

### 13.2.5. 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.

### 13.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the Special Function Registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 13.2 lists the SFRs implemented in the CIP-51 System Controller.

The SFRs are accessed whenever the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the datasheet, as indicated in Table 13.3, for a detailed description of each register.

#### 13.2.6.1.SFR Paging

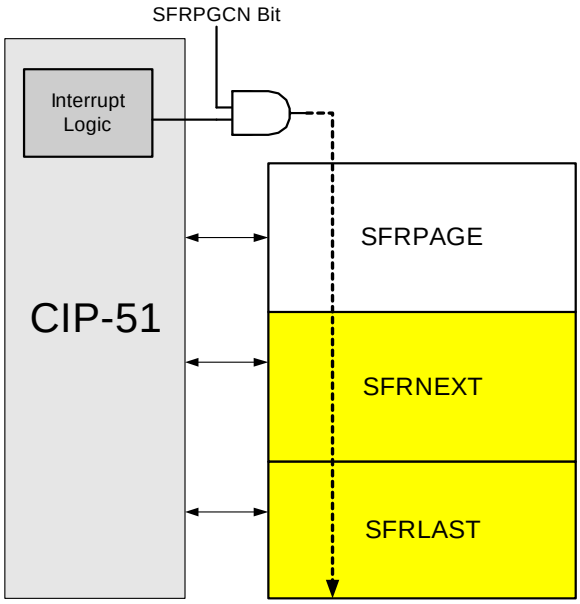
The CIP-51 features *SFR paging,* allowing the device to map many SFRs into the 0x80 to 0xFF memory address space. The SFR memory space has 256 *pages*. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFRs. The C8051F06x family of devices utilizes five SFR pages: 0, 1, 2, 3, and F. SFR pages are selected using the Special Function Register Page Selection register, SFRPAGE (see Figure 13.10). 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).

#### 13.2.6.2.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 13.3. 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.

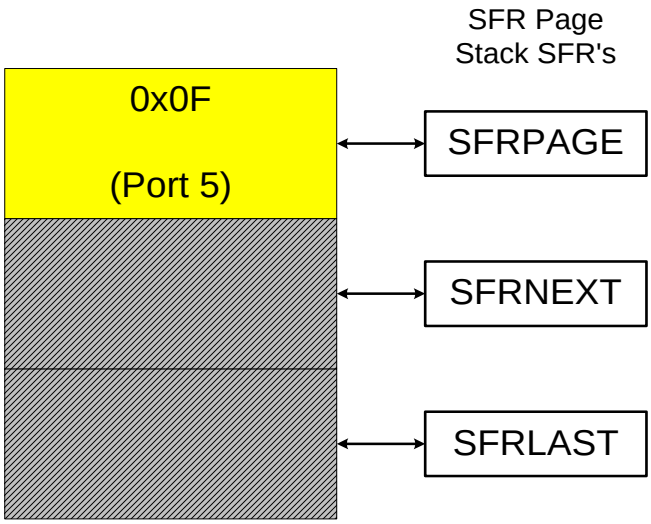
A summary of the SFR locations (address and SFR page) is provided in Table 13.2. in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Note that certain SFRs are accessible from ALL SFR pages, and are denoted by the “**(ALL PAGES)**” designation. For example, the Port I/O registers P0, P1, P2, and P3 all have the “**(ALL PAGES)**” designation, indicating these SFRs are accessible from all SFR pages regardless of the SFRPAGE register value.

#### 13.2.6.3.SFR Page Stack Example

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

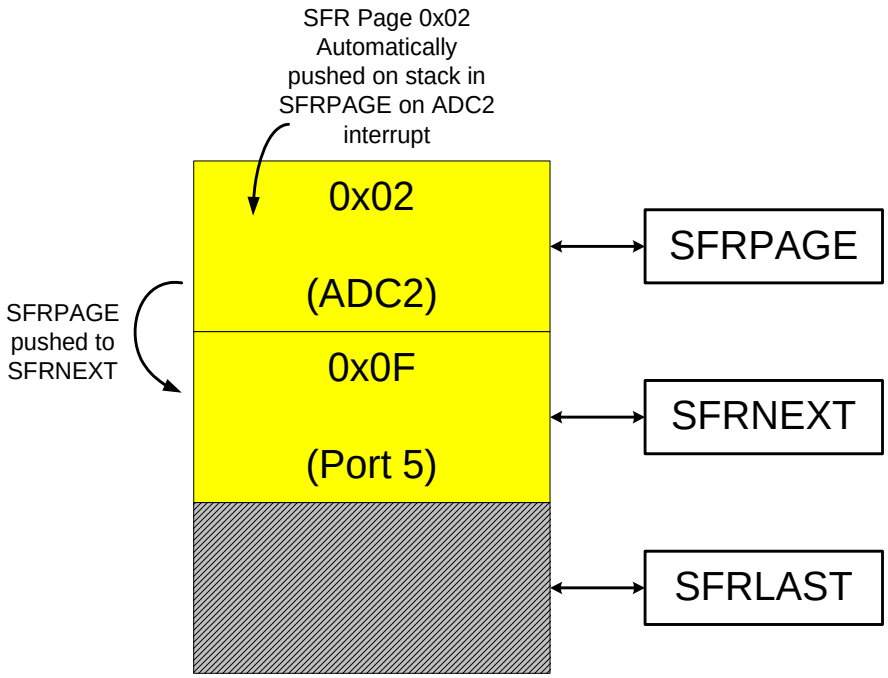
In this example, the SFR Page Control is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to Port 5 (SFR “P5”, located at address 0xD8 on SFR Page 0x0F). The device is also using the Programmable Counter Array (PCA) and the 10-bit 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 13.4 below.

**Figure 13.4. SFR Page Stack While Using SFR Page 0x0F To Access Port 5**



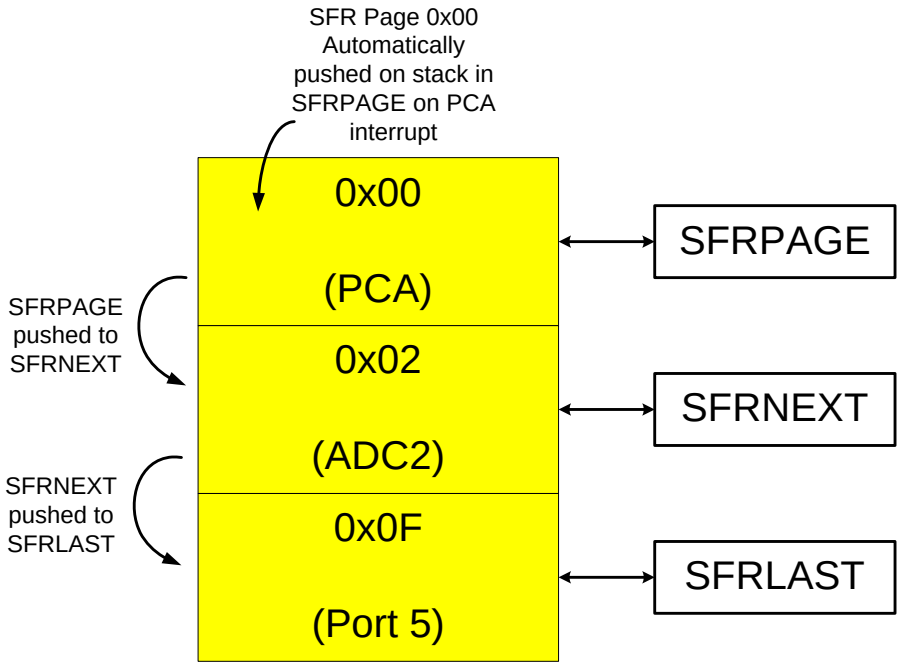
While CIP-51 executes in-line code (writing values to Port 5 in this example), ADC2 Window Comparator Interrupt occurs. The CIP-51 vectors to the ADC2 Window Comparator ISR and pushes the current SFR Page value (SFR Page 0x0F) into SFRNEXT in the SFR Page Stack. The SFR page needed to access ADC2's SFRs 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 SFRs. 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 SFRs that are not on SFR Page 0x02. See Figure 13.5 below.

**Figure 13.5. SFR Page Stack After ADC2 Window Comparator Interrupt Occurs**



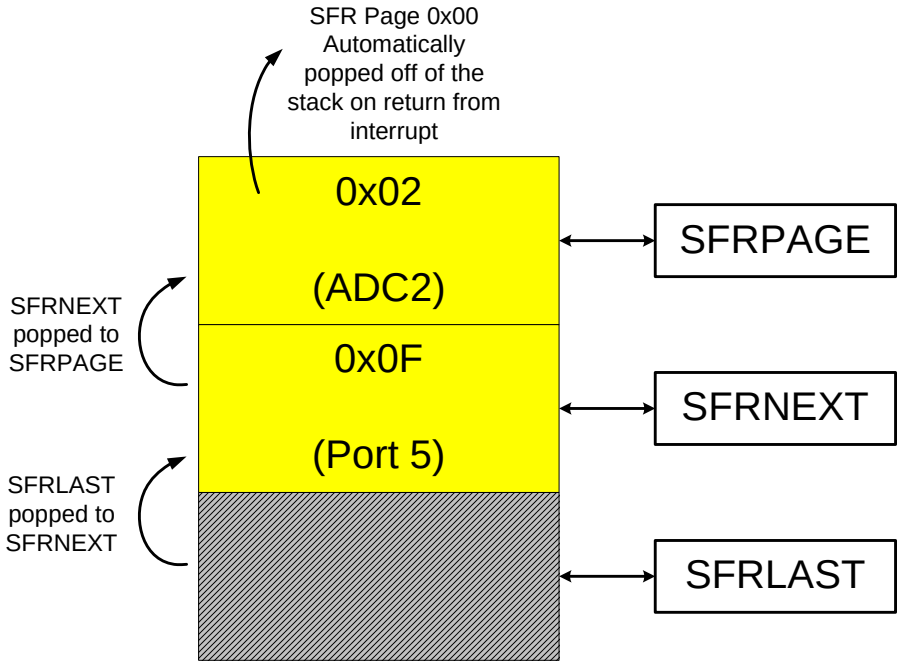
While in the ADC2 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a *high* priority interrupt, while the ADC2 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place 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 13.6 below.

**Figure 13.6. SFR Page Stack Upon PCA Interrupt Occurring During an ADC2 ISR**



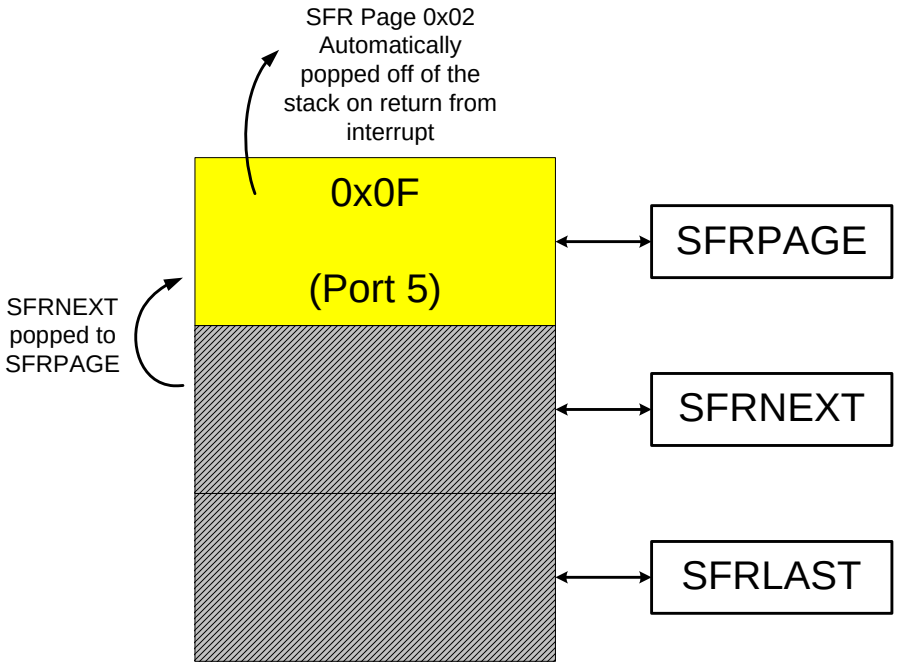
On exit from the PCA interrupt service routine, the CIP-51 will return to the ADC2 Window Comparator ISR. On execution of the RETI instruction, SFR Page 0x00 used to access the PCA registers will be automatically popped off of the SFR Page Stack, and the contents of the SFRNEXT register will be moved to the SFRPAGE register. Software in the ADC2 ISR can continue to access SFRs 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 13.7 below.

**Figure 13.7. 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 13.8 below.

**Figure 13.8. 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 Figure 13.9.

**Figure 13.9. SFRPGCN: SFR Page Control Register**

| 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 SFRs 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 SFRs for the peripheral or function that is the source of the interrupt.

**Figure 13.10. SFRPAGE: SFR Page 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: 0x84
SFR Page: All Pages

Bits7-0: SFR Page Bits: Byte Represents the SFR Page the C8051 MCU uses when reading or modifying SFRs.
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 SFRs of the corresponding peripheral/function that caused the interrupt, and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt).
SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and **not** by reading/writing to the SFRPAGE register)

**Figure 13.11. 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 SFR-LAST is the third entry. The SFR stack bytes may be used to alter the context in the SFR Page Stack, and will not cause the stack to ‘push’ or ‘pop’. Only interrupts and return from interrupt cause push and pop 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.

**Figure 13.12. 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 SFR-LAST is the third entry. The SFR stack bytes may be used to alter the context in the SFR Page Stack, and will not cause the stack to ‘push’ or ‘pop’. Only interrupts and return from interrupt cause push and pop 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 13.2. 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 1 2 3 F | SPI0CN<br>CAN0CN<br><br>DMA0CF<br>P7 | PCA0L<br><br>DMA0CTL | PCA0H<br><br>DMA0CTH | PCA0CPL0<br><br>DMA0CSL | PCA0CPH0<br><br>DMA0CSH | PCA0CPL1<br><br>DMA0BND | PCA0CPH1<br><br>DMA0ISW | WDTCN<br>**(ALL PAGES)** |
| F0 | 0 1 2 3 F | B<br>**(ALL PAGES)** | | | | | | EIP1<br>**(ALL PAGES)** | EIP2<br>**(ALL PAGES)** |
| E8 | 0 1 2 3 F | ADC0CN<br>ADC1CN<br>ADC2CN<br><br>P6 | PCA0CPL2 | PCA0CPH2 | PCA0CPL3 | PCA0CPH3 | PCA0CPL4 | PCA0CPH4 | RSTSRC |
| E0 | 0 1 2 3 F | ACC<br>**(ALL PAGES)** | PCA0CPL5<br><br>XBR0 | PCA0CPH5<br><br>XBR1 | XBR2 | XBR3 | | EIE1<br>**(ALL PAGES)** | EIE2<br>**(ALL PAGES)** |
| D8 | 0 1 2 3 F | PCA0CN<br>CAN0DATL<br><br>DMA0CN<br>P5 | PCA0MD<br>CAN0DATH<br><br>DMA0DAL | PCA0CPM0<br>CAN0ADR<br><br>DMA0DAH | PCA0CPM1<br>CAN0TST<br><br>DMA0DSL | PCA0CPM2<br><br>DMA0DSH | PCA0CPM3<br><br>DMA0IPT | PCA0CPM4<br><br>DMA0IDT | PCA0CPM5 |
| D0 | 0 1 2 3 F | PSW<br>**(ALL PAGES)** | REF0CN<br>REF1CN<br>REF2CN | DAC0L<br>DAC1L | DAC0H<br>DAC1H | DAC0CN<br>DAC1CN | | | |
| C8 | 0 1 2 3 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 | | SMB0CR |
| C0 | 0 1 2 3 F | SMB0CN<br>CAN0STA | SMB0STA | SMB0DAT | SMB0ADR | ADC0GTL<br><br>ADC2GTL | ADC0GTH<br><br>ADC2GTH | ADC0LTL<br><br>ADC2LTL | ADC0LTH<br><br>ADC2LTH |
| B8 | 0 1 2 3 F | IP<br>**(ALL PAGES)** | SADEN0 | AMX2CF<br><br>ADC0CPT | AMX0SL<br><br>AMX2SL<br><br>ADC0CCF | ADC0CF<br>ADC1CF<br>ADC2CF | | ADC0L<br>ADC1L<br>ADC2L | ADC0H<br>ADC1H<br>ADC2H |
| | | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

**Table 13.2. Special Function Register (SFR) Memory Map**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| B0 | 0<br>1<br>2<br>3<br>F | P3<br>**(ALL PAGES)** | | | | | | | FLSCL<br><br><br>FLACL |
| A8 | 0<br>1<br>2<br>3<br>F | IE<br>**(ALL PAGES)** | SADDR0 | | | | P1MDIN | P2MDIN | |
| A0 | 0<br>1<br>2<br>3<br>F | P2<br>**(ALL PAGES)** | EMI0TC | EMI0CN | EMI0CF | P0MDOUT | P1MDOUT | P2MDOUT | P3MDOUT |
| 98 | 0<br>1<br>2<br>3<br>F | SCON0<br>SCON1 | SBUF0<br>SBUF1 | SPI0CFG | SPI0DAT | P4MDOUT | SPI0CKR<br><br>P5MDOUT | P6MDOUT | P7MDOUT |
| 90 | 0<br>1<br>2<br>3<br>F | P1<br>**(ALL PAGES)** | SSTA0 | | | | | SFRPGCN | CLKSEL |
| 88 | 0<br>1<br>2<br>3<br>F | TCON<br>CPT0CN<br>CPT1CN<br>CPT2CN | TMOD<br>CPT0MD<br>CPT1MD<br>CPT2MD | TL0<br><br><br>OSCICN | TL1<br><br><br>OSCICL | TH0<br><br><br>OSCXCN | TH1 | CKCON | PSCTL |
| 80 | 0<br>1<br>2<br>3<br>F | P0<br>**(ALL PAGES)** | SP<br>**(ALL PAGES)** | DPL<br>**(ALL PAGES)** | DPH<br>**(ALL PAGES)** | SFRPAGE<br>**(ALL PAGES)** | SFRNEXT<br>**(ALL PAGES)** | SFRLAST<br>**(ALL PAGES)** | PCON<br>**(ALL PAGES)** |
| | | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

**Table 13.3. Special Function Registers**

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| B | 0xF0 | All Pages | B Register | page 150 |
| ACC | 0xE0 | All Pages | Accumulator | page 150 |
| ADC0CCF | 0xBB | F | ADC0 Calibration Coefficient | page 68 |
| ADC0CF | 0xBC | 0 | ADC0 Configuration | page 58 |
| ADC0CN | 0xE8 | 0 | ADC0 Control | page 60 |
| ADC0CPT | 0xBA | F | ADC0 Calibration Pointer | page 68 |
| ADC0GTH | 0xC5 | 0 | ADC0 Greater-Than High | page 69 |
| ADC0GTL | 0xC4 | 0 | ADC0 Greater-Than Low | page 69 |
| ADC0H | 0xBF | 0 | ADC0 Data Word High | page 63 |
| ADC0L | 0xBE | 0 | ADC0 Data Word Low | page 63 |
| ADC0LTH | 0xC7 | 0 | ADC0 Less-Than High | page 70 |
| ADC0LTL | 0xC6 | 0 | ADC0 Less-Than Low | page 70 |
| ADC1CF | 0xBC | 1 | ADC1 Configuration | page 59 |
| ADC1CN | 0xE8 | 1 | ADC1 Control | page 61 |
| ADC1H | 0xBF | 1 | ADC1 Data Word High | page 65 |
| ADC1L | 0xBE | 1 | ADC1 Data Word Low | page 65 |
| ADC2CF | 0xBC | 2 | ADC2 Configuration | page 94[*5] |
| ADC2CN | 0xE8 | 2 | ADC2 Control | page 96[*5] |
| ADC2GTH | 0xC5 | 2 | ADC2 Greater-Than High | page 97[*5] |
| ADC2GTL | 0xC4 | 2 | ADC2 Greater-Than Low | page 97[*5] |
| ADC2H | 0xBF | 2 | ADC2 Data Word High | page 95[*5] |
| ADC2L | 0xBE | 2 | ADC2 Data Word Low | page 95[*5] |
| ADC2LTH | 0xC7 | 2 | ADC2 Less-Than High | page 98[*5] |
| ADC2LTL | 0xC6 | 2 | ADC2 Less-Than Low | page 98[*5] |
| AMX0SL | 0xBB | 0 | ADC0 Multiplexer Channel Select | page 57 |
| AMX2CF | 0xBA | 2 | ADC2 Analog Multiplexer Configuration | page 94[*5] |
| AMX2SL | 0xBB | 2 | ADC2 Analog Multiplexer Channel Select | page 93[*5] |
| CAN0ADR | 0xDA | 1 | CAN0 Address | page 232[*5] |
| CAN0CN | 0xF8 | 1 | CAN0 Control | page 232[*5] |
| CAN0DATH | 0xD9 | 1 | CAN0 Data High | page 231[*5] |
| CAN0DATL | 0xD8 | 1 | CAN0 Data Low | page 231[*5] |
| CAN0STA | 0xC0 | 1 | CAN0 Status | page 233[*5] |
| CAN0TST | 0xDB | 1 | CAN0 Test | page 233[*5] |
| CKCON | 0x8E | 0 | Clock Control | page 293 |
| CLKSEL | 0x97 | F | Oscillator Clock Selection Register | page 173 |
| CPT0CN | 0x88 | 1 | Comparator 0 Control | page 120 |
| CPT0MD | 0x89 | 1 | Comparator 0 Configuration | page 121 |
| CPT1CN | 0x88 | 2 | Comparator 1 Control | page 120 |
| CPT1MD | 0x89 | 2 | Comparator 1 Configuration | page 121 |
| CPT2CN | 0x88 | 3 | Comparator 2 Control | page 120 |

**Table 13.3. Special Function Registers (Continued)**

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| CPT2MD | 0x89 | 3 | Comparator 2 Configuration | page 121 |
| DAC0CN | 0xD4 | 0 | DAC0 Control | page 106[*5] |
| DAC0H | 0xD3 | 0 | DAC0 High | page 105[*5] |
| DAC0L | 0xD2 | 0 | DAC0 Low | page 105[*5] |
| DAC1CN | 0xD4 | 1 | DAC1 Control | page 108[*5] |
| DAC1H | 0xD3 | 1 | DAC1 High | page 107[*5] |
| DAC1L | 0xD2 | 1 | DAC1 Low | page 107[*5] |
| DMA0BND | 0xFD | 3 | DMA0 Instruction Boundary | page 83 |
| DMA0CF | 0xF8 | 3 | DMA0 Configuration | page 81 |
| DMA0CN | 0xD8 | 3 | DMA0 Control | page 80 |
| DMA0CSH | 0xFC | 3 | DMA0 Repeat Counter Status High Byte | page 85 |
| DMA0CSL | 0xFB | 3 | DMA0 Repeat Counter Status Low Byte | page 85 |
| DMA0CTH | 0xFA | 3 | DMA0 Repeat Counter Limit High Byte | page 85 |
| DMA0CTL | 0xF9 | 3 | DMA0 Repeat Counter Limit Low Byte | page 85 |
| DMA0DAH | 0xDA | 3 | DMA0 Data Address Beginning High Byte | page 84 |
| DMA0DAL | 0xD9 | 3 | DMA0 Data Address Beginning Low Byte | page 84 |
| DMA0DSH | 0xDC | 3 | DMA0 Data Address Pointer High Byte | page 84 |
| DMA0DSL | 0xDB | 3 | DMA0 Data Address Pointer Low Byte | page 84 |
| DMA0IDT | 0xDE | 3 | DMA0 Instruction Write Data | page 82 |
| DMA0IPT | 0xDD | 3 | DMA0 Instruction Write Address | page 82 |
| DMA0ISW | 0xFE | 3 | DMA0 Instruction Status | page 83 |
| DPH | 0x83 | All Pages | Data Pointer High | page 148 |
| DPL | 0x82 | All Pages | Data Pointer Low | page 148 |
| EIE1 | 0xE6 | All Pages | Extended Interrupt Enable 1 | page 156 |
| EIE2 | 0xE7 | All Pages | Extended Interrupt Enable 2 | page 157 |
| EIP1 | 0xF6 | All Pages | Extended Interrupt Priority 1 | page 158 |
| EIP2 | 0xF7 | All Pages | Extended Interrupt Priority 2 | page 159 |
| EMI0CF | 0xA3 | 0 | EMIF Configuration | page 189[*1] |
| EMI0CN | 0xA2 | 0 | EMIF Control | page 189[*1] |
| EMI0TC | 0xA1 | 0 | EMIF Timing Control | page 194[*1] |
| FLACL | 0xB7 | F | Flash Access Limit | page 182 |
| FLSCL | 0xB7 | 0 | Flash Scale | page 184 |
| IE | 0xA8 | All Pages | Interrupt Enable | page 154 |
| IP | 0xB8 | All Pages | Interrupt Priority | page 155 |
| OSCICL | 0x8B | F | Internal Oscillator Calibration | page 172 |
| OSCICN | 0x8A | F | Internal Oscillator Control | page 172 |
| OSCXCN | 0x8C | F | External Oscillator Control | page 174 |
| P0 | 0x80 | All Pages | Port 0 Latch | page 214 |
| P0MDOUT | 0xA4 | F | Port 0 Output Mode Configuration | page 214 |
| P1 | 0x90 | All Pages | Port 1 Latch | page 215 |
| P1MDIN | 0xAD | F | Port 1 Input Mode | page 215 |
| P1MDOUT | 0xA5 | F | Port 1 Output Mode Configuration | page 216 |
| P2 | 0xA0 | All Pages | Port 2 Latch | page 216 |

**Table 13.3. Special Function Registers (Continued)**

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

| Register | Address | SFR Page | Description | Page No. |
|---|---|---|---|---|
| P2MDIN | 0xAE | F | Port 2 Input Mode | page 217 |
| P2MDOUT | 0xA6 | F | Port 2 Output Mode Configuration | page 217 |
| P3 | 0xB0 | All Pages | Port 3 Latch | page 218*1 |
| P3MDOUT | 0xA7 | F | Port 3 Output Mode Configuration | page 218*1 |
| P4 | 0xC8 | F | Port 4 Latch | page 221*1 |
| P4MDOUT | 0x9C | F | Port 4 Output Mode Configuration | page 221*1 |
| P5 | 0xD8 | F | Port 5 Latch | page 222*1 |
| P5MDOUT | 0x9D | F | Port 5 Output Mode Configuration | page 222*1 |
| P6 | 0xE8 | F | Port 6 Latch | page 223*1 |
| P6MDOUT | 0x9E | F | Port 6 Output Mode Configuration | page 223*1 |
| P7 | 0xF8 | F | Port 7 Latch | page 224*1 |
| P7MDOUT | 0x9F | F | Port 7 Output Mode Configuration | page 224*1 |
| PCA0CN | 0xD8 | 0 | PCA Control | page 312 |
| PCA0CPH0 | 0xFC | 0 | PCA Capture 0 High | page 316 |
| PCA0CPH1 | 0xFE | 0 | PCA Capture 1 High | page 316 |
| PCA0CPH2 | 0xEA | 0 | PCA Capture 2 High | page 316 |
| PCA0CPH3 | 0xEC | 0 | PCA Capture 3 High | page 316 |
| PCA0CPH4 | 0xEE | 0 | PCA Capture 4 High | page 316 |
| PCA0CPH5 | 0xE2 | 0 | PCA Capture 5 High | page 316 |
| PCA0CPL0 | 0xFB | 0 | PCA Capture 0 Low | page 316 |
| PCA0CPL1 | 0xFD | 0 | PCA Capture 1 Low | page 316 |
| PCA0CPL2 | 0xE9 | 0 | PCA Capture 2 Low | page 316 |
| PCA0CPL3 | 0xEB | 0 | PCA Capture 3 Low | page 316 |
| PCA0CPL4 | 0xED | 0 | PCA Capture 4 Low | page 316 |
| PCA0CPL5 | 0xE1 | 0 | PCA Capture 5 Low | page 316 |
| PCA0CPM0 | 0xDA | 0 | PCA Module 0 Mode Register | page 314 |
| PCA0CPM1 | 0xDB | 0 | PCA Module 1 Mode Register | page 314 |
| PCA0CPM2 | 0xDC | 0 | PCA Module 2 Mode Register | page 314 |
| PCA0CPM3 | 0xDD | 0 | PCA Module 3 Mode Register | page 314 |
| PCA0CPM4 | 0xDE | 0 | PCA Module 4 Mode Register | page 314 |
| PCA0CPM5 | 0xDF | 0 | PCA Module 5 Mode Register | page 314 |
| PCA0H | 0xFA | 0 | PCA Counter High | page 315 |
| PCA0L | 0xF9 | 0 | PCA Counter Low | page 315 |
| PCA0MD | 0xD9 | 0 | PCA Mode | page 313 |
| PCON | 0x87 | All Pages | Power Control | page 161 |
| PSCTL | 0x8F | 0 | Program Store R/W Control | page 185 |
| PSW | 0xD0 | All Pages | Program Status Word | page 149 |
| RCAP2H | 0xCB | 0 | Timer/Counter 2 Capture/Reload High | page 301 |
| RCAP2L | 0xCA | 0 | Timer/Counter 2 Capture/Reload Low | page 301 |
| RCAP3H | 0xCB | 1 | Timer/Counter 3 Capture/Reload High | page 301 |
| RCAP3L | 0xCA | 1 | Timer/Counter 3 Capture/Reload Low | page 301 |
| RCAP4H | 0xCB | 2 | Timer/Counter 4 Capture/Reload High | page 301 |
| RCAP4L | 0xCA | 2 | Timer/Counter 4 Capture/Reload Low | page 301 |

**Table 13.3. 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 0 | page 62 |
| REF1CN | 0xD1 | 1 | Voltage Reference Control 1 | page 62 |
| REF2CN | 0xD1 | 2 | Voltage Reference Control 2 | page 112[*2], page 114[*3], page 116[*5] |
| RSTSRC | 0xEF | 0 | Reset Source | page 168 |
| SADDR0 | 0xA9 | 0 | UART 0 Slave Address | page 276 |
| SADEN0 | 0xB9 | 0 | UART 0 Slave Address Enable | page 276 |
| SBUF0 | 0x99 | 0 | UART 0 Data Buffer | page 276 |
| SBUF1 | 0x99 | 1 | UART 1 Data Buffer | page 283 |
| SCON0 | 0x98 | 0 | UART 0 Control | page 274 |
| SCON1 | 0x98 | 1 | UART 1 Control | page 282 |
| SFRLAST | 0x86 | All Pages | SFR Page Stack Access Register | page 140 |
| SFRNEXT | 0x85 | All Pages | SFR Page Register | page 140 |
| SFRPAGE | 0x84 | All Pages | SFR Page Register | page 139 |
| SFRPGCN | 0x96 | F | SFR Page Control Register | page 139 |
| SMB0ADR | 0xC3 | 0 | SMBus Slave Address | page 246 |
| SMB0CN | 0xC0 | 0 | SMBus Control | page 243 |
| SMB0CR | 0xCF | 0 | SMBus Clock Rate | page 244 |
| SMB0DAT | 0xC2 | 0 | SMBus Data | page 245 |
| SMB0STA | 0xC1 | 0 | SMBus Status | page 247 |
| SP | 0x81 | All Pages | Stack Pointer | page 148 |
| SPI0CFG | 0x9A | 0 | SPI Configuration | page 258 |
| SPI0CKR | 0x9D | 0 | SPI Clock Rate Control | page 260 |
| SPI0CN | 0xF8 | 0 | SPI Control | page 259 |
| SPI0DAT | 0x9B | 0 | SPI Data | page 261 |
| SSTA0 | 0x91 | 0 | UART 0 Status | page 275 |
| TCON | 0x88 | 0 | Timer/Counter Control | page 291 |
| TH0 | 0x8C | 0 | Timer/Counter 0 High | page 294 |
| TH1 | 0x8D | 0 | Timer/Counter 1 High | page 294 |
| TL0 | 0x8A | 0 | Timer/Counter 0 Low | page 294 |
| TL1 | 0x8B | 0 | Timer/Counter 1 Low | page 294 |
| TMOD | 0x89 | 0 | Timer/Counter Mode | page 292 |
| TMR2CF | 0xC9 | 0 | Timer/Counter 2 Configuration | page 300 |
| TMR2CN | 0xC8 | 0 | Timer/Counter 2 Control | page 299 |
| TMR2H | 0xCD | 0 | Timer/Counter 2 High | page 302 |
| TMR2L | 0xCC | 0 | Timer/Counter 2 Low | page 301 |
| TMR3CF | 0xC9 | 1 | Timer/Counter 3 Configuration | page 300 |
| TMR3CN | 0xC8 | 1 | Timer/Counter 3 Control | page 299 |
| TMR3H | 0xCD | 1 | Timer/Counter 3 High | page 302 |
| TMR3L | 0xCC | 1 | Timer/Counter 3 Low | page 301 |
| TMR4CF | 0xC9 | 2 | Timer/Counter 4 Configuration | page 300 |
| TMR4CN | 0xC8 | 2 | Timer/Counter 4 Control | page 299 |
| TMR4H | 0xCD | 2 | Timer/Counter 4 High | page 302 |

**Table 13.3. 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 | page 301 |
| WDTCN | 0xFF | All Pages | Watchdog Timer Control | page 167 |
| XBR0 | 0xE1 | F | Port I/O Crossbar Control 0 | page 210 |
| XBR1 | 0xE2 | F | Port I/O Crossbar Control 1 | page 211 |
| XBR2 | 0xE3 | F | Port I/O Crossbar Control 2 | page 212 |
| XBR3 | 0xE4 | F | Port I/O Crossbar Control 3 | page 213 |

*1 Refers to a register in the C8051F060/2/4/6 only.

*2 Refers to a register in the C8051F060/2 only.

*3 Refers to a register in the C8051F061/3 only.

*4 Refers to a register in the C8051F060/1/2/3 only.

*5 Refers to a register in the C8051F064/5/6/7 only.

### 13.2.7. 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.

**Figure 13.13. 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.

**Figure 13.14. 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.

**Figure 13.15. 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.

**Figure 13.16. PSW: Program Status Word**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 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.

**Figure 13.17. 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.

**Figure 13.18. 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.

## 13.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 22 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs 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-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.

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.

### 13.3.1. MCU Interrupt Sources and Vectors

The MCUs support 22 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 13.4. 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).

### 13.3.2. External Interrupts

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 13.4. Interrupt Summary**

| Interrupt Source | Interrupt Vector | Priority Order | Pending Flag | Bit addressable | Cleared by HW | Enable Flag | Priority Control |
|---|---|---|---|---|---|---|---|
| Reset | 0x0000 | Top | None | N/A | N/A | Always Enabled | Always Highest |
| External Interrupt 0 (/INT0) | 0x0003 | 0 | IE0 (TCON.1) | Y | Y | EX0 (IE.0) | PX0 (IP.0) |
| Timer 0 Overflow | 0x000B | 1 | TF0 (TCON.5) | Y | Y | ET0 (IE.1) | PT0 (IP.1) |
| External Interrupt 1 (/INT1) | 0x0013 | 2 | IE1 (TCON.3) | Y | Y | EX1 (IE.2) | PX1 (IP.2) |
| Timer 1 Overflow | 0x001B | 3 | TF1 (TCON.7) | Y | Y | ET1 (IE.3) | PT1 (IP.3) |
| UART0 | 0x0023 | 4 | RI0 (SCON0.0)<br>TI0 (SCON0.1) | Y | | ES0 (IE.4) | PS0 (IP.4) |
| Timer 2 | 0x002B | 5 | TF2 (TMR2CN.7) | Y | | 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 | | ESPI0 (EIE1.0) | PSPI0 (EIP1.0) |
| SMBus Interface | 0x003B | 7 | SI (SMB0CN.3) | Y | | ESMB0 (EIE1.1) | PSMB0 (EIP1.1) |
| ADC0 Window Comparator | 0x0043 | 8 | AD0WINT (ADC0CN.1) | Y | | EWADC0 (EIE1.2) | PWADC0 (EIP1.2) |
| Programmable Counter Array | 0x004B | 9 | CF (PCA0CN.7)<br>CCFn (PCA0CN.n) | Y | | EPCA0 (EIE1.3) | PPCA0 (EIP1.3) |
| Comparator 0 | 0x0053 | 10 | CP0FIF/CP0RIF (CPT0CN.4/.5) | Y | | CP0IE (EIE1.4) | PCP0 (EIP1.4) |
| Comparator 1 | 0x005B | 11 | CP1FIF/CP1RIF (CPT1CN.4/.5) | Y | | CP1IE (EIE1.5) | PCP1 (EIP1.5) |
| Comparator 2 | 0x0063 | 12 | CP2FIF/CP2RIF (CPT2CN.4/.5) | Y | | CP2IE (EIE1.6) | PCP2 (EIP1.6) |
| ADC0 End of Conversion | 0x006B | 13 | ADC0INT (ADC0CN.5) | Y | | EADC0 (EIE1.7) | PADC0 (EIP1.7) |
| Timer 3 | 0x0073 | 14 | TF3 (TMR3CN.7) | Y | | ET3 (EIE2.0) | PT3 (EIP2.0) |
| ADC1 End of Conversion | 0x007B | 15 | ADC1INT (ADC1CN.5) | Y | | EADC1 (EIE2.1) | PADC1 (EIP2.1) |
| Timer 4 | 0x0083 | 16 | TF4 (TMR4CN.7) | Y | | ET4 (EIE2.2) | PT4 (EIP2.2) |
| ADC2 Window Comparator | 0x008B | 17 | AD2WINT (ADC2CN.1) | Y | | EWADC2 (EIE2.3) | PWADC2 (EIP2.3) |
| ADC2 End of Conversion | 0x0093 | 18 | AD2INT (ADC2CN.5) | Y | | EADC2 (EIE2.4) | PADC2 (EIP2.4) |
| CAN Interrupt | 0x009B | 19 | CAN0CN.7 | Y | Y | ECAN0 (EIE2.5) | PCAN0 (EIP2.5) |
| UART1 | 0x00A3 | 20 | RI1 (SCON1.0)<br>TI1 (SCON1.1) | Y | | ES1 (EIP2.6) | PS1 (EIP2.6) |
| DMA0 Interrupt | 0x00AB | 21 | DMA0INT (DMA0CN.6) | Y | | EDMA0 (EIE2.7) | PDMA0 (EIP2.7) |

### 13.3.3. 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 13.4.

### 13.3.4. 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. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

### 13.3.5. 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).

**Figure 13.19. 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 interrupt requests generated by the TF2 flag.

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 all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit2: EX1: Enable External Interrupt 1.
This bit sets the masking of external interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the /INT1 pin.

Bit1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit0: EX0: Enable External Interrupt 0.
This bit sets the masking of external interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the /INT0 pin.

**Figure 13.20. 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 level.
1: Timer 2 interrupt set to high priority level.
Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupt set to high priority level.
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 level.
1: Timer 1 interrupt set to high priority level.
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 level.
1: External Interrupt 1 set to high priority level.
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 level.
1: Timer 0 interrupt set to high priority level.
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 level.
1: External Interrupt 0 set to high priority level.

**Figure 13.21. EIE1: Extended Interrupt Enable 1**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| EADC0 | CP2IE | CP1IE | CP0IE | EPCA0 | EWADC0 | ESMB0 | ESPI0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE6
SFR Page: All Pages

Bit7: EADC0: Enable ADC0 End of Conversion Interrupt.
This bit sets the masking of the ADC0 End of Conversion Interrupt.
0: Disable ADC0 Conversion Interrupt.
1: Enable interrupt requests generated by the ADC1 Conversion Interrupt.

Bit6: CP2IE: Enable Comparator (CP2) Interrupt.
This bit sets the masking of the CP2 interrupt.
0: Disable CP2 interrupts.
1: Enable interrupt requests generated by the CP2IF flag.

Bit6: CP1IE: Enable Comparator (CP1) Interrupt.
This bit sets the masking of the CP1 interrupt.
0: Disable CP1 interrupts.
1: Enable interrupt requests generated by the CP1IF flag.

Bit6: CP0IE: Enable Comparator (CP0) Interrupt.
This bit sets the masking of the CP0 interrupt.
0: Disable CP0 interrupts.
1: Enable interrupt requests generated by the CP0IF flag.

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

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 Interrupt requests generated by ADC0 Window Comparisons.

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

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

**Figure 13.22. EIE2: Extended Interrupt Enable 2**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| EDMA0 | ES1 | ECAN0 | EADC2 | EWADC2 | ET4 | EADC1 | ET3 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE7
SFR Page: All Pages

Bit7: EDMA0: Enable DMA0 Interrupt.
This bit sets the masking of the DMA0 Interrupt.
0: Disable DMA0 interrupt.
1: Enable DMA0 interrupt.

Bit6: ES1: Enable UART1 Interrupt.
This bit sets the masking of the UART1 Interrupt.
0: Disable UART1 interrupt.
1: Enable UART1 interrupt.

Bit5: ECAN0: Enable CAN Controller Interrupt.
This bit sets the masking of the CAN Controller Interrupt.
0: Disable CAN Controller Interrupt.
1: Enable interrupt requests generated by the CAN Controller.

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 interrupt.
1: Enable interrupt requests generated by the ADC2 End of Conversion Interrupt.

Bit3: EWADC2: Enable Window Comparison ADC1 Interrupt.
This bit sets the masking of ADC2 Window Comparison interrupt.
0: Disable ADC2 Window Comparison Interrupt.
1: Enable Interrupt requests generated by ADC2 Window Comparisons.

Bit2: ET4: Enable Timer 4 Interrupt
This bit sets the masking of the Timer 4 interrupt.
0: Disable Timer 4 interrupt.
1: Enable interrupt requests generated by the TF4 flag.

Bit1: EADC1: Enable ADC1 End of Conversion Interrupt.
This bit sets the masking of the ADC1 End of Conversion Interrupt.
0: Disable ADC1 Conversion Interrupt.
1: Enable interrupt requests generated by the ADC1 Conversion Interrupt.

Bit0: ET3: Enable Timer 3 Interrupt.
This bit sets the masking of the Timer 3 interrupt.
0: Disable all Timer 3 interrupts.
1: Enable interrupt requests generated by the TF3 flag.

**Figure 13.23. EIP1: Extended Interrupt Priority 1**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PADC0 | PCP2 | PCP1 | PCP0 | PPCA0 | PWADC0 | PSMB0 | PSPI0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xF6
SFR Page: All Pages

Bit7: PADC0: ADC 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 level.
1: ADC0 End of Conversion interrupt set to high priority level.

Bit6: PCP2: Comparator2 (CP2) Interrupt Priority Control.
This bit sets the priority of the CP2 interrupt.
0: CP2 interrupt set to low priority level.
1: CP2 interrupt set to high priority level.

Bit5: PCP1: Comparator1 (CP1) Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 interrupt set to low priority level.
1: CP1 interrupt set to high priority level.

Bit4: PCP0: Comparator0 (CP0) Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 interrupt set to low priority level.
1: CP0 interrupt set to high priority level.

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 level.
1: PCA0 interrupt set to high priority level.

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

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 level.
1: SMBus interrupt set to high priority level.

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 level.
1: SPI0 interrupt set to high priority level.

**Figure 13.24. EIP2: Extended Interrupt Priority 2**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PDMA0 | PS1 | PCAN0 | PADC2 | PWADC2 | PT4 | PADC1 | PT3 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xF7
SFR Page: All Pages

Bit7: PDMA0: DMA0 Interrupt Priority Control.
This bit sets the priority of the DMA0 interrupt.
0: DMA0 interrupt set to low priority.
1: DMA0 interrupt set to high priority.

Bit6: PS1: 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: PCAN0: CAN Interrupt Priority Control.
This bit sets the priority of the CAN Interrupt.
0: CAN Interrupt set to low priority level.
1: CAN Interrupt set to high priority level.

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 Comparator Interrupt Priority Control.
0: ADC2 Window interrupt set to low priority.
1: ADC2 Window 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: PADC1: ADC End of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC1 End of Conversion Interrupt.
0: ADC1 End of Conversion interrupt set to low priority level.
1: ADC1 End of Conversion interrupt set to high priority level.

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 level.
1: Timer 3 interrupt set to high priority level.

## 13.4. 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 internal oscillator 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. Figure 13.25 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 oscillator saves even more power, but requires a reset to restart the MCU.

### 13.4.1. 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 /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 0x0000.

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 14.7 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.

### 13.4.2. 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 internal 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 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to sleep for longer than the MCD timeout of 100 µs.

**Figure 13.25. 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

Bits7-2: 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 internal 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.)
**See Note in Section “**13.4.1. Idle Mode” on page 160**.**

# 14. 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.

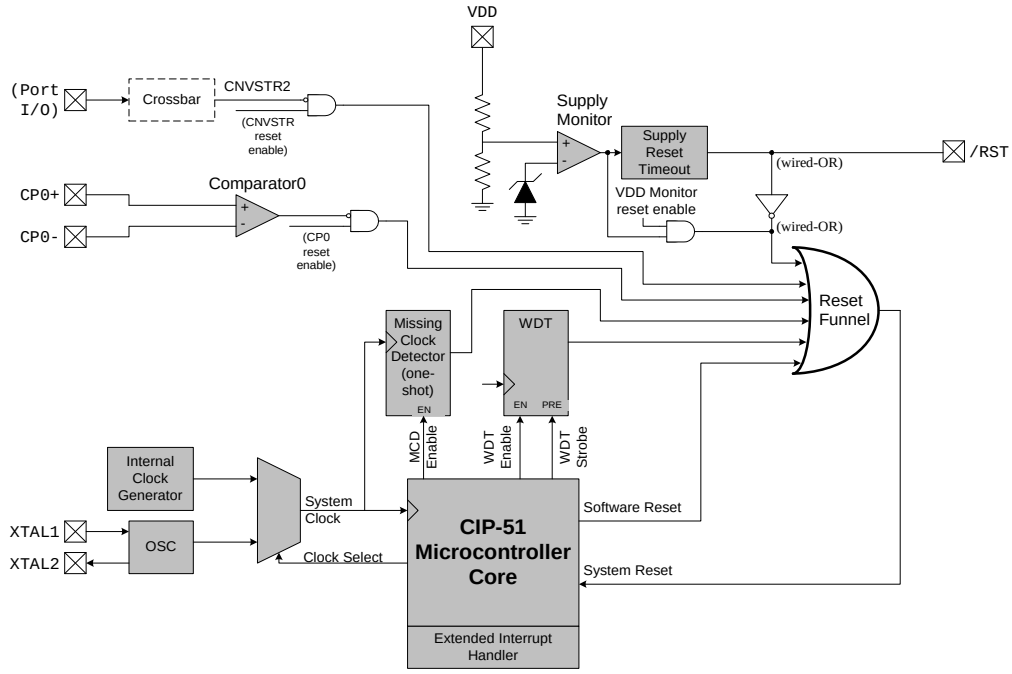
All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost even though the data on the stack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

The I/O port latches are reset to 0xFF (all logic 1's), activating internal weak pull-ups which take the external I/O pins to a high state. The external I/O pins do not go high immediately, but will go high within four system clock cycles after entering the reset state. This allows power to be conserved while the part is held in reset. For VDD Monitor resets, the /RST pin is driven low until the end of the VDD 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 "15. Oscillators" on page 171 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled using its longest timeout interval (see Section "14.7. Watchdog Timer Reset" on page 165). 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 /RST pin, external CNVSTR2 signal, software command, Comparator0, Missing Clock Detector, and Watchdog Timer. Each reset source is described in the following sections.

**Figure 14.1. Reset Sources**

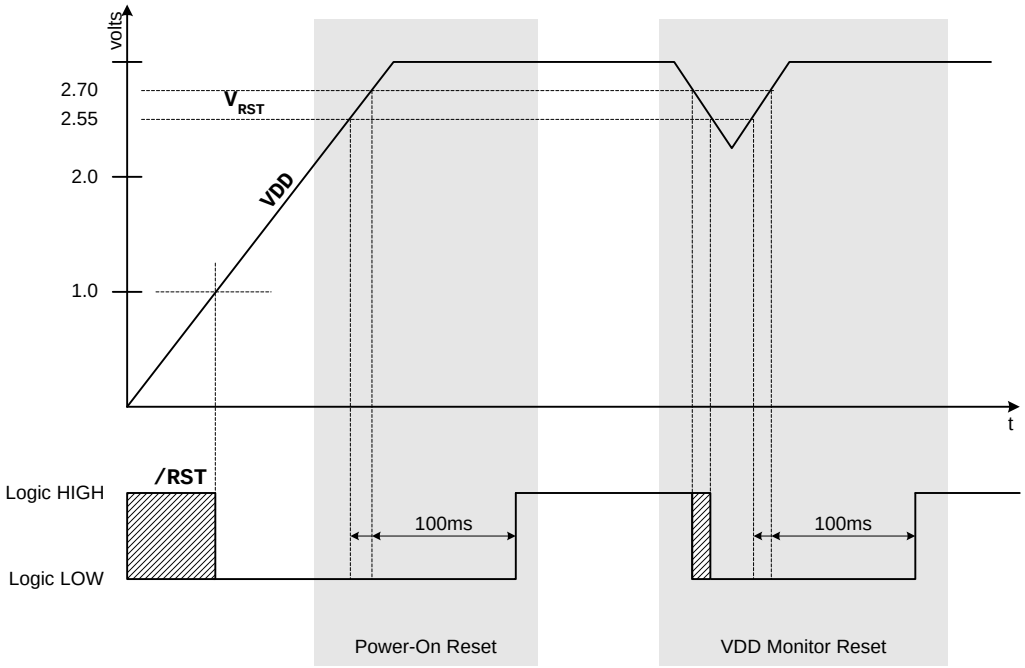


## 14.1. Power-on Reset

The C8051F060/1/2/3/4/5/6/7 family incorporates a power supply monitor that holds the MCU in the reset state until VDD rises above the $V_{RST}$ level during power-up. See Figure 14.2 for timing diagram, and refer to Table 14.1 for the Electrical Characteristics of the power supply monitor circuit. The /RST pin is asserted low until the end of the 100 ms VDD Monitor timeout in order to allow the VDD supply to stabilize. The VDD Monitor reset is enabled and disabled using the external VDD monitor enable pin (MONEN).

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 14.2. Reset Timing**




## 14.2. Power-fail Reset

When a power-down transition or power irregularity causes VDD to drop below $V_{RST}$, the power supply monitor will drive the /RST pin low and return the CIP-51 to the reset state. When VDD 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 14.2). Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if VDD dropped below the level required for data retention. If the PORSF flag is set to logic 1, the data may no longer be valid.

## 14.3. 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.

## 14.4. 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 “15. Oscillators” on page 171) enables the Missing Clock Detector.

## 14.5. 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 “12. Comparators” on page 117) 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.

## 14.6. External CNVSTR2 Pin Reset

The external CNVSTR2 signal can be configured as a reset input by writing a ‘1’ to the CNVRSEF flag (RSTSRC.6). The CNVSTR2 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 205. Note that the Crossbar must be configured for the CNVSTR2 signal to be routed to the appropriate Port I/O. The Crossbar should be configured and enabled before the CNVRSEF is set. CNVSTR2 cannot be used to start ADC2 conversions when it is configured as a reset source. When configured as a reset, CNVSTR2 is active-low and level sensitive. After a CNVSTR2 reset, the CNVRSEF flag (RSTSRC.6) will read ‘1’ signifying CNVSTR2 as the reset source; otherwise, this bit reads ‘0’. The state of the /RST pin is unaffected by this reset.

## 14.7. 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 Figure 14.3.

### 14.7.1. 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.

### 14.7.2. 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. Interrupts should be disabled during this procedure to avoid delay between the two writes.

### 14.7.3. 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.

### 14.7.4. 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.

**Figure 14.3. WDTCN: Watchdog Timer Control Register**

| 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.

**Figure 14.4. RSTSRC: Reset Source Register**

| 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 Reset Source Enable and Flag
Write: 0: CNVSTR2 is not a reset source.
1: CNVSTR2 is a reset source (active low).
Read: 0: Source of prior reset was not CNVSTR2.
1: Source of prior reset was CNVSTR2.

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. /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 VDD 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 VDD monitor as a reset source.
0: De-select the VDD monitor as a reset source.
1: Select the VDD monitor as a reset source.
**Important: At power-on, the VDD monitor is enabled/disabled using the external VDD monitor enable pin (MONEN). The PORSF bit does not disable or enable the VDD monitor circuit. It simply selects the VDD 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 VDD 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 VDD monitor reset.
1: Source of last reset was a power-on or VDD 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. /RST is driven low.
Read: 0: Source of prior reset was not /RST pin.
1: Source of prior reset was /RST pin.

**Table 14.1. Reset Electrical Characteristics**

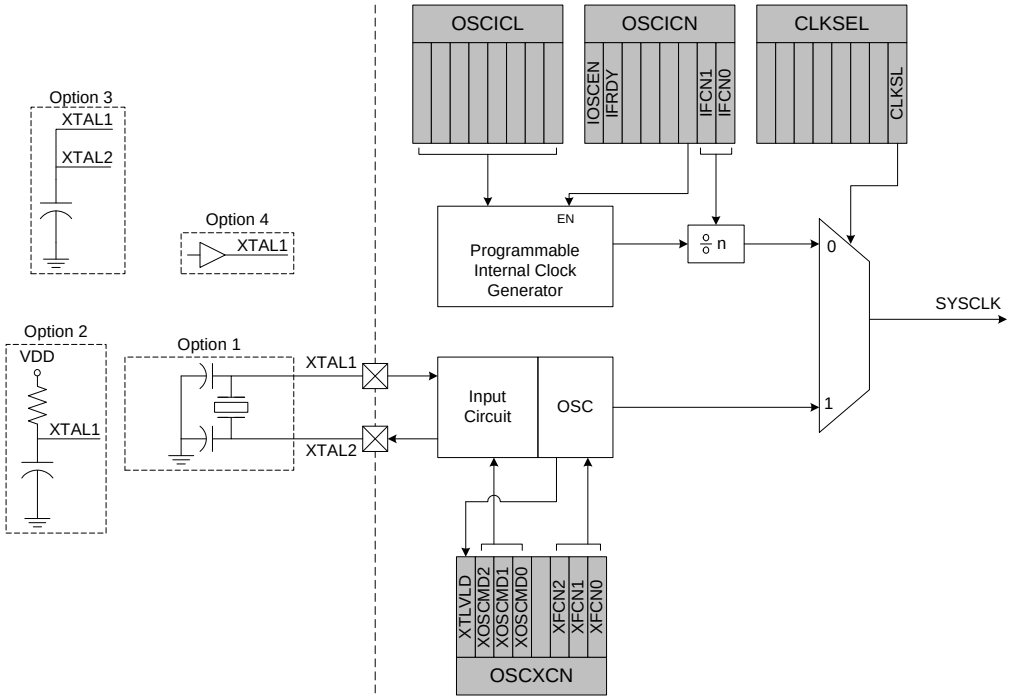
**-40 to +85 °C unless otherwise specified.**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| /RST Output Low Voltage | $I_{OL}$ = 8.5 mA, VDD = 2.7 V to 3.6 V | | | 0.6 | V |
| /RST Input High Voltage | | 0.7 x VDD | | | V |
| /RST Input Low Voltage | | | | 0.3 x VDD | |
| /RST Input Leakage Current | /RST = 0.0 V | | 50 | | µA |
| VDD for /RST Output Valid | | 1.0 | | | V |
| AV+ for /RST Output Valid | | 1.0 | | | V |
| VDD POR Threshold ($V_{RST}$) | | 2.40 | 2.55 | 2.70 | V |
| Minimum /RST Low Time to Generate a System Reset | | 10 | | | ns |
| Reset Time Delay | /RST rising edge after VDD crosses $V_{RST}$ threshold | 80 | 100 | 120 | ms |
| Missing Clock Detector Time-out | Time from last system clock to reset initiation | 100 | 220 | 500 | µs |

# 15. Oscillators

C8051F060/1/2/3/4/5/6/7 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 15.1. The system clock can be sourced by the external oscillator circuit, the internal oscillator, or a scaled version of the internal oscillator. The internal oscillator's electrical specifications are given in Table 15.1.

**Figure 15.1. Oscillator Diagram**




## 15.1. Programmable Internal Oscillator

All C8051F060/1/2/3/4/5/6/7 devices include a programmable 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 Figure 15.2.

OSCICL is factory calibrated to obtain a 24.5 MHz base frequency ($f_{BASE}$).

Electrical specifications for the precision internal oscillator are given in Table 15.1. The programmed internal oscillator frequency must not exceed 25 MHz. 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.

.

**Figure 15.2. OSCICL: Internal Oscillator Calibration Register**

| 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.

**Figure 15.3. OSCICN: Internal Oscillator Control Register**

| 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

Bit7: IOSCEN: Internal Oscillator Enable Bit
0: Internal Oscillator Disabled
1: Internal Oscillator Enabled
Bit6: IFRDY: Internal Oscillator Frequency Ready Flag
0: Internal Oscillator not running at programmed frequency.
1: Internal Oscillator running at programmed frequency.
Bits5-2: Reserved.
Bits1-0: IFCN1-0: Internal Oscillator Frequency Control Bits
00: SYSCLK derived from Internal Oscillator divided by 8.
01: SYSCLK derived from Internal Oscillator divided by 4.
10: SYSCLK derived from Internal Oscillator divided by 2.
11: SYSCLK derived from Internal Oscillator divided by 1.

**Table 15.1. Internal Oscillator Electrical Characteristics**

**-40°C to +85°C unles otherwise specified.**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Calibrated Internal Oscillator Frequency | | 24 | 24.5 | 25 | MHz |
| Internal Oscillator Supply Current (3.0V Supply) | OSCICN.7 = 1 | | 550 | | µA |

## 15.2. 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 15.1. 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 15.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see Figure 15.5).

## 15.3. System Clock Selection

The CLKSL bit in register CLKSEL selects which oscillator generates the system clock. CLKSL must be set to ‘1’ for the system clock to run from the external oscillator; however the external oscillator may still clock peripherals (timers, PCA) when the internal oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal and external oscillator, so long as the selected oscillator 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.

**Figure 15.4. CLKSEL: Oscillator Clock Selection Register**

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

SFR Address: 0x97
SFR Page: F

Bits7-1: Reserved.
Bit0: CLKSL: System Clock Source Select Bit.
0: SYSCLK derived from the Internal Oscillator, and scaled as per the IFCN bits in OSCICN.
1: SYSCLK derived from the External Oscillator circuit.

**Figure 15.5. OSCXCN: External Oscillator Control Register**

| 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: Unused. 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 15.1, Option 1; XOSCMD = 11x).
Choose XFCN value to match crystal frequency.

**RC MODE** (Circuit from Figure 15.1, 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 = Pull-up resistor value in kΩ

**C MODE** (Circuit from Figure 15.1, Option 3; XOSCMD = 10x).
Choose K Factor (KF) for the oscillation frequency desired:
**f = KF / (C * VDD)**, where
f = frequency of oscillation in MHz
C = capacitor value on XTAL1, XTAL2 pins in pF
VDD = Power Supply on MCU in volts

## 15.4. 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 15.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in Figure 15.5 (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. Introducing a blanking interval of 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 least1 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 and external noise. 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. Crystal loading capacitors should be referenced to AGND.

## 15.5. 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 15.1, 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 * 50 ] = 0.1 MHz = 100 kHz
Referring to the table in Figure 15.5, the required XFCN setting is 010.

## 15.6. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 15.1, 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 VDD = 3.0 V and C = 50 pF:

f = KF / ( C * VDD ) = KF / ( 50 * 3 )
f = KF / 150

If a frequency of roughly 50 kHz is desired, select the K Factor from the table in Figure 15.5 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.

# 16. Flash Memory

The C8051F060/1/2/3/4/5/6/7 devices include on-chip, reprogrammable Flash memory for program code and non-volatile data storage. The C8051F060/1/2/3/4/5 include 64 k + 128 bytes of Flash, and the C8051F066/7 include 32 k + 128 bytes of Flash. The Flash memory can be programmed in-system, a single byte at a time, 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. The bytes would typically be erased (set to 0xFF) before being reprogrammed. Flash write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. The CPU is stalled during write/erase operations while the device peripherals remain active. Interrupts that occur during Flash write/erase operations are held, and are then serviced in their priority order once the Flash operation has completed. Refer to Table 16.1 for the electrical characteristics of the Flash memory.

## 16.1. 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 “26. JTAG (IEEE 1149.1)” on page 317.

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.

**NOTE: To ensure the integrity of Flash memory contents, it is strongly recommended that the on-chip VDD monitor be enabled by connecting the VDD monitor enable pin (MONEN) to VDD 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 163 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**. The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). The following steps illustrate the algorithm for programming Flash from user software.

Step 1. Disable interrupts.
Step 2. Set FLWE (FLSCL.0) to enable Flash writes/erases via user software.
Step 3. Set PSEE (PSCTL.1) to enable Flash erases.
Step 4. Set PSWE (PSCTL.0) to redirect MOVX commands to write to Flash.
Step 5. Use the MOVX command to write a data byte to any location within the 512-byte page to be erased.
Step 6. Clear PSEE to disable Flash erases
Step 7. Use the MOVX command to write a data byte to the desired byte location within the erased 512-byte page. Repeat this step until all desired bytes are written (within the target page).

Step 8. Clear the PSWE bit to redirect MOVX write commands to the XRAM data space.

Step 9. Re-enable interrupts.

Write/Erase timing is automatically controlled by hardware. Note that code execution in the 8051 is stalled while the Flash is being programmed or erased.

**Table 16.1. Flash Electrical Characteristics**

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Flash Size * | C8051F060/1/2/3/4/5 | | 65664 † | | Bytes |
| Flash Size * | C8051F066/7 | | 32896 | | Bytes |
| Endurance | | 20 k | 100 k | | Erase/Write |
| Erase Cycle Time | | 10 | 12 | 14 | ms |
| Write Cycle Time | | 40 | 50 | 60 | µs |

* Includes 128-byte Scratch Pad Area

† 1024 Bytes at location 0xFC00 to 0xFFFF are reserved.

## 16.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction (as described in the previous section) and read using the MOVC instruction.

An additional 128-byte sector of Flash memory is included for non-volatile data storage. Its smaller sector size makes it particularly well suited as general purpose, non-volatile scratchpad 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 sector is double-mapped over the normal Flash memory area; its address ranges from 0x00 to 0x7F (see Figure 16.1 and Figure 16.2). To access this 128-byte sector, the SFLE bit in PSCTL must be set to logic 1. Code execution from this 128-byte scratchpad sector is not supported.

## 16.3. 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) and the Program Store Erase Enable (PSCTL.1) 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 8k-byte block of memory. Clearing a bit to logic 0 in a 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 C8051F060/1/2/3/4/5, the security lock bytes are located at 0xFBFE (Write/Erase Lock) and 0xFBFF (Read Lock), as shown in Figure 16.1. On the C8051F066/7, the security lock bytes are located at 0x7FFE (Write/Erase Lock) and 0x7FFF (Read Lock), as shown in Figure 16.2. The 512-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.

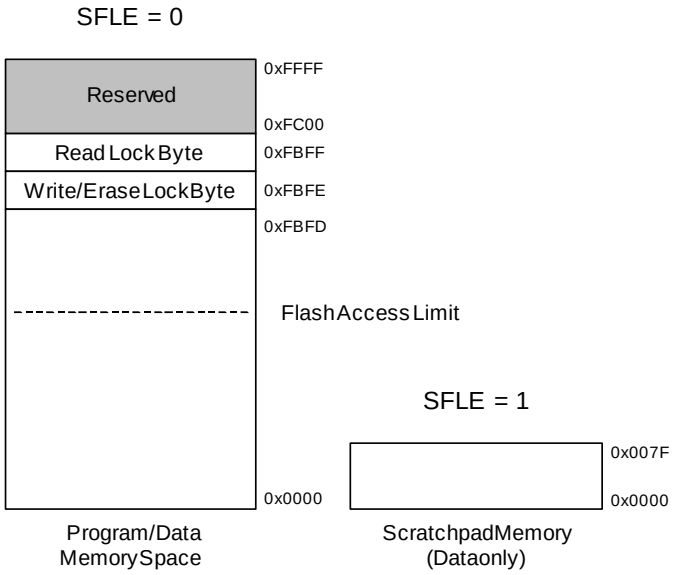
**Important Note: To ensure protection from external access, the block containing the lock bytes must be Write/Erase locked. On the 64 k byte devices (C8051F060/1/2/3/4/5), the page containing the security bytes is 0xFA00-0xFBFF, and is locked by clearing bit 7 of the Write/Erase Lock Byte. On the 32 k byte devices (C8051F066/7), the page containing the security bytes is 0x7E00-0x7FFF, 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.**

**Figure 16.1. C8051F060/1/2/3/4/5 Flash Program Memory Map and Security Bytes**

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

| Bit | Memory Block |
|---|---|
| 7 | 0xE000-0xFBFD |
| 6 | 0xC000-0xDFFF |
| 5 | 0xA000-0xBFFF |
| 4 | 0x8000-0x9FFF |
| 3 | 0x6000-0x7FFF |
| 2 | 0x4000-0x5FFF |
| 1 | 0x2000-0x3FFF |
| 0 | 0x0000-0x1FFF |




**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 block containing the security bytes is locked, the security bytes may be written but not erased.

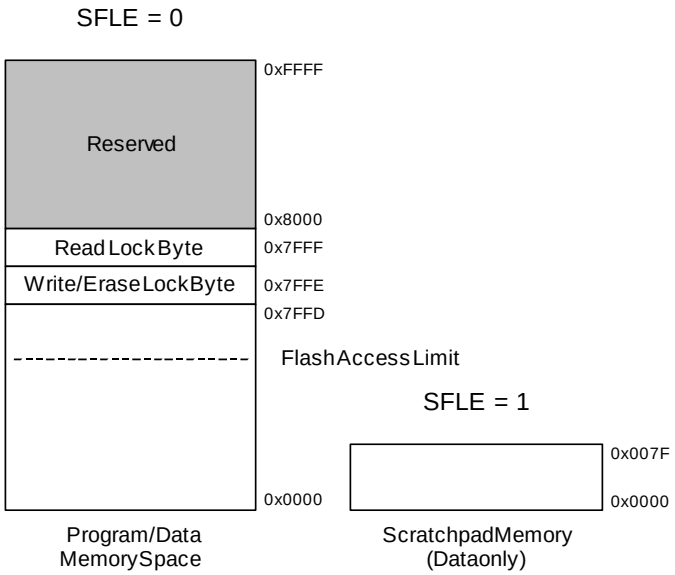
**Flash Access Limit**

The Flash Access Limit is defined by the setting of the FLACL register, as described in Figure 16.3. 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 16.2. C8051F066/7 Flash Program 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 | 0x6000-0x7FFD |
| 2 | 0x4000-0x5FFF |
| 1 | 0x2000-0x3FFF |
| 0 | 0x0000-0x1FFF |




**Flash Read Lock Byte**

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

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 block containing the security bytes 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 Figure 16.3. 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.

The Flash Access Limit security feature (see Figure 16.3) protects proprietary program code and data from being read by software running on the C8051F060/1/2/3/4/5/6/7. 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 16-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 start-

ing at 0x0000 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 always return a data value of 0x00.) 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 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 16-bit FAL address is calculated as 0xNN00, where NN is the contents of the FAL Security Register. Thus, the FAL can be located on 256-byte boundaries anywhere in program memory space. However, the 512-byte erase sector size essentially requires that a 512 boundary be used. The contents of a non-initialized FAL security byte is 0x00, thereby setting the FAL address to 0x0000 and allowing read access to all locations in program memory space by default.

**Figure 16.3. 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 high byte of the 16-bit program memory read/write/erase limit address. The entire 16-bit access limit address value is calculated as 0xNN00 where NN is replaced by contents of FLACL. 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 512-byte Flash page boundary**.

### 16.3.1. Summary of Flash Security Options

There are three Flash access methods supported on the C8051F060/1/2/3/4/5/6/7; 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.

**Figure 16.4. FLSCL: Flash Memory Control**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| FOSE | FRAE | Reserved | Reserved | Reserved | Reserved | Reserved | FLWE | 10000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | SFR Address: |

SFR Address: 0xB7
SFR Page: 0

Bit 7: FOSE: Flash One-Shot Timer Enable
This is the timer that turns off the sense amps after a Flash read.
0: Flash One-Shot Timer disabled.
1: Flash One-Shot Timer enabled (recommended setting.)

Bit 6: FRAE: Flash Read Always Enable
0: Flash reads occur as necessary (recommended setting.).
1: Flash reads occur every system clock cycle.

Bits 5-1: RESERVED. Read = 00000b. Must Write 00000b.

Bit 0: FLWE: Flash Write/Erase Enable
This bit must be set to allow Flash writes/erases from user software.
0: Flash writes/erases disabled.
1: Flash writes/erases enabled.

**Figure 16.5. 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 128-byte Scratchpad Flash sector. When SFLE is set to logic 1, Flash accesses out of the address range 0x00-0x7F should not be attempted. 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 Scratchpad sector.

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.

# 17. External Data Memory Interface and On-Chip XRAM

The C8051F060/1/2/3/4/5/6/7 MCUs include 4 k bytes of on-chip RAM mapped into the external data memory space (XRAM). In addition, the C8051F060/2/4/6 include 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 Figure 17.1). Note: the MOVX instruction can also be used for writing to the Flash memory. See Section “16. Flash Memory” on page 177 for details. The MOVX instruction accesses XRAM by default.

## 17.1. Accessing XRAM

The XRAM memory space (both internal and external) 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 or written. 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 four steps:

1. Enable the EMIF on the 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 four steps is explained in detail in the following sections. The Port enable bit, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in Figure 17.2.

## 17.3. Port Selection and Configuration

When enabled, the External Memory Interface appears on Ports 7, 6, 5, and 4 in non-multiplexed mode, or Ports 7, 6, and 4 in multiplexed mode.

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. See Section “18. Port Input/Output” on page 203 for more information about the Port operation and configuration. **The Port latches should be explicitly configured to ‘park’ the External Memory Interface pins in a dormant state when not in use, 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. See Section “18. Port Input/Output” on page 203 for more information about Port output mode configuration.

**Figure 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

**Figure 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 not mapped to port pins.
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 4 kB boundary are directed on-chip. Accesses above the 4 kB 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 4 kB boundary are directed on-chip. Accesses above the 4 kB 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.

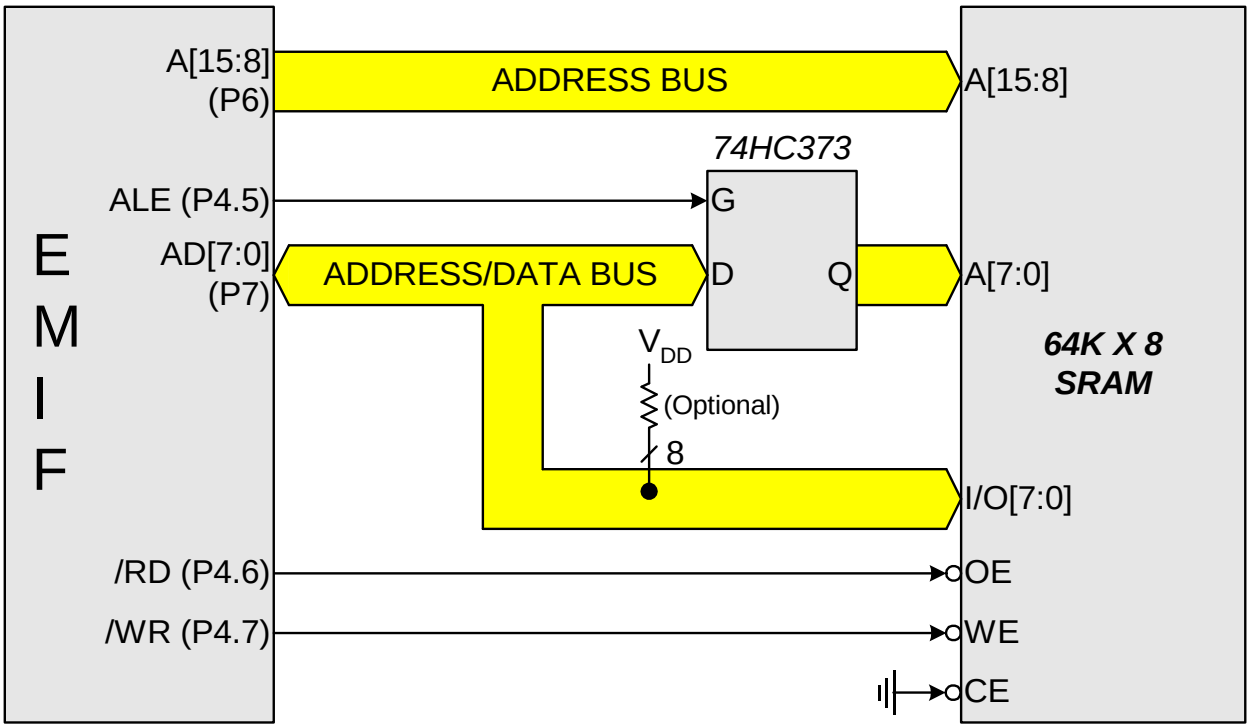
### 17.4.1. Multiplexed Configuration

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 17.3.

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 199 for more information.

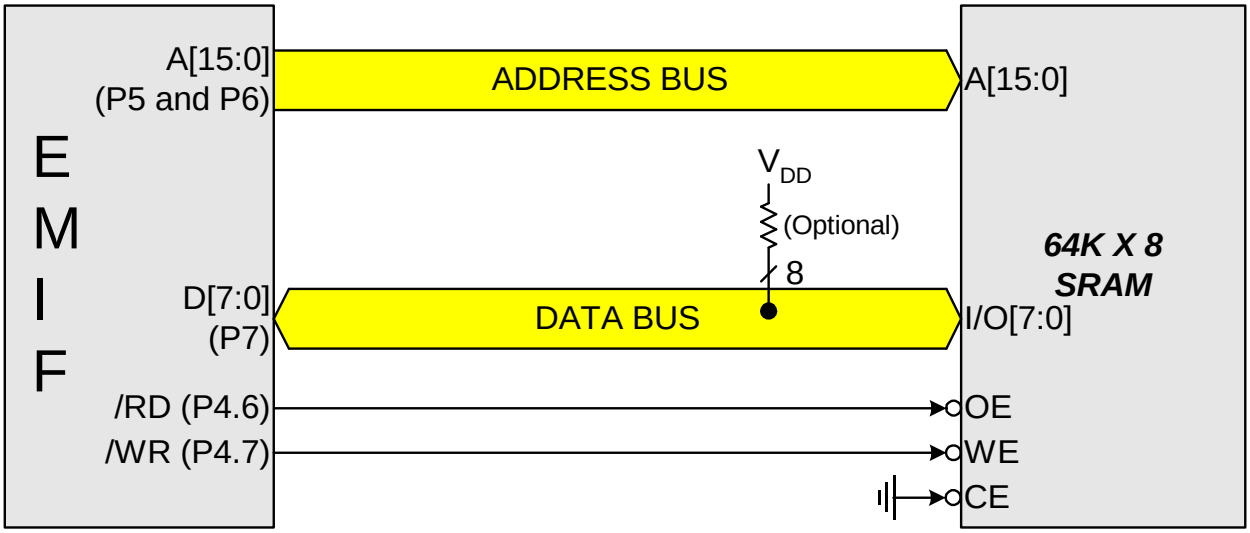
**Figure 17.3. Multiplexed Configuration Example**



### 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 17.4. See Section “17.6.1. Non-multiplexed Mode” on page 196 for more information about Non-multiplexed operation.

**Figure 17.4. 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 17.5, based on the EMIF Mode bits in the EMI0CF register (Figure 17.2). These modes are summarized below. More information about the different modes can be found in Section “17.6. Timing” on page 194.

### 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 4 k byte boundaries. As an example, the addresses 0x1000 and 0x2000 both evaluate to address 0x0000 in on-chip XRAM space.

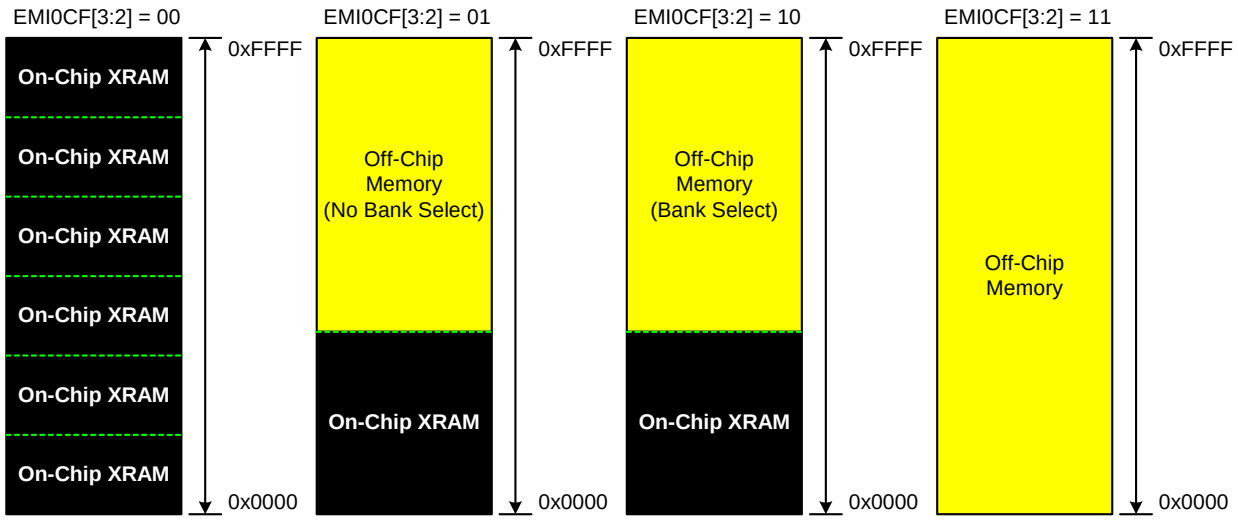
- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

### 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 4 kB boundary will access on-chip XRAM space.
- Effective addresses beyond the 4 kB 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. 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 17.5. 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 4 kB boundary will access on-chip XRAM space.
- Effective addresses beyond the 4 kB 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 4 kB 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. 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 Figure 17.6, 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.

**Figure 17.6. 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.

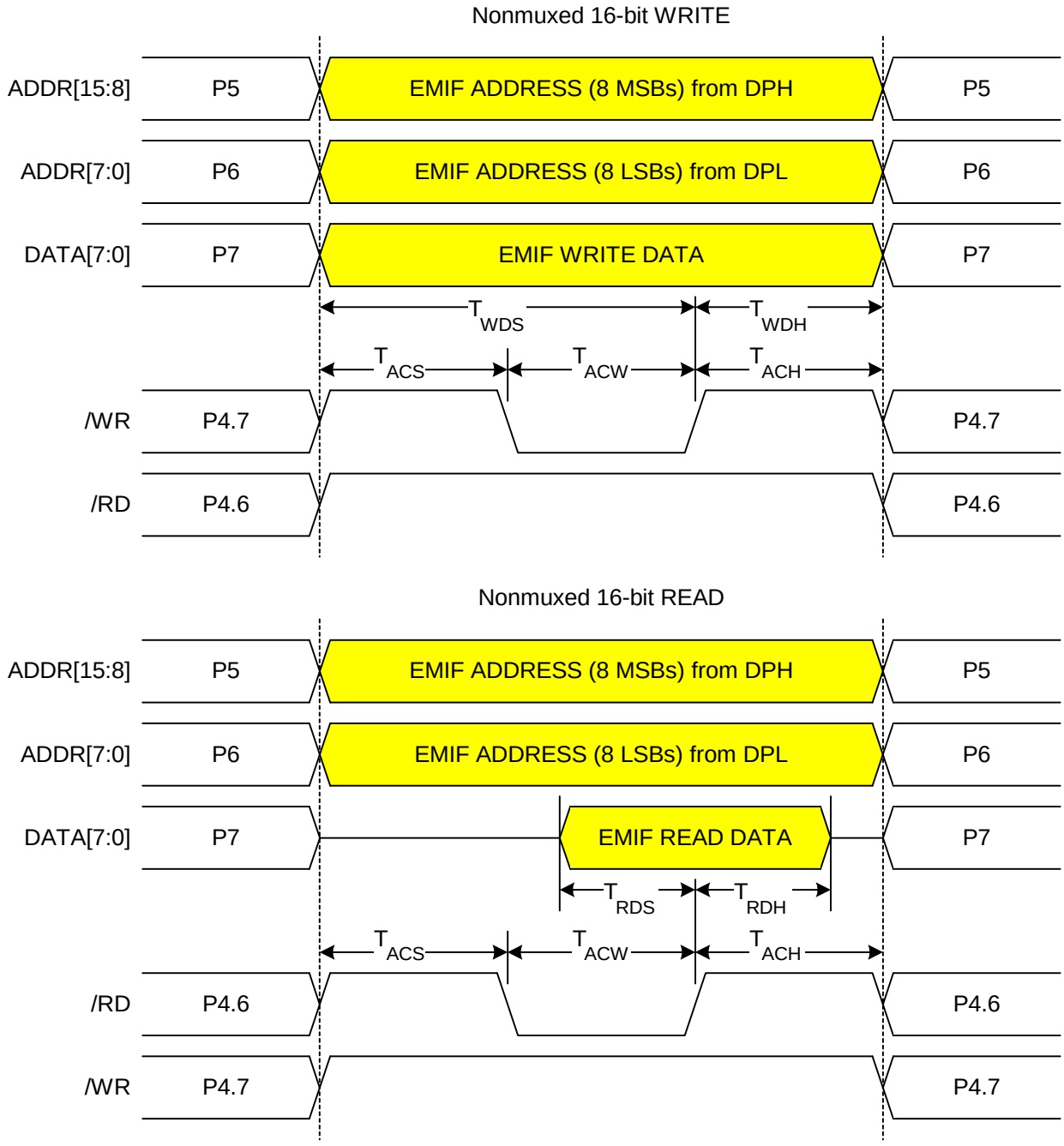
Bits1-0: EAH1-0: EMIF Address Hold Time Bits.
00: Address hold time = 0 SYSCLK cycles.
01: Address hold time = 1 SYSCLK cycle.
10: Address hold time = 2 SYSCLK cycles.
11: Address hold time = 3 SYSCLK cycles.

Table 17.1 lists the AC parameters for the External Memory Interface, and Figure 17.7 through Figure 17.12 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

### 17.6.1. Non-multiplexed Mode

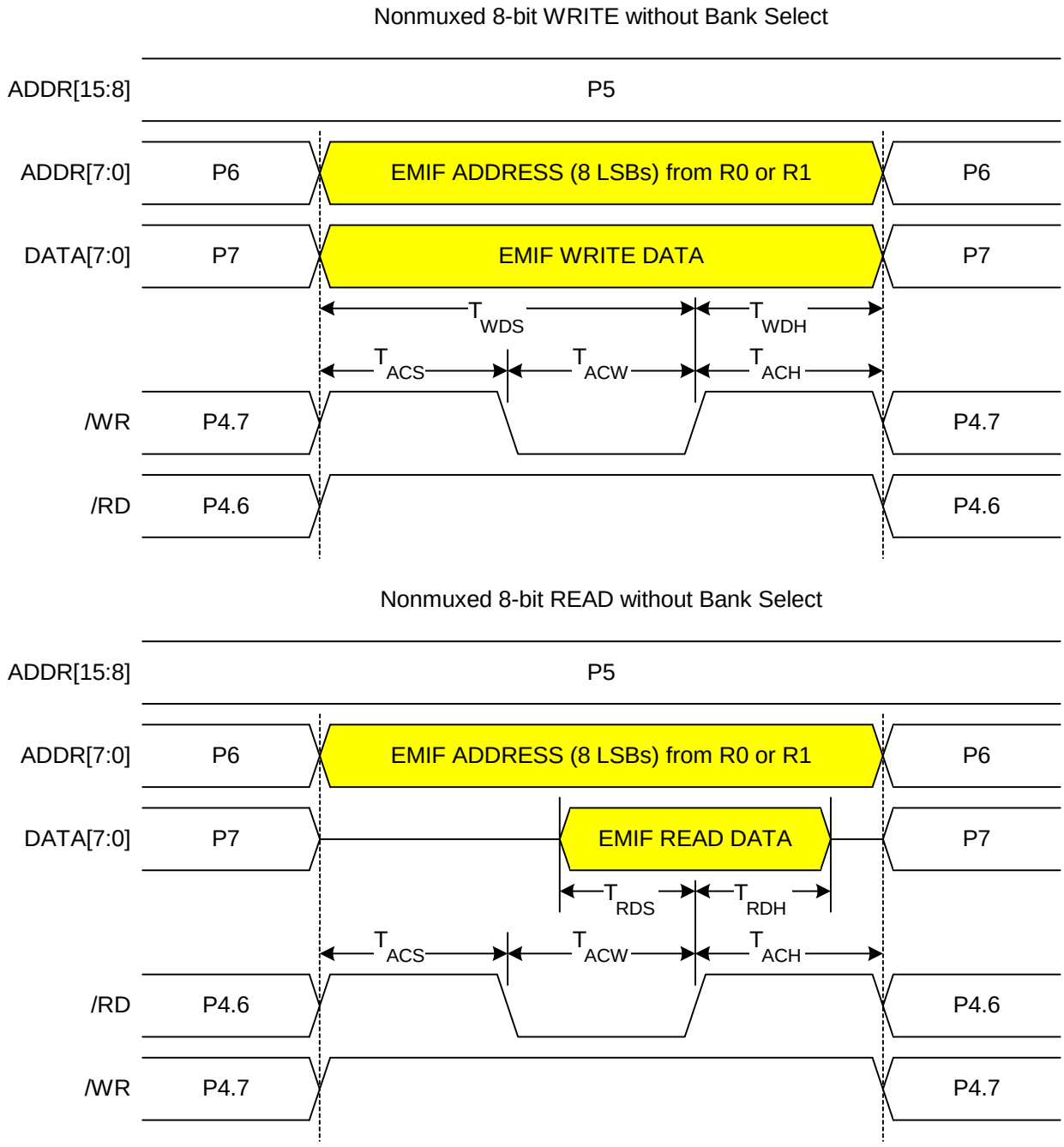
#### 17.6.1.1.16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’.

**Figure 17.7. Non-multiplexed 16-bit MOVX Timing**



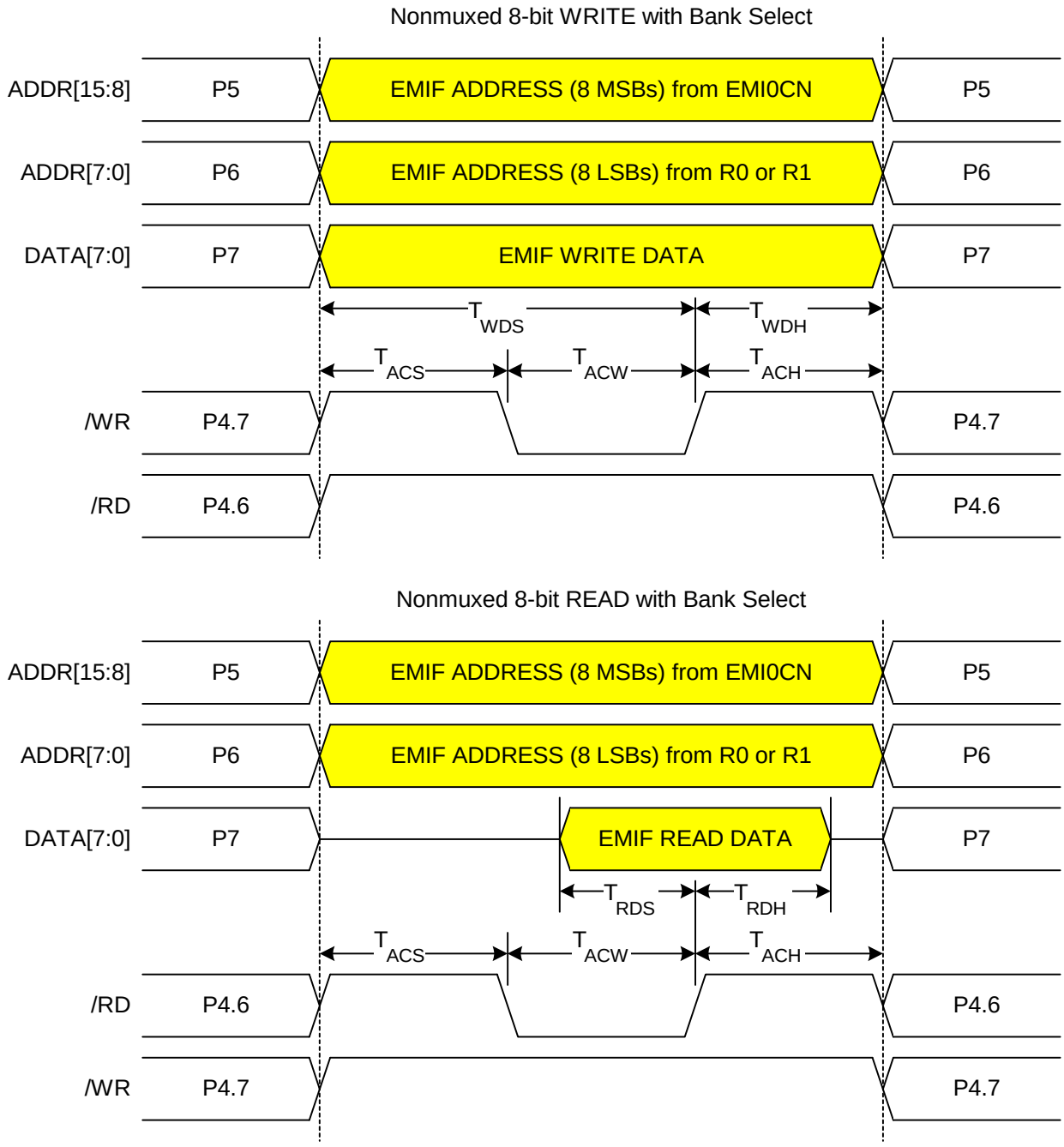
**17.6.1.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘101’ or ‘111’.**

**Figure 17.8. Non-multiplexed 8-bit MOVX without Bank Select Timing**



**17.6.1.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = '110'.**

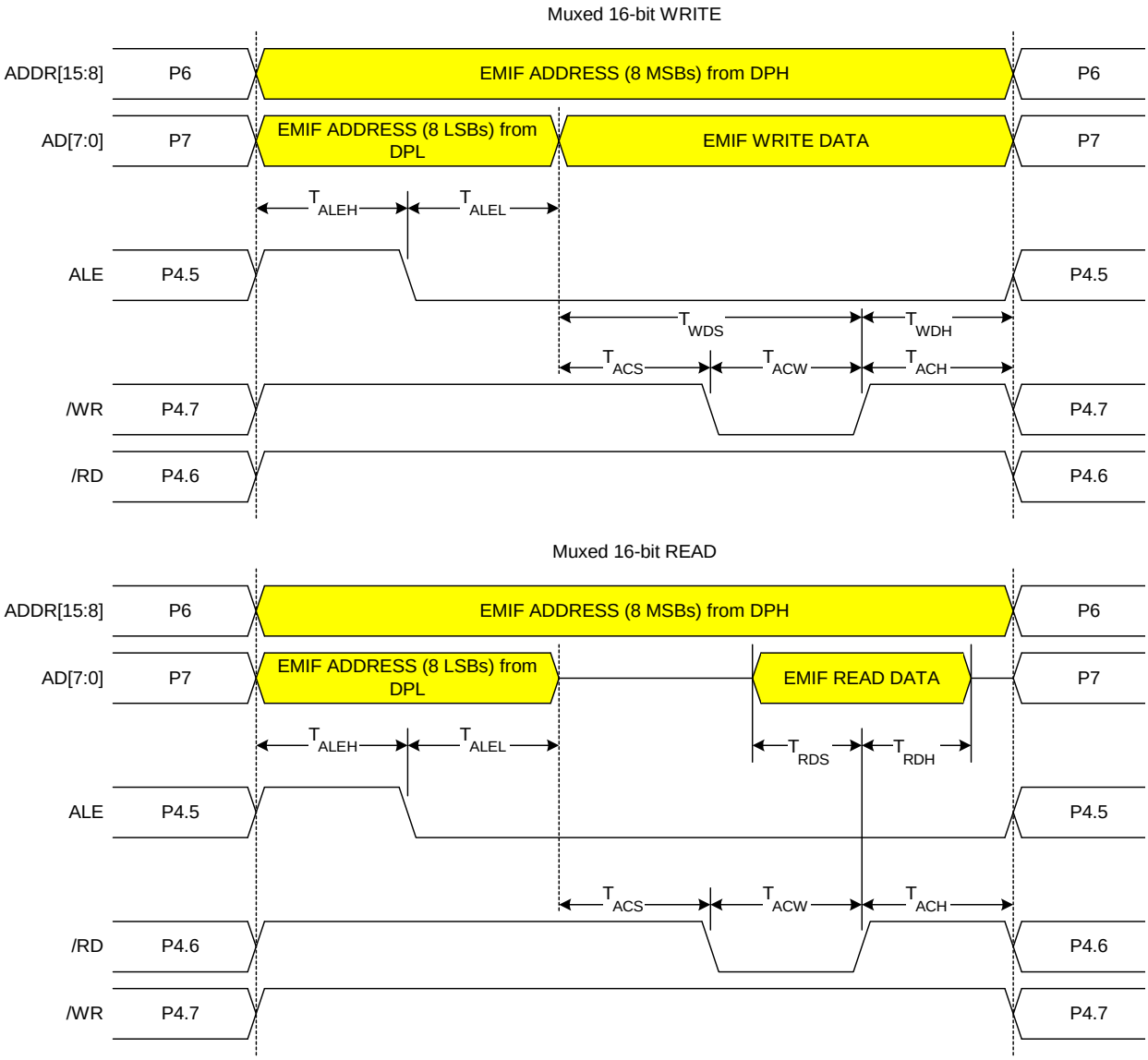
**Figure 17.9. Non-multiplexed 8-bit MOVX with Bank Select Timing**



### 17.6.2. Multiplexed Mode

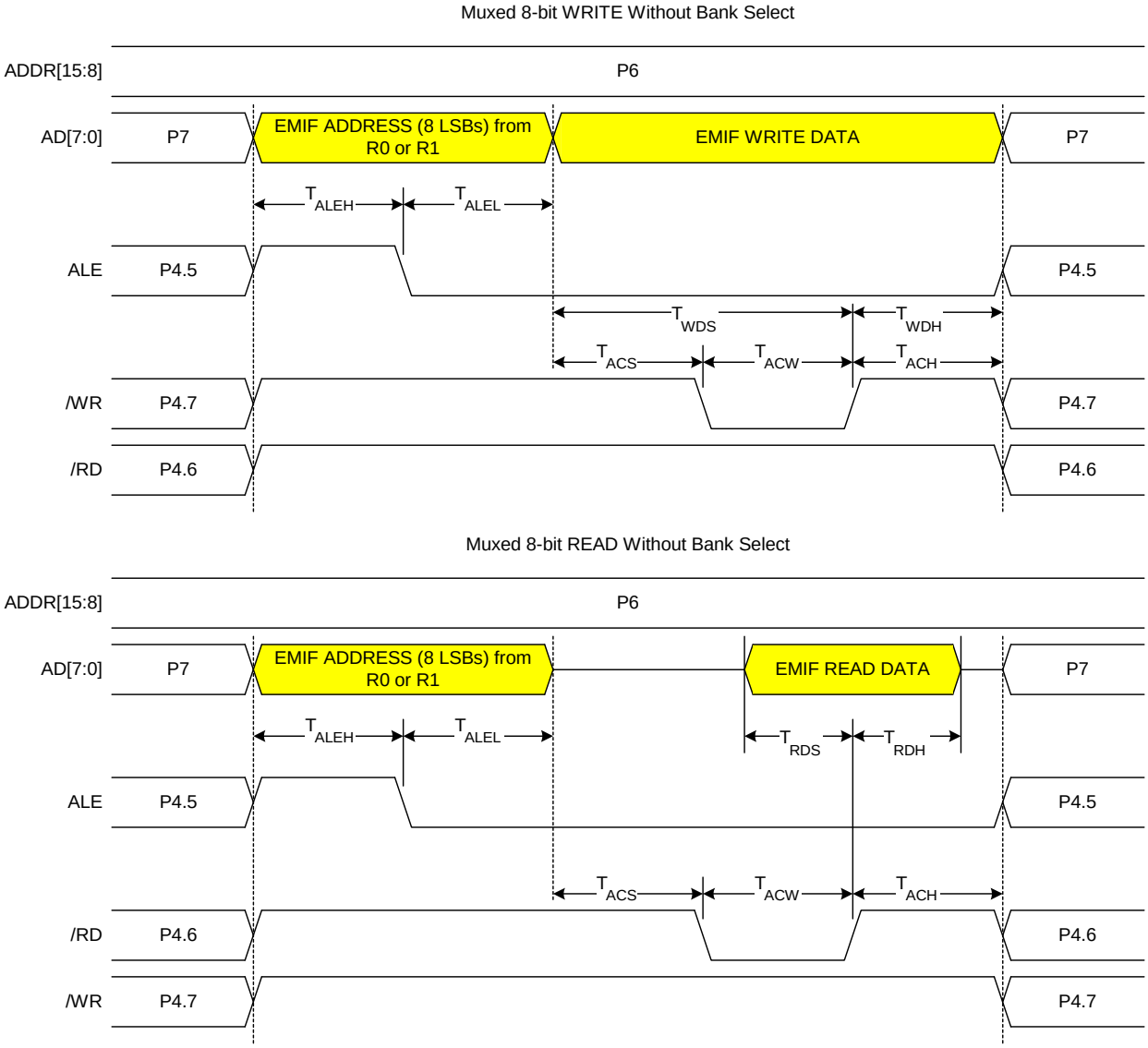
#### 17.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

**Figure 17.10. Multiplexed 16-bit MOVX Timing**



### 17.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

**Figure 17.11. Multiplexed 8-bit MOVX without Bank Select Timing**



### 17.6.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘010’.

**Figure 17.12. Multiplexed 8-bit MOVX with Bank Select Timing**

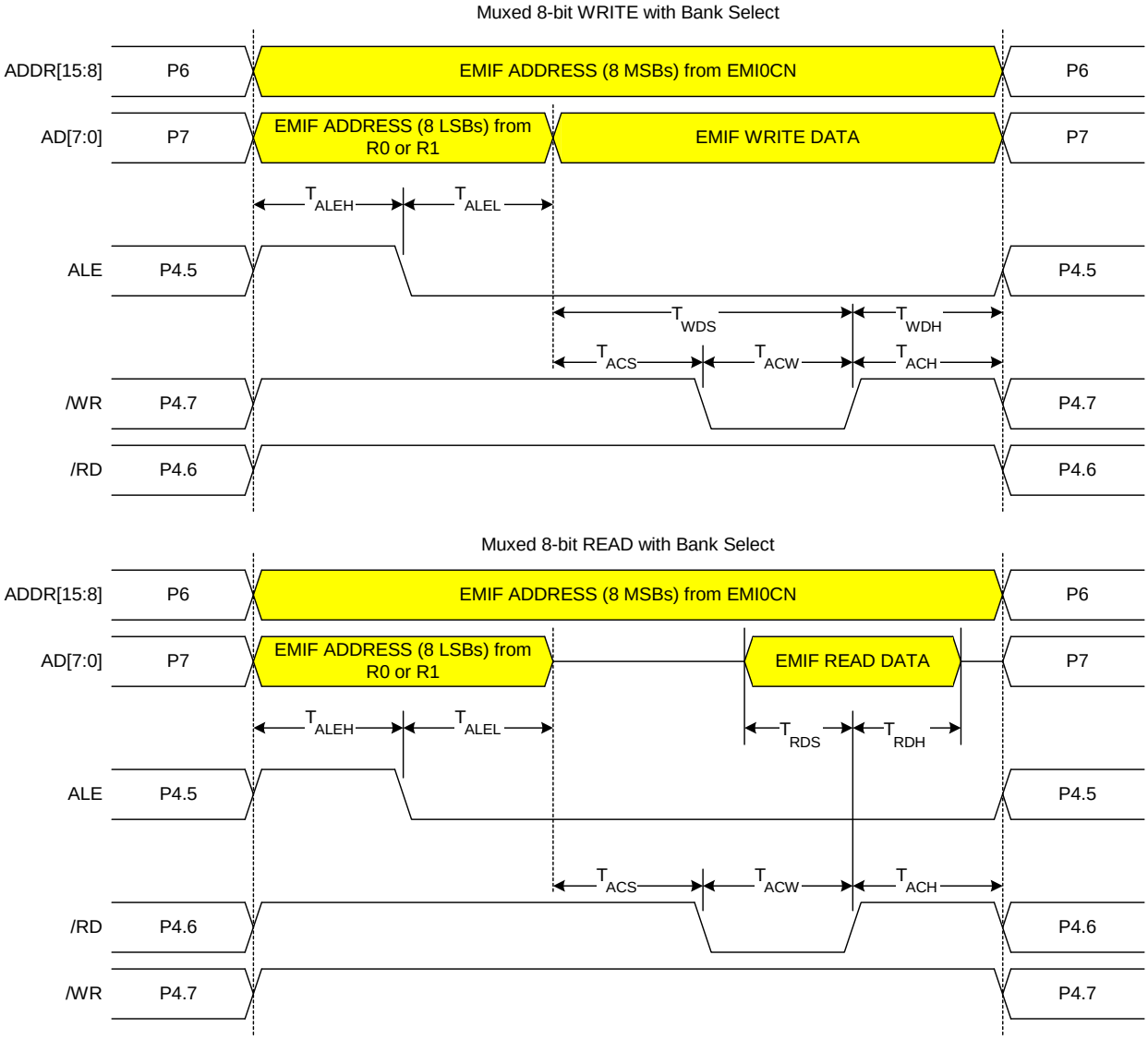


**Table 17.1. AC Parameters for External Memory Interface**

| Parameter | Description | Min | Max | Units |
|---|---|---|---|---|
| $T_{SYSCLK}$ | System Clock Period | 40 | | ns |
| $T_{ACS}$ | Address / Control Setup Time | 0 | $3*T_{SYSCLK}$ | ns |
| $T_{ACW}$ | Address / Control Pulse Width | $1*T_{SYSCLK}$ | $16*T_{SYSCLK}$ | ns |
| $T_{ACH}$ | Address / Control Hold Time | 0 | $3*T_{SYSCLK}$ | ns |
| $T_{ALEH}$ | Address Latch Enable High Time | $1*T_{SYSCLK}$ | $4*T_{SYSCLK}$ | ns |
| $T_{ALEL}$ | Address Latch Enable Low Time | $1*T_{SYSCLK}$ | $4*T_{SYSCLK}$ | ns |
| $T_{WDS}$ | Write Data Setup Time | $1*T_{SYSCLK}$ | $19*T_{SYSCLK}$ | ns |
| $T_{WDH}$ | Write Data Hold Time | 0 | $3*T_{SYSCLK}$ | ns |
| $T_{RDS}$ | Read Data Setup Time | 20 | | ns |
| $T_{RDH}$ | Read Data Hold Time | 0 | | ns |

# 18. Port Input/Output

The C8051F06x family of devices are fully integrated mixed-signal System on a Chip MCUs with 59 digital I/O pins (C8051F060/2/4/6) or 24 digital I/O pins (C8051F061/3/5/7), organized as 8-bit Ports. All ports are both bit- and byte-addressable through their corresponding Port Data registers. All Port pins support configurable Open-Drain or Push-Pull output modes and weak pull-ups. Additionally, Port 0 pins are 5 V-tolerant. 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**

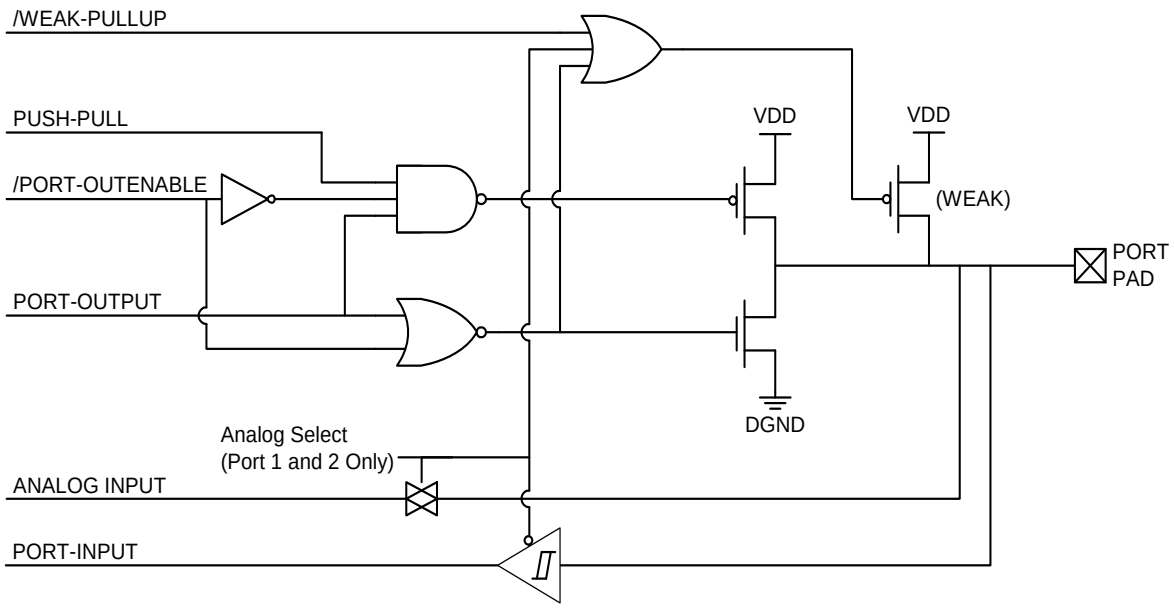



**Table 18.1. Port I/O DC Electrical Characteristics**

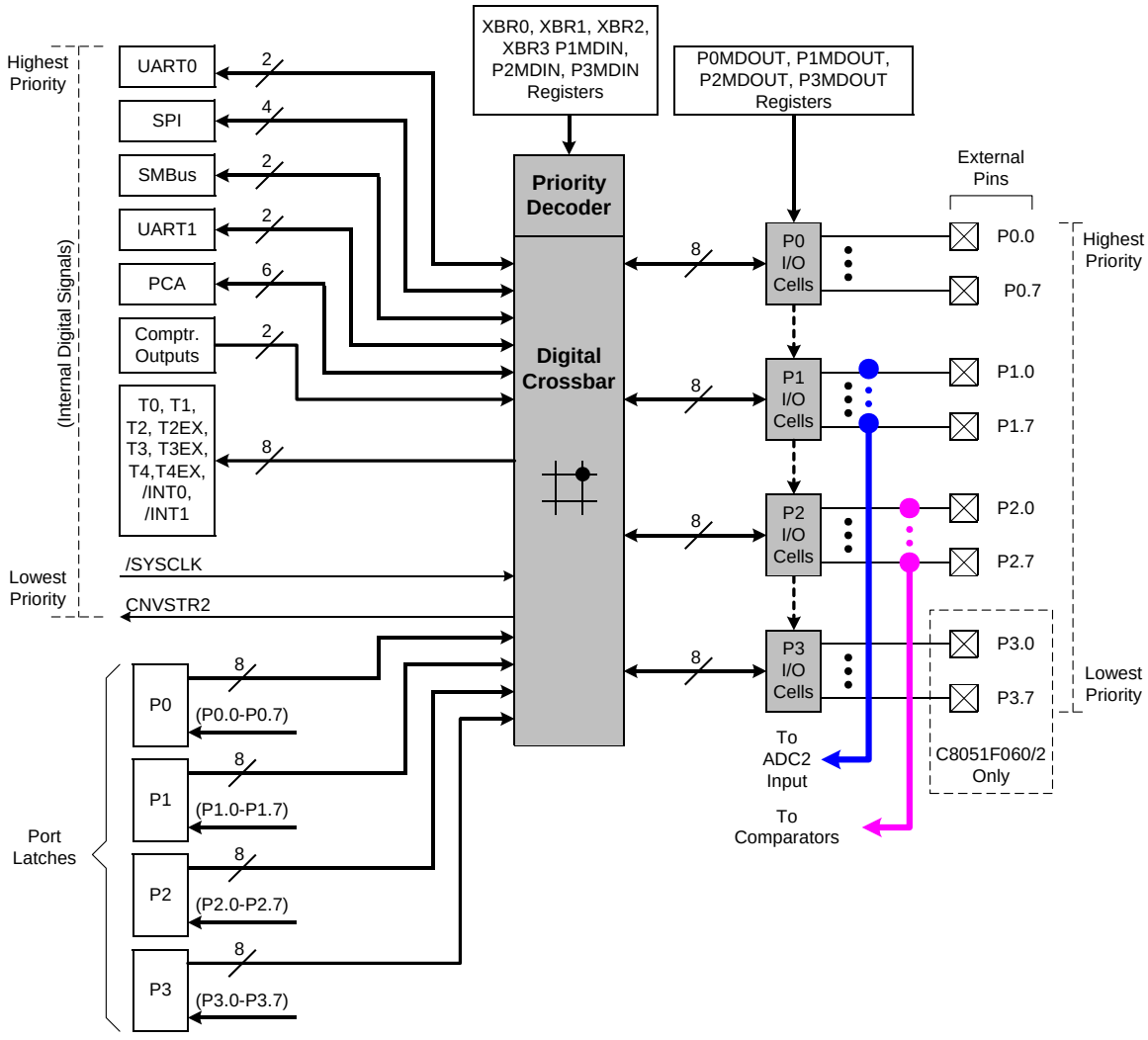
VDD = 2.7 to 3.6 V, -40 to +85 °C unless otherwise specified.

| 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 | VDD - 0.7<br>VDD - 0.1 | | | V |
| Output Low Voltage ($V_{OL}$) | $I_{OL}$ = 8.5 mA<br>$I_{OL}$ = 10 µA | | | 0.6<br>0.1 | V |
| Input High Voltage (VIH) | | 0.7 x VDD | | | |
| Input Low Voltage (VIL) | | | | 0.3 x VDD | |
| Input Leakage Current | DGND < Port Pin < VDD, Pin Tri-state<br>Weak Pull-up Off<br>Weak Pull-up On | | <br><br>10 | ± 1 | <br>µA<br>µA |
| Input Capacitance | | | 5 | | pF |

The C8051F06x family of devices have a wide array of digital resources which are available through the four lower I/O Ports: P0, P1, P2, and (on the C8051F060/2/4/6) 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 2 can be used as analog inputs to the analog Voltage comparators. On the C8051F060/1/2/3, the pins of Port 1 can be used as analog inputs for ADC2.

The upper Ports (available on C8051F060/2/4/6) can be byte-accessed as GPIO pins, or used as part of an External Memory Interface which is active during a MOVX instruction whose target address resides in off-chip memory. See Section “17. External Data Memory Interface and On-Chip XRAM” on page 187 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 (on the C8051F060/2/4/6) or P2.7 (on the C8051F061/3/5/7) 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, XBR2, and XBR3, shown in Figure 18.5, Figure 18.6, Figure 18.7, and Figure 18.8. 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. 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 periph-

**Figure 18.3. Priority Crossbar Decode Table**

**(P1MDIN = 0xFF; P2MDIN = 0xFF)**

| | 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 |
| CP2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | | CP2E: XBR3.3 |
| T0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | | T0E: XBR1.1 |
| /INT0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | | INT0E: XBR1.2 |
| T1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | T1E: XBR1.3 |
| /INT1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | INT1E: XBR1.4 |
| T2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | T2E: XBR1.5 |
| T2EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | T2EXE: XBR1.6 |
| T3 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | T3E: XBR3.0 |
| T3EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | T3EXE: XBR3.1 |
| T4 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | T4E: XBR2.3 |
| T4EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | T4EXE: XBR2.4 |
| /SYSCLK | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | SYSCKE: XBR1.7 |
| CNVSTR2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | CNVSTE2: XBR3.2 |
| | | | | | | | | | AIN2.0 | AIN2.1 | AIN2.2 | AIN2.3 | AIN2.4 | AIN2.5 | AIN2.6 | AIN2.7 | | | | CP1+ | CP1- | CP2+ | CP2- | CP0+ | CP0- | | | | | | | | | |

eral'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 the SMBus, UART0 or UART1 are selected (i.e. SMBus, SPI, UART). It would be impossible, for example, to assign TX0 to a Port pin without assigning RX0 as well. The SPI can operate in 3 or 4-wire mode (with or without NSS). 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 Figure 18.9, Figure 18.11, Figure 18.14, and Figure 18.17), a set of SFRs 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.

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 VDD. 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 Figure 18.10, Figure 18.13, Figure 18.16, and Figure 18.18). 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 Pull-ups

By default, each Port pin has an internal weak pull-up device enabled which provides a resistive connection (about 100 kΩ) between the pin and VDD. The weak pull-up devices can be globally disabled by writing a logic 1 to the Weak Pull-up Disable bit, (WEAKPUD, XBR2.7). The weak pull-up is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pull-up device. The weak pull-up device can also be explicitly disabled on a Port 1 pin by configuring the pin as an Analog Input, as described below.

### 18.1.5. Configuring Port 1 and 2 pins as Analog Inputs

The pins on Port 1 can serve as analog inputs to the ADC2 analog MUX (C8051F060/1/2/3 only) and the pins on Port 2 can serve as analog inputs to the Comparators (all 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 VDD / 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 pull-up 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 PnMDOUT bits of pins configured as Analog Inputs should explicitly be set to logic 0 (Open-Drain output mode), and the associated Port 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 or the Comparators, however, it is strongly recommended. See the analog peripheral’s corresponding section in this datasheet for further information.

### 18.1.6. Crossbar Pin Assignment Example

In this example (Figure 18.4), we configure the Crossbar to allocate Port pins for UART0, the SMBus, all 6 PCA modules, /INT0, and /INT1 (12 pins total). Additionally, 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:

XBR0, XBR1, and XBR2 are set such that UART0EN = 1, SMB0EN = 1, PCA0ME = '110', INT0E = 1, and INT1E = 1. Thus: XBR0 = 0x3D, XBR1 = 0x14, and XBR2 = 0x40.

1. 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).
2. We enable the Crossbar by setting XBARE = 1: XBR2 = 0x40.
   - 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.
   - PCA0 is next in priority order, so P0.4 through P1.1 are assigned to CEX0 through CEX5
   - P1MDIN is set to 0xE3, which configures P1.2, P1.3, and P1.4 as Analog Inputs, causing the Crossbar to skip these pins.
   - /INT0 is next in priority order, so it is assigned to the next non-skipped pin, which is P1.5.
   - /INT1 is next in priority order, so it is assigned to P1.6.
3. We set the UART0 TX pin (TX0, P0.0) output and the CEX0-3 outputs to Push-Pull by setting P0MDOUT = 0xF1.
4. We explicitly disable the output drivers on the 3 Analog Input pins by setting the corresponding bits in the P1MDOUT register to '0', and in P1 to '1'. Additionally, the CEX5-4 output pins are set to Push-Pull mode. Therefore, P1MDOUT = 0x03 (configure unused pins to Open-Drain) and P1 = 0xFF (a logic 1 selects the high-impedance state).

**Figure 18.4. Crossbar Example:**

**(P1MDIN = 0xE3; XBR0 = 0x3D; XBR1 = 0x14; XBR2 = 0x40)**

| | 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 | | | | ● | | ● | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 |
| CP2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | | CP2E: XBR3.3 |
| T0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | | T0E: XBR1.1 |
| /INT0 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | | INT0E: XBR1.2 |
| T1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | | T1E: XBR1.3 |
| /INT1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | | INT1E: XBR1.4 |
| T2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | | T2E: XBR1.5 |
| T2EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | T2EXE: XBR1.6 |
| T3 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | T3E: XBR3.0 |
| T3EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | T3EXE: XBR3.1 |
| T4 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | T4E: XBR2.3 |
| T4EX | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | T4EXE: XBR2.4 |
| /SYSCLK | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | SYSCKE: XBR1.7 |
| CNVSTR2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | CNVSTE2: XBR3.2 |
| | | | | | | | | | AIN2.0 | AIN2.1 | **AIN2.2** | **AIN2.3** | **AIN2.4** | AIN2.5 | AIN2.6 | AIN2.7 | | | CP1+ | CP1- | CP2+ | CP2- | CP0+ | CP0- | | | | | | | | | |

**Figure 18.5. 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.
4-wire mode:
1: SPI0 SCK, MISO, MOSI, and NSS routed to 4 Port pins.
3-wire mode:
1: SPI0 SCK, MISO and MOSI routed to 3 Port pins.

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.

**Figure 18.6. 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 routed to Port pin.

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: /INT1 routed to Port pin.

Bit1: T0E: T0 Input Enable Bit.
0: T0 unavailable at Port pin.
1: T1 routed to Port pin.

Bit0: CP1E: CP1 Output Enable Bit.
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

**Figure 18.7. 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 | - | T4EXE | T4E | UART1E | - | - | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE3
SFR Page: F

Bit7: WEAKPUD: Weak Pull-Up Disable Bit.
0: Weak pull-ups globally enabled.
1: Weak pull-ups 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: UNUSED. Read = 0, Write = don't care.
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.
Bits1-0: Reserved

**Figure 18.8. XBR3: Port I/O Crossbar Register 3**

| R | R | R | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| CTXOUT | - | - | | CP2E | CNVST2E | T3EXE | T3E | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xE4
SFR Page: F

Bit7: CTXOUT: CAN Transmit Pin (CTX) Output Mode.
0: CTX pin output mode is configured as open-drain.
1: CTX pin output mode is configured as push-pull.
Bit6-4: Reserved
Bit3: CP2E: CP2 Output Enable Bit.
0: CP2 unavailable at Port pin.
1: CP2 routed to Port pin.
Bit2: CNVST2E: ADC2 External Convert Start Input Enable Bit.
0: CNVST2 for ADC2 unavailable at Port pin.
1: CNVST2 for ADC2 routed to Port pin.
Bit1: T3EXE: T3EX Input Enable Bit.
0: T3EX unavailable at Port pin.
1: T3EX routed to Port pin.
Bit0: T3E: T3 Input Enable Bit.
0: T3 unavailable at Port pin.
1: T3 routed to Port pin.

**Figure 18.9. P0: Port0 Data Register**

| 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, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P0MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P0.n pin is logic low.
1: P0.n pin is logic high.

**Figure 18.10. P0MDOUT: Port0 Output Mode 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: 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.

**Figure 18.11. P1: Port1 Data Register**

| 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, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P1MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P1.n pin is logic low.
1: P1.n pin is logic high.

Note:
On the C8051F060/1/2/3, 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 Figure 18.12**)**. Note that in analog mode, the output mode of the pin is determined by the Port 1 latch and P1MDOUT (Figure 18.13). See Section "7. 10-Bit ADC (ADC2, C8051F060/1/2/3)" on page 87 for more information about ADC2.

**Figure 18.12. P1MDIN: Port1 Input Mode Register**

| 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 pull-up 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. The state of the weak pull-up is determined by the WEAKPUD bit (XBR2.7, see Figure 18.7).

**Figure 18.13. P1MDOUT: Port1 Output Mode 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: 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.

**Figure 18.14. P2: Port2 Data Register**

| 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, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P2MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P2.n pin is logic low.
1: P2.n pin is logic high.

**Figure 18.15. P2MDIN: Port2 Input Mode Register**

| 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: 0xAE
SFR Page: F

Bits7-0: P2MDIN.[7:0]: Port 2 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 pull-up 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. The state of the weak pull-up is determined by the WEAKPUD bit (XBR2.7, see Figure 18.7).

**Figure 18.16. P2MDOUT: Port2 Output Mode 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: 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.

**Figure 18.17. P3: Port3 Data Register**

| 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, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P3MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P3.n pin is logic low.
1: P3.n pin is logic high.

Note: Although P3 is not brought out to pins on the C8051F061/3/5/7 devices, the Port Data register is still present and can be used by software. See “Configuring Ports which are not Pinned Out” on page 219.

**Figure 18.18. P3MDOUT: Port3 Output Mode 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: 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 (C8051F060/2/4/6 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 Figure 18.19, Figure 18.21, Figure 18.23, and Figure 18.25), a set of SFRs which are byte-addressable. Note that Port 4 has only three pins: P4.5, P4.6, and P4.7. 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.

### 18.2.1. Configuring Ports which are not Pinned Out

Although P3, P4, P5, P6, and P7 are not brought out to pins on the C8051F061/3/5/7 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 pull-up devices enabled by setting WEAKPUD (XBR2.7) to a logic 0.
2. Configure the output modes of P3, P4, P5, P6, and P7 to “Push-Pull” by writing 0xFF to the associated output mode register (PnMDOUT).
3. Force the output states of P3, P4, P5, P6, and P7 to logic 0 by writing zeros to the Port Data registers: P3 = 0x00, 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 VDD. 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 Figure 18.20, Figure 18.22, Figure 18.24, and Figure 18.26). For example, to place Port pin 5.3 in push-pull mode (digital output), set P5MDOUT.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 Pull-ups

By default, each Port pin has an internal weak pull-up device enabled which provides a resistive connection (about 100 kΩ) between the pin and VDD. The weak pull-up devices can be globally disabled by writ-

ing a logic 1 to the Weak Pull-up Disable bit, (WEAKPUD, XBR2.7). The weak pull-up is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pull-up device.

### 18.2.5. External Memory Interface

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 187 for more information about the External Memory Interface.

**Figure 18.19. P4: Port4 Data Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| P4.7 | P4.6 | P4.5 | - | - | - | - | - | 11111111 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Bit Addressable |

SFR Address: 0xC8
SFR Page: F

Bits7-5: P4.[7:5]: Port4 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (open, if corresponding P4MDOUT.n bit = 0). See Figure 18.20.
Read - Returns states of I/O pins.
0: P4.n pin is logic low.
1: P4.n pin is logic high.
Bits 4-0: Reserved. Write to ‘11111’.

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 187 for more information.

**Figure 18.20. P4MDOUT: Port4 Output Mode 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: 0x9C
SFR Page: F

Bits7-5: P4MDOUT.[7:5]: Port4 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.
Bits 4-0: Reserved. Write to ‘00000’.

**Figure 18.21. P5: Port5 Data Register**

| 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, if corresponding P5MDOUT bit = 0). See Figure 18.22.
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 187 for more information about the External Memory Interface.

**Figure 18.22. P5MDOUT: Port5 Output Mode 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: 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.

**Figure 18.23. P6: Port6 Data Register**

| 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, if corresponding P6MDOUT bit = 0). See Figure 18.24.
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 187 for more information about the External Memory Interface.

**Figure 18.24. P6MDOUT: Port6 Output Mode 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: 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.

**Figure 18.25. P7: Port7 Data Register**

| 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, if corresponding P7MDOUT bit = 0). See Figure 18.26.
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 187 for more information about the External Memory Interface.

**Figure 18.26. P7MDOUT: Port7 Output Mode 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: 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.

# 19. Controller Area Network (CAN0, C8051F060/1/2/3)

***IMPORTANT DOCUMENTATION NOTE:*** *The Bosch CAN Controller is integrated in the C8051F060/1/2/3 devices. This section of the data sheet gives a description of the CAN controller as an overview and offers a description of how the Silicon Labs CIP-51 MCU interfaces with the on-chip Bosch CAN controller. In order to use the CAN controller, please refer to Bosch's C_CAN User's Manual (revision 1.2) as an accompanying manual to Silicon Labs' C8051F060/1/2/3/4/5/6/7 Data sheet.*

The C8051F060/1/2/3 family of devices feature a Control Area Network (CAN) controller that enables serial communication using the CAN protocol. Silicon Labs CAN controller facilitates communication on a CAN network in accordance with the Bosch specification 2.0A (basic CAN) and 2.0B (full CAN). The CAN controller consists of a CAN Core, Message RAM (separate from the CIP-51 RAM), a message handler state machine, and control registers. Silicon Labs CAN is a protocol controller and does not provide physical layer drivers (i.e., transceivers). Figure 19.2 shows an example typical configuration on a CAN bus.

Silicon Labs CAN operates at bit rates of up to 1 Mbit/second, though this can be limited by the physical layer chosen to transmit data on the CAN bus. The CAN processor has 32 Message Objects that can be configured to transmit or receive data. Incoming data, message objects and their identifier masks are stored in the CAN message RAM. All protocol functions for transmission of data and acceptance filtering is performed by the CAN controller and not by the CIP-51 MCU. In this way, minimal CPU bandwidth is needed to use CAN communication. The CIP-51 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFR) in the CIP-51. The CAN controller's clock ($f_{sys}$, or CAN_CLK in the C_CAN User's Guide) is equal to the CIP-51 MCU's clock (SYSCLK).

Figure 19.1. CAN Controller Diagram

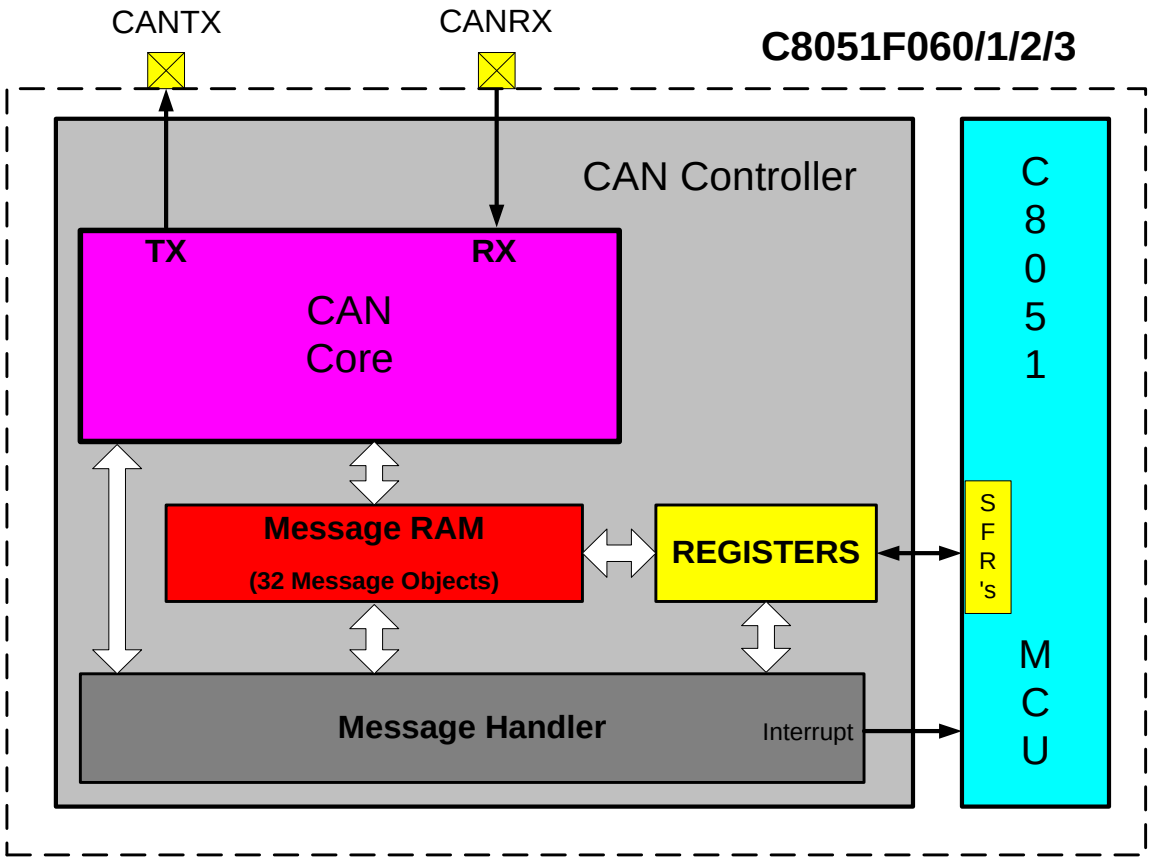
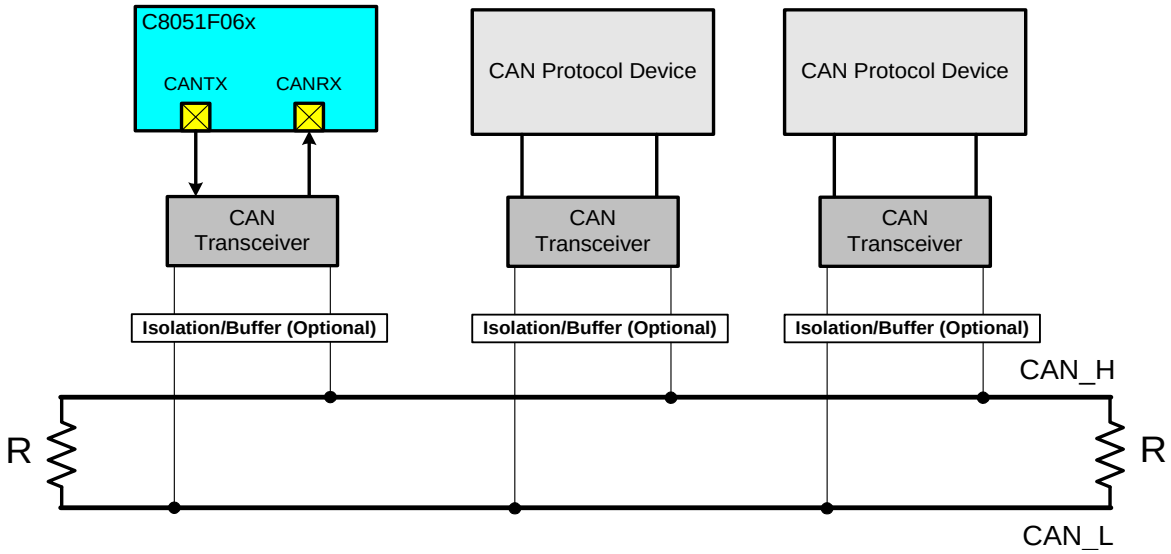



Figure 19.2. Typical CAN Bus Configuration



## 19.1. Bosch CAN Controller Operation

The CAN Controller featured in the C8051F060/1/2/3 devices is a full implementation of Bosch's full CAN module and fully complies with CAN specification 2.0B.

The function and use of the CAN Controller is detailed in the *Bosch CAN User's Guide*. The User's Guide should be used as a reference to configure and use the CAN controller. This Silicon Labs datasheet describes how to access the CAN controller.

The CAN Control Register (CAN0CN), CAN Test Register (CAN0TST), and CAN Status Register (CAN0STA) in the CAN controller can be accessed directly or indirectly via CIP-51 SFRs. All other CAN registers must be accessed via an indirect indexing method. See "Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers" on page 229.

## 19.2. CAN Registers

CAN registers are classified as follows:

1. CAN Controller Protocol Registers: CAN control, interrupt, error control, bus status, test modes.
2. Message Object Interface Registers: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The C8051 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. Message Handler Registers: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).
4. C8051 MCU Special Function Registers (SFR): Five registers located in the C8051 MCU memory map that allow direct access to certain CAN Controller Protocol Registers, and Indexed indirect access to all CAN registers.

### 19.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes. The CAN controller protocol registers are accessible using C8051 MCU SFRs by an indexed method, and some can be accessed directly by addressing the SFRs in the C8051 SFR map for convenience.

The registers are: CAN Control Register (CAN0CN), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register. CAN0STA, CAN0CN, and CAN0TST can be accessed via C8051 MCU SFRs. All others are accessed indirectly using the CAN address indexed method via CAN0ADR, CAN0DATH, and CAN0DATL.

**Please refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.**

### 19.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers. These registers are accessed via the C8051's CAN0ADR and CAN0DAT registers using the indirect indexed address method.

**Please refer to the Bosch CAN User's Guide for information on the function and use of the Message Object Interface Registers.**

### 19.2.3. Message Handler Registers

The Message Handler Registers are *read only* registers. Their flags can be read via the indexed access method with CAN0ADR, CAN0DATH, and CAN0DATL. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

**Please refer to the Bosch CAN User's Guide for information on the function and use of the Message Handler Registers.**

### 19.2.4. CIP-51 MCU Special Function Registers

C8051F060/1/2/3 peripherals are modified, monitored, and controlled using Special Function Registers (SFRs). Most of the CAN Controller registers cannot be accessed *directly* using the SFRs. Three of the CAN Controller's registers may be accessed directly with SFRs. All other CAN Controller registers are accessed indirectly using three CIP-51 MCU SFRs: the CAN Data Registers (CAN0DATH and CAN0DATL) and CAN Address Register (CAN0ADR). In this way, there are a total of five CAN registers used to configure and run the CAN Controller.

### 19.2.5. Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers

Each CAN Controller Register has an index number (see Table  below). The CAN register address space is 128 words (256 bytes). A CAN register is accessed via the CAN Data Registers (CAN0DATH and CAN0DATL) when a CAN register's index number is placed into the CAN Address Register (CAN0ADR). For example, if the Bit Timing Register is to be configured with a new value, CAN0ADR is loaded with 0x03. The low byte of the desired value is accessed using CAN0DATL and the high byte of the bit timing register is accessed using CAN0DATH. CAN0DATL is bit addressable for convenience. To load the value 0x2304 into the Bit Timing Register:

```
CAN0ADR = 0x03;     // Load Bit Timing Register's index (Table 18.1)
CAN0DATH = 0x23;    // Move the upper byte into data reg high byte
CAN0DATL = 0x04;    // Move the lower byte into data reg low byte
```

Note: CAN0CN, CAN0STA, and CAN0TST may be accessed either by using the index method, or by direct access with the CIP-51 MCU SFRs. CAN0CN is located at SFR location 0xF8/SFR page 1 (Figure 19.6), CAN0TST at 0xDB/SFR page 1 (Figure 19.7), and CAN0STA at 0xC0/SFR page 1 (Figure 19.8).

### 19.2.6. CAN0ADR Autoincrement Feature

For ease of programming message objects, CAN0ADR features autoincrementing for the index ranges 0x08 to 0x12 (Interface Registers 1) and 0x20 to 0x2A (Interface Registers 2). When the CAN0ADR register has an index in these ranges, **the CAN0ADR will autoincrement by 1 to point to the next CAN register 16-bit word upon a read/write of CAN0DATL**. This speeds programming of the frequently programmed interface registers when configuring message objects.

**NOTE: Table  below supersedes Figure 5 in section 3, "Programmer's Model" of the Bosch CAN User's Guide.**

**Table 19.1. CAN Register Index and Reset Values**

| CAN Register Index | Register name | Reset Value | Notes |
|---|---|---|---|
| 0x00 | CAN Control Register | 0x0001 | Accessible in CIP-51 SFR Map |
| 0x01 | Status Register | 0x0000 | Accessible in CIP-51 SFR Map |
| 0x02 | Error Register | 0x0000 | Read Only |
| 0x03 | Bit Timing Register | 0x2301 | Write Enabled by CCE Bit in CAN0CN |

**Table 19.1. CAN Register Index and Reset Values (Continued)**

| CAN Register Index | Register name | Reset Value | Notes |
|---|---|---|---|
| 0x04 | Interrupt Register | 0x0000 | Read Only |
| 0x05 | Test Register | 0x0000 | Bit 7 (RX) is determined by CAN bus |
| 0x06 | BRP Extension Register | 0x0000 | Write Enabled by TEST bit in CAN0CN |
| 0x08 | IF1 Command Request | 0x0001 | CAN0ADR autoincrements in IF1 index space (0x08 - 0x12) upon write to CAN0DATL |
| 0x09 | IF1 Command Mask | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x0A | IF1 Mask 1 | 0xFFFF | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x0B | IF1 Mask 2 | 0xFFFF | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x0C | IF1 Arbitration 1 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x0D | IF1 Arbitration 2 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x0E | IF1 Message Control | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x0F | IF1 Data A1 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x10 | IF1 Data A2 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x11 | IF1 Data B1 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x12 | IF1 Data B2 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x20 | IF2 Command Request | 0x0001 | CAN0ADR autoincrements in IF1 index space (0x08 - 0x12) upon write to CAN0DATL |
| 0x21 | IF2 Command Mask | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x22 | IF2 Mask 1 | 0xFFFF | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x23 | IF2 Mask 2 | 0xFFFF | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x24 | IF2 Arbitration 1 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x25 | IF2 Arbitration 2 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x26 | IF2 Message Control | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x27 | IF2 Data A1 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x28 | IF2 Data A2 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x29 | IF2 Data B1 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x2A | IF2 Data B2 | 0x0000 | CAN0ADR autoincrement upon write to CAN0DATL |
| 0x40 | Transmission Request 1 | 0x0000 | Transmission request flags for message objects (read only) |
| 0x41 | Transmission Request 2 | 0x0000 | Transmission request flags for message objects (read only) |
| 0x48 | New Data 1 | 0x0000 | New Data flags for message objects (read only) |
| 0x49 | New Data 2 | 0x0000 | New Data flags for message objects (read only) |
| 0x50 | Interrupt Pending 1 | 0x0000 | Interrupt pending flags for message objects (read only) |
| 0x51 | Interrupt Pending 2 | 0x0000 | Interrupt pending flags for message objects (read only) |
| 0x58 | Message Valid 1 | 0x0000 | Message valid flags for message objects (read only) |

**Table 19.1. CAN Register Index and Reset Values (Continued)**

| CAN Register Index | Register name | Reset Value | Notes |
|---|---|---|---|
| 0x59 | Message Valid 2 | 0x0000 | Message valid flags for message objects (read only) |

**Figure 19.3. CAN0DATH: CAN Data Access 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: 0xD9
SFR Page: 1

Bit7-0: CAN0DATH: CAN Data Access Register High Byte.
The CAN0DAT Registers are used to read/write register values and data to and from the CAN Registers pointed to with the index number in the CAN0ADR Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

**Figure 19.4. CAN0DATL: CAN Data Access Register Low Byte**

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

SFR Address: 0xD8
SFR Page: 1

Bit7-0: CAN0DATL: CAN Data Access Register Low Byte.
The CAN0DAT Registers are used to read/write register values and data to and from the CAN Registers pointed to with the index number in the CAN0ADR Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

**Figure 19.5. CAN0ADR: CAN Address Index 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: 0xDA
SFR Page: 1

Bit7-0: CAN0ADR: CAN Address Index Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

Note: When the value of CAN0ADR is 0x08-0x12 and 0x20-2A (IF1 and IF2 registers), this register will autoincrement by 1 upon a write to CAN0DATL. See Section "19.2.6. CAN0ADR Autoincrement Feature" on page 229.

**All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide.**

**Figure 19.6. CAN0CN: CAN Control Register**

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

SFR Address: 0xF8
SFR Page: 1

Bit 4: CANIF: CAN Interrupt Flag. Write = don't care.
0: CAN interrupt has not occured.
1: CAN interrupt has occured and is active.

CANIF is controlled by the CAN controller and is cleared by hardware once all interrupt conditions have been cleared in the CAN controller. See section 3.4.1 in the *Bosch CAN User's Guide* (page 24) for more information concerning CAN controller interrupts.

***All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide with the exception of the CANIF bit.**

This register may be accessed directly in the CIP-51 SFR register space, or through the indirect, index method (See Section "19.2.5. Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers" on page 229).

**Figure 19.7. CAN0TST: CAN Test Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| Please see the Bosch CAN User's Guide for a complete definition of this register | | | | | | | | |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xDB
SFR Page: 1

**All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide.**
This register may be accessed directly in the CIP-51 SFR register space, or through the indirect, index method (See Section "19.2.5. Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers" on page 229).

**Figure 19.8. CAN0STA: CAN Status Register**

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| Please see the Bosch CAN User's Guide for a complete definition of this register | | | | | | | | |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xC0
SFR Page: 1

**All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide.**
This register may be accessed directly in the CIP-51 SFR register space, or through the indirect, index method (See Section "19.2.5. Using CAN0ADR, CAN0DATH, and CANDATL To Access CAN Registers" on page 229).

# 20. 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 20.1. SMBus0 Block Diagram**

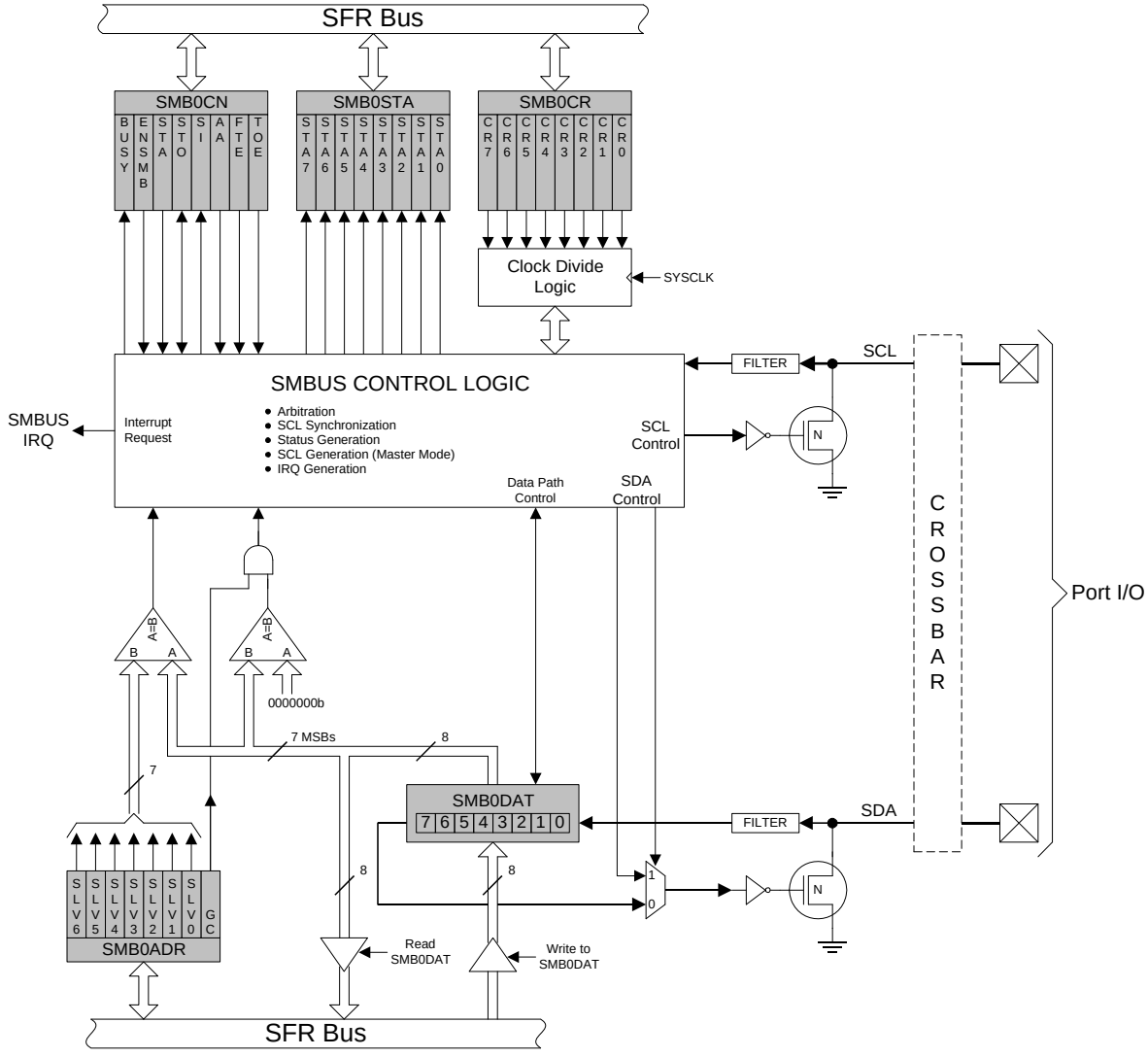
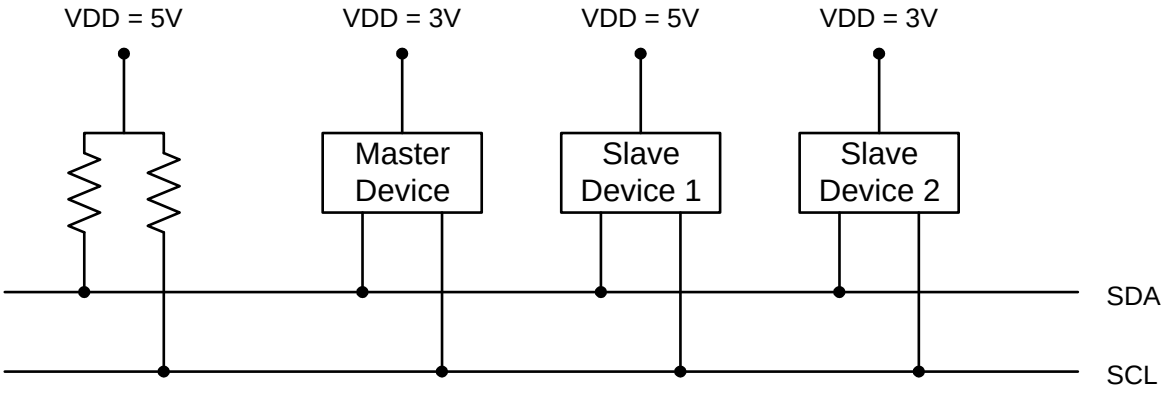


Figure 20.2 shows a typical SMBus configuration. The SMBus0 interface will work at any voltage between 3.0 V and 5.0 V and different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pull-up 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 20.2. Typical SMBus Configuration**




## 20.1. 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.

## 20.2. SMBus Protocol

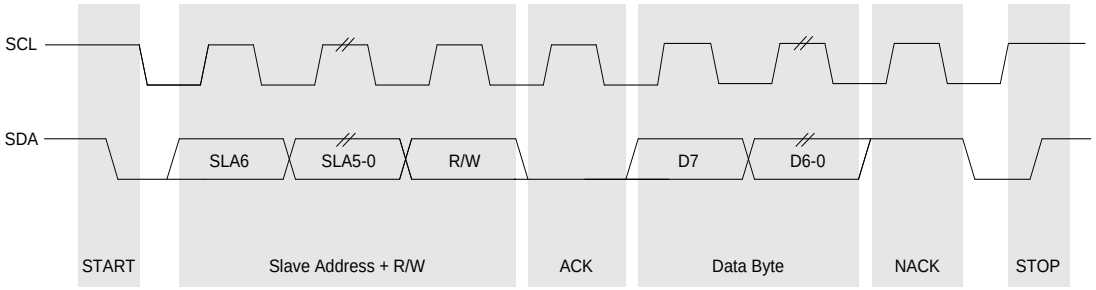
Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses 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 20.3). 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 20.3 illustrates a typical SMBus transaction.

**Figure 20.3. SMBus Transaction**




### 20.2.1. 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 20.2.4). 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.

### 20.2.2. 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.

### 20.2.3. 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.

### 20.2.4. 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.

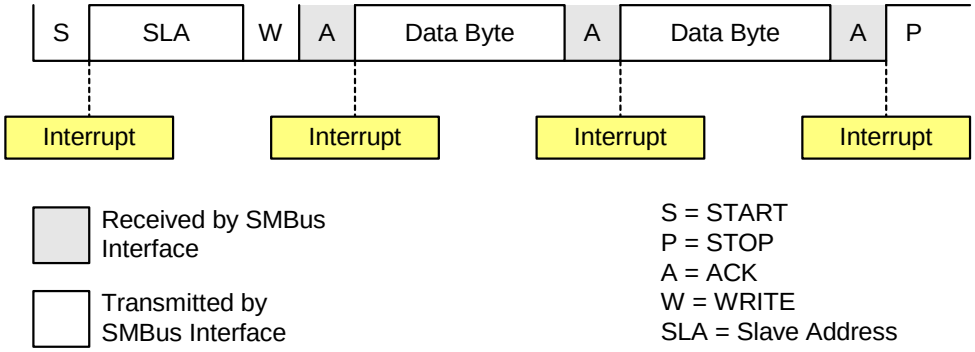
## 20.3. SMBus Transfer Modes

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 20.1 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.

### 20.3.1. 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.

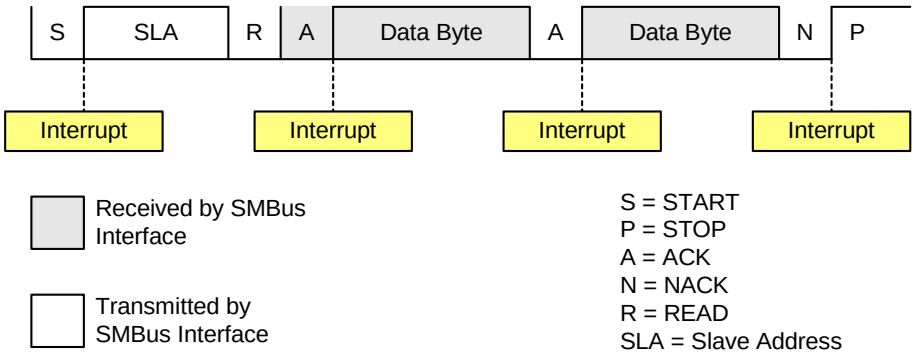
**Figure 20.4. Typical Master Transmitter Sequence**




### 20.3.2. 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.

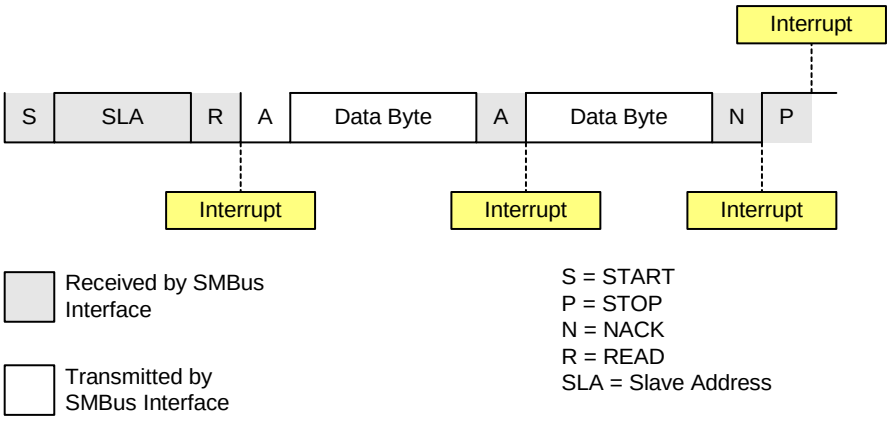
**Figure 20.5. Typical Master Receiver Sequence**



### 20.3.3. 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 20.6. Typical Slave Transmitter Sequence**

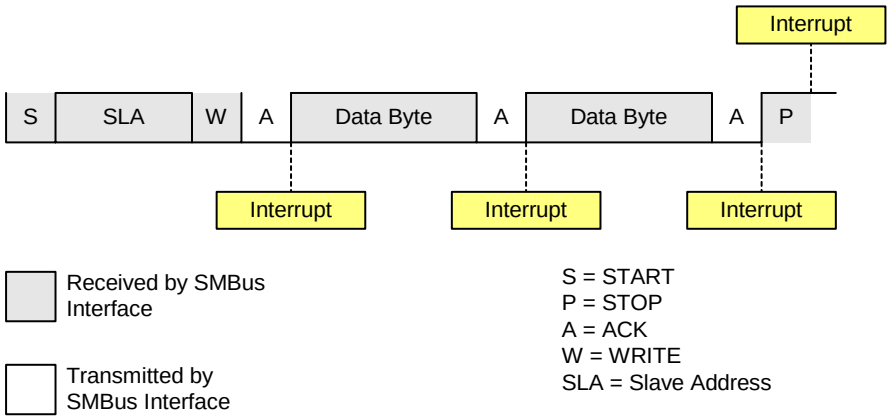



### 20.3.4. Slave Receiver Mode

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 20.7. Typical Slave Receiver Sequence**



## 20.4. SMBus Special Function Registers

The SMBus0 serial interface is accessed and controlled through five SFRs: 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.

### 20.4.1. 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 Figure 20.9, SMBus0 Clock Rate Register).

When the TOE bit in SMB0CN is set to logic 1, Timer 4 is used to detect SCL low timeouts. If Timer 4 is enabled (see Section “24.2. Timer 2, Timer 3, and Timer 4” on page 295), Timer 4 is forced to reload when SCL is high, and forced to count when SCL is low. With Timer 4 enabled and configured to overflow after

25 ms (and TOE set), a Timer 4 overflow indicates a SCL low timeout; the Timer 4 interrupt service routine can then be used to reset SMBus0 communication in the event of an SCL low timeout.

**Figure 20.8. SMB0CN: SMBus0 Control Register**

| 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 4, if enabled.

### 20.4.2. Clock Rate Register

**Figure 20.9. SMB0CR: SMBus0 Clock Rate 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: 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 Hz:

$$SMB0CR < ((288 - 0.85 \cdot SYSCLK)/(1.125 \cdot 10^{6}))$$

The resulting SCL signal high and low times are given by the following equations:

$$T_{LOW} = (256 - SMB0CR)/SYSCLK$$

$$T_{HIGH} \cong (258 - SMB0CR)/SYSCLK + 625ns$$

Using the same value of SMB0CR from above, the Bus Free Timeout period is given in the following equation:

$$T_{BFT} \cong 10 \times \frac{(256 - SMB0CR) + 1}{SYSCLK}$$

### 20.4.3. 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.

**Figure 20.10. SMB0DAT: SMBus0 Data 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: 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.

### 20.4.4. 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.

**Figure 20.11. SMB0ADR: SMBus0 Address Register**

| 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.

### 20.4.5. 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.

**Figure 20.12. SMB0STA: SMBus0 Status Register**

| 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 20.1. 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. |

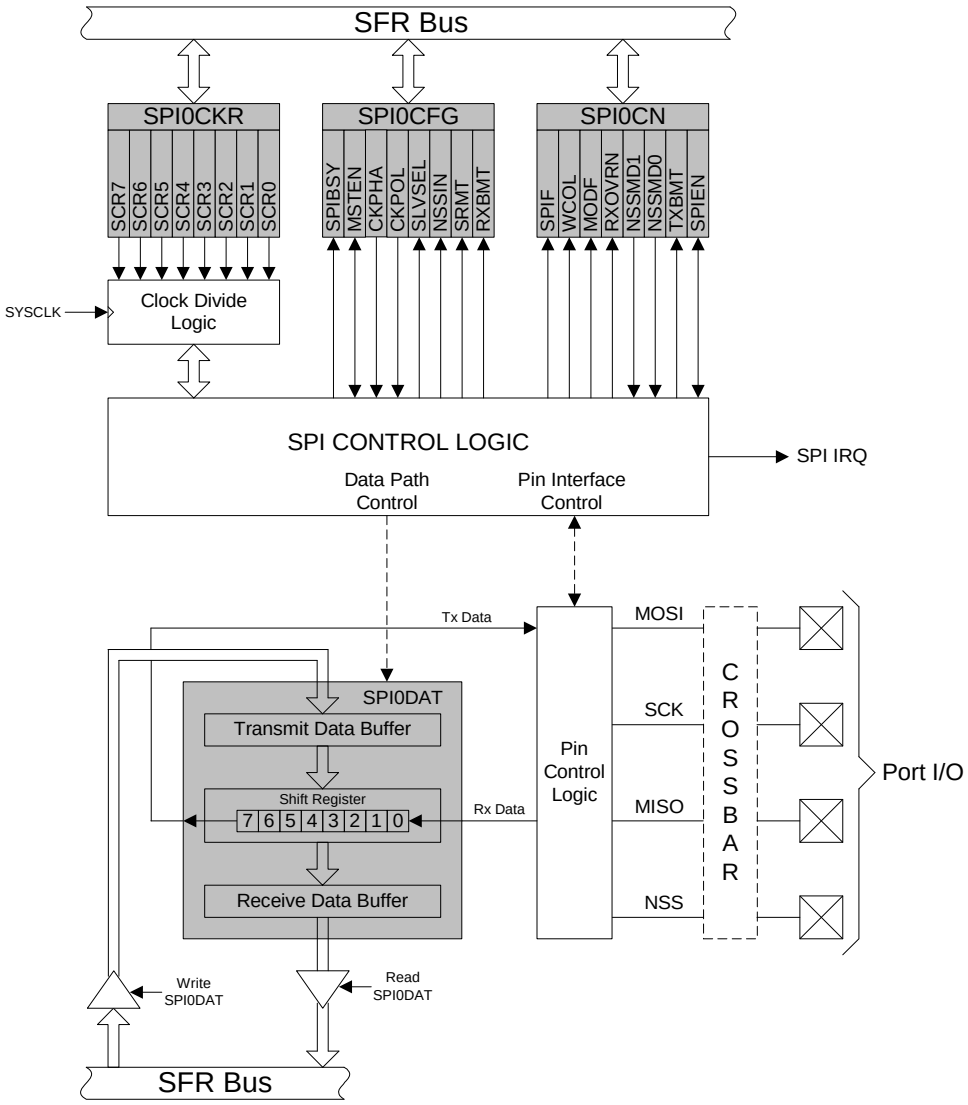
**Table 20.1. SMB0STA Status Codes and States**

| 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. |

# 21. 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 21.1. SPI Block Diagram**



## 21.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 21.1.1. 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.

### 21.1.2. 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.

### 21.1.3. 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.

### 21.1.4. 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 21.2, Figure 21.3, and Figure 21.4 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 203 for general purpose port I/O and crossbar information.

## 21.2. 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 21.2 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 21.3 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 21.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

Figure 21.2. Multiple-Master Mode Connection Diagram

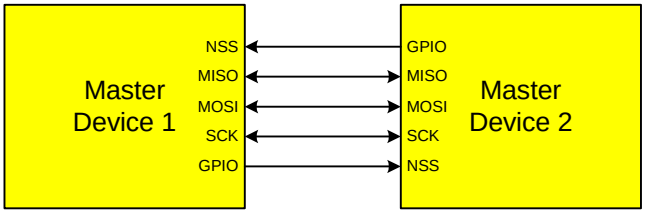



Figure 21.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

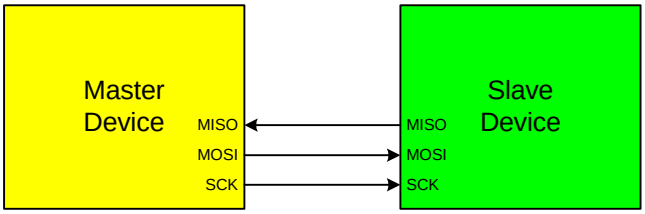
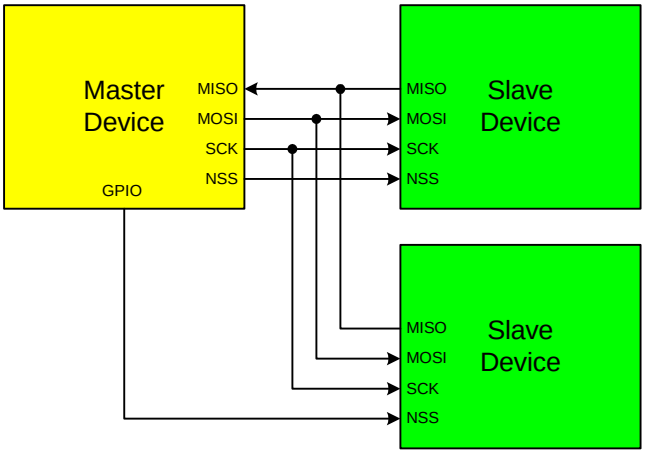



Figure 21.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram



## 21.3. 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 21.4 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 21.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

## 21.4. 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.

## 21.5. 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 21.5. For slave mode, the clock and data relationships are shown in Figure 21.6 and Figure 21.7. 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, C8051F31x, C8051F32x, and C8051F33x

The SPI0 Clock Rate Register (SPI0CKR) as shown in Figure 21.10 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 21.5. Master Mode Data/Clock Timing**

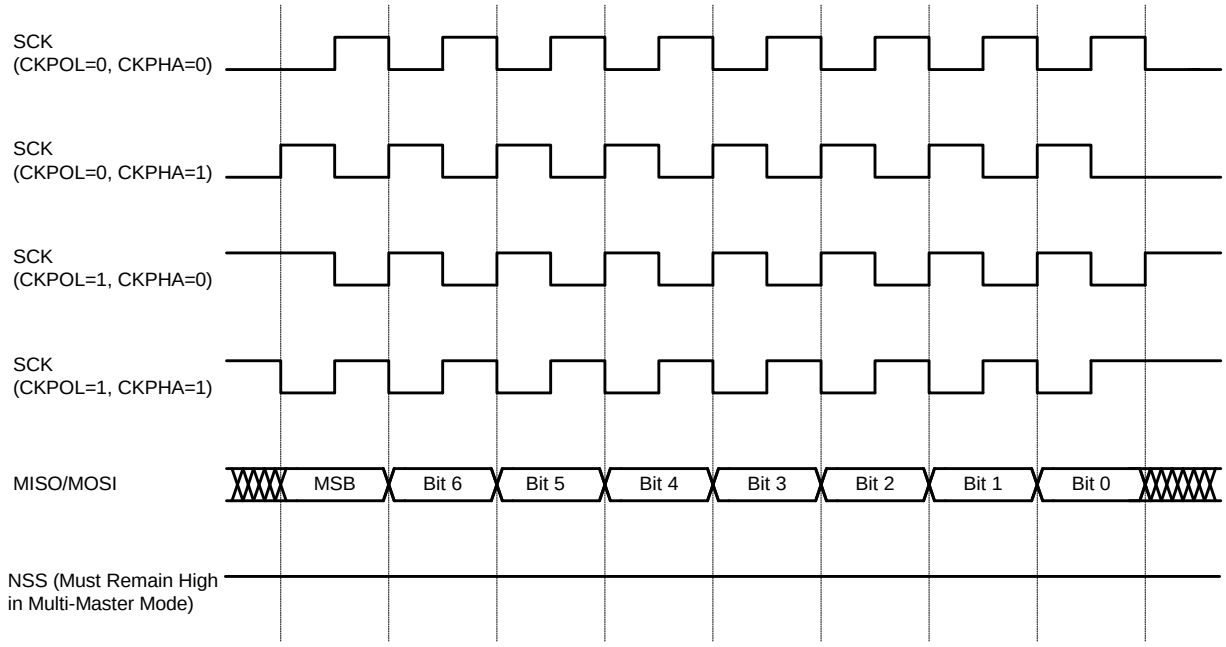


**Figure 21.6. Slave Mode Data/Clock Timing (CKPHA = 0)**

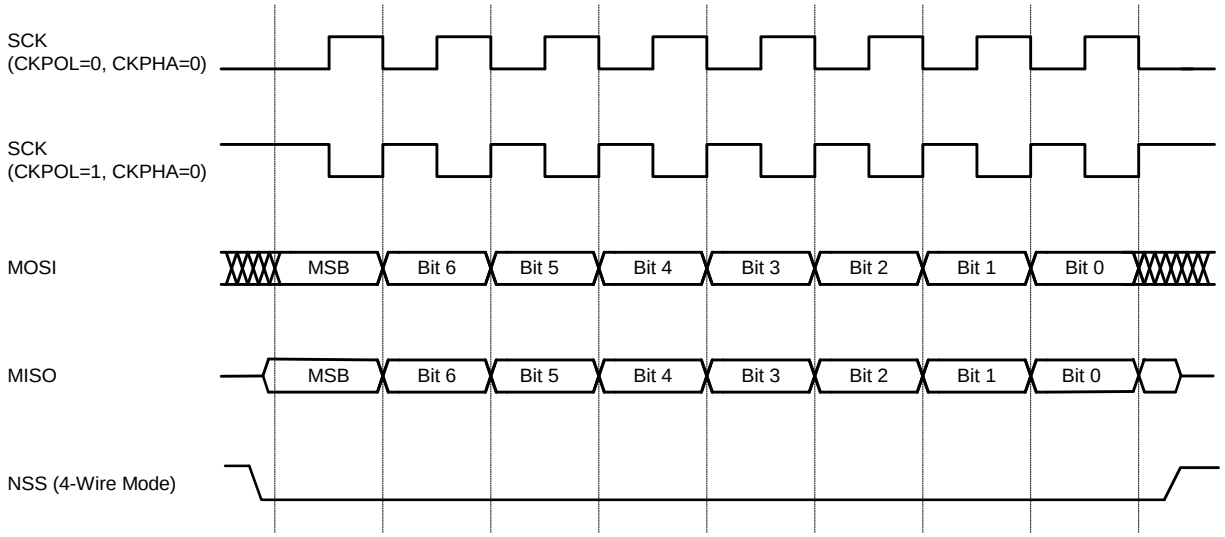
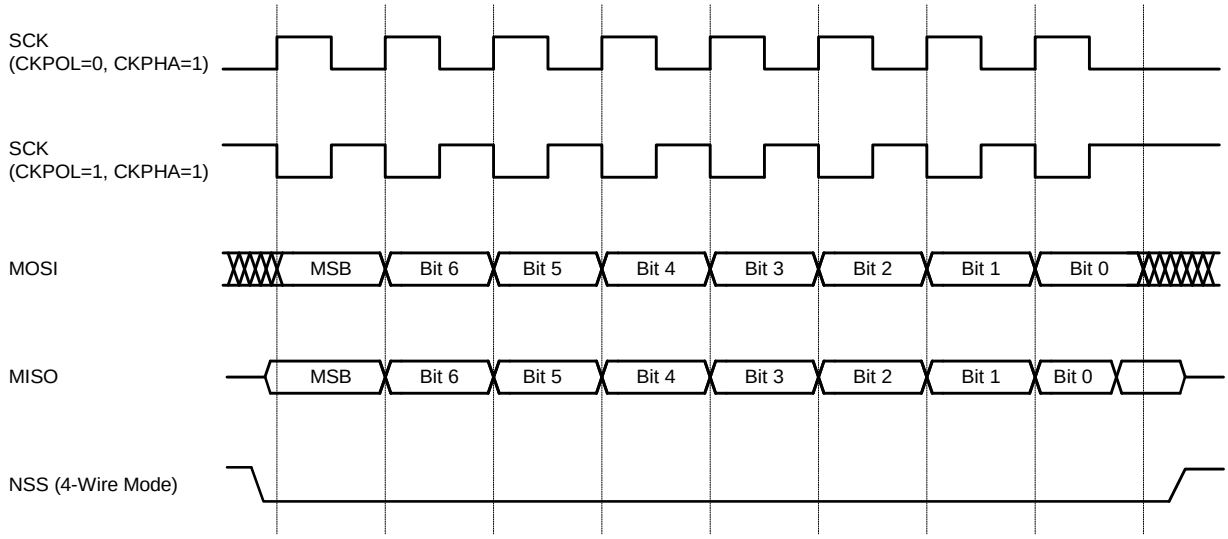



**Figure 21.7. Slave Mode Data/Clock Timing (CKPHA = 1)**



## 21.6. 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.

**Figure 21.8. SPI0CFG: SPI0 Configuration Register**

| 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.

†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 21.1 for timing parameters.

**Figure 21.9. SPI0CN: SPI0 Control Register**

| 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 “21.2. SPI0 Master Mode Operation” on page 253 and Section “21.3. SPI0 Slave Mode Operation” on page 255).
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.

**Figure 21.10. SPI0CKR: SPI0 Clock Rate Register**

| 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.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

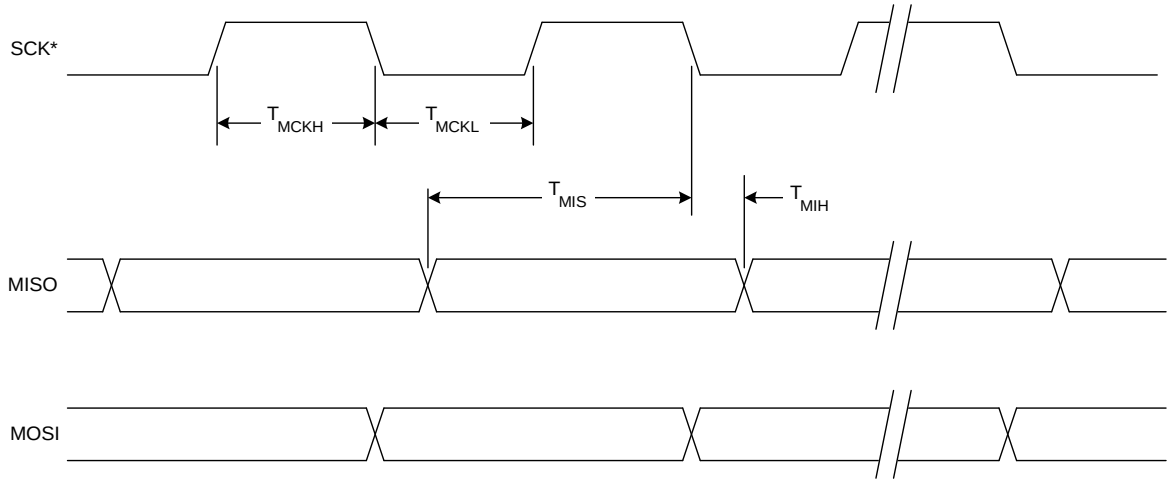
**Figure 21.11. SPI0DAT: SPI0 Data Register**

| 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: 0x9B
SFR Page: 0

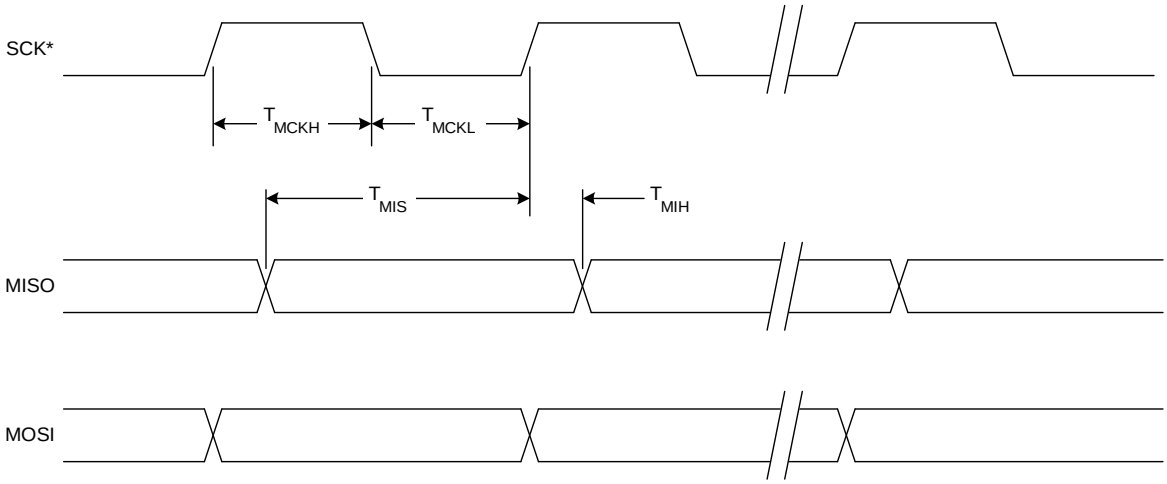
Bits 7-0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

**Figure 21.12. SPI Master Timing (CKPHA = 0)**



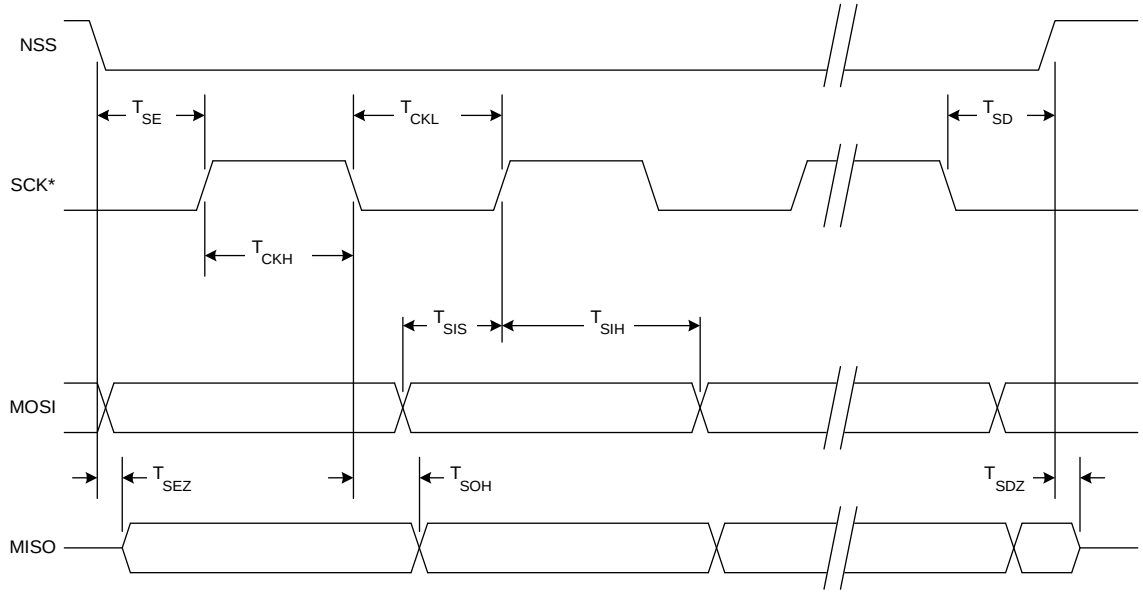

* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 21.13. SPI Master Timing (CKPHA = 1)**



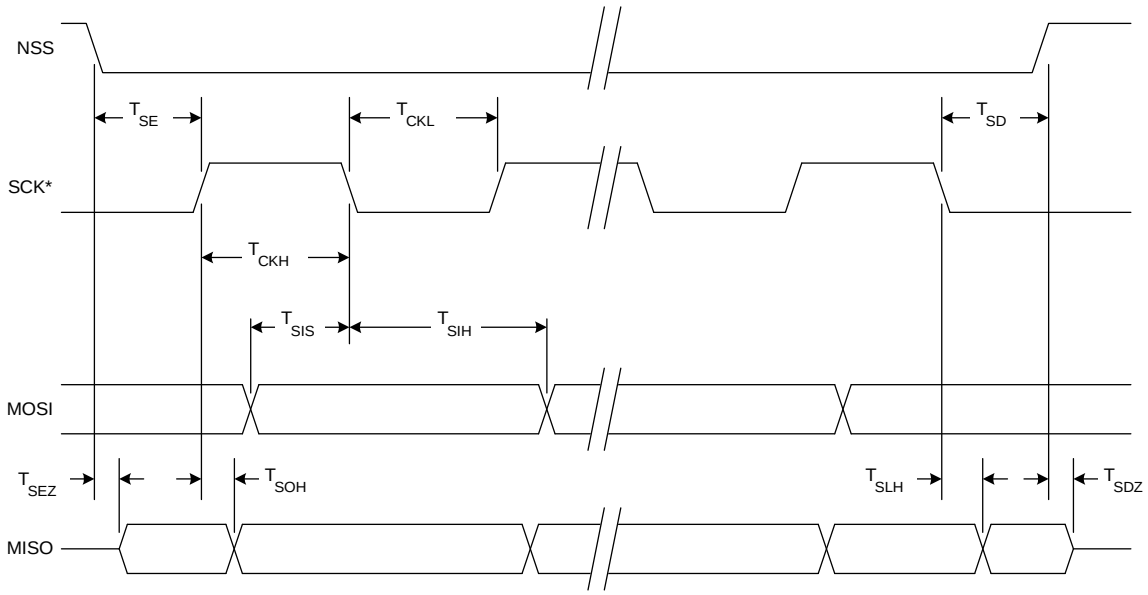

* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 21.14. SPI Slave Timing (CKPHA = 0)**




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 21.15. SPI Slave Timing (CKPHA = 1)**




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Table 21.1. SPI Slave Timing Parameters**

| Parameter | Description | Min | Max | Units |
|---|---|---|---|---|
| **Master Mode Timing**† (See Figure 21.12 and Figure 21.13) | | | | |
| $T_{MCKH}$ | SCK High Time | $1*T_{SYSCLK}$ | | ns |
| $T_{MCKL}$ | SCK Low Time | $1*T_{SYSCLK}$ | | ns |
| $T_{MIS}$ | MISO Valid to SCK Shift Edge | $1*T_{SYSCLK}$ + 20 | | ns |
| $T_{MIH}$ | SCK Shift Edge to MISO Change | 0 | | ns |
| **Slave Mode Timing**† (See Figure 21.14 and Figure 21.15) | | | | |
| $T_{SE}$ | NSS Falling to First SCK Edge | $2*T_{SYSCLK}$ | | ns |
| $T_{SD}$ | Last SCK Edge to NSS Rising | $2*T_{SYSCLK}$ | | ns |
| $T_{SEZ}$ | NSS Falling to MISO Valid | | $4*T_{SYSCLK}$ | ns |
| $T_{SDZ}$ | NSS Rising to MISO High-Z | | $4*T_{SYSCLK}$ | ns |
| $T_{CKH}$ | SCK High Time | $5*T_{SYSCLK}$ | | ns |
| $T_{CKL}$ | SCK Low Time | $5*T_{SYSCLK}$ | | ns |
| $T_{SIS}$ | MOSI Valid to SCK Sample Edge | $2*T_{SYSCLK}$ | | ns |
| $T_{SIH}$ | SCK Sample Edge to MOSI Change | $2*T_{SYSCLK}$ | | ns |
| $T_{SOH}$ | SCK Shift Edge to MISO Change | | $4*T_{SYSCLK}$ | ns |
| $T_{SLH}$ | Last SCK Edge to MISO Change (CKPHA = 1 ONLY) | $6*T_{SYSCLK}$ | $8*T_{SYSCLK}$ | ns |

†$T_{SYSCLK}$ is equal to one period of the device system clock (SYSCLK).

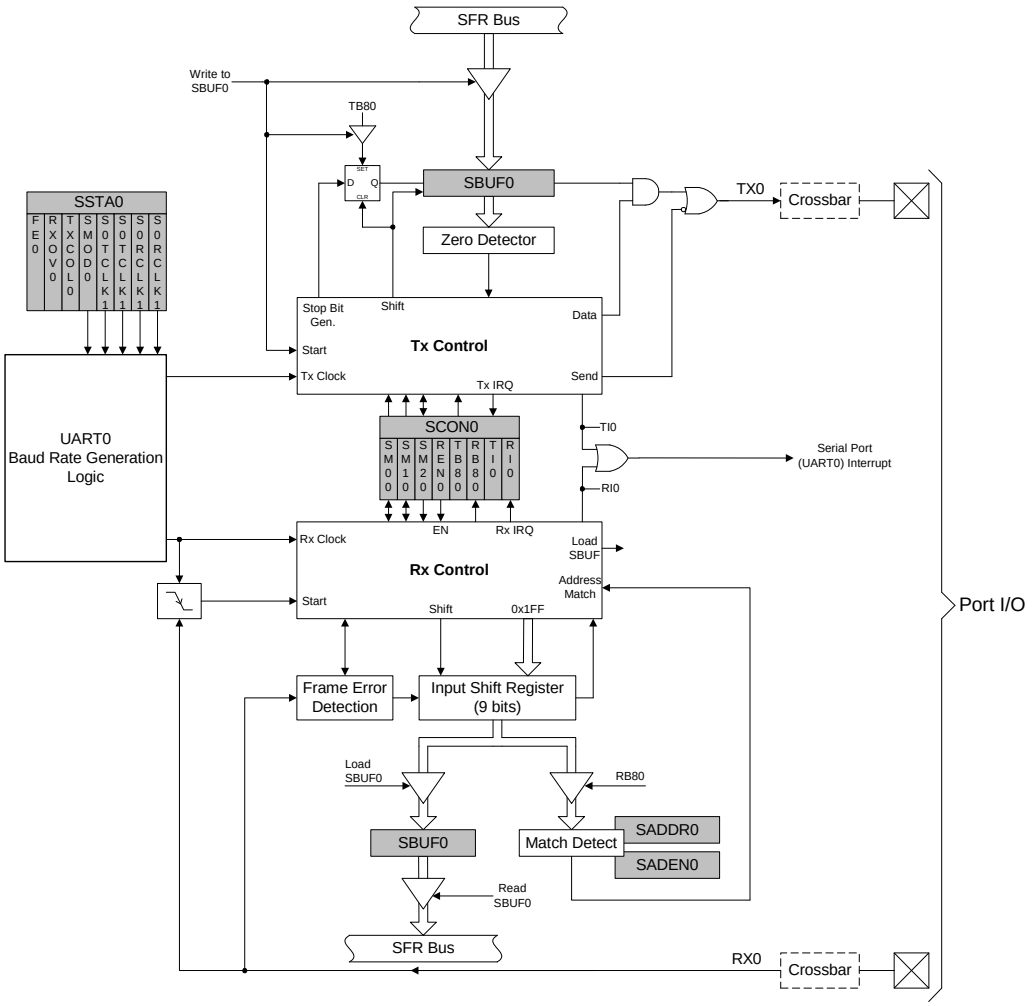
# 22. UART0

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 SFRs, 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 22.1. UART0 Block Diagram**



## 22.1. 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 22.1.

**Table 22.1. 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 |

### 22.1.1. 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 22.3).

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 22.2), 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 pull-up will typically be required.

**Figure 22.2. UART0 Mode 0 Timing Diagram**

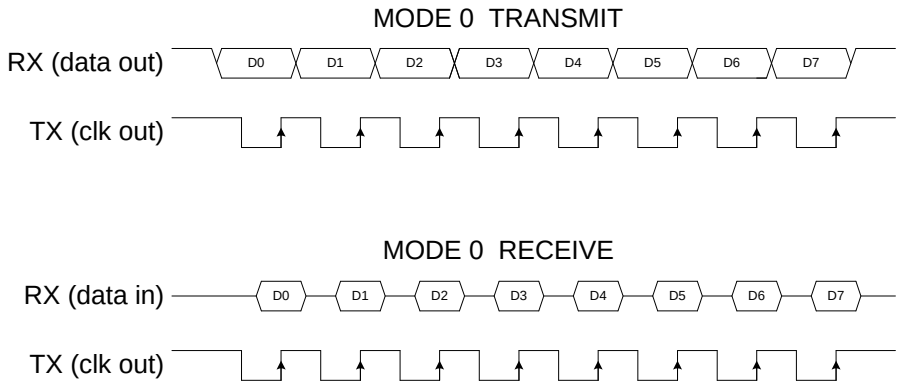
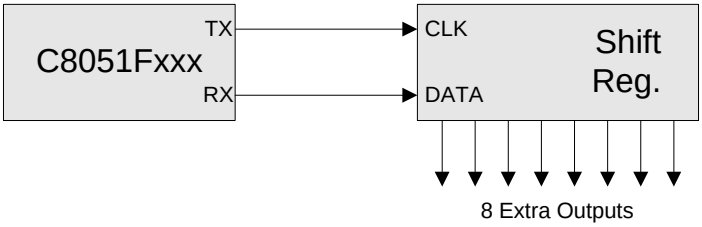



**Figure 22.3. UART0 Mode 0 Interconnect**



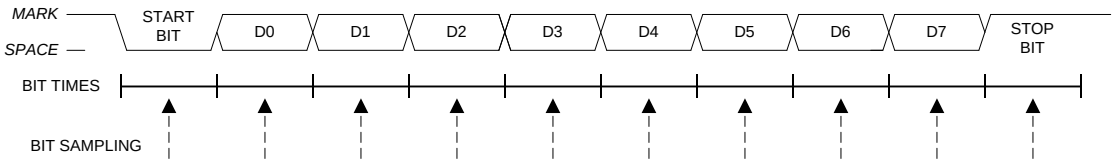

### 22.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

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 22.4. 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 Figure 22.9). 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 22.1.

**Equation 22.1. 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 “24.1. Timer 0 and Timer 1” on page 287. The Timer 1 overflow rate is calculated as shown in Equation 22.2.

**Equation 22.2. 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 22.3.

**Equation 22.3. 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 22.4.

**Equation 22.4. Timer 2, 3, or 4 Overflow Rate**

$$\text{Timer234_OverflowRate} = \text{TnCLK}/(65536 - \text{RCAPn})$$

### 22.1.3. 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 22.2). 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 22.2.

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 22.5. Mode 2 Baud Rate**

$$BaudRate = 2^{SMOD0} \times \left(\frac{SYSCLK}{64}\right)$$

**Figure 22.5. UART0 Modes 2 and 3 Timing Diagram**

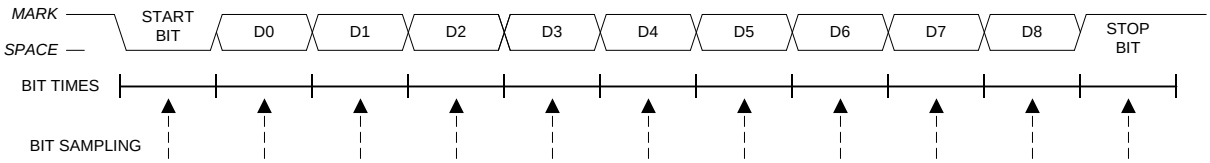
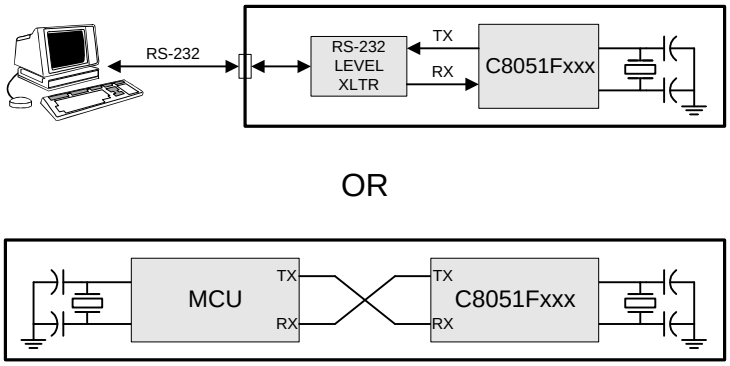


**Figure 22.6. UART0 Modes 1, 2, and 3 Interconnect Diagram**




### 22.1.4. 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 22.1 and Equation 22.3. Multiprocessor communications and hardware address recognition are supported, as described in Section 22.2.

## 22.2. 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.

### 22.2.1. Configuration of a Masked Address

The UART0 address is configured via two SFRs: 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.

### 22.2.2. 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 |

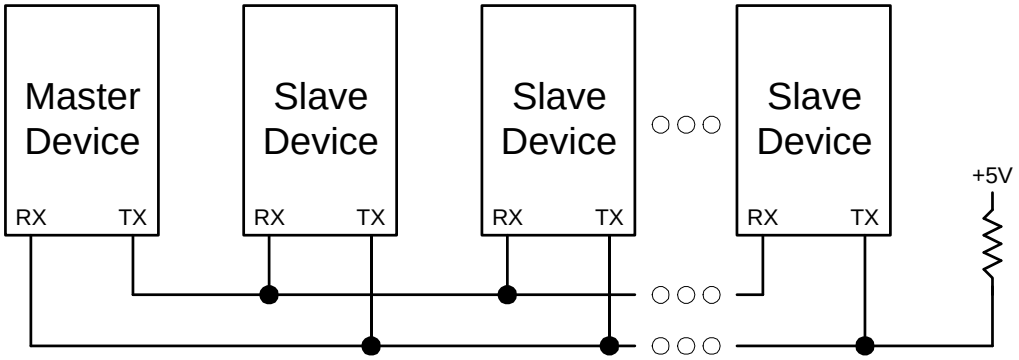
| **Example 6, SLAVE #3** | |
|---|---|
| SADDR0 | = 00110101 |
| SADEN0 | = 11000000 |
| Broadcast Address | = 11110101 |

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 22.7. UART Multi-Processor Mode Interconnect Diagram**




## 22.3. Frame and Transmission Error Detection

**All Modes:**
The Transmit Collision bit (TXCOL0 bit in register SCON0) reads ‘1’ if user software writes data to the SBUF0 register while a transmit is in progress. Note that the TXCOL0 bit is also used as the SM20 bit when written by user software. This bit does not generate an interrupt.

**Modes 1, 2, and 3:**

The Receive Overrun bit (RXOV0 in register SCON0) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. Note that the RXOV0 bit is also used as the SM10 bit when written by user software. The Frame Error bit (FE0 in register SSTA0) reads ‘1’ if an invalid (low) STOP bit is detected. Note that the FE0 bit is also used as the SM00 bit when written by user software. The RXOV0 and FE0 bits do not generate interrupts.

**Table 22.2. Oscillator Frequencies for Standard Baud Rates**

| Oscillator frequency (MHz) | Divide Factor | Timer 1 Reload Value* | Timer 2, 3, or 4 Reload Value | Resulting Baud Rate (Hz)** |
|---|---|---|---|---|
| 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 |
| 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 |
| 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 |

* Assumes SMOD0=1 and T1M=1.
** Numbers in parenthesis show the actual baud rate.

**Figure 22.8. SCON0: UART0 Control Register**

| 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.

**Figure 22.9. SSTA0: UART0 Status and Clock Selection Register**

| 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.

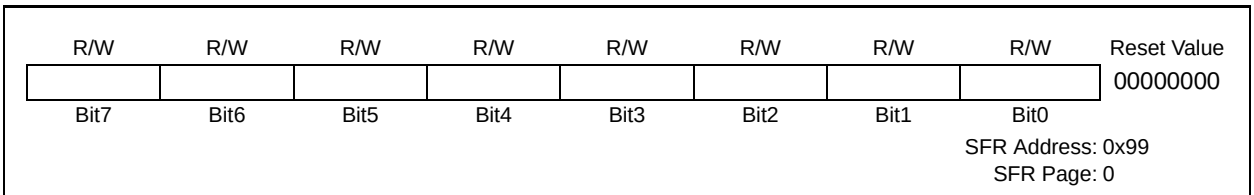
| S0TCLK1 | S0TCLK0 | **Serial Transmit Baud Rate Clock Source** |
|---|---|---|
| 0 | 0 | Timer 1 generates UART0 TX Baud Rate |
| 0 | 1 | Timer 2 Overflow generates UART0 TX baud rate |
| 1 | 0 | Timer 3 Overflow generates UART0 TX baud rate |
| 1 | 1 | Timer 4 Overflow generates UART0 TX baud rate |

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 |

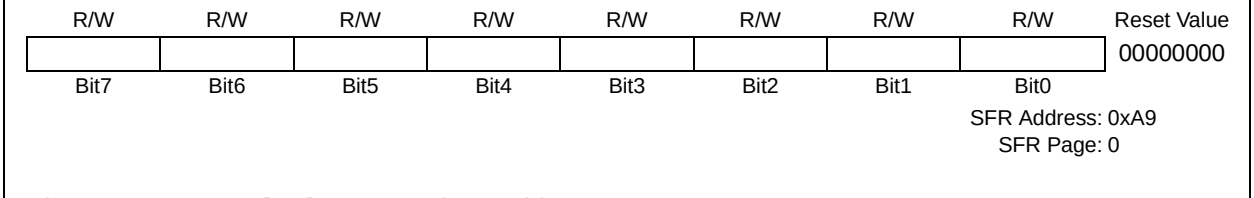
† Note: FE0, RXOV0, and TXCOL0 are flags only, and no interrupt is generated by these conditions.

**Figure 22.10. SBUF0: UART0 Data Buffer Register**



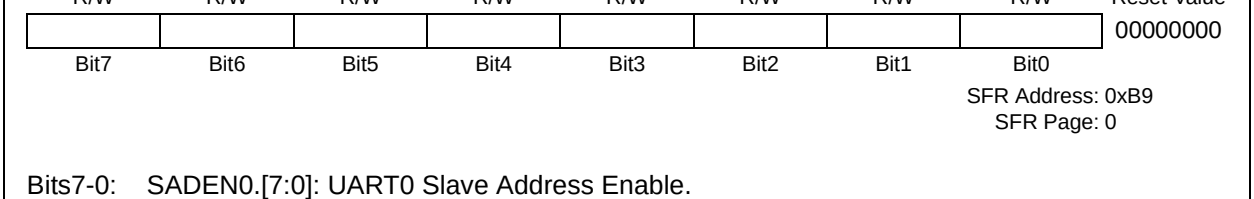

Bits7-0: SBUF0.[7:0]: UART0 Buffer Bits 7-0 (MSB-LSB).
This is actually two registers; a transmit and a receive buffer register. When data is moved to SBUF0, it goes to the transmit buffer and is held for serial transmission. Moving a byte to SBUF0 is what initiates the transmission. When data is moved from SBUF0, it comes from the receive buffer.

**Figure 22.11. SADDR0: UART0 Slave Address Register**




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”.

**Figure 22.12. SADEN0: UART0 Slave Address Enable Register**




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.

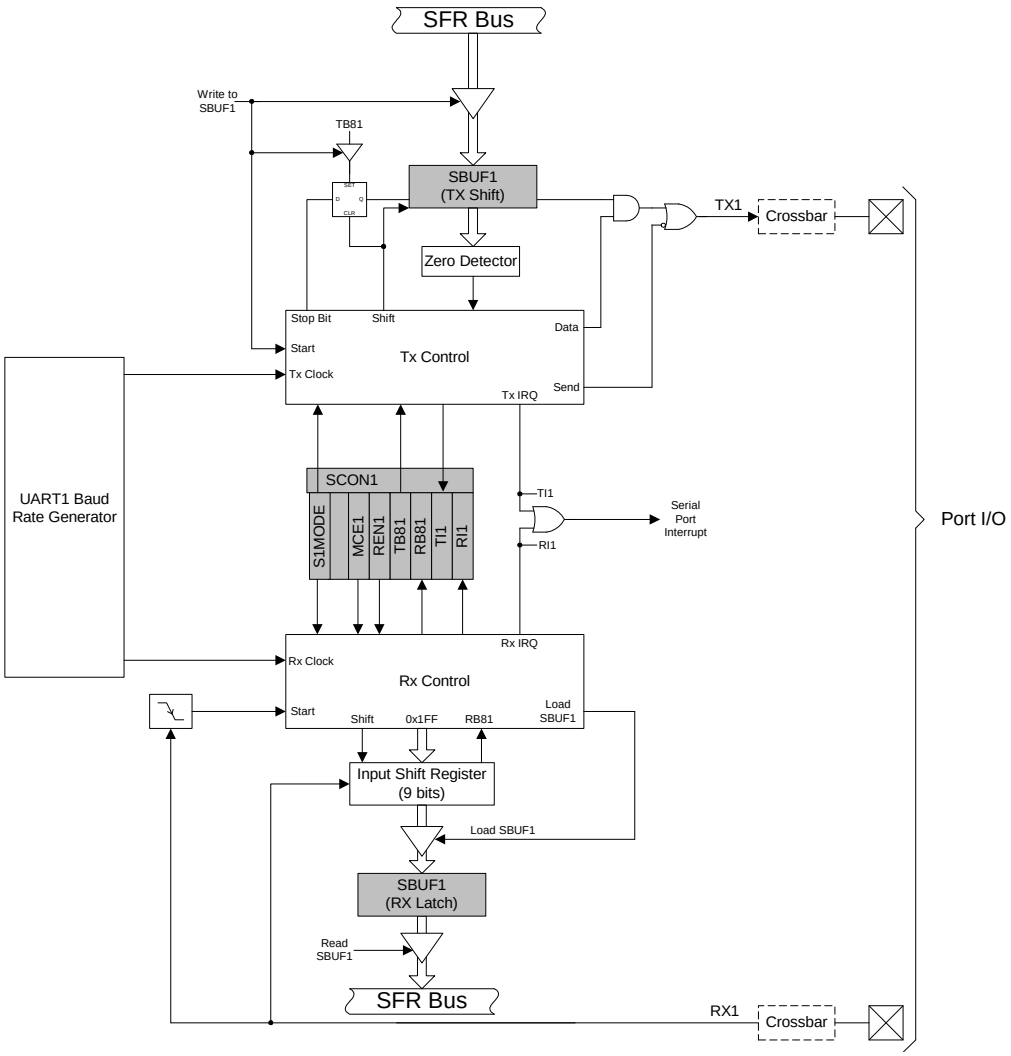
# 23. UART1

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 “23.1. Enhanced Baud Rate Generation” on page 278). 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).

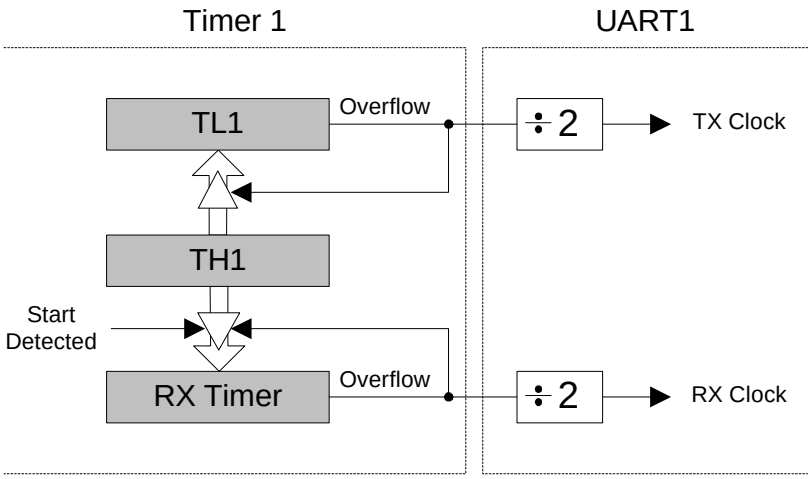
**Figure 23.1. UART1 Block Diagram**



## 23.1. 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 23.2), 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 23.2. UART1 Baud Rate Logic**




Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “24.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 289). 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 23.1.

**Equation 23.1. UART1 Baud Rate**

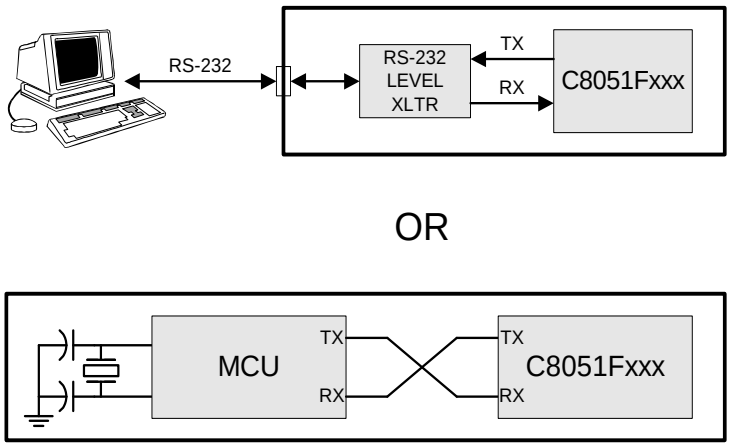
$$UartBaudRate = \frac{T1_{CLK}}{(256 - T1H)} \times \frac{1}{2}$$

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 “24.1. Timer 0 and Timer 1” on page 287. A quick reference for typical baud rates and system clock frequencies is given in Table 23.1 through Table 23.6. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1 (see Section “24.1. Timer 0 and Timer 1” on page 287 for more details).

## 23.2. 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 23.3. UART Interconnect Diagram**



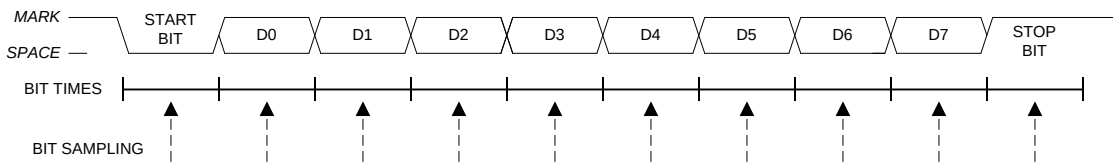

### 23.2.1. 8-Bit UART

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 23.4. 8-Bit UART Timing Diagram**

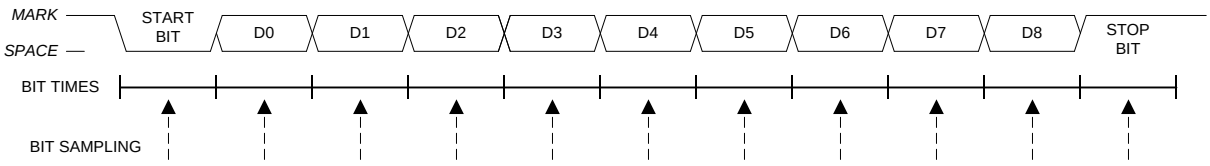


### 23.2.2. 9-Bit UART

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 23.5. 9-Bit UART Timing Diagram**



## 23.3. 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 23.6. UART Multi-Processor Mode Interconnect Diagram**

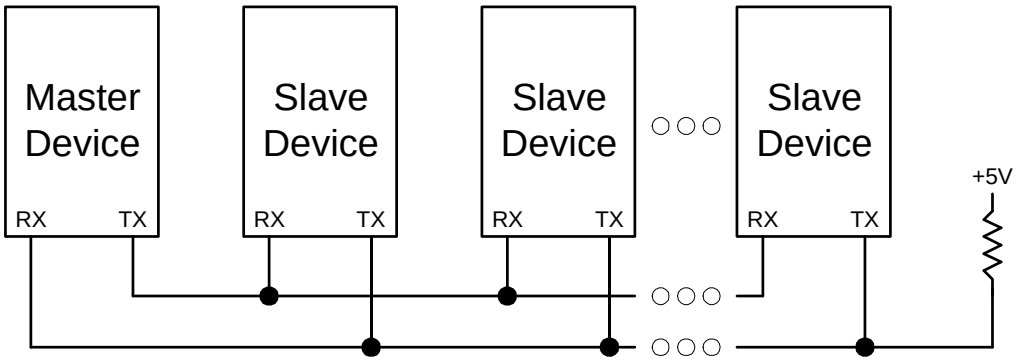


**Figure 23.7. SCON1: Serial Port 1 Control Register**

| 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: 8-bit UART with Variable Baud Rate.
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.
S1MODE = 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.
S1MODE = 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.

**Figure 23.8. SBUF1: Serial (UART1) Port Data Buffer 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: 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 23.1. Timer Settings for Standard Baud Rates Using the Internal Oscillator**

| | Frequency: 24.5 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 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

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

**Table 23.2. Timer Settings for Standard Baud Rates Using an External 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

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

**Table 23.3. Timer Settings for Standard Baud Rates Using an External 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

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

**Table 23.4. Timer Settings for Standard Baud Rates Using an External Oscillator**

| | Frequency: 18.432 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% | 80 | SYSCLK | XX | 1 | 0xD8 |
| | 115200 | 0.00% | 160 | SYSCLK | XX | 1 | 0xB0 |
| | 57600 | 0.00% | 320 | SYSCLK | XX | 1 | 0x60 |
| | 28800 | 0.00% | 640 | SYSCLK / 4 | 01 | 0 | 0xB0 |
| | 14400 | 0.00% | 1280 | SYSCLK / 4 | 01 | 0 | 0x60 |
| | 9600 | 0.00% | 1920 | SYSCLK / 12 | 00 | 0 | 0xB0 |
| | 2400 | 0.00% | 7680 | SYSCLK / 48 | 10 | 0 | 0xB0 |
| | 1200 | 0.00% | 15360 | SYSCLK / 48 | 10 | 0 | 0x60 |
| SYSCLK from Internal Osc. | 230400 | 0.00% | 80 | EXTCLK / 8 | 11 | 0 | 0xFB |
| | 115200 | 0.00% | 160 | EXTCLK / 8 | 11 | 0 | 0xF6 |
| | 57600 | 0.00% | 320 | EXTCLK / 8 | 11 | 0 | 0xEC |
| | 28800 | 0.00% | 640 | EXTCLK / 8 | 11 | 0 | 0xD8 |
| | 14400 | 0.00% | 1280 | EXTCLK / 8 | 11 | 0 | 0xB0 |
| | 9600 | 0.00% | 1920 | EXTCLK / 8 | 11 | 0 | 0x88 |

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

**Table 23.5. Timer Settings for Standard Baud Rates Using an External Oscillator**

| | Frequency: 11.0592 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% | 48 | SYSCLK | XX | 1 | 0xE8 |
| | 115200 | 0.00% | 96 | SYSCLK | XX | 1 | 0xD0 |
| | 57600 | 0.00% | 192 | SYSCLK | XX | 1 | 0xA0 |
| | 28800 | 0.00% | 384 | SYSCLK | XX | 1 | 0x40 |
| | 14400 | 0.00% | 768 | SYSCLK / 12 | 00 | 0 | 0xE0 |
| | 9600 | 0.00% | 1152 | SYSCLK / 12 | 00 | 0 | 0xD0 |
| | 2400 | 0.00% | 4608 | SYSCLK / 12 | 00 | 0 | 0x40 |
| | 1200 | 0.00% | 9216 | SYSCLK / 48 | 10 | 0 | 0xA0 |
| SYSCLK from Internal Osc. | 230400 | 0.00% | 48 | EXTCLK / 8 | 11 | 0 | 0xFD |
| | 115200 | 0.00% | 96 | EXTCLK / 8 | 11 | 0 | 0xFA |
| | 57600 | 0.00% | 192 | EXTCLK / 8 | 11 | 0 | 0xF4 |
| | 28800 | 0.00% | 384 | EXTCLK / 8 | 11 | 0 | 0xE8 |
| | 14400 | 0.00% | 768 | EXTCLK / 8 | 11 | 0 | 0xD0 |
| | 9600 | 0.00% | 1152 | EXTCLK / 8 | 11 | 0 | 0xB8 |

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

**Table 23.6. Timer Settings for Standard Baud Rates Using an External Oscillator**

| | Frequency: 3.6864 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% | 16 | SYSCLK | XX | 1 | 0xF8 |
| | 115200 | 0.00% | 32 | SYSCLK | XX | 1 | 0xF0 |
| | 57600 | 0.00% | 64 | SYSCLK | XX | 1 | 0xE0 |
| | 28800 | 0.00% | 128 | SYSCLK | XX | 1 | 0xC0 |
| | 14400 | 0.00% | 256 | SYSCLK | XX | 1 | 0x80 |
| | 9600 | 0.00% | 384 | SYSCLK | XX | 1 | 0x40 |
| | 2400 | 0.00% | 1536 | SYSCLK / 12 | 00 | 0 | 0xC0 |
| | 1200 | 0.00% | 3072 | SYSCLK / 12 | 00 | 0 | 0x80 |
| SYSCLK from Internal Osc. | 230400 | 0.00% | 16 | EXTCLK / 8 | 11 | 0 | 0xFF |
| | 115200 | 0.00% | 32 | EXTCLK / 8 | 11 | 0 | 0xFE |
| | 57600 | 0.00% | 64 | EXTCLK / 8 | 11 | 0 | 0xFC |
| | 28800 | 0.00% | 128 | EXTCLK / 8 | 11 | 0 | 0xF8 |
| | 14400 | 0.00% | 256 | EXTCLK / 8 | 11 | 0 | 0xF0 |
| | 9600 | 0.00% | 384 | EXTCLK / 8 | 11 | 0 | 0xE8 |

X = Don't care

†SCA1-SCA0 and T1M bit definitions can be found in Section 24.1.

# 24. 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 ADC's, DAC's, 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. Timers 2, 3, 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 |
| 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 Figure 24.6 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.

## 24.1. 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 "13.3.5. Interrupt Register Descriptions" on page 154); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section 13.3.5). 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.

### 24.1.1. 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 205 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 Figure 24.6).

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 “13.3.5. Interrupt Register Descriptions” on page 154), facilitating pulse width measurements.

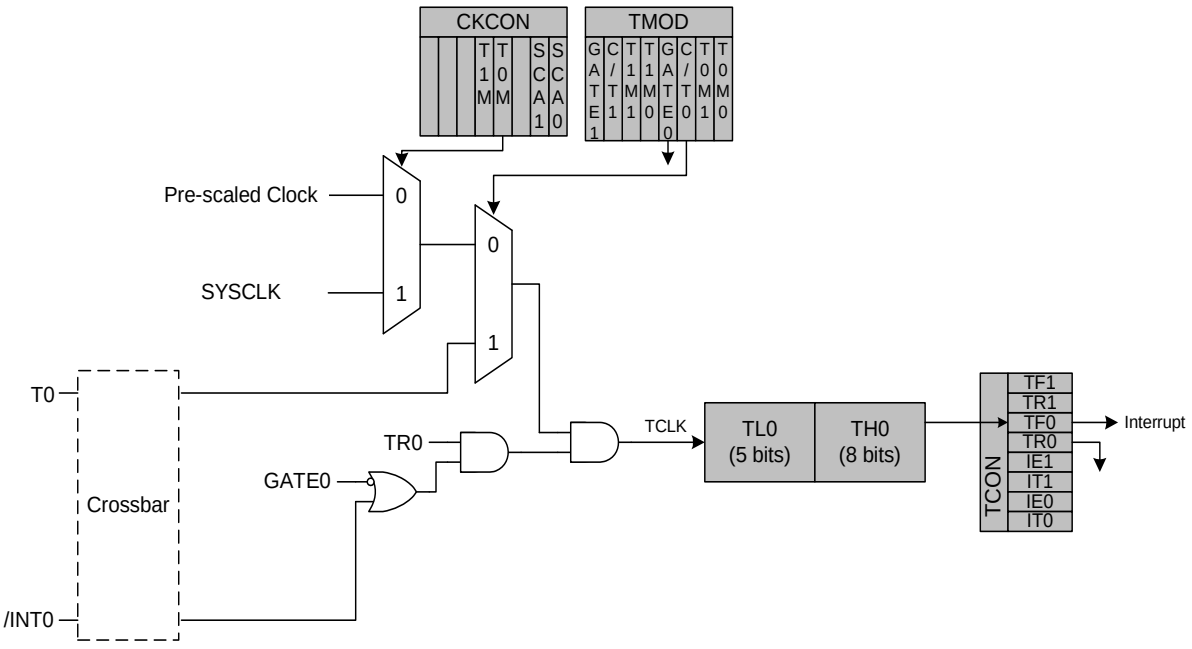
| TR0 | GATE0 | /INT0 | Counter/Timer |
|---|---|---|---|
| 0 | X | X | Disabled |
| 1 | 0 | X | Enabled |
| 1 | 1 | 0 | Disabled |
| 1 | 1 | 1 | Enabled |

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 24.1. T0 Mode 0 Block Diagram**



### 24.1.2. Mode 1: 16-bit Counter/Timer

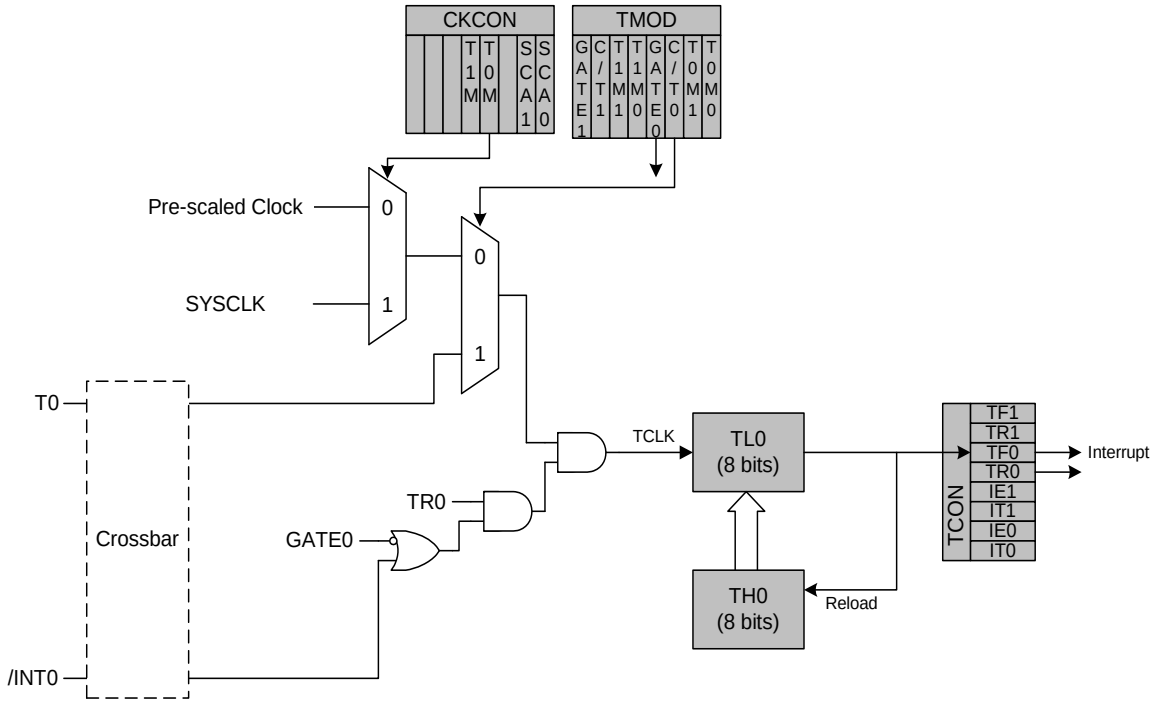
Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

### 24.1.3. 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 24.2. T0 Mode 2 Block Diagram**



### 24.1.4. 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 24.3. T0 Mode 3 Block Diagram**

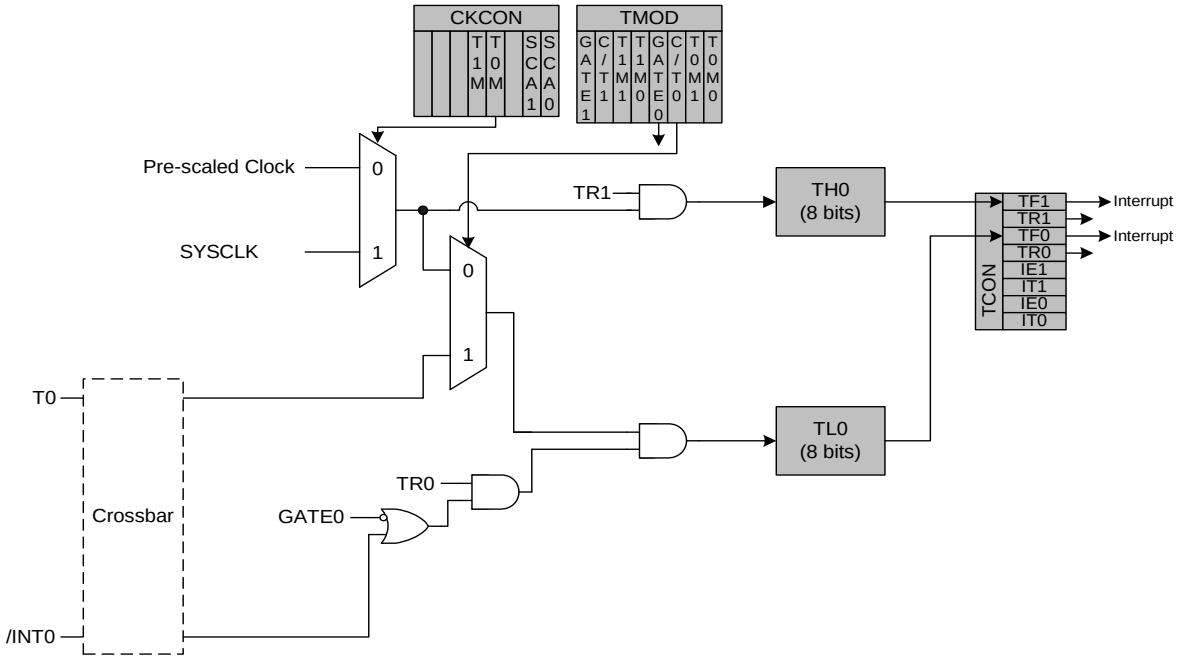


**Figure 24.4. TCON: Timer Control Register**

| 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.

**Figure 24.5. TMOD: Timer Mode Register**

| 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 |

**Figure 24.6. CKCON: Clock Control Register**

| 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 | |

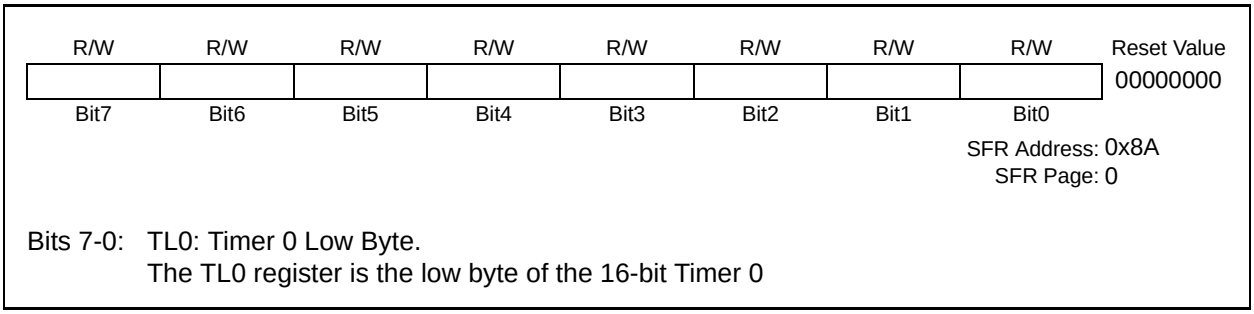
SFR Address: 0x8E
SFR Page: 0

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† |

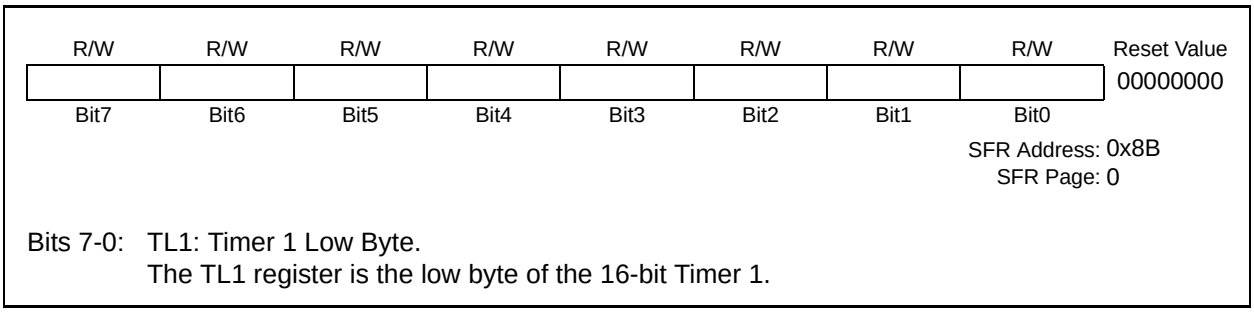
†Note: External clock divided by 8 is synchronized with the system clock, and external clock must be less than or equal to the system clock frequency to operate the timer in this mode.

**Figure 24.7. TL0: Timer 0 Low Byte**



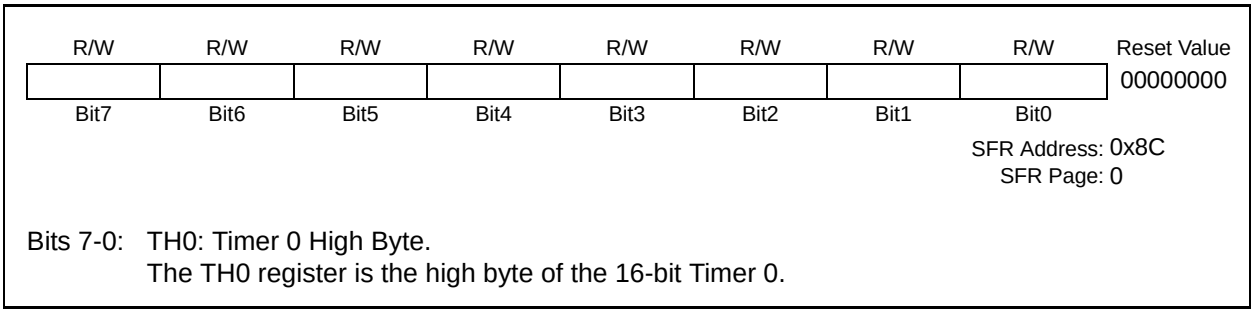

Bits 7-0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0

**Figure 24.8. TL1: Timer 1 Low Byte**



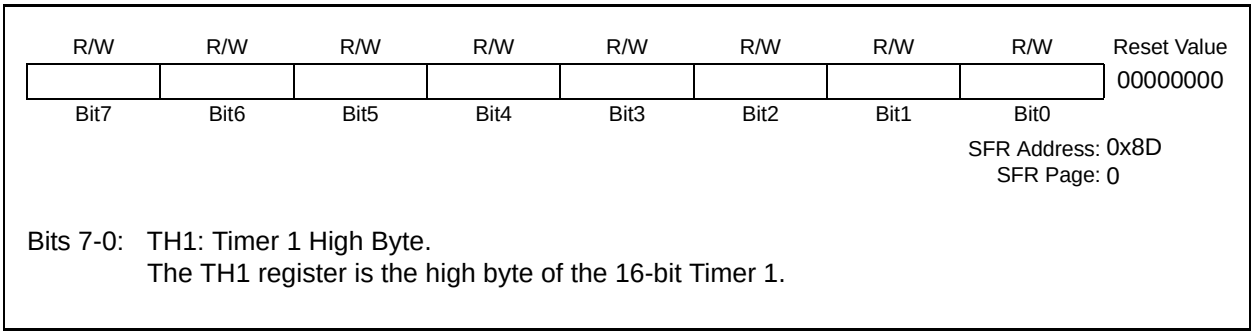

Bits 7-0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

**Figure 24.9. TH0: Timer 0 High Byte**




Bits 7-0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

**Figure 24.10. TH1: Timer 1 High Byte**




Bits 7-0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

## 24.2. Timer 2, Timer 3, and Timer 4

Timers 2, 3, and 4 are 16-bit counter/timers, each formed by two 8-bit SFRs: TMRnL (low byte) and TMRnH (high byte) where n = 2, 3, and 4 for timers 2, 3, and 4 respectively. These timers feature auto-reload, capture, and toggle output 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, 3, and 4 Configuration registers (TMRnCF). These timers may also be used to generate a square-wave at an external pin. 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 selected timer clock source 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. Refer to Section “18.1. Ports 0 through 3 and the Priority Crossbar Decoder” on page 205 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 (TnCON.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 Figure 24.14). 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).

### 24.2.1. Configuring Timer 2, 3, and 4 to Count Down

Timers 2, 3, and 4 have the ability to count down. When the timer’s respective Decrement Enable Bit (DCENn) in the Timer Configuration Register (See Figure 24.14) 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. 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.**

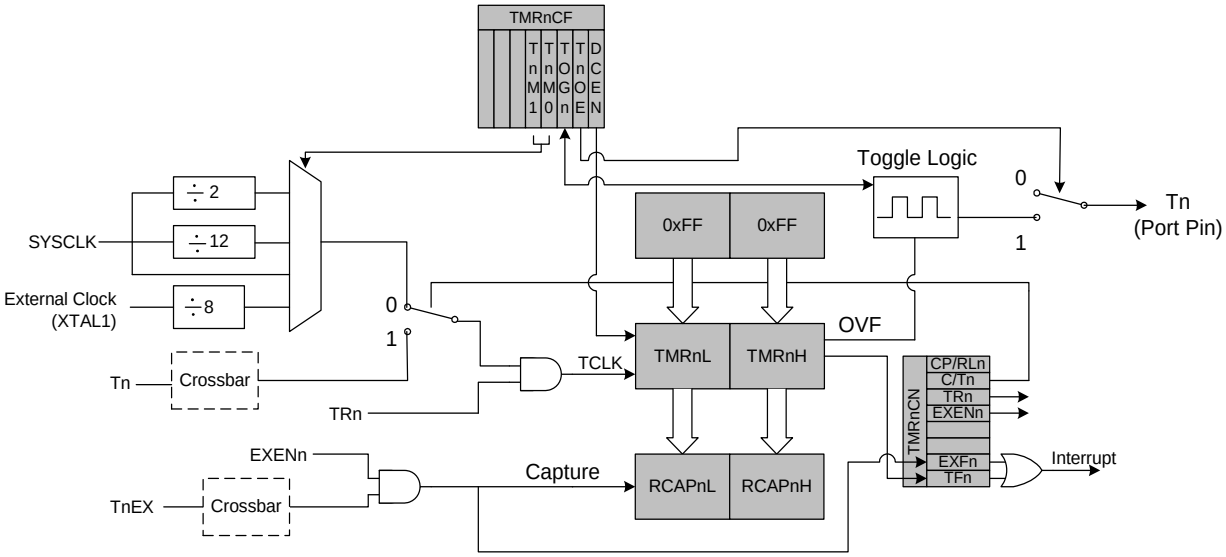
### 24.2.2. Capture Mode

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 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 “13.3. Interrupt Handler” on page 151 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 (TnCON.2) to logic 1. The Timer 2, 3, and 4 respective External Enable EXENn (TnCON.3) must also be set to logic 1 to enable a captures. If EXENn is cleared, transitions on TnEX will be ignored.

**Figure 24.11. T2, 3, and 4 Capture Mode Block Diagram**



### 24.2.3. 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 (configured as an input in the digital crossbar) will cause a timer reload (in addition to timer overflows causing 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. See Section 24.2.1 for information concerning configuration of a timer to count down.

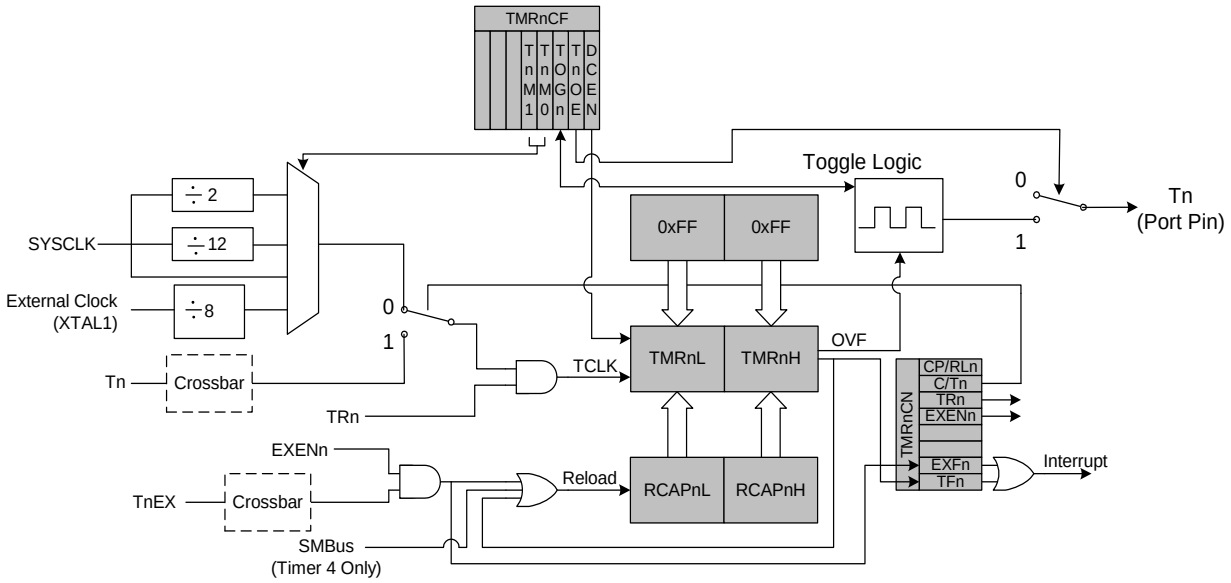
When counting down, the counter/timer will set its overflow/underflow flag (TFn) and cause an interrupt (if enabled) when the value in the timer (TMRnH and TMRnL registers) matches the 16-bit value in the Reload/Capture Registers (RCAPnH and RCAPnL). This is considered an underflow event, and will cause 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 thought of as the most significant bit (MSB) of a 17-bit counter.

.

**Figure 24.12. T2, 3, and 4 Auto-reload Mode Block Diagram**



### 24.2.4. Toggle Output Mode

Timer 2, 3, and 4 have the capability to toggle the state of their respective output port pins (T2, T3, 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 203). 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 24.1. Toggle Mode Square Wave Frequency**

$$F_{sq} = \frac{F_{TCLK}}{2 \cdot (65536 - RCAPn)}$$

**Figure 24.13. TMRnCN: Timer 2, 3, and 4 Control Registers**

| 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, and 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.

**Figure 24.14. TMRnCF: Timer 2, 3, and 4 Configuration Registers**

| | | | 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 an external clock signal routed to Tn (port pin) divided by 8. Clock source is selected as follows:
00: SYSCLK/12
01: SYSCLK
10: EXTERNAL CLOCK/8
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.

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 "Toggle Mode Square Wave Frequency" on page 298.)*
*Configure Port Pin to output squarewave (See* Section "18. Port Input/Output" on page 203*)*
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.

**Figure 24.15. 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.

**Figure 24.16. 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.

**Figure 24.17. 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

**Figure 24.18. 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

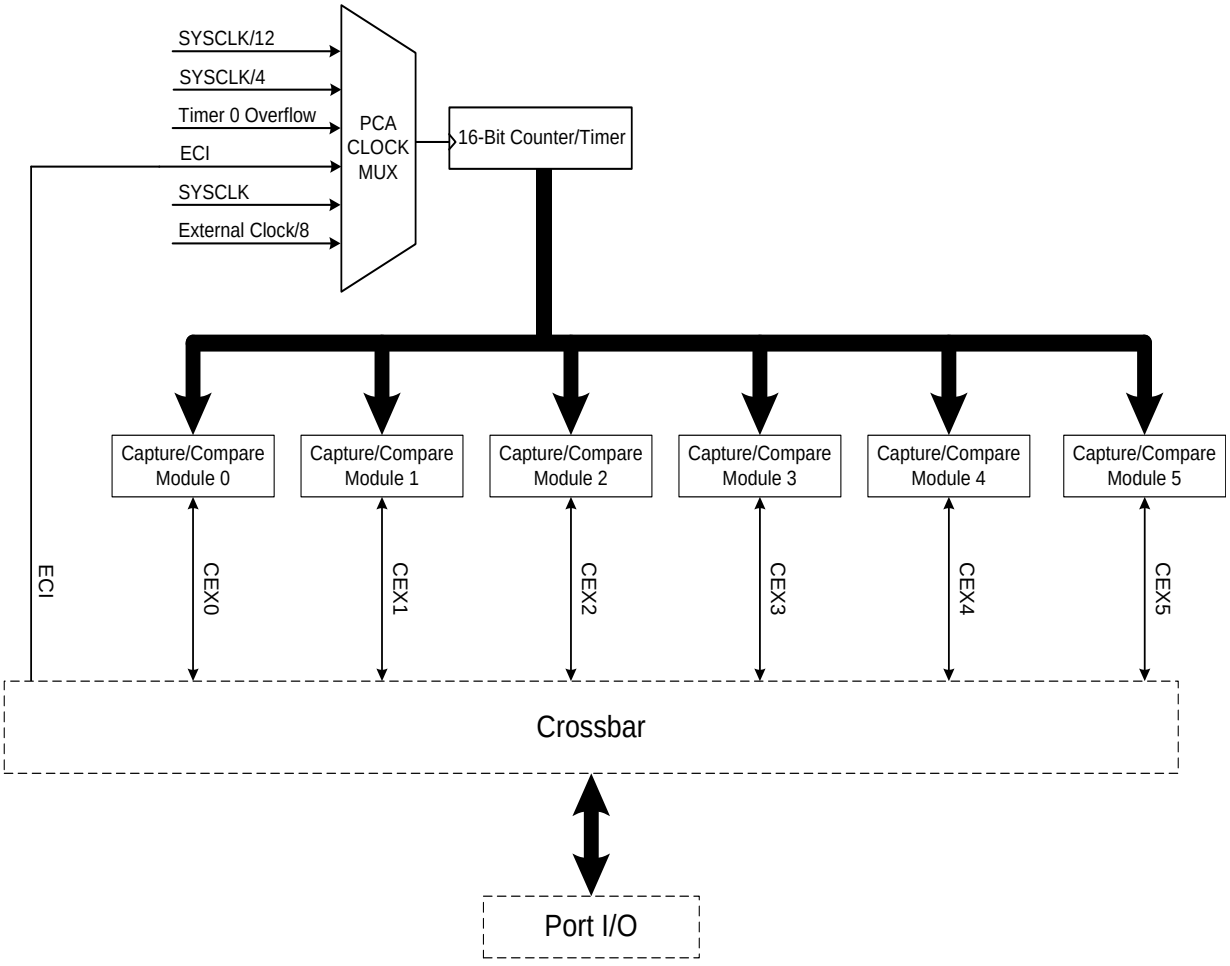
SFR Page: TMR2H: page 0; TMR3H: page 1; TMR4H: page 2

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

# 25. 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 205). The counter/timer is driven by a programmable time-base 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 25.2). The PCA is configured and controlled through the system controller's Special Function Registers. The basic PCA block diagram is shown in Figure 25.1.

**Figure 25.1. PCA Block Diagram**



## 25.1. PCA Counter/Timer

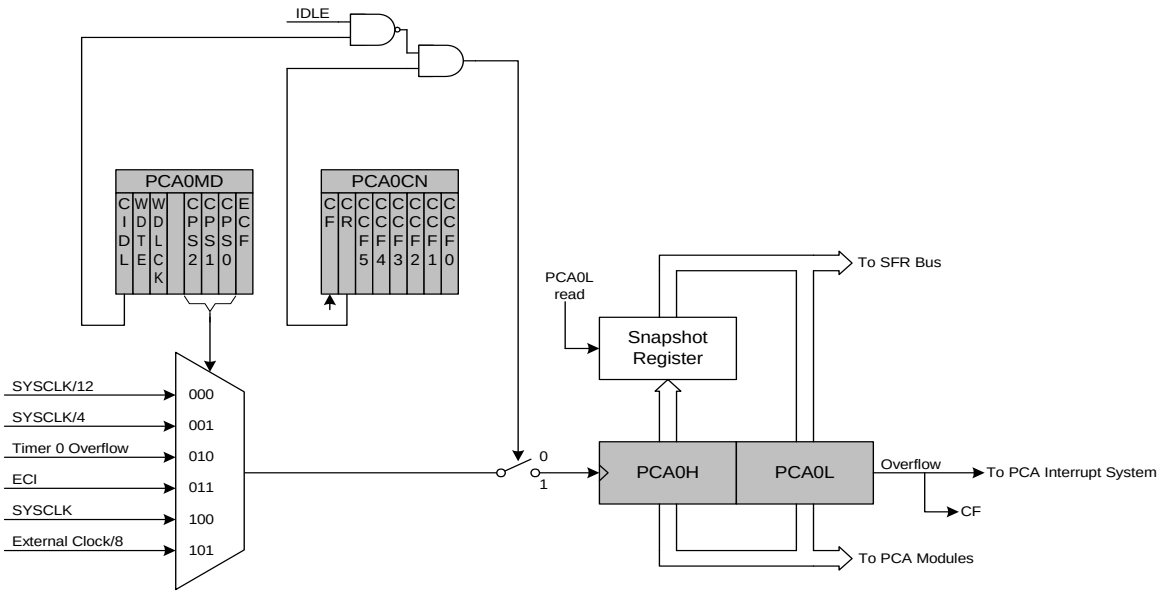
The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. 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 25.1.

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 25.1. 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 (synchronized with system clock) |

**Figure 25.2. PCA Counter/Timer Block Diagram**



## 25.2. 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.

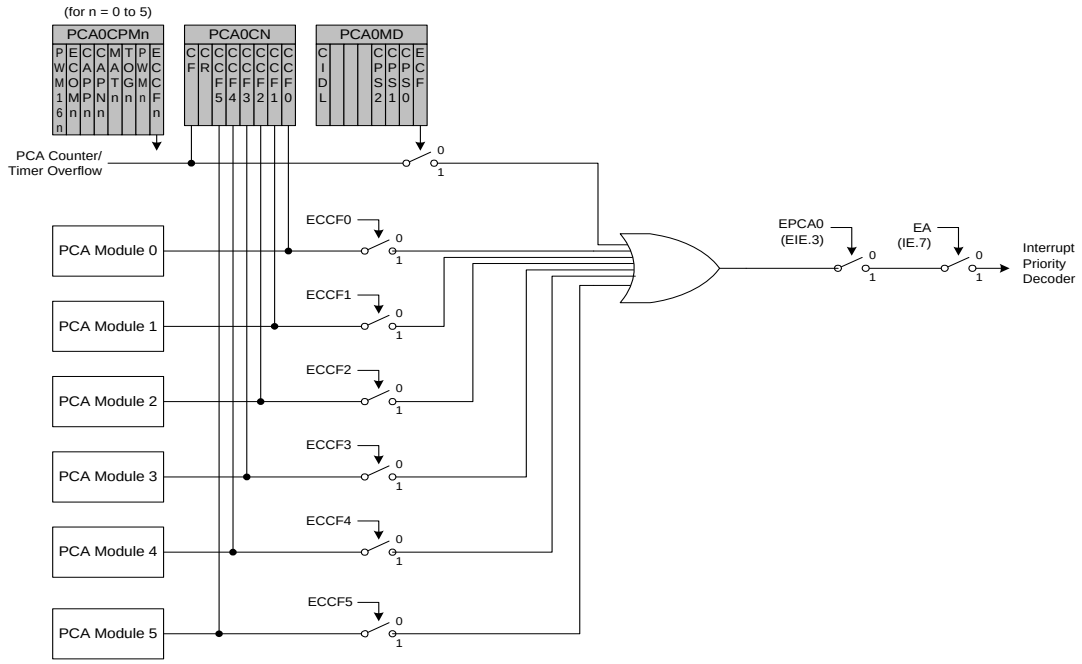
Table 25.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA0 capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic 1. See Figure 25.3 for details on the PCA interrupt configuration.

**Table 25.2. 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

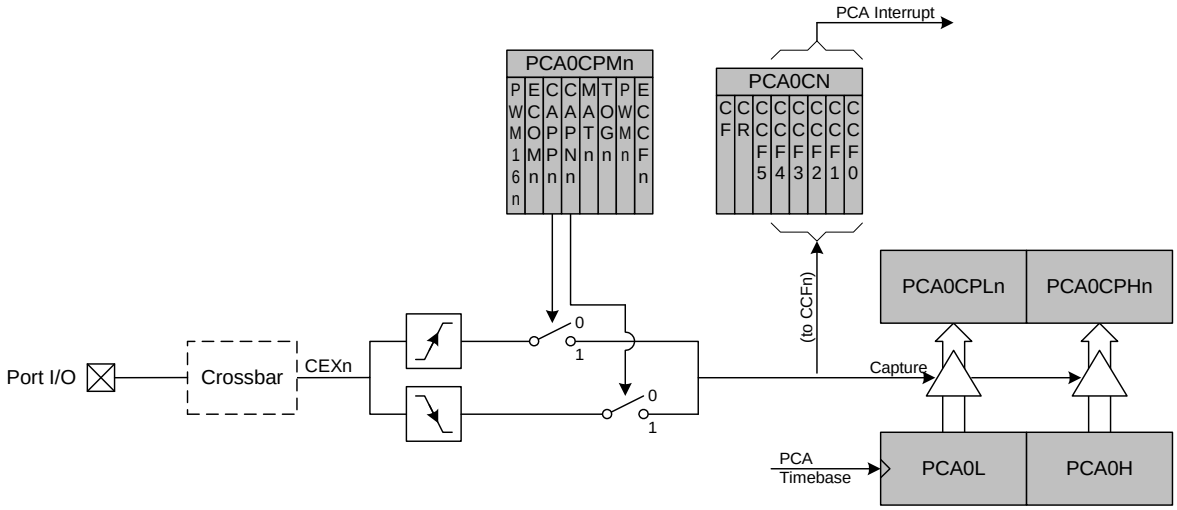
**Figure 25.3. PCA Interrupt Block Diagram**



### 25.2.1. 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 25.4. PCA Capture Mode Diagram**



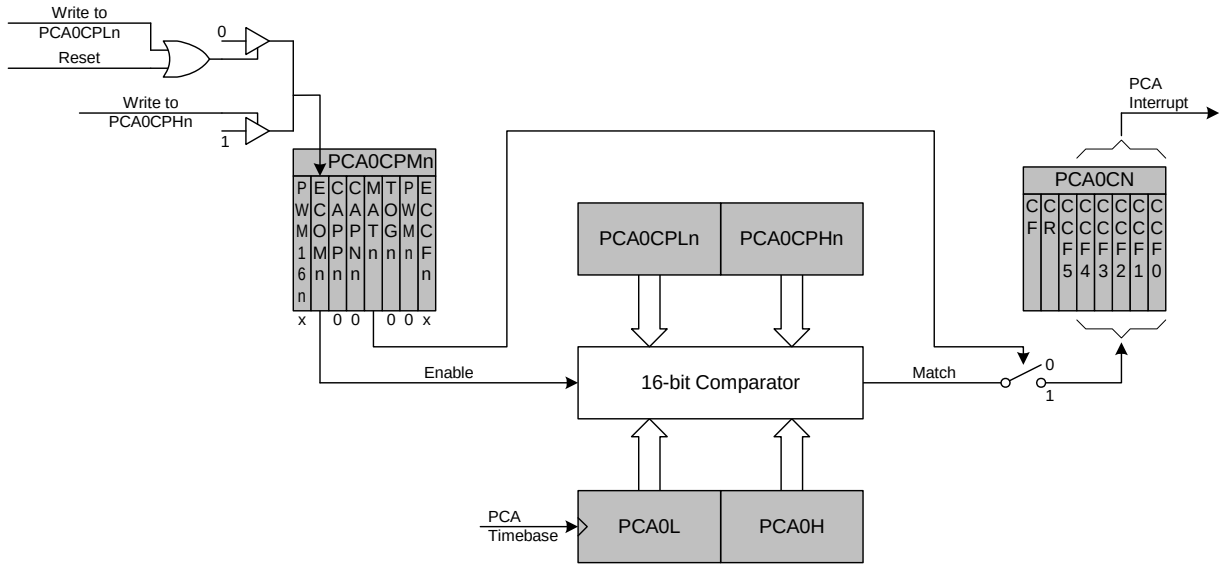

Note: The signal at CEXn must be high or low for at least 2 system clock cycles in order to be valid.

### 25.2.2. 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 25.5. PCA Software Timer Mode Diagram**

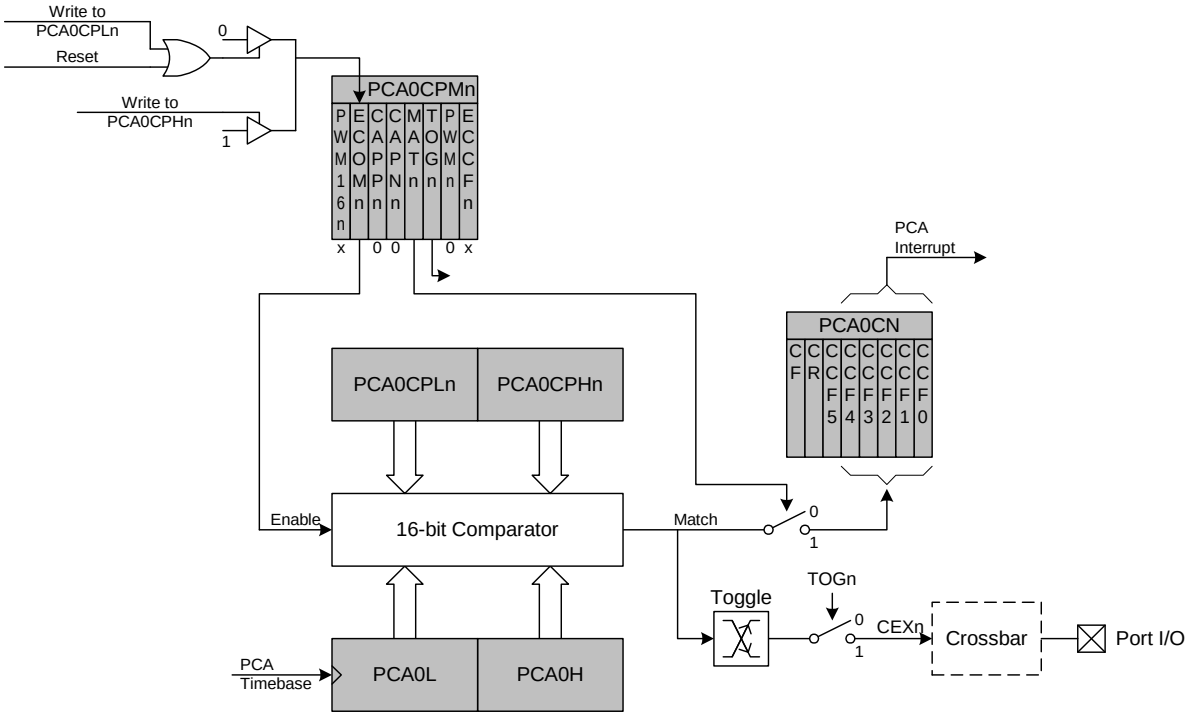


### 25.2.3. High Speed Output Mode

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 25.6. PCA High Speed Output Mode Diagram**



### 25.2.4. 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 25.1.

**Equation 25.1. Square Wave Frequency Output**

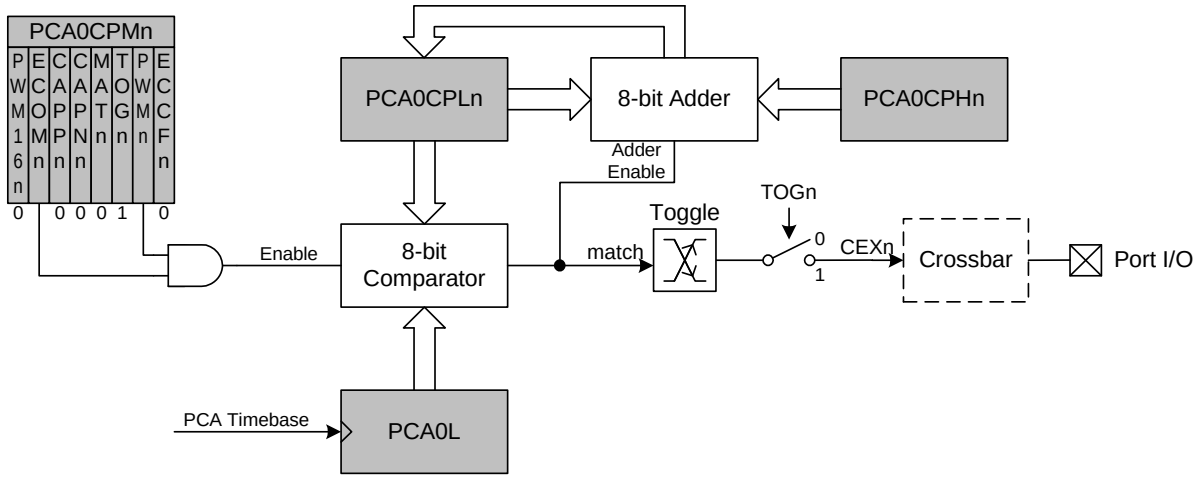
$$F_{sqr} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

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 25.7. PCA Frequency Output Mode**



### 25.2.5. 8-Bit Pulse Width Modulator Mode

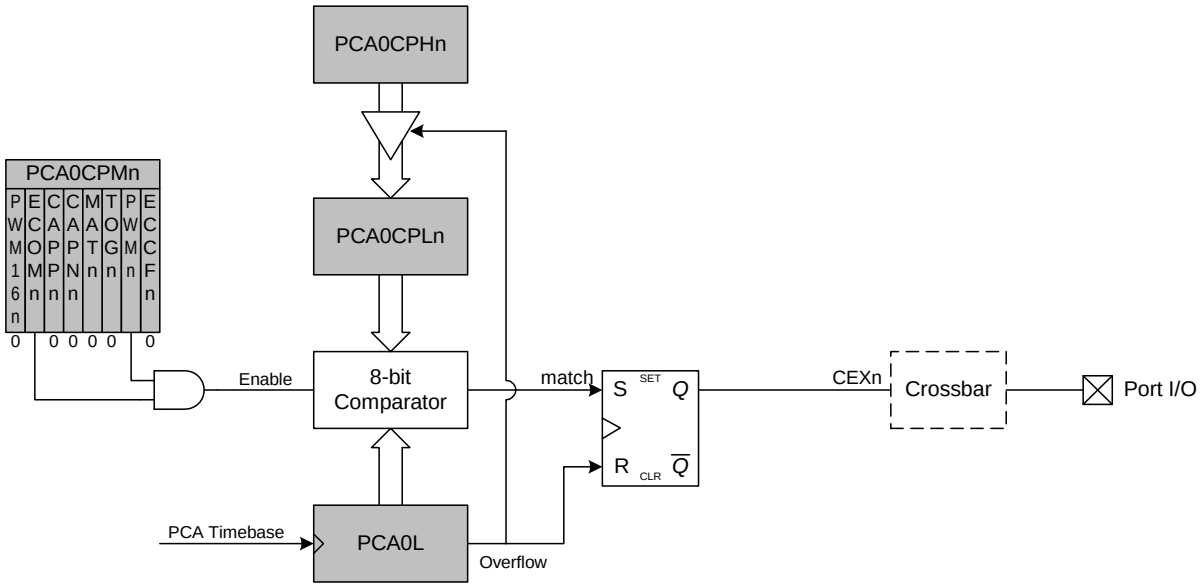
Each module can be used independently to generate pulse width modulated (PWM) outputs on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA0 counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPLn capture/compare register. 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 25.8). 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 25.2.

**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 25.2. 8-Bit PWM Duty Cycle**

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

**Figure 25.8. PCA 8-Bit PWM Mode Diagram**



### 25.2.6. 16-Bit Pulse Width Modulator Mode

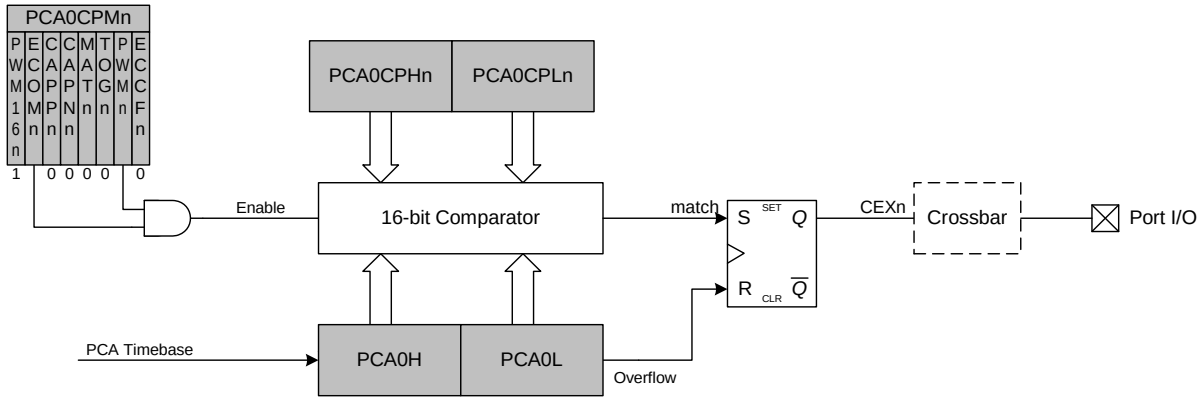
Each PCA0 module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA0 clocks for the low time of the PWM signal. When the PCA0 counter matches the module contents, the output on CEXn is asserted high; when the counter overflows, CEXn is asserted 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 25.3.

**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 25.3. 16-Bit PWM Duty Cycle**

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

**Figure 25.9. PCA 16-Bit PWM Mode**



## 25.3. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of PCA0.

**Figure 25.10. PCA0CN: PCA Control Register**

| 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.

**Figure 25.11. PCA0MD: PCA0 Mode Register**

| 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† |
| 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.

†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 in this mode.

**Figure 25.12. PCA0CPMn: PCA0 Capture/Compare Mode Registers**

| 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.

**Figure 25.13. 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.

**Figure 25.14. 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.

**Figure 25.15. 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.

**Figure 25.16. 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.

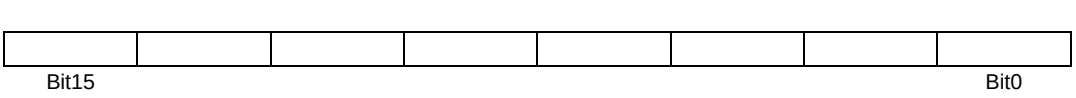
# 26. JTAG (IEEE 1149.1)

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 26.1 can be commanded. There are three DR's associated with JTAG Boundary-Scan, and four associated with Flash read/write operations on the MCU.

**Figure 26.1. 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 |

## 26.1. Boundary Scan

The DR in the Boundary Scan path is a 126-bit shift register for the C8051F060/2/4/6 and a 118-bit shift register for the C8051F061/3/5/7. 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 26.1. Boundary Data Register Bit Definitions (C8051F060/2/4/6)**

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

| Bit | Action | Target |
|---|---|---|
| 0 | Capture | Reset Enable from MCU |
| | Update | Reset Enable to /RST pin |
| 1 | Capture | Reset Input from /RST pin |
| | Update | Not used |
| 2 | Capture | CAN RX Output Enable to pin |
| | Update | CAN RX Output Enable to pin |
| 3 | Capture | CAN RX Input from pin |
| | Update | CAN RX Output to pin |
| 4 | Capture | CAN TX Output Enable to pin |
| | Update | CAN TX Output Enable to pin |
| 5 | Capture | CAN TX Input from pin |
| | Update | CAN TX Output to pin |
| 6 | Capture | External Clock from XTAL1 pin |
| | Update | Not used |
| 7 | Capture | Weak Pullup Enable from MCU |
| | Update | Weak Pullup Enable to Port Pins |
| 8, 10, 12, 14, 16, 18, 20, 22 | Capture | P0.n output enable from MCU (e.g. Bit 8 = P0.0, Bit 10 = P0.1, etc.) |
| | Update | P0.n output enable to pin (e.g. Bit 8 = P0.0oe, Bit 10 = P0.1oe, etc.) |
| 9, 11, 13, 15, 17, 19, 21, 23 | Capture | P0.n input from pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.) |
| | Update | P0.n output to pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.) |
| 24, 26, 28, 30, 32, 34, 36, 38 | Capture | P1.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P1.n output enable to pin (follows P0.n numbering scheme) |
| 25, 27, 29, 31, 33, 35, 37, 39 | Capture | P1.n input from pin (follows P0.n numbering scheme) |
| | Update | P1.n output to pin (follows P0.n numbering scheme) |
| 40, 42, 44, 46, 48, 50, 52, 54 | Capture | P2.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P2.n output enable to pin (follows P0.n numbering scheme) |
| 41, 43, 45, 47, 49, 51, 53, 55 | Capture | P2.n input from pin (follows P0.n numbering scheme) |
| | Update | P2.n output to pin (follows P0.n numbering scheme) |
| 56, 58, 60, 62, 64, 66, 68, 70 | Capture | P3.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P3.n output enable to pin (follows P0.n numbering scheme) |
| 57, 59, 61, 63, 65, 67, 69, 71 | Capture | P3.n input from pin (follows P0.n numbering scheme) |
| | Update | P3.n output to pin (follows P0.n numbering scheme) |
| 72, 74, 76 | Capture | P4.5, P4.6, P4.7 (respectively) output enable from MCU |
| | Update | P4.5, P4.6, P4.7 (respectively) output enable to pin |
| 73, 75, 77 | Capture | P4.5, P4.6, P4.7 (respectively) input from pin |
| | Update | P4.5, P4.6, P4.7 (respectively) output to pin |
| 78, 80, 82, 84, 86, 88, 90, 92 | Capture | P5.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P5.n output enable to pin (follows P0.n numbering scheme) |

**Table 26.1. Boundary Data Register Bit Definitions (C8051F060/2/4/6) (Continued)**

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

| Bit | Action | Target |
|---|---|---|
| 79, 81, 83, 85, 87, 89, 91, 93 | Capture | P5.n input from pin (follows P0.n numbering scheme) |
| | Update | P5.n output to pin (follows P0.n numbering scheme) |
| 94, 96, 98, 100, 102, 104, 106, 108 | Capture | P6.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P6.n output enable to pin (follows P0.n numbering scheme) |
| 95, 97, 99, 101, 103, 105, 107, 109 | Capture | P6.n input from pin (follows P0.n numbering scheme) |
| | Update | P6.n output to pin (follows P0.n numbering scheme) |
| 110, 112, 114, 116, 118, 120, 122, 124 | Capture | P7.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P7.n output enable to pin (follows P0.n numbering scheme) |
| 111, 113, 115, 117, 119, 121, 123, 125 | Capture | P7.n input from pin (follows P0.n numbering scheme) |
| | Update | P7.n output to pin (follows P0.n numbering scheme) |

**Table 26.2. Boundary Data Register Bit Definitions (C8051F061/3/5/7)**

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

| Bit | Action | Target |
|---|---|---|
| 0 | Capture | Not used |
| | Update | Not used |
| 1 | Capture | Not used |
| | Update | Not used |
| 2 | Capture | CAN RX Output Enable to pin |
| | Update | CAN RX Output Enable to pin |
| 3 | Capture | CAN RX Input from pin |
| | Update | CAN RX Output to pin |
| 4 | Capture | CAN TX Output Enable to pin |
| | Update | CAN TX Output Enable to pin |
| 5 | Capture | CAN TX Input from pin |
| | Update | CAN TX Output to pin |
| 6 | Capture | External Clock from XTAL1 pin |
| | Update | Not used |
| 7 | Capture | Weak Pullup Enable from MCU |
| | Update | Weak Pullup Enable to Port Pins |
| 8, 10, 12, 14, 16, 18, 20, 22 | Capture | P0.n output enable from MCU (e.g. Bit 8 = P0.0, Bit 10 = P0.1, etc.) |
| | Update | P0.n output enable to pin (e.g. Bit 8 = P0.0oe, Bit 10 = P0.1oe, etc.) |
| 9, 11, 13, 15, 17, 19, 21, 23 | Capture | P0.n input from pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.) |
| | Update | P0.n output to pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.) |
| 24, 26, 28, 30, 32, 34, 36, 38 | Capture | P1.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P1.n output enable to pin (follows P0.n numbering scheme) |
| 25, 27, 29, 31, 33, 35, 37, 39 | Capture | P1.n input from pin (follows P0.n numbering scheme) |
| | Update | P1.n output to pin (follows P0.n numbering scheme) |
| 40, 42, 44, 46, 48, 50, 52, 54 | Capture | P2.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P2.n output enable to pin (follows P0.n numbering scheme) |
| 41, 43, 45, 47, 49, 51, 53, 55 | Capture | P2.n input from pin (follows P0.n numbering scheme) |
| | Update | P2.n output to pin (follows P0.n numbering scheme) |
| 56, 58, 60, 62, 64, 66, 68, 70 | Capture | P3.n output enable from MCU (follows P0.n numbering scheme) |
| | Update | P3.n output enable to pin (follows P0.n numbering scheme) |
| 57, 59, 61, 63, 65, 67, 69, 71 | Capture | P3.n input from pin (follows P0.n numbering scheme) |
| | Update | P3.n output to pin (follows P0.n numbering scheme) |
| 72 | Capture | Reset Enable from MCU |
| | Update | Reset Enable to /RST pin |
| 73 | Capture | Reset Input from /RST pin |
| | Update | Not used |
| 74, 76, 78, 80, 82, 84 | Capture | P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) output enable from MCU† |
| | Update | P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) output enable to pin† |
| 75, 77, 79, 81, 83, 85 | Capture | P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) input from pin† |
| | Update | P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) output to pin† |
| 86, 88, 90, 92, 94, 96, 98, 100 | Capture | P6.n output enable from MCU (follows P0.n numbering scheme)† |
| | Update | P6.n output enable to pin (follows P0.n numbering scheme)† |

**Table 26.2. Boundary Data Register Bit Definitions (C8051F061/3/5/7) (Continued)**

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

| Bit | Action | Target |
|---|---|---|
| 87, 89, 91, 93, 95, 97, 99, 101 | Capture | P6.n input from pin (follows P0.n numbering scheme)† |
| | Update | P6.n output to pin (follows P0.n numbering scheme)† |
| 102, 104, 106, 108, 110, 112, 114, 116 | Capture | P7.n output enable from MCU (follows P0.n numbering scheme)† |
| | Update | P7.n output enable to pin (follows P0.n numbering scheme)† |
| 103, 105, 107, 109, 111, 113, 115, 117 | Capture | P7.n input from pin (follows P0.n numbering scheme)† |
| | Update | P7.n output to pin (follows P0.n numbering scheme)† |
| † Not connected to pins in this device package. | | |

### 26.1.1. 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.

### 26.1.2. SAMPLE Instruction

The SAMPLE instruction is accessed via the IR. The Boundary DR provides observability and presetting of the scan-path latches.

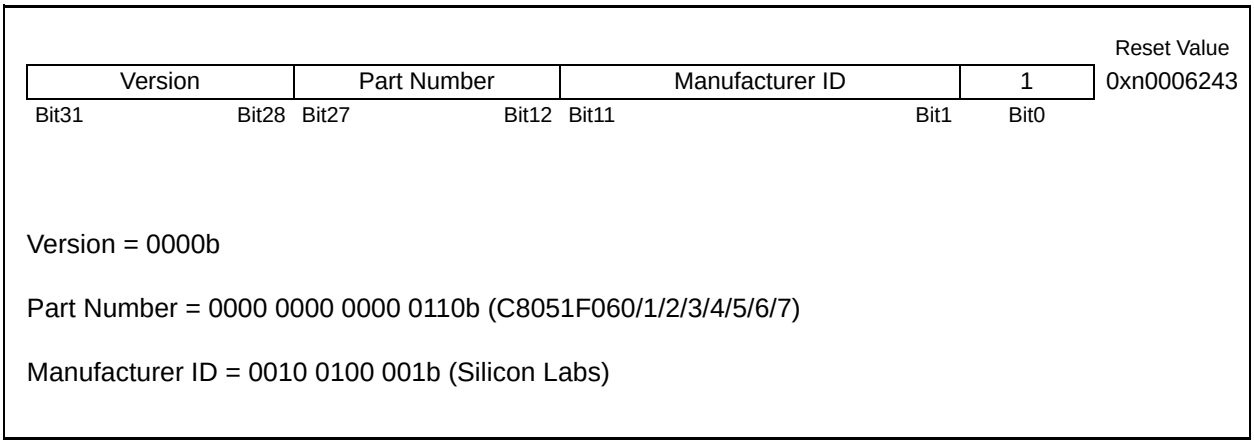
### 26.1.3. BYPASS Instruction

The BYPASS instruction is accessed via the IR. It provides access to the standard JTAG Bypass data register.

### 26.1.4. IDCODE Instruction

The IDCODE instruction is accessed via the IR. It provides access to the 32-bit Device ID register.

**Figure 26.2. DEVICEID: JTAG Device ID Register**




Version = 0000b

Part Number = 0000 0000 0000 0110b (C8051F060/1/2/3/4/5/6/7)

Manufacturer ID = 0010 0100 001b (Silicon Labs)

## 26.2. 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 Write-Data 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 at 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).

**Figure 26.3. FLASHCON: JTAG Flash Control Register**

| SFLE | WRMD2 | WRMD1 | WRMD0 | RDMD3 | RDMD2 | RDMD1 | RDMD0 | Reset Value |
|---|---|---|---|---|---|---|---|---|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | 00000000 |

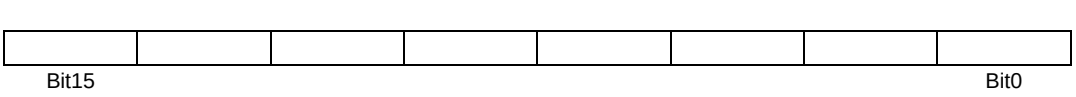
This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

Bit7: SFLE: Scratchpad Flash Memory Access Enable
When this bit is set, Flash reads and writes through the JTAG port are directed to the 128-byte Scratchpad Flash sector. When SFLE is set to logic 1, Flash accesses out of the address range 0x00-0x7F should not be attempted. Reads/Writes out of this range will yield undefined results.
0: Flash access from JTAG directed to the Program/Data Flash sector.
1: Flash access from JTAG directed to the Scratchpad sector.

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 = 0x7BFE - 0x7BFF, the entire user space will be erased (i.e. entire Flash memory except for Reserved area 0x7C00 - 0x7FFF).
(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 to 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.)

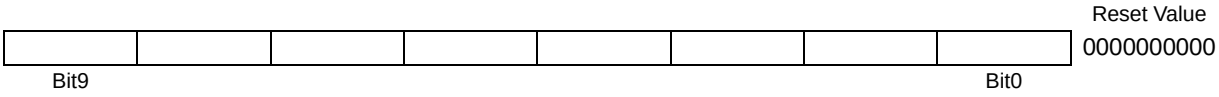
**Figure 26.5. FLASHADR: JTAG Flash Address Register**




This register holds the address for all JTAG Flash read, write, and erase operations. This register auto-increments
after each read or write, regardless of whether the operation succeeded or failed.

Bits15-0: Flash Operation 16-bit Address.

**Figure 26.4. FLASHDAT: JTAG Flash Data Register**




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

## 26.3. Debug Support

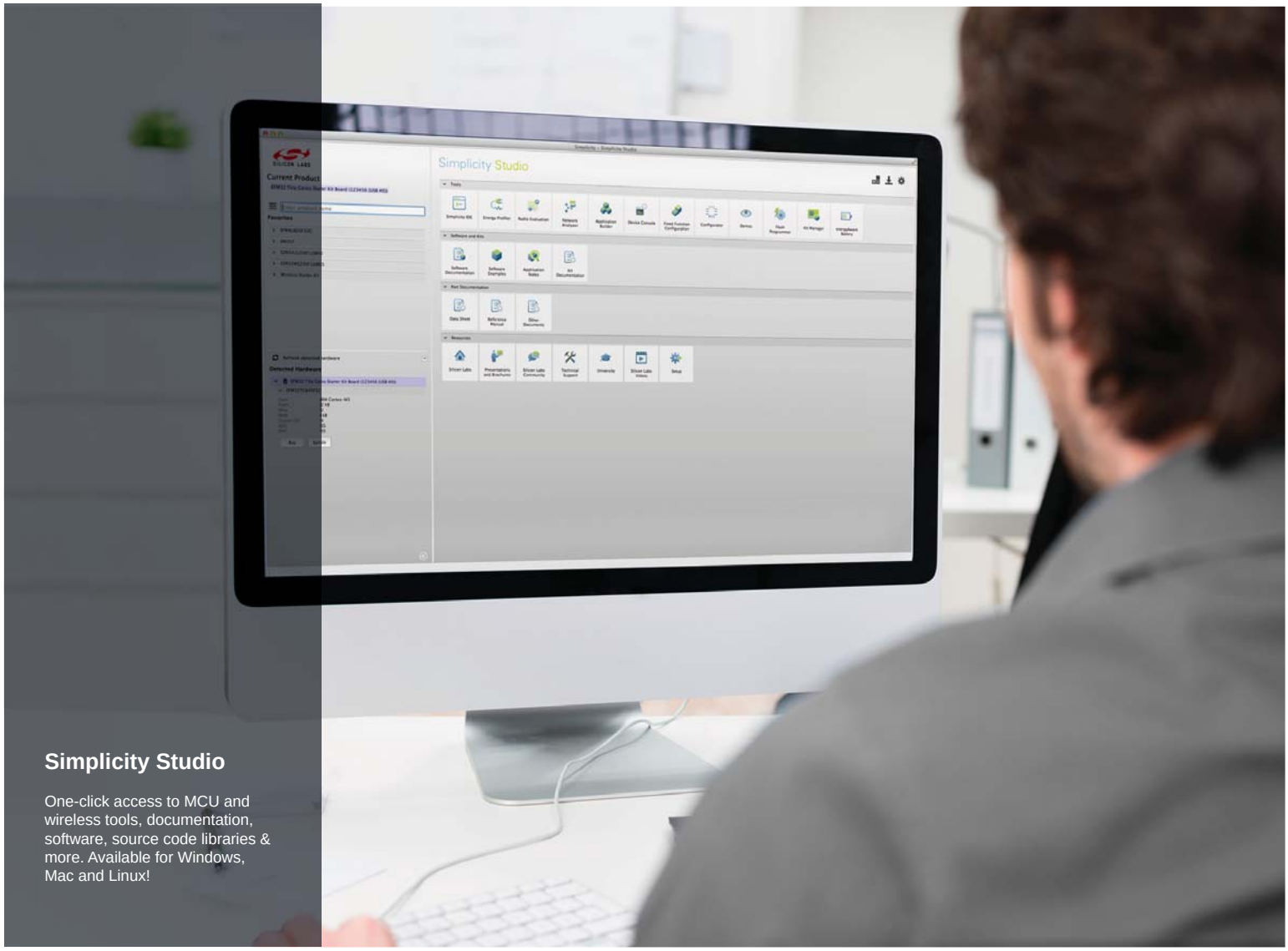
Each MCU has on-chip JTAG and debug logic that provides non-intrusive, full speed, in-circuit debug support using the production part installed in the end application, via the four pin JTAG I/F. Silicon Labs' debug system supports inspection and modification of memory and registers, breakpoints, and 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 C8051F060DK 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 C8051F06x family. Each kit includes development software for the PC, a Serial Adapter (for connection to JTAG) and a target application board with a C8051F060 installed. Serial cables and wall-mount power supply are also included.

# Document Change List

## Revision 1.1 to Revision 1.2

- Added four part numbers: C8051F064, C8051F065, C8051F066, and C8051F067.
- Modified all sections to describe functionality of the four new parts.
- Revised and expanded Flash Chapter with clearer descriptions of Flash security features.
- UART0 Chapter, Section 22.3: “FE0 in register SCON0” changed to “FE0 in register SSTA0”.
- UART0 Chapter: Updated and clarified baud rate equations.
- Port I/O Chapter, Section 18.2: Added a note in text body that Port 4-7 registers are all on SFR Page F.
- Comparators Chapter: Updated Table 12.1 “Comparator Electrical Characteristics”.
- CIP51 Chapter: Section 13.4.1: Added note regarding IDLE mode operation.
- ADC2 Chapter: AD2LJST bit removed from ADC2CF register description (AD2LJST is in the ADC2CN register).
- ADC2 Chapter: Updated Table 7.1 “ADC2 Electrical Characteristics” and Figure 7.2 “Temperature Sensor Transfer Function” with temperature sensor information.
- ADC0/ADC1 Chapter: Tracking/Conversion timing when ADnTM = 1 is shown in Figure 5.4 and Table 5.1. References to “18” or “16” SAR clocks of tracking were removed.
- DACs Chapter, Table 8.1 “DAC Electrical Characteristics”: Changed “Gain Error” to “Full-Scale Error”.
- SMBus Chapter, Figure 20.9 SMB0CR: Changed “1.125” to “1.125 * 10^6”.
- PCA Chapter, Figure 25.12 PCA0CPMn: Bit 0 name changed to “ECCFn” (from incorrect “EECFn”).
- JTAG Chapter, Figure 26.3 FLASHCON: Bit 7 description corrected. Bit 7 is SFLE, allowing access to the Scratchpad memory area.
- CAN Chapter: Added text “The CAN controller’s clock ($f_{sys}$, or CAN_CLK in the C_CAN User’s Guide) is equal to the CIP-51 MCU’s clock (SYSCLK).”
- Table 4.1 “Pin Descriptions”, MONEN: Added text “Recommended configuration is to connect directly to VDD.”
- Timers Chapter: All references to “DCEN” and “DECEN” corrected to “DCENn”.
- Timers Chapter, Equation 24.1: Equation was corrected to “Fsq = Ftclk / (2*(65536-RCAPn))”. This equation is valid for a timer counting up or down.
- Timers Chapter, Figure 24.14 TMRnCF: Corrected Bit 1 description. For square-wave output, CP/RLn = 0, C/Tn = 0, TnOE = 1.
- VREF Chapters: Added VREF Power Supply Current to VREF Electrical Characteristics Tables.
- PCA Chapter: Added Note about writing PCA0CPLn and PCA0CPHn to sections for SW Timer Mode, High-Speed Output Mode, Frequency Output Mode, 8-bit PWM Mode, and 16-bit PWM Mode.
- Oscillators Chapter, Table 15.1 “Internal Oscillator Electrical Characteristics”: Updated typical supply current.
- Table 3.1 “Global DC Electrical Characteristics”, Updated supply current numbers with additional characterization data.
- ADC0/ADC1 Chapter: Table 5.2 “ADC0 and ADC1 Electrical Characteristics”, Updated supply current numbers with additional characterization data.
- ADC0/ADC1 Chapter: Table 5.3 “Voltage Reference 0 and 1 Electrical Characteristics”, Updated Output Voltage numbers with characterization data.
- Figure 4.3 “TQFP-100 Package Drawing”, Added “L” Dimension.
- Figure 4.6 “TQFP-64 Package Drawing”, Added “L” Dimension.




## 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**

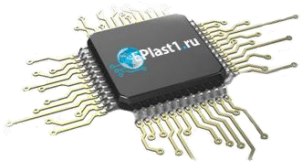
Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.




**Как с нами связаться**

**Телефон:** 8 (812) 309 58 32 (многоканальный)
**Факс:** 8 (812) 320-02-42
**Электронная почта:** org@eplast1.ru
**Адрес:** 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.