SILICON LABORATORIES

C8051F020/1/2/3

8K ISP FLASH MCU Family

ANALOG PERIPHERALS

- **SAR ADC**
    - 12-Bit (C8051F020/1)
    - 10-Bit (C8051F022/3)
    - ± 1 LSB INL
    - Programmable Throughput up to 100 ksps
    - Up to 8 External Inputs; Programmable as Single-Ended or Differential
    - Programmable Amplifier Gain: 16, 8, 4, 2, 1, 0.5
    - Data-Dependent Windowed Interrupt Generator
    - Built-in Temperature Sensor (± 3°C)
- **8-bit ADC**
    - Programmable Throughput up to 500 ksps
    - 8 External Inputs
    - Programmable Amplifier Gain: 4, 2, 1, 0.5
- **Two 12-bit DACs**
    - Can Synchronize Outputs to Timers for Jitter-Free Waveform Generation
- **Two Analog Comparators**
- **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
- Low-Cost, **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
- 22 Vectored Interrupt Sources

MEMORY

- 4352 Bytes Internal Data RAM (4k + 256)
- 64k Bytes FLASH; In-System programmable in 512-byte Sectors
- External 64k Byte Data Memory Interface (programmable multiplexed or non-multiplexed modes)

DIGITAL PERIPHERALS

- 8 Byte-Wide Port I/O (C8051F020/2); 5V tolerant
- 4 Byte-Wide Port I/O (C8051F021/3); 5V tolerant
- Hardware SMBus™ (I^{2}C™ Compatible), SPI™, and Two UART Serial Ports Available Concurrently
- Programmable 16-bit Counter/Timer Array with 5 Capture/Compare Modules
- 5 General Purpose 16-bit Counter/Timers
- Dedicated Watch-Dog Timer; Bi-directional Reset Pin

CLOCK SOURCES

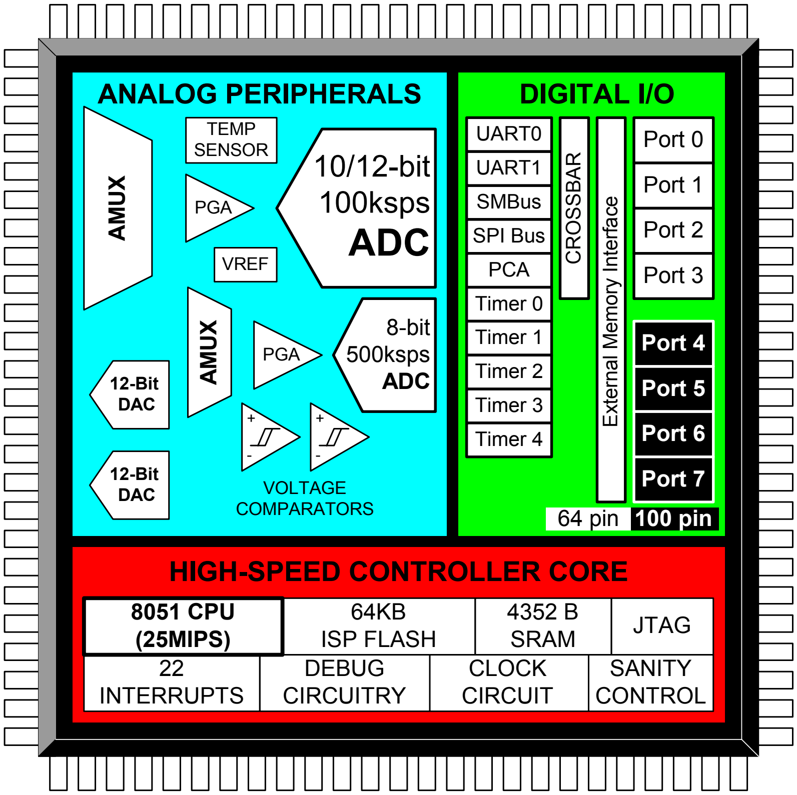
- Internal Programmable Oscillator: 2-to-16 MHz
- External Oscillator: Crystal, RC, C, or Clock
- Real-Time Clock Mode using Timer 3 or PCA

SUPPLY VOLTAGE .... 2.7V TO 3.6V

- Typical Operating Current: 10 mA @ 20 MHz
- Multiple Power Saving Sleep and Shutdown Modes

100-Pin TQFP and 64-Pin TQFP Packages Available

Temperature Range: -40°C to +85°C


ANALOG PERIPHERALS
AMUX
TEMP SENSOR
PGA
VREF
10/12-bit 100ksps ADC
AMUX
PGA
8-bit 500ksps ADC
12-Bit DAC
12-Bit DAC
VOLTAGE COMPARATORS
DIGITAL I/O
UART0
UART1
SMBus
SPI Bus
PCA
Timer 0
Timer 1
Timer 2
Timer 3
Timer 4
CROSSBAR
External Memory Interface
Port 0
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
64 pin
100 pin
HIGH-SPEED CONTROLLER CORE
8051 CPU (25MIPS)
64KB ISP FLASH
4352 B SRAM
JTAG
22 INTERRUPTS
DEBUG CIRCUITRY
CLOCK CIRCUIT
SANITY CONTROL

Copyright © 2003 by Silicon Laboratories

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

Notes

TABLE OF CONTENTS

1. SYSTEM OVERVIEW 17
- 1.1. CIP-51™ Microcontroller Core 22
  - 1.1.1. Fully 8051 Compatible 22
  - 1.1.2. Improved Throughput 22
  - 1.1.3. Additional Features 23
- 1.2. On-Chip Memory 24
- 1.3. JTAG Debug and Boundary Scan 25
- 1.4. Programmable Digital I/O and Crossbar 26
- 1.5. Programmable Counter Array 27
- 1.6. Serial Ports 27
- 1.7. 12-Bit Analog to Digital Converter 28
- 1.8. 8-Bit Analog to Digital Converter 29
- 1.9. Comparators and DACs 30

2. ABSOLUTE MAXIMUM RATINGS 31

3. GLOBAL DC ELECTRICAL CHARACTERISTICS 32

4. PINOUT AND PACKAGE DEFINITIONS 33

5. ADC0 (12-BIT ADC, C8051F020/1 ONLY) 43
- 5.1. Analog Multiplexer and PGA 43
- 5.2. ADC Modes of Operation 44
  - 5.2.1. Starting a Conversion 44
  - 5.2.2. Tracking Modes 45
  - 5.2.3. Settling Time Requirements 46
- 5.3. ADC0 Programmable Window Detector 53

6. ADC0 (10-BIT ADC, C8051F022/3 ONLY) 59
- 6.1. Analog Multiplexer and PGA 59
- 6.2. ADC Modes of Operation 60
  - 6.2.1. Starting a Conversion 60
  - 6.2.2. Tracking Modes 61
  - 6.2.3. Settling Time Requirements 62
- 6.3. ADC0 Programmable Window Detector 69

7. ADC1 (8-BIT ADC) 75
- 7.1. Analog Multiplexer and PGA 75
- 7.2. ADC1 Modes of Operation 76
  - 7.2.1. Starting a Conversion 76
  - 7.2.2. Tracking Modes 76
  - 7.2.3. Settling Time Requirements 78

8. DACS, 12-BIT VOLTAGE MODE 83
- 8.1. DAC Output Scheduling 83
  - 8.1.1. Update Output On-Demand 84
  - 8.1.2. Update Output Based on Timer Overflow 84
- 8.2. DAC Output Scaling/Justification 84

9. VOLTAGE REFERENCE (C8051F020/2) 91

10. VOLTAGE REFERENCE (C8051F021/3)..........93
11. COMPARATORS..........95
12. CIP-51 MICROCONTROLLER..........101
12.1. Instruction Set..........102
12.1.1. Instruction and CPU Timing..........102
12.1.2. MOVX Instruction and Program Memory..........102
12.2. Memory Organization..........107
12.2.1. Program Memory..........107
12.2.2. Data Memory..........108
12.2.3. General Purpose Registers..........108
12.2.4. Bit Addressable Locations..........108
12.2.5. Stack..........108
12.2.6. Special Function Registers..........109
12.2.7. Register Descriptions..........113
12.3. Interrupt Handler..........116
12.3.1. MCU Interrupt Sources and Vectors..........116
12.3.2. External Interrupts..........116
12.3.3. Interrupt Priorities..........118
12.3.4. Interrupt Latency..........118
12.3.5. Interrupt Register Descriptions..........119
12.4. Power Management Modes..........125
12.4.1. Idle Mode..........125
12.4.2. Stop Mode..........125
13. RESET SOURCES..........127
13.1. Power-on Reset..........128
13.2. Power-fail Reset..........128
13.3. External Reset..........129
13.4. Software Forced Reset..........129
13.5. Missing Clock Detector Reset..........129
13.6. Comparator0 Reset..........129
13.7. External CNVSTR Pin Reset..........129
13.8. Watchdog Timer Reset..........129
13.8.1. Enable/Reset WDT..........130
13.8.2. Disable WDT..........130
13.8.3. Disable WDT Lockout..........130
13.8.4. Setting WDT Interval..........130
14. OSCILLATORS..........135
14.1. External Crystal Example..........138
14.2. External RC Example..........138
14.3. External Capacitor Example..........138
15. FLASH MEMORY..........139
15.1. Programming The FLASH Memory..........139
15.2. Non-volatile Data Storage..........140
15.3. Security Options..........140
16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM..........145

16.1. Accessing XRAM....145
    16.1.1. 16-Bit MOVX Example....145
    16.1.2. 8-Bit MOVX Example....145
16.2. Configuring the External Memory Interface....146
16.3. Port Selection and Configuration....146
16.4. Multiplexed and Non-multiplexed Selection....148
    16.4.1. Multiplexed Configuration....148
    16.4.2. Non-multiplexed Configuration....149
16.5. Memory Mode Selection....150
    16.5.1. Internal XRAM Only....150
    16.5.2. Split Mode without Bank Select....150
    16.5.3. Split Mode with Bank Select....151
    16.5.4. External Only....151
16.6. Timing....151
    16.6.1. Non-multiplexed Mode....153
        16.6.1.1. 16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’....153
        16.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘101’ or ‘111’....154
        16.6.1.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘110’....155
    16.6.2. Multiplexed Mode....156
        16.6.2.1. 16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’....156
        16.6.2.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’....157
        16.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘010’....158
17. PORT INPUT/OUTPUT....161
17.1. Ports 0 through 3 and the Priority Crossbar Decoder....163
    17.1.1. Crossbar Pin Assignment and Allocation....163
    17.1.2. Configuring the Output Modes of the Port Pins....164
    17.1.3. Configuring Port Pins as Digital Inputs....165
    17.1.4. External Interrupts (IE6 and IE7)....165
    17.1.5. Weak Pull-ups....165
    17.1.6. Configuring Port 1 Pins as Analog Inputs (AIN1.[7:0])....165
    17.1.7. External Memory Interface Pin Assignments....166
    17.1.8. Crossbar Pin Assignment Example....168
17.2. Ports 4 through 7 (C8051F020/2 only)....177
    17.2.1. Configuring Ports which are not Pinned Out....177
    17.2.2. Configuring the Output Modes of the Port Pins....177
    17.2.3. Configuring Port Pins as Digital Inputs....178
    17.2.4. Weak Pull-ups....178
    17.2.5. External Memory Interface....178
18. SYSTEM MANAGEMENT BUS / I2C BUS (SMBUS0)....183
18.1. Supporting Documents....184
18.2. SMBus Protocol....185
    18.2.1. Arbitration....185
    18.2.2. Clock Low Extension....185
    18.2.3. SCL Low Timeout....186
    18.2.4. SCL High (SMBus Free) Timeout....186

18.3. SMBus Transfer Modes....................187
    18.3.1. Master Transmitter Mode....................187
    18.3.2. Master Receiver Mode....................187
    18.3.3. Slave Transmitter Mode....................188
    18.3.4. Slave Receiver Mode....................188
18.4. SMBus Special Function Registers....................189
    18.4.1. Control Register....................189
    18.4.2. Clock Rate Register....................192
    18.4.3. Data Register....................193
    18.4.4. Address Register....................193
    18.4.5. Status Register....................194

19. SERIAL PERIPHERAL INTERFACE BUS (SPI0)....................197
19.1. Signal Descriptions....................198
    19.1.1. Master Out, Slave In (MOSI)....................198
    19.1.2. Master In, Slave Out (MISO)....................198
    19.1.3. Serial Clock (SCK)....................198
    19.1.4. Slave Select (NSS)....................198
19.2. SPI0 Operation....................199
19.3. Serial Clock Timing....................200
19.4. SPI Special Function Registers....................201

20. UART0....................205
20.1. UART0 Operational Modes....................206
    20.1.1. Mode 0: Synchronous Mode....................206
    20.1.2. Mode 1: 8-Bit UART, Variable Baud Rate....................207
    20.1.3. Mode 2: 9-Bit UART, Fixed Baud Rate....................208
    20.1.4. Mode 3: 9-Bit UART, Variable Baud Rate....................209
20.2. Multiprocessor Communications....................210
20.3. Frame and Transmission Error Detection....................211

21. UART1....................215
21.1. UART1 Operational Modes....................216
    21.1.1. Mode 0: Synchronous Mode....................216
    21.1.2. Mode 1: 8-Bit UART, Variable Baud Rate....................217
    21.1.3. Mode 2: 9-Bit UART, Fixed Baud Rate....................218
    21.1.4. Mode 3: 9-Bit UART, Variable Baud Rate....................219
21.2. Multiprocessor Communications....................220
21.3. Frame and Transmission Error Detection....................221

22. TIMERS....................225
22.1. Timer 0 and Timer 1....................227
    22.1.1. Mode 0: 13-bit Counter/Timer....................227
    22.1.2. Mode 1: 16-bit Counter/Timer....................228
    22.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload....................229
    22.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)....................230
22.2. Timer 2....................234
    22.2.1. Mode 0: 16-bit Counter/Timer with Capture....................235
    22.2.2. Mode 1: 16-bit Counter/Timer with Auto-Reload....................236

22.2.3. Mode 2: Baud Rate Generator 237
22.3. Timer 3 240
22.4. Timer 4 243
22.4.1. Mode 0: 16-bit Counter/Timer with Capture 244
22.4.2. Mode 1: 16-bit Counter/Timer with Auto-Reload 245
22.4.3. Mode 2: Baud Rate Generator 246
23. PROGRAMMABLE COUNTER ARRAY 249
23.1. PCA Counter/Timer 250
23.2. Capture/Compare Modules 252
23.2.1. Edge-triggered Capture Mode 253
23.2.2. Software Timer (Compare) Mode 254
23.2.3. High Speed Output Mode 255
23.2.4. Frequency Output Mode 256
23.2.5. 8-Bit Pulse Width Modulator Mode 257
23.2.6. 16-Bit Pulse Width Modulator Mode 258
23.3. Register Descriptions for PCA0 259
24. JTAG (IEEE 1149.1) 265
24.1. Boundary Scan 266
24.1.1. EXTEST Instruction 267
24.1.2. SAMPLE Instruction 267
24.1.3. BYPASS Instruction 267
24.1.4. IDCODE Instruction 267
24.2. Flash Programming Commands 268
24.3. Debug Support 271

Notes

LIST OF FIGURES AND TABLES

1. SYSTEM OVERVIEW 17
Table 1.1. Product Selection Guide 17
Figure 1.1. C8051F020 Block Diagram..18
Figure 1.2. C8051F021 Block Diagram..19
Figure 1.3. C8051F022 Block Diagram..20
Figure 1.4. C8051F023 Block Diagram..21
Figure 1.5. Comparison of Peak MCU Execution Speeds 22
Figure 1.6. On-Board Clock and Reset 23
Figure 1.7. On-Chip Memory Map 24
Figure 1.8. Development/In-System Debug Diagram 25
Figure 1.9. Digital Crossbar Diagram..26
Figure 1.10. PCA Block Diagram...27
Figure 1.11. 12-Bit ADC Block Diagram..28
Figure 1.12. 8-Bit ADC Diagram 29
Figure 1.13. Comparator and DAC Diagram..30
2. ABSOLUTE MAXIMUM RATINGS 31
Table 2.1. Absolute Maximum Ratings*...31
3. GLOBAL DC ELECTRICAL CHARACTERISTICS 32
Table 3.1. Global DC Electrical Characteristics...32
4. PINOUT AND PACKAGE DEFINITIONS...33
Table 4.1. Pin Definitions...33
Figure 4.1. TQFP-100 Pinout Diagram...38
Figure 4.2. TQFP-100 Package Drawing..39
Figure 4.3. TQFP-64 Pinout Diagram...40
Figure 4.4. TQFP-64 Package Drawing..41
5. ADC0 (12-BIT ADC, C8051F020/1 ONLY) 43
Figure 5.1. 12-Bit ADC0 Functional Block Diagram..43
Figure 5.2. Temperature Sensor Transfer Function 44
Figure 5.3. 12-Bit ADC Track and Conversion Example Timing..............................45
Figure 5.4. ADC0 Equivalent Input Circuits 46
Figure 5.5. AMX0CF: AMUX0 Configuration Register (C8051F020/1) 47
Figure 5.6. AMX0SL: AMUX0 Channel Select Register (C8051F020/1)..................48
Figure 5.7. ADC0CF: ADC0 Configuration Register (C8051F020/1).......................49

Figure 5.17. 12-Bit ADC0 Window Interrupt Example: Right Justified Differential Data.....55
Figure 5.18. 12-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data....56
Figure 5.19. 12-Bit ADC0 Window Interrupt Example: Left Justified Differential Data.......57
Table 5.1. 12-Bit ADC0 Electrical Characteristics (C8051F020/1)....................................58
6. ADC0 (10-BIT ADC, C8051F022/3 ONLY) 59
Figure 6.1. 10-Bit ADC0 Functional Block Diagram..59
Figure 6.2. Temperature Sensor Transfer Function..60
Figure 6.3. 10-Bit ADC Track and Conversion Example Timing..61
Figure 6.4. ADC0 Equivalent Input Circuits 62
Figure 6.5. AMX0CF: AMUX0 Configuration Register (C8051F022/3) 63
Figure 6.6. AMX0SL: AMUX0 Channel Select Register (C8051F022/3)...........................64
Figure 6.7. ADC0CF: ADC0 Configuration Register (C8051F022/3).................................65
Figure 6.8. ADC0CN: ADC0 Control Register (C8051F022/3) 66
Figure 6.9. ADC0H: ADC0 Data Word MSB Register (C8051F022/3) 67
Figure 6.10. ADC0L: ADC0 Data Word LSB Register (C8051F022/3)................................67
Figure 6.11. ADC0 Data Word Example (C8051F022/3) 68
Figure 6.12. ADC0GTH: ADC0 Greater-Than Data High Byte Register (C8051F022/3).....69
Figure 6.13. ADC0GTL: ADC0 Greater-Than Data Low Byte Register (C8051F022/3) 69
Figure 6.14. ADC0LTH: ADC0 Less-Than Data High Byte Register (C8051F022/3) 69
Figure 6.15. ADC0LTL: ADC0 Less-Than Data Low Byte Register (C8051F022/3) 69
Figure 6.16. 10-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data .70
Figure 6.17. 10-Bit ADC0 Window Interrupt Example: Right Justified Differential Data.....71
Figure 6.18. 10-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data....72
Figure 6.19. 10-Bit ADC0 Window Interrupt Example: Left Justified Differential Data.......73
Table 6.1. 10-Bit ADC0 Electrical Characteristics (C8051F022/3)....................................74
7. ADC1 (8-BIT ADC) 75
Figure 7.1. ADC1 Functional Block Diagram..75
Figure 7.2. ADC1 Track and Conversion Example Timing 77
Figure 7.3. ADC1 Equivalent Input Circuit...78
Figure 7.4. ADC1CF: ADC1 Configuration Register (C8051F020/1/2/3)...........................79
Figure 7.5. AMX1SL: AMUX1 Channel Select Register (C8051F020/1/2/3) 79
Figure 7.6. ADC1CN: ADC1 Control Register (C8051F020/1/2/3) 80
Figure 7.7. ADC1: ADC1 Data Word Register 81
Figure 7.8. ADC1 Data Word Example..81
Table 7.1. ADC1 Electrical Characteristics...82
8. DACS, 12-BIT VOLTAGE MODE 83
Figure 8.1. DAC Functional Block Diagram 83
Figure 8.2. DAC0H: DAC0 High Byte Register 85
Figure 8.3. DAC0L: DAC0 Low Byte Register 85
Figure 8.4. DAC0CN: DAC0 Control Register..86
Figure 8.5. DAC1H: DAC1 High Byte Register 87
Figure 8.6. DAC1L: DAC1 Low Byte Register 87
Figure 8.7. DAC1CN: DAC1 Control Register..88
Table 8.1. DAC Electrical Characteristics...89
9. VOLTAGE REFERENCE (C8051F020/2)..91

Figure 9.1. Voltage Reference Functional Block Diagram..91
Figure 9.2. REF0CN: Reference Control Register 92
Table 9.1. Voltage Reference Electrical Characteristics 92
10. VOLTAGE REFERENCE (C8051F021/3)..93
Figure 10.1. Voltage Reference Functional Block Diagram 93
Figure 10.2. REF0CN: Reference Control Register 94
Table 10.1. Voltage Reference Electrical Characteristics 94
11. COMPARATORS...95
Figure 11.1. Comparator Functional Block Diagram 95
Figure 11.2. Comparator Hysteresis Plot...96
Figure 11.3. CPT0CN: Comparator0 Control Register 97
Figure 11.4. CPT1CN: Comparator1 Control Register 98
Table 11.1. Comparator Electrical Characteristics...99
12. CIP-51 MICROCONTROLLER..101
Figure 12.1. CIP-51 Block Diagram...101
Table 12.1. CIP-51 Instruction Set Summary..103
Figure 12.2. Memory Map 107
Table 12.2. Special Function Register (SFR) Memory Map...109
Table 12.3. Special Function Registers 109
Figure 12.3. SP: Stack Pointer...113
Figure 12.4. DPL: Data Pointer Low Byte 113
Figure 12.5. DPH: Data Pointer High Byte 113
Figure 12.6. PSW: Program Status Word..114
Figure 12.7. ACC: Accumulator...115
Figure 12.8. B: B Register 115
Table 12.4. Interrupt Summary...117
Figure 12.9. IE: Interrupt Enable..119
Figure 12.10. IP: Interrupt Priority 120
Figure 12.11. EIE1: Extended Interrupt Enable 1 121
Figure 12.12. EIE2: Extended Interrupt Enable 2 122
Figure 12.13. EIP1: Extended Interrupt Priority 1...123
Figure 12.14. EIP2: Extended Interrupt Priority 2...124
Figure 12.15. PCON: Power Control..126
13. RESET SOURCES 127
Figure 13.1. Reset Sources 127
Figure 13.2. Reset Timing 128
Figure 13.3. WDTCN: Watchdog Timer Control Register 131
Figure 13.4. RSTSRC: Reset Source Register...132
Table 13.1. Reset Electrical Characteristics 133
14. OSCILLATORS..135
Figure 14.1. Oscillator Diagram 135
Figure 14.2. OSCICN: Internal Oscillator Control Register 136
Table 14.1. Internal Oscillator Electrical Characteristics...136
Figure 14.3. OSCXCN: External Oscillator Control Register...137
15. FLASH MEMORY 139

Table 15.1. FLASH Electrical Characteristics 140
Figure 15.1. FLASH Program Memory Map and Security Bytes 141
Figure 15.2. FLACL: FLASH Access Limit 142
Figure 15.3. FLSCL: FLASH Memory Control 143
Figure 15.4. PSCTL: Program Store Read/Write Control 144
16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM......................145
Figure 16.1. EMI0CN: External Memory Interface Control 147
Figure 16.2. EMI0CF: External Memory Configuration...147
Figure 16.3. Multiplexed Configuration Example...148
Figure 16.4. Non-multiplexed Configuration Example 149
Figure 16.5. EMIF Operating Modes...150
Figure 16.6. EMI0TC: External Memory Timing Control 152
Figure 16.7. Non-multiplexed 16-bit MOVX Timing 153
Figure 16.8. Non-multiplexed 8-bit MOVX without Bank Select Timing...........................154
Figure 16.9. Non-multiplexed 8-bit MOVX with Bank Select Timing................................155
Figure 16.10. Multiplexed 16-bit MOVX Timing...156
Figure 16.11. Multiplexed 8-bit MOVX without Bank Select Timing 157
Figure 16.12. Multiplexed 8-bit MOVX with Bank Select Timing.......................................158
Table 16.1. AC Parameters for External Memory Interface...159
17. PORT INPUT/OUTPUT 161
Figure 17.1. Port I/O Cell Block Diagram...161
Table 17.1. Port I/O DC Electrical Characteristics 161
Figure 17.2. Lower Port I/O Functional Block Diagram...162
Figure 17.3. Priority Crossbar Decode Table 163
Figure 17.4. Priority Crossbar Decode Table 166
Figure 17.5. Priority Crossbar Decode Table 167
Figure 17.6. Crossbar Example: 169
Figure 17.7. XBR0: Port I/O Crossbar Register 0 170
Figure 17.8. XBR1: Port I/O Crossbar Register 1 171
Figure 17.9. XBR2: Port I/O Crossbar Register 2 172
Figure 17.10. P0: Port0 Data Register 173
Figure 17.11. P0MDOUT: Port0 Output Mode Register...173
Figure 17.12. P1: Port1 Data Register 174
Figure 17.13. P1MDIN: Port1 Input Mode Register 174
Figure 17.14. P1MDOUT: Port1 Output Mode Register...175
Figure 17.15. P2: Port2 Data Register 175
Figure 17.16. P2MDOUT: Port2 Output Mode Register...175
Figure 17.17. P3: Port3 Data Register 176
Figure 17.18. P3MDOUT: Port3 Output Mode Register...176
Figure 17.19. P3IF: Port3 Interrupt Flag Register 177
Figure 17.20. P74OUT: Ports 7 - 4 Output Mode Register 179
Figure 17.21. P4: Port4 Data Register 180
Figure 17.22. P5: Port5 Data Register 180
Figure 17.23. P6: Port6 Data Register 181
Figure 17.24. P7: Port7 Data Register 181

18. SYSTEM MANAGEMENT BUS / I2C BUS (SMBUS0) 183
Figure 18.1. SMBus0 Block Diagram 183
Figure 18.2. Typical SMBus Configuration 184
Figure 18.3. SMBus Transaction 185
Figure 18.4. Typical Master Transmitter Sequence...187
Figure 18.5. Typical Master Receiver Sequence 187
Figure 18.6. Typical Slave Transmitter Sequence...188
Figure 18.7. Typical Slave Receiver Sequence 188
Figure 18.8. SMB0CN: SMBus0 Control Register 191
Figure 18.9. SMB0CR: SMBus0 Clock Rate Register..192
Figure 18.10. SMB0DAT: SMBus0 Data Register 193
Figure 18.11. SMB0ADR: SMBus0 Address Register..193
Figure 18.12. SMB0STA: SMBus0 Status Register..194
Table 18.1. SMB0STA Status Codes and States 195
19. SERIAL PERIPHERAL INTERFACE BUS (SPI0) 197
Figure 19.1. SPI Block Diagram...197
Figure 19.2. Typical SPI Interconnection...198
Figure 19.3. Full Duplex Operation..199
Figure 19.4. Data/Clock Timing Diagram 200
Figure 19.5. SPI0CFG: SPI0 Configuration Register..201
Figure 19.6. SPI0CN: SPI0 Control Register 202
Figure 19.7. SPI0CKR: SPI0 Clock Rate Register...203
Figure 19.8. SPI0DAT: SPI0 Data Register 203
20. UART0 205
Figure 20.1. UART0 Block Diagram..205
Table 20.1. UART0 Modes 206
Figure 20.2. UART0 Mode 0 Interconnect..206
Figure 20.3. UART0 Mode 0 Timing Diagram 206
Figure 20.4. UART0 Mode 1 Timing Diagram 207
Figure 20.5. UART Modes 2 and 3 Timing Diagram..208
Figure 20.6. UART Modes 1, 2, and 3 Interconnect Diagram 209
Figure 20.7. UART Multi-Processor Mode Interconnect Diagram 210
Table 20.2. Oscillator Frequencies for Standard Baud Rates 212
Figure 20.8. SCON0: UART0 Control Register..213
Figure 20.9. SBUF0: UART0 Data Buffer Register..214
Figure 20.10. SADDR0: UART0 Slave Address Register 214
Figure 20.11. SADEN0: UART0 Slave Address Enable Register 214
21. UART1 215
Figure 21.1. UART1 Block Diagram..215
Table 21.1. UART1 Modes 216
Figure 21.2. UART1 Mode 0 Interconnect..216
Figure 21.3. UART1 Mode 0 Timing Diagram 216
Figure 21.4. UART1 Mode 1 Timing Diagram 217
Figure 21.5. UART Modes 2 and 3 Timing Diagram..218
Figure 21.6. UART Modes 1, 2, and 3 Interconnect Diagram 219

Figure 21.7. UART Multi-Processor Mode Interconnect Diagram.......................................220
Table 21.2. Oscillator Frequencies for Standard Baud Rates...222
Figure 21.8. SCON1: UART1 Control Register...223
Figure 21.9. SBUF1: UART1 Data Buffer Register...224
Figure 21.10. SADDR1: UART1 Slave Address Register 224
Figure 21.11. SADEN1: UART1 Slave Address Enable Register 224
22. TIMERS...225
Figure 22.1. CKCON: Clock Control Register..226
Figure 22.2. T0 Mode 0 Block Diagram...228
Figure 22.3. T0 Mode 2 (8-bit Auto-Reload) Block Diagram..229
Figure 22.4. T0 Mode 3 (Two 8-bit Timers) Block Diagram...230
Figure 22.5. TCON: Timer Control Register...231
Figure 22.6. TMOD: Timer Mode Register...232
Figure 22.7. TL0: Timer 0 Low Byte 233
Figure 22.8. TL1: Timer 1 Low Byte 233
Figure 22.9. TH0 Timer 0 High Byte 233
Figure 22.10. TH1: Timer 1 High Byte 233
Figure 22.11. T2 Mode 0 Block Diagram..235
Figure 22.12. T2 Mode 1 Block Diagram..236
Figure 22.13. T2 Mode 2 Block Diagram..237
Figure 22.14. T2CON: Timer 2 Control Register..238
Figure 22.15. RCAP2L: Timer 2 Capture Register Low Byte 239
Figure 22.16. RCAP2H: Timer 2 Capture Register High Byte 239
Figure 22.17. TL2: Timer 2 Low Byte 239
Figure 22.18. TH2 Timer 2 High Byte 239
Figure 22.19. Timer 3 Block Diagram...240
Figure 22.20. TMR3CN: Timer 3 Control Register 241
Figure 22.21. TMR3RLL: Timer 3 Reload Register Low Byte 241
Figure 22.22. TMR3RLH: Timer 3 Reload Register High Byte 242
Figure 22.23. TMR3L: Timer 3 Low Byte 242
Figure 22.24. TMR3H: Timer 3 High Byte 242
Figure 22.25. T4 Mode 0 Block Diagram..244
Figure 22.26. T4 Mode 1 Block Diagram..245
Figure 22.27. T4 Mode 2 Block Diagram..246
Figure 22.28. T4CON: Timer 4 Control Register..247
Figure 22.29. RCAP4L: Timer 4 Capture Register Low Byte 248
Figure 22.30. RCAP4H: Timer 4 Capture Register High Byte 248
Figure 22.31. TL4: Timer 4 Low Byte 248
Figure 22.32. TH4 Timer 4 High Byte 248
23. PROGRAMMABLE COUNTER ARRAY 249
Figure 23.1. PCA Block Diagram...249
Figure 23.2. PCA Counter/Timer Block Diagram..250
Table 23.1. PCA Timebase Input Options..250
Figure 23.3. PCA Interrupt Block Diagram..252
Table 23.2. PCA0CPM Register Settings for PCA Capture/Compare Modules..................252

Figure 23.4. PCA Capture Mode Diagram 253
Figure 23.5. PCA Software Timer Mode Diagram...254
Figure 23.6. PCA High Speed Output Mode Diagram 255
Figure 23.7. PCA Frequency Output Mode 256
Figure 23.8. PCA 8-Bit PWM Mode Diagram 257
Figure 23.9. PCA 16-Bit PWM Mode 258
Figure 23.10. PCA0CN: PCA Control Register 259
Figure 23.11. PCA0MD: PCA0 Mode Register 260
Figure 23.12. PCA0CPMn: PCA0 Capture/Compare Mode Registers 261
Figure 23.13. PCA0L: PCA0 Counter/Timer Low Byte 262
Figure 23.14. PCA0H: PCA0 Counter/Timer High Byte 262
Figure 23.15. PCA0CPLn: PCA0 Capture Module Low Byte 263
Figure 23.16. PCA0CPHn: PCA0 Capture Module High Byte......................................263
24. JTAG (IEEE 1149.1)..265
Figure 24.1. IR: JTAG Instruction Register 265
Table 24.1. Boundary Data Register Bit Definitions...266
Figure 24.2. DEVICEID: JTAG Device ID Register 267
Figure 24.3. FLASHCON: JTAG Flash Control Register...269
Figure 24.4. FLASHADR: JTAG Flash Address Register 270
Figure 24.5. FLASHDAT: JTAG Flash Data Register..270

Notes

1. SYSTEM OVERVIEW

The C8051F020/1/2/3 devices are fully integrated mixed-signal System-on-a-Chip MCUs with 64 digital I/O pins (C8051F020/2) or 32 digital I/O pins (C8051F021/3). 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)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 12-bit (C8051F020/1) or 10-bit (C8051F022/3) 100 ksps 8-channel ADC with PGA and analog multiplexer
- True 8-bit ADC 500 ksps 8-channel ADC with PGA and analog multiplexer
- Two 12-bit DACs with programmable update scheduling
- 64k bytes of in-system programmable FLASH memory
- 4352 (4096 + 256) bytes of on-chip RAM
- External Data Memory Interface with 64k byte address space
- SPI, SMBus/I^{2}C, and (2) UART serial interfaces implemented in hardware
- Five general purpose 16-bit Timers
- Programmable Counter/Timer Array with five capture/compare modules
- On-chip Watchdog Timer, VDD Monitor, and Temperature Sensor

With on-chip VDD monitor, Watchdog Timer, and clock oscillator, the C8051F020/1/2/3 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 V-to-3.6 V operation over the industrial temperature range (-45° C to +85° C). The Port I/Os, /RST, and JTAG pins are tolerant for input signals up to 5 V. The C8051F020/2 are available in a 100-pin TQFP package (see block diagrams in Figure 1.1 and Figure 1.3). The C8051F021/3 are available in a 64-pin TQFP package (see block diagrams in Figure 1.2 and Figure 1.4).

Table 1.1. Product Selection Guide

	MIPS (Peak)	FLASH Memory	RAM	External Memory Interface	SMBus/I^{2}C	SPI	UARTS	Timers (16-bit)	Programmable Counter Array	Digital Port I/O's	12-bit 100ksps ADC Inputs	10-bit 100ksps ADC Inputs	8-bit 500ksps ADC Inputs	Voltage Reference	Temperature Sensor	DAC Resolution (bits)	DAC Outputs	Analog Comparators	Package
C8051F020	25	64k	4352	✓	✓	✓	2	5	✓	64	8	-	8	✓	✓	12	2	2	100TQFP
C8051F021	25	64k	4352	✓	✓	✓	2	5	✓	32	8	-	8	✓	✓	12	2	2	64TQFP
C8051F022	25	64k	4352	✓	✓	✓	2	5	✓	64	-	8	8	✓	✓	12	2	2	100TQFP
C8051F023	25	64k	4352	✓	✓	✓	2	5	✓	32	-	8	8	✓	✓	12	2	2	64TQFP

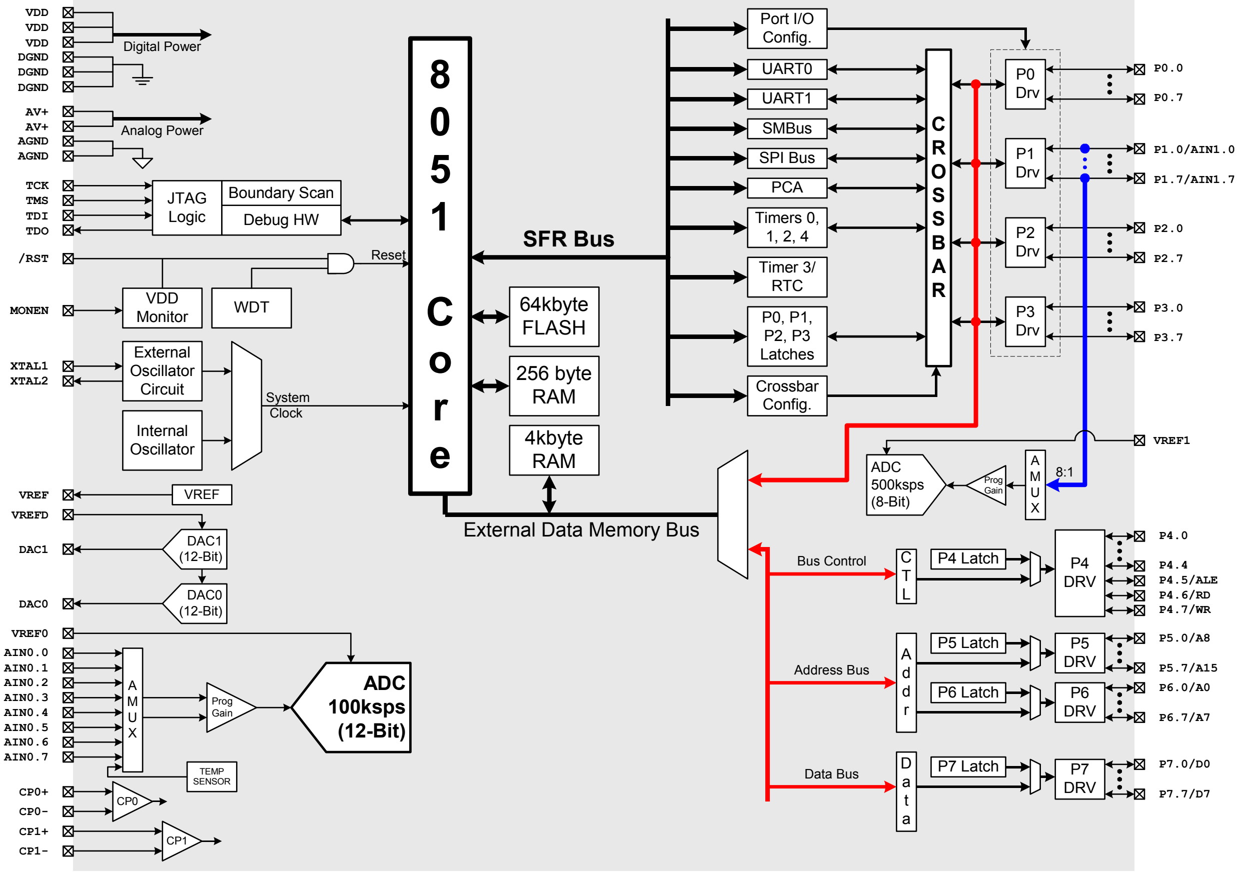
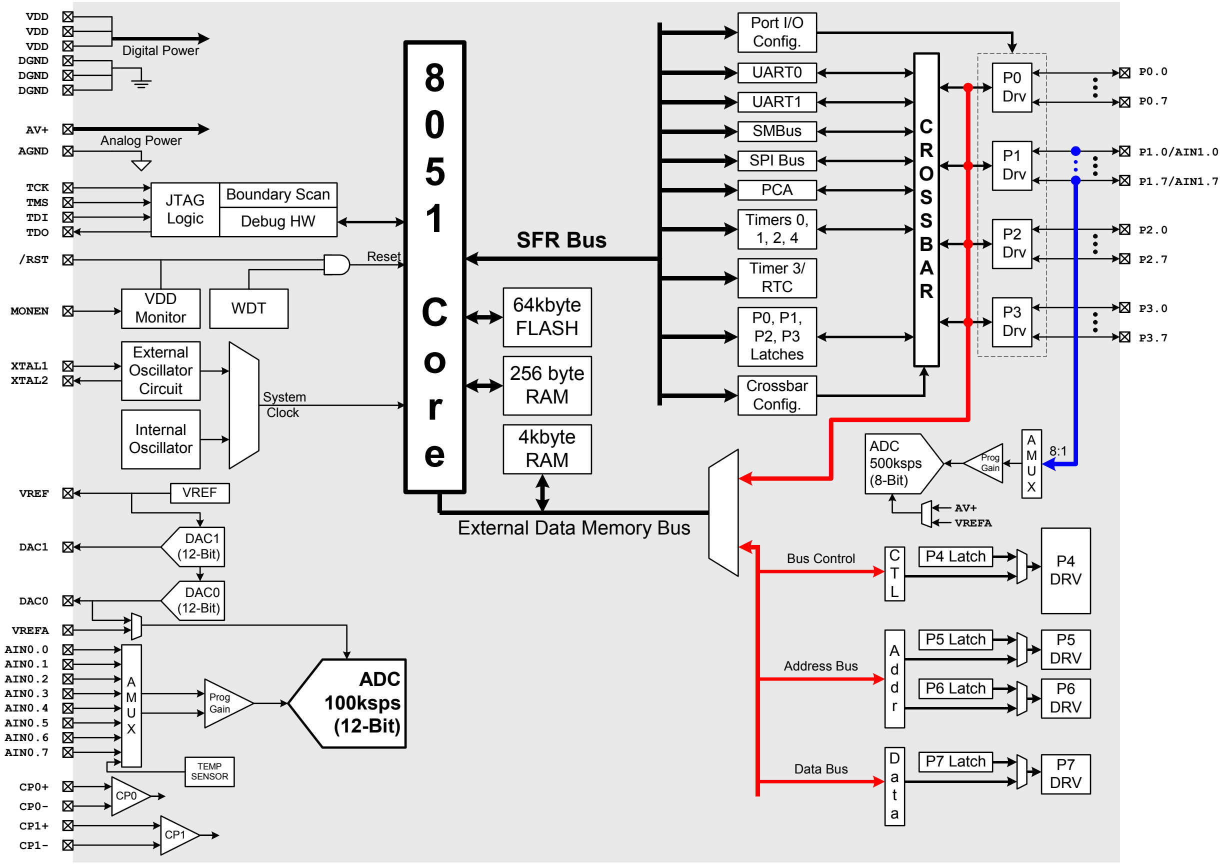
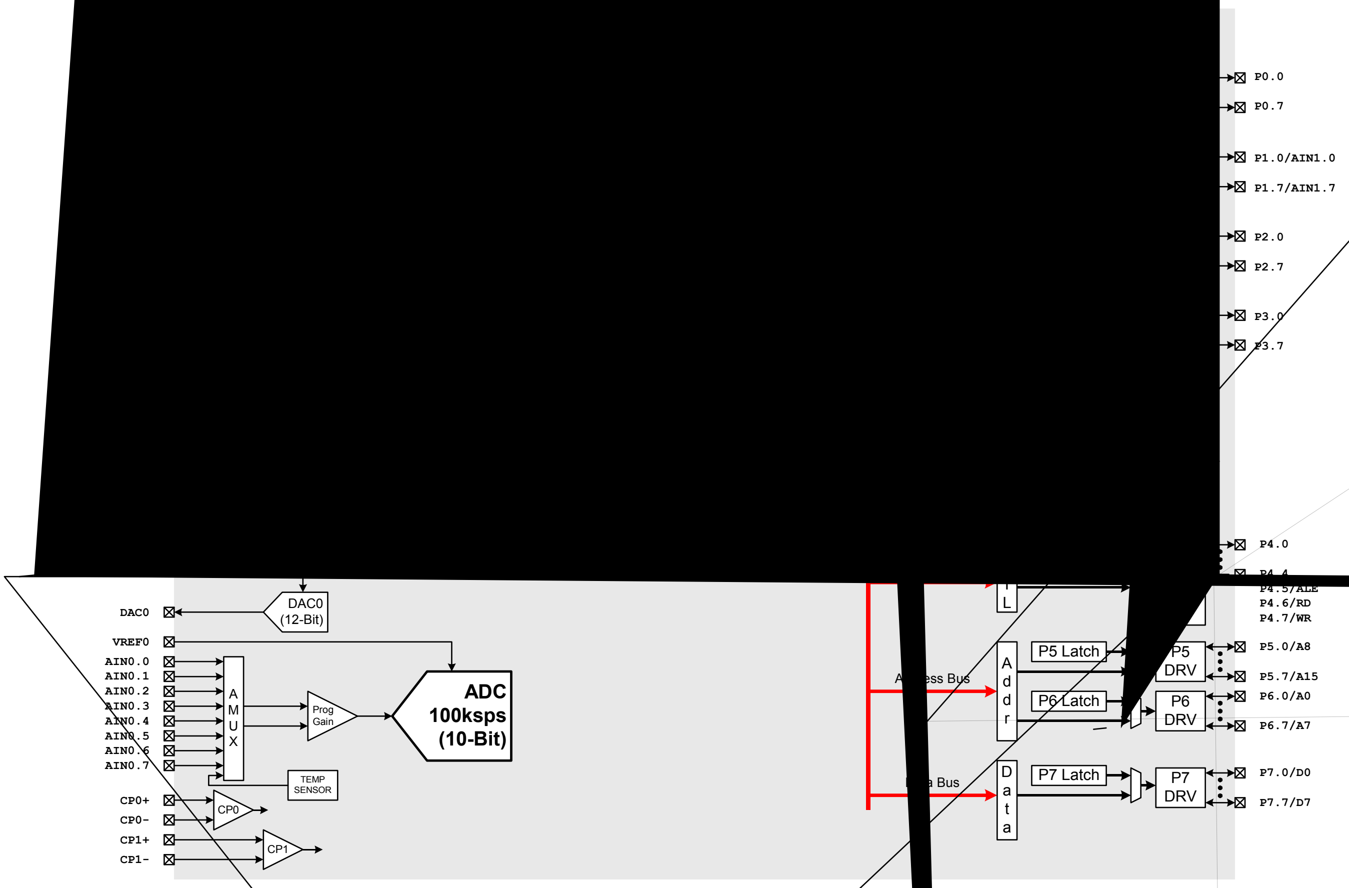

VDD
VDD
VDD
DGND
DGND
DGND
Digital Power
AV+
AV+
AGND
AGND
Analog Power
TCK
TMS
TDI
TDO
JTAG Logic
Boundary Scan
Debug HW
/RST
Reset
MONEN
VDD Monitor
WDT
XTAL1
XTAL2
External Oscillator Circuit
Internal Oscillator
System Clock
VREF
VREF
VREFD
DAC1
DAC1 (12-Bit)
DAC0
DAC0 (12-Bit)
VREF0
AIN0.0
AIN0.1
AIN0.2
AIN0.3
AIN0.4
AIN0.5
AIN0.6
AIN0.7
AMUX
Prog Gain
ADC 100ksps (12-Bit)
TEMP SENSOR
CP0+
CP0-
CP0
CP1+
CP1-
CP1
8051 Core
SFR Bus
64kbyte FLASH
256 byte RAM
4kbyte RAM
External Data Memory Bus
Port I/O Config.
UART0
UART1
SMBus
SPI Bus
PCA
Timers 0, 1, 2, 4
Timer 3/ RTC
P0, P1, P2, P3 Latches
Crossbar Config.
CROSSBAR
P0 Drv
P1 Drv
P2 Drv
P3 Drv
P0.0
P0.7
P1.0/AIN1.0
P1.7/AIN1.7
P2.0
P2.7
P3.0
P3.7
VREF1
ADC 500ksps (8-Bit)
Prog Gain
AMUX
8:1
Bus Control
CTL
P4 Latch
P4 DRV
P4.0
P4.4
P4.5/ALE
P4.6/RD
P4.7/WR
Address Bus
Addr
P5 Latch
P5 DRV
P5.0/A8
P5.7/A15
P6 Latch
P6 DRV
P6.0/A0
P6.7/A7
Data Bus
Data
P7 Latch
P7 DRV
P7.0/D0
P7.7/D7

Figure 1.1. C8051F020 Block Diagram

Figure 1.2. C8051F021 Block Diagram

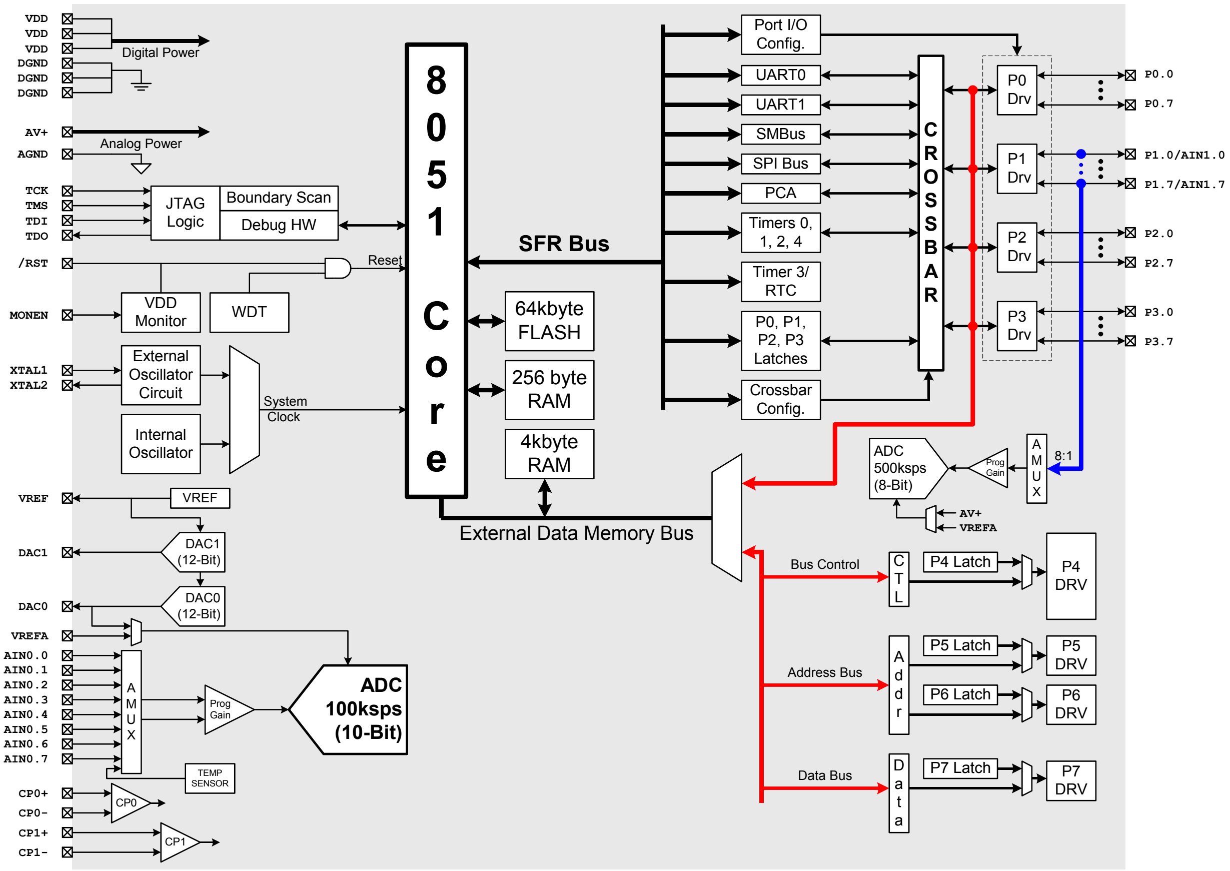

VDD
VDD
VDD
DGND
DGND
DGND
Digital Power
AV+
AGND
Analog Power
TCK
TMS
TDI
TDO
JTAG Logic
Boundary Scan
Debug HW
/RST
Reset
MONEN
VDD Monitor
WDT
XTAL1
XTAL2
External Oscillator Circuit
Internal Oscillator
System Clock
VREF
VREF
DAC1
DAC1 (12-Bit)
DAC0
DAC0 (12-Bit)
VREFA
AIN0.0
AIN0.1
AIN0.2
AIN0.3
AIN0.4
AIN0.5
AIN0.6
AIN0.7
AMUX
Prog Gain
ADC 100ksps (12-Bit)
TEMP SENSOR
CP0+
CP0-
CP1+
CP1-
CP0
CP1
8051 Core
SFR Bus
64kbyte FLASH
256 byte RAM
4kbyte RAM
External Data Memory Bus
Port I/O Config.
UART0
UART1
SMBus
SPI Bus
PCA
Timers 0, 1, 2, 4
Timer 3/ RTC
P0, P1, P2, P3 Latches
Crossbar Config.
CROSSBAR
P0 Drv
P1 Drv
P2 Drv
P3 Drv
P0.0
P0.7
P1.0/AIN1.0
P1.7/AIN1.7
P2.0
P2.7
P3.0
P3.7
ADC 500ksps (8-Bit)
Prog Gain
AMUX
8:1
AV+
VREFA
Bus Control
CTL
P4 Latch
P4 DRV
Address Bus
Addr
P5 Latch
P5 DRV
P6 Latch
P6 DRV
Data Bus
Data
P7 Latch
P7 DRV

[redacted]



P0.0
P0.7
P1.0/AIN1.0
P1.7/AIN1.7
P2.0
P2.7
P3.0
P3.7
P4.0
P4.4
P4.5/ALE
P4.6/RD
P4.7/WR
DAC0
DAC0
(12-Bit)
VREF0
AIN0.0
AIN0.1
AIN0.2
AIN0.3
AIN0.4
AIN0.5
AIN0.6
AIN0.7
AMUX
Prog
Gain
ADC
100ksps
(10-Bit)
TEMP
SENSOR
CP0+
CP0-
CP1+
CP1-
CP0
CP1
L
Address Bus
Addr
Data Bus
Data
P5 Latch
P6 Latch
P7 Latch
P5
DRV
P6
DRV
P7
DRV
P5.0/A8
P5.7/A15
P6.0/A0
P6.7/A7
P7.0/D0
P7.7/D7

Figure 1.4. C8051F023 Block Diagram


VDD
VDD
VDD
DGND
DGND
DGND
Digital Power
AV+
AGND
Analog Power
TCK
TMS
TDI
TDO
JTAG Logic
Boundary Scan
Debug HW
/RST
Reset
MONEN
VDD Monitor
WDT
XTAL1
XTAL2
External Oscillator Circuit
Internal Oscillator
System Clock
VREF
VREF
DAC1
DAC1 (12-Bit)
DAC0
DAC0 (12-Bit)
VREFA
AIN0.0
AIN0.1
AIN0.2
AIN0.3
AIN0.4
AIN0.5
AIN0.6
AIN0.7
AMUX
Prog Gain
ADC 100ksps (10-Bit)
TEMP SENSOR
CP0+
CP0-
CP1+
CP1-
CP0
CP1
8051 Core
SFR Bus
64kbyte FLASH
256 byte RAM
4kbyte RAM
External Data Memory Bus
Port I/O Config.
UART0
UART1
SMBus
SPI Bus
PCA
Timers 0, 1, 2, 4
Timer 3/ RTC
P0, P1, P2, P3 Latches
Crossbar Config.
CROSSBAR
P0 Drv
P1 Drv
P2 Drv
P3 Drv
P0.0
P0.7
P1.0/AIN1.0
P1.7/AIN1.7
P2.0
P2.7
P3.0
P3.7
ADC 500ksps (8-Bit)
Prog Gain
AMUX
8:1
AV+
VREFA
Bus Control
Address Bus
Data Bus
CTL
Addr
Data
P4 Latch
P5 Latch
P6 Latch
P7 Latch
P4 DRV
P5 DRV
P6 DRV
P7 DRV

1.1. CIP-51™ Microcontroller Core

1.1.1. Fully 8051 Compatible

The C8051F020 family utilizes Silicon Labs' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The core has all the peripherals included with a standard 8052, including five 16-bit counter/timers, two full-duplex UARTs, 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space, and 8/4 byte-wide I/O Ports.

1.1.2. Improved Throughput

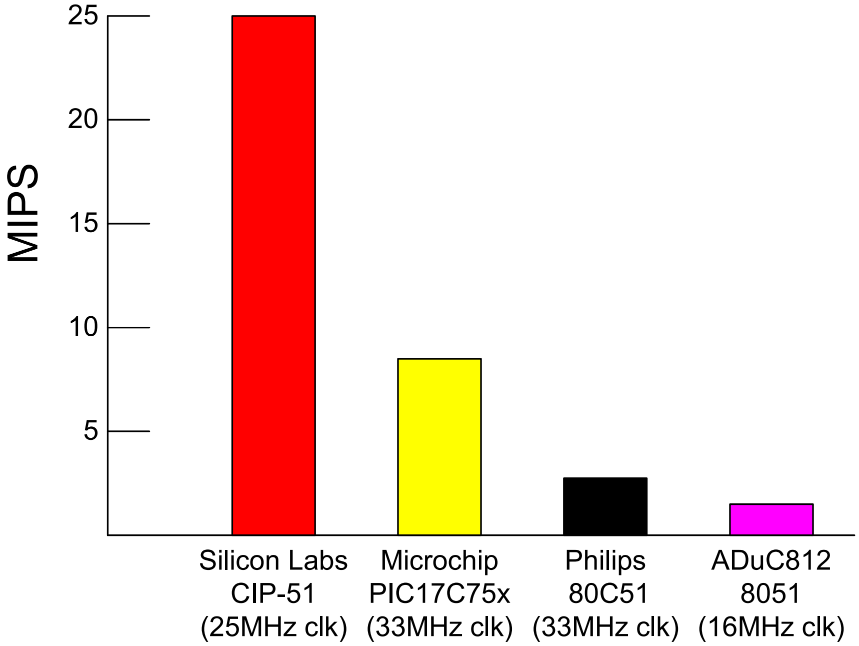
The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute with a maximum system clock of 12-to-24 MHz. By contrast, the CIP-51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

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

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

With the CIP-51's maximum system clock at 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

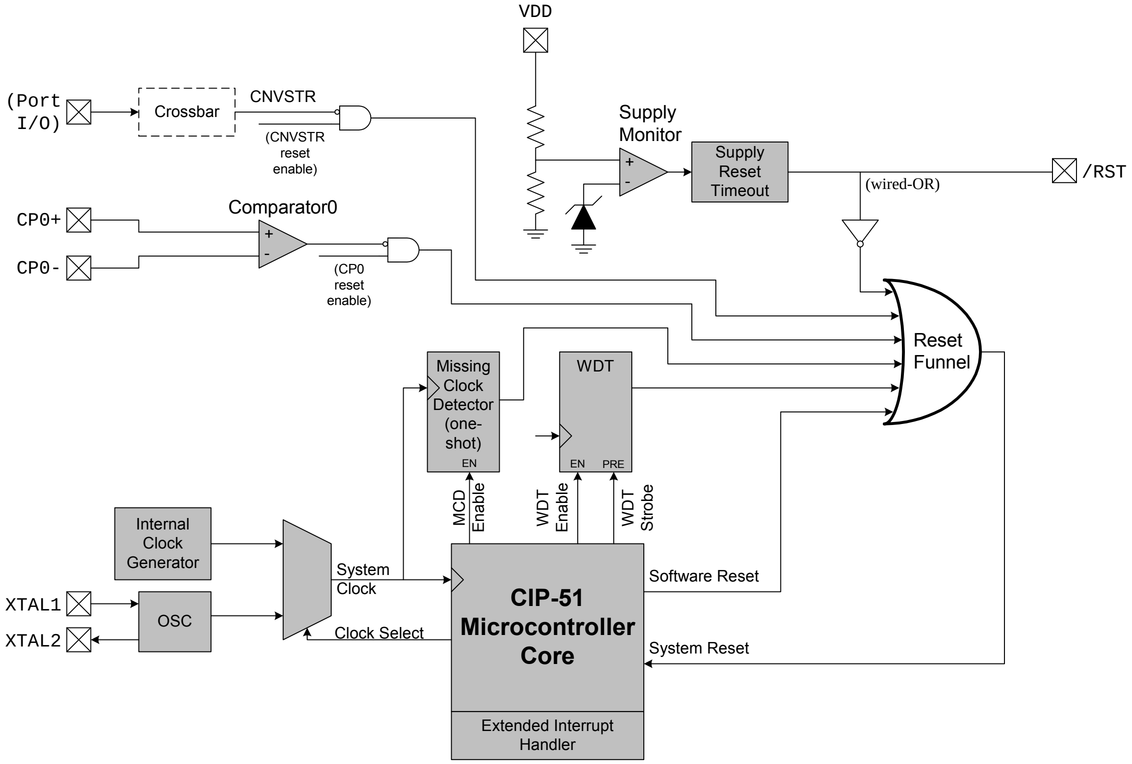
The C8051F020 MCU family includes several key enhancements to the CIP-51 core and peripherals to improve overall performance and ease of use in end applications.

The extended interrupt handler provides 22 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing the numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

There are up to seven reset sources for the MCU: an on-board VDD monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator0, a forced software reset, the CNVSTR 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 16 MHz) internal oscillator as needed.

Figure 1.6. On-Board Clock and Reset


VDD
(Port I/O)
Crossbar
CNVSTR
(CNVSTR reset enable)
Supply Monitor
Supply Reset Timeout
(wired-OR)
/RST
CP0+
CP0-
Comparator0
(CP0 reset enable)
Reset Funnel
Missing Clock Detector (one-shot)
EN
WDT
EN
PRE
MCD Enable
WDT Enable
WDT Strobe
Internal Clock Generator
System Clock
XTAL1
XTAL2
OSC
Clock Select
CIP-51 Microcontroller Core
Software Reset
System Reset
Extended Interrupt Handler

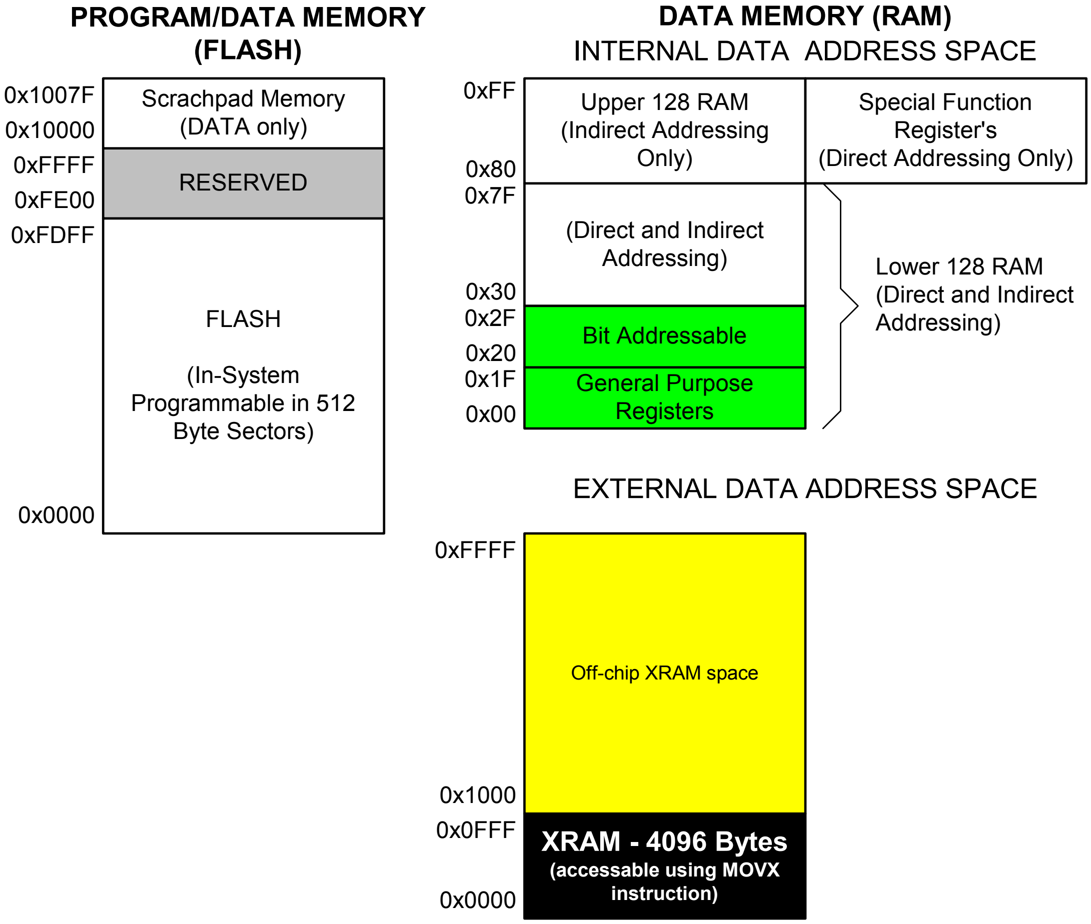
1.2. On-Chip Memory

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

The CIP-51 in the C8051F020/1/2/3 MCUs additionally has an on-chip 4k byte RAM block and an external memory interface (EMIF) for accessing off-chip data memory. The on-chip 4k byte block can be addressed over the entire 64k external data memory address range (overlapping 4k boundaries). External data memory address space can be mapped to on-chip memory only, off-chip memory only, or a combination of the two (addresses up to 4k directed to on-chip, above 4k directed to EMIF). The EMIF is also configurable for multiplexed or non-multiplexed address/data lines.

The MCU's program memory consists of 64k bytes of FLASH. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. The 512 bytes from addresses 0xFE00 to 0xFFFF are reserved for factory use. There is also a single 128 byte sector at address 0x10000 to 0x1007F, which may be useful as a small table for software constants. See Figure 1.7 for the MCU system memory map.

Figure 1.7. On-Chip Memory Map


PROGRAM/DATA MEMORY (FLASH)
0x1007F
0x10000
Scrachpad Memory (DATA only)
0xFFFF
0xFE00
RESERVED
0xFDFF
FLASH
(In-System Programmable in 512 Byte Sectors)
0x0000
DATA MEMORY (RAM)
INTERNAL DATA ADDRESS SPACE
0xFF
0x80
Upper 128 RAM (Indirect Addressing Only)
Special Function Register's (Direct Addressing Only)
0x7F
(Direct and Indirect Addressing)
0x30
0x2F
Bit Addressable
0x20
0x1F
General Purpose Registers
0x00
Lower 128 RAM (Direct and Indirect Addressing)
EXTERNAL DATA ADDRESS SPACE
0xFFFF
Off-chip XRAM space
0x1000
0x0FFF
XRAM - 4096 Bytes (accessable using MOVX instruction)
0x0000

1.3. JTAG Debug and Boundary Scan

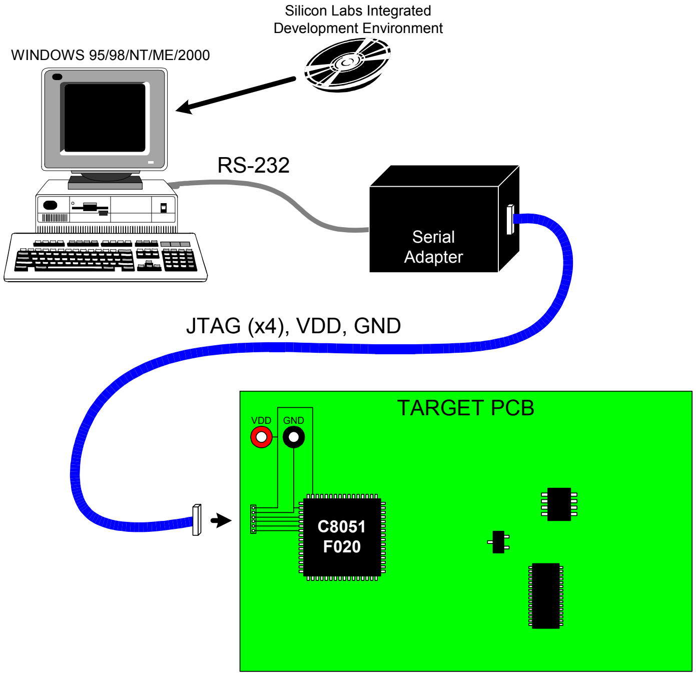
The C8051F020 family has on-chip JTAG boundary scan and debug circuitry that provides *non-intrusive, full speed, in-circuit debugging using the production part installed in the end application*, via the four-pin JTAG interface. The JTAG port is fully compliant to IEEE 1149.1, providing full boundary scan for test and manufacturing purposes.

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

The C8051F020DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F020/1/2/3 MCUs. The kit includes software with a developer's studio and debugger, an integrated 8051 assembler, and an RS-232 to JTAG serial adapter. It also has a target application board with the associated MCU installed, plus the RS-232 and JTAG cables, and wall-mount power supply. The Development Kit requires a Windows 95/98/NT/ME/2000 computer with one available RS-232 serial port. As shown in Figure 1.8, the PC is connected via RS-232 to the Serial Adapter. A six-inch ribbon cable connects the Serial Adapter to the user's application board, picking up the four JTAG pins and VDD and GND. The Serial Adapter takes its power from the application board; it requires roughly 20 mA at 2.7-3.6 V. For applications where there is not sufficient power available from the target system, the provided power supply can be connected directly to the Serial Adapter.

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 analog peripherals.

Figure 1.8. Development/In-System Debug Diagram



1.4. Programmable Digital I/O and Crossbar

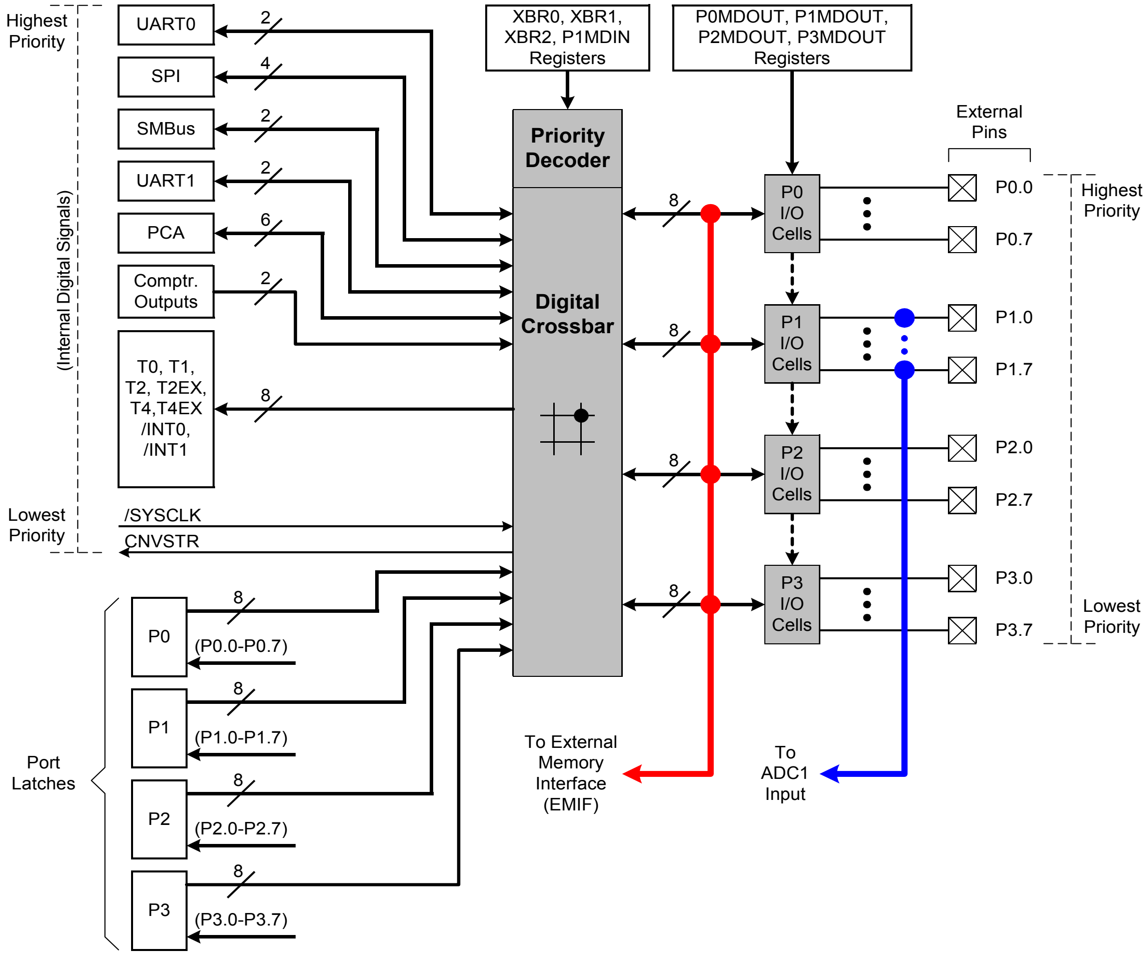
The standard 8051 Ports (0, 1, 2, and 3) are available on the MCUs. The C8051F020/2 have 4 additional ports (4, 5, 6, and 7) for a total of 64 general-purpose port I/O. The Port I/O behave like the standard 8051 with a few enhancements.

Each Port I/O pin can be configured as either a push-pull or open-drain output. Also, the "weak 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 essentially 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, all combinations of functions are supported.

The on-chip counter/timers, serial buses, HW interrupts, ADC Start of Conversion input, 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

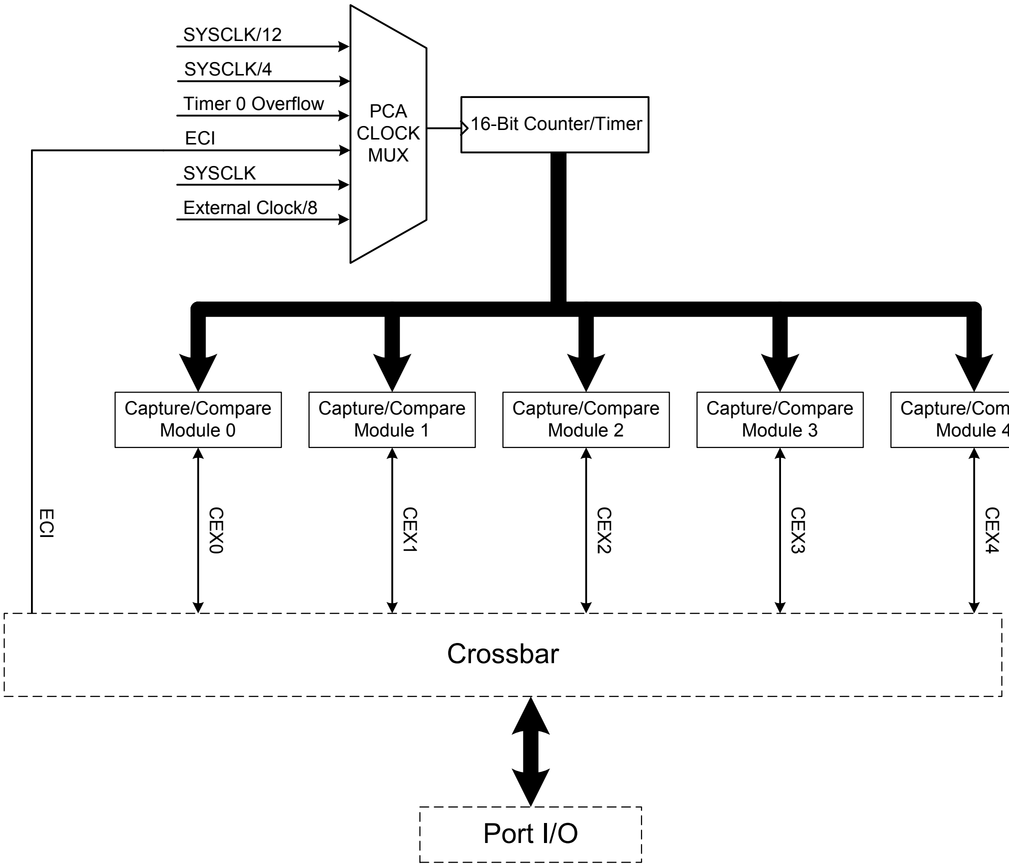

Highest Priority
UART0
2
SPI
4
SMBus
2
UART1
2
PCA
6
Comptr. Outputs
2
T0, T1, T2, T2EX, T4,T4EX /INT0, /INT1
8
(Internal Digital Signals)
Lowest Priority
/SYSCLK
CNVSTR
XBR0, XBR1, XBR2, P1MDIN Registers
P0MDOUT, P1MDOUT, P2MDOUT, P3MDOUT Registers
Priority Decoder
Digital Crossbar
8
8
8
8
P0 I/O Cells
P1 I/O Cells
P2 I/O Cells
P3 I/O Cells
External Pins
P0.0
P0.7
P1.0
P1.7
P2.0
P2.7
P3.0
P3.7
Highest Priority
Lowest Priority
Port Latches
P0
8
(P0.0-P0.7)
P1
8
(P1.0-P1.7)
P2
8
(P2.0-P2.7)
P3
8
(P3.0-P3.7)
To External Memory Interface (EMIF)
To ADC1 Input

1.5. Programmable Counter Array

The C8051F020 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 5 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


SYSCLK/12
SYSCLK/4
Timer 0 Overflow
ECI
SYSCLK
External Clock/8
PCA CLOCK MUX
16-Bit Counter/Timer
Capture/Compare Module 0
Capture/Compare Module 1
Capture/Compare Module 2
Capture/Compare Module 3
Capture/Compare Module 4
ECI
CEX0
CEX1
CEX2
CEX3
CEX4
Crossbar
Port I/O

1.6. Serial Ports

The C8051F020 MCU Family includes two Enhanced Full-Duplex UARTs, SPI Bus, and SMBus/I^{2}C. 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.

1.7. 12-Bit Analog to Digital Converter

The C8051F020/1 has an on-chip 12-bit SAR ADC (ADC0) with a 9-channel input multiplexer and programmable gain amplifier. With a maximum throughput of 100 ksps, the ADC offers true 12-bit accuracy with an INL of ±1LSB. C8051F022/3 devices include a 10-bit SAR ADC with similar specifications and configuration options. The ADC0 voltage reference is selected between the DAC0 output and an external VREF pin. On C8051F020/2 devices, ADC0 has its own dedicated VREF0 input pin; on C8051F021/3 devices, the ADC0 shares the VREFA input pin with the 8-bit ADC1. The on-chip 15 ppm/°C voltage reference may generate the voltage reference for other system components or the on-chip ADCs via the VREF output pin.

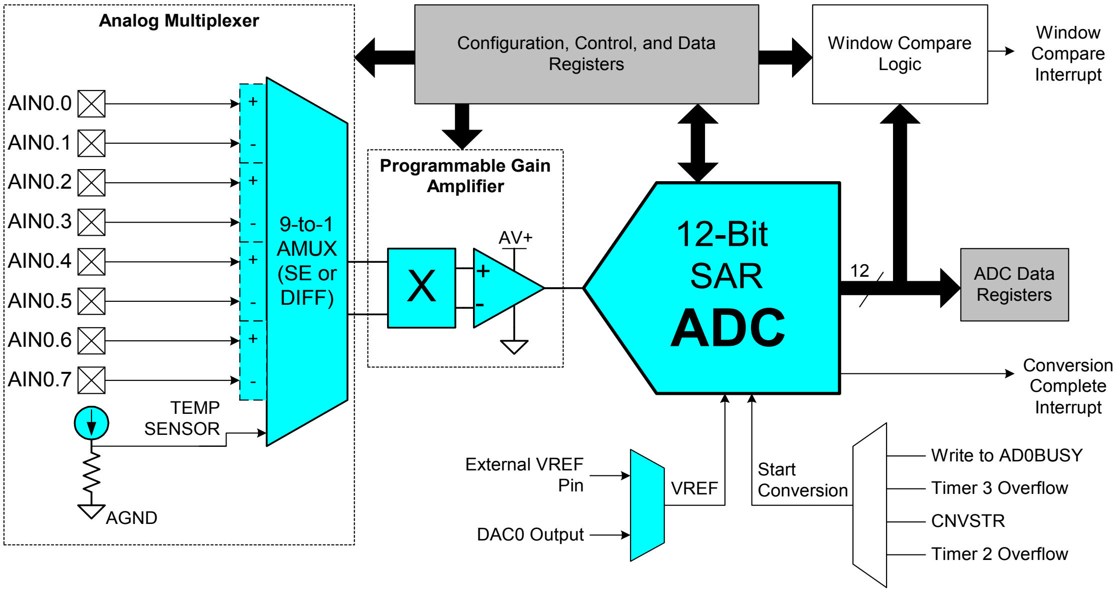
The ADC is under full control of the CIP-51 microcontroller via its associated Special Function Registers. One input channel is tied to an internal temperature sensor, while the other eight channels are available externally. Each pair of the eight external input channels can be configured as either two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

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

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

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

Figure 1.11. 12-Bit ADC Block Diagram


Analog Multiplexer
AIN0.0
AIN0.1
AIN0.2
AIN0.3
AIN0.4
AIN0.5
AIN0.6
AIN0.7
TEMP SENSOR
AGND
+
-
9-to-1 AMUX (SE or DIFF)
Configuration, Control, and Data Registers
Window Compare Logic
Window Compare Interrupt
Programmable Gain Amplifier
X
AV+
12-Bit SAR ADC
12
ADC Data Registers
Conversion Complete Interrupt
External VREF Pin
DAC0 Output
VREF
Start Conversion
Write to AD0BUSY
Timer 3 Overflow
CNVSTR
Timer 2 Overflow

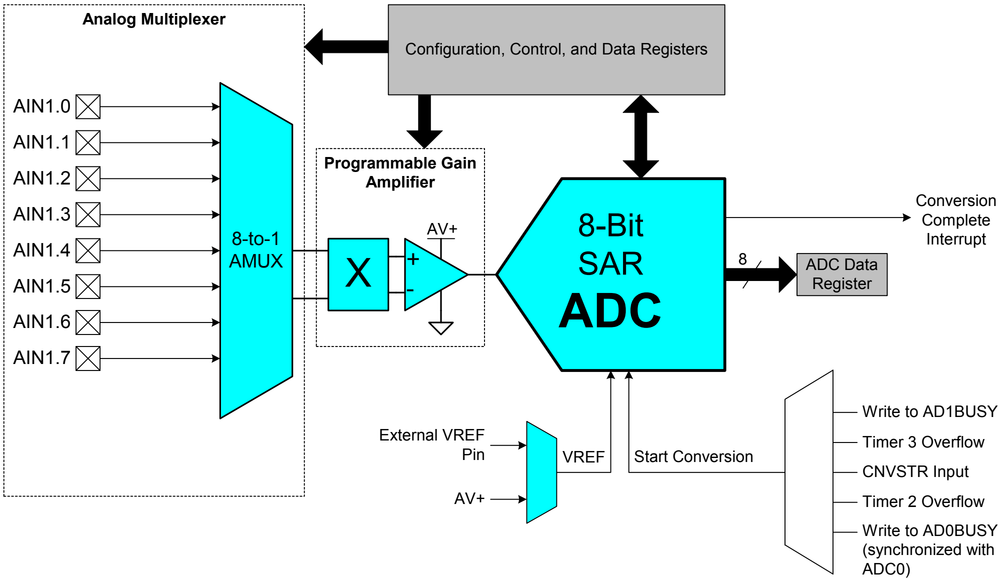
1.8. 8-Bit Analog to Digital Converter

The C8051F020/1/2/3 has an on-board 8-bit SAR ADC (ADC1) with an 8-channel input multiplexer and programmable gain amplifier. This ADC features a 500 ksps maximum throughput and true 8-bit accuracy with an INL of ±1LSB. Eight input pins are available for measurement. The ADC is under full control of the CIP-51 microcontroller via the Special Function Registers. The ADC1 voltage reference is selected between the analog power supply (AV+) and an external VREF pin. On C8051F020/2 devices, ADC1 has its own dedicated VREF1 input pin; on C8051F021/3 devices, ADC1 shares the VREFA input pin with the 12/10-bit ADC0. User software may put ADC1 into shutdown mode to save power.

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

A flexible conversion scheduling system allows ADC1 conversions to be initiated by software commands, timer overflows, or an external input signal. ADC1 conversions may also be synchronized with ADC0 software-commanded conversions. Conversion completions are indicated by a status bit and an interrupt (if enabled), and the resulting 8-bit data word is latched into an SFR upon completion.

Figure 1.12. 8-Bit ADC Diagram


Analog Multiplexer
Configuration, Control, and Data Registers
AIN1.0
AIN1.1
AIN1.2
AIN1.3
AIN1.4
AIN1.5
AIN1.6
AIN1.7
8-to-1 AMUX
Programmable Gain Amplifier
X
AV+
+
-
8-Bit SAR ADC
Conversion Complete Interrupt
8
ADC Data Register
External VREF Pin
AV+
VREF
Start Conversion
Write to AD1BUSY
Timer 3 Overflow
CNVSTR Input
Timer 2 Overflow
Write to AD0BUSY (synchronized with ADC0)

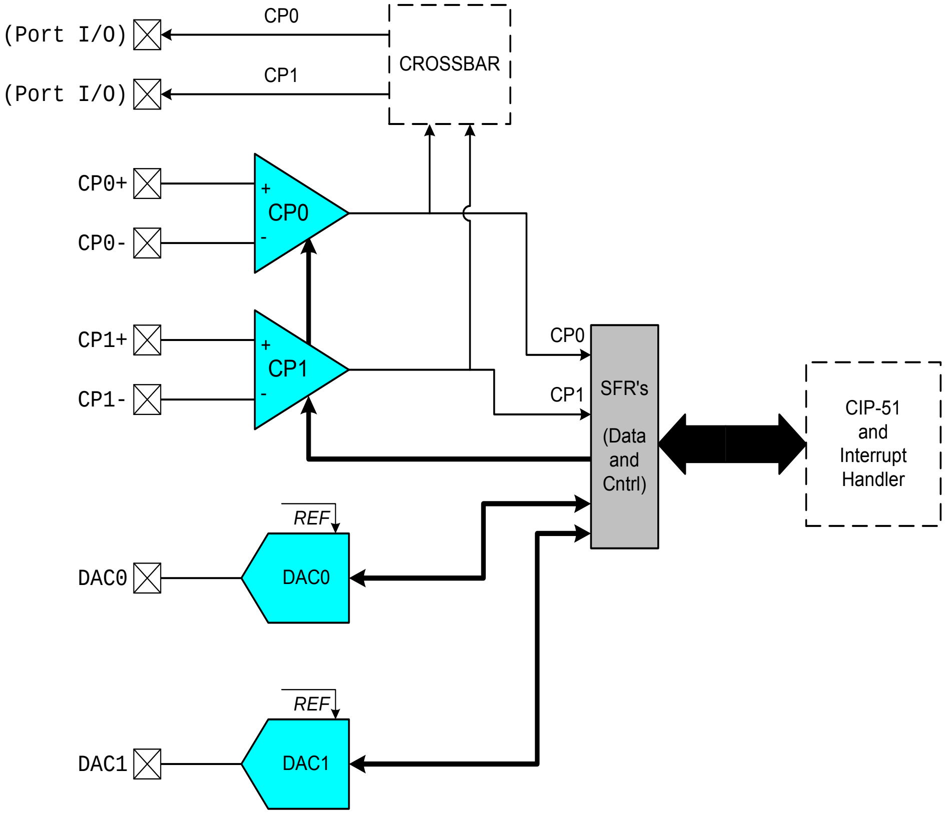
1.9. Comparators and DACs

Each C8051F020/1/2/3 MCU has two 12-bit DACs and two comparators on chip. The MCU data and control interface to each comparator and DAC is via the Special Function Registers. The MCU can place any DAC or comparator in low power shutdown mode.

The comparators have software programmable hysteresis. Each comparator can generate an interrupt on its rising edge, falling edge, or both; these interrupts are capable of waking up the MCU from sleep mode. The comparators' output state can also be polled in software. The comparator outputs can be programmed to appear on the Port I/O pins via the Crossbar.

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 a Timer 2, 3, or 4 overflow. The DAC voltage reference is supplied via the dedicated VREFD input pin on C8051F020/2 devices or via the internal voltage reference on C8051F021/3 devices. The DACs are especially useful as references for the comparators or offsets for the differential inputs of the ADC.

Figure 1.13. Comparator and DAC Diagram


(Port I/O)
(Port I/O)
CP0
CP1
CROSSBAR
CP0+
CP0-
CP0
+
-
CP1+
CP1-
CP1
+
-
CP0
CP1
SFR's
(Data
and
Cntrl)
CIP-51
and
Interrupt
Handler
REF
DAC0
DAC0
REF
DAC1
DAC1

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 and Port I/O) with respect to DGND		-0.3		VDD + 0.3	V
Voltage on any Port I/O Pin or /RST with respect to DGND		-0.3		5.8	V
Voltage on VDD with respect to DGND		-0.3		4.2	V
Maximum Total current through VDD, AV+, DGND, and AGND				800	mA
Maximum output current sunk by any Port pin				100	mA
Maximum output current sunk by any other I/O pin				50	mA
Maximum output current sourced by any Port pin				100	mA
Maximum output current sourced by any other I/O pin				50	mA

* 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 1.1. Global DC Electrical Characteristics

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

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Analog Supply Voltage		2.7[†]	3.0	3.6	V
Analog Supply Current	AV+=2.7 V, Internal REF, ADC, DAC, Comparators all active		1.7		mA
Analog Supply Current with analog sub-systems inactive	AV+=2.7 V, Internal REF, ADC, DAC, Comparators all disabled, oscillator disabled, VDD Monitor disabled		0.2		µA
Analog-to-Digital Supply Delta (|VDD - AV+|)				0.5	V
Digital Supply Voltage		2.7	3.0	3.6	V
Digital Supply Current with CPU active	VDD=2.7 V, Clock=25 MHz VDD=2.7 V, Clock=1 MHz VDD=2.7 V, Clock=32 kHz		10 0.5 20		mA mA µA
Digital Supply Current with CPU inactive (not accessing FLASH)	VDD=2.7 V, Clock=25 MHz VDD=2.7 V, Clock=1 MHz VDD=2.7 V, Clock=32 kHz		5 0.2 10		mA mA µA
Digital Supply Current (shutdown)	VDD=2.7 V, Oscillator not running, VDD Monitor disabled		0.2		µA
Digital Supply RAM Data Retention Voltage			1.5		V
Specified Operating Temperature Range		-40		+85	°C
SYSCLK (system clock frequency)		0[‡]		25	MHz
Tsysl (SYSCLK low time)		18			ns
Tsysh (SYSCLK high time)		18			ns

[†] Analog Supply AV+ must be greater than 1 V for VDD monitor to operate.

[‡] 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
	F020 F022	F021 F023		
VDD	37, 64, 90	24, 41, 57		Digital Supply Voltage. Must be tied to +2.7 to +3.6 V.
DGND	38, 63, 89	25, 40, 56		Digital Ground. Must be tied to Ground.
AV+	11, 14	6		Analog Supply Voltage. Must be tied to +2.7 to +3.6 V.
AGND	10, 13	5		Analog Ground. Must be tied to Ground.
TMS	1	58	D In	JTAG Test Mode Select with internal pull-up.
TCK	2	59	D In	JTAG Test Clock with internal pull-up.
TDI	3	60	D In	JTAG Test Data Input with internal pull-up. TDI is latched on the rising edge of TCK.
TDO	4	61	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	5	62	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	17	A In	Crystal Input. This pin is the return for the internal oscillator circuit for a crystal or ceramic resonator. For a precision internal clock, connect a crystal or ceramic resonator from XTAL1 to XTAL2. If overdriven by an external CMOS clock, this becomes the system clock.
XTAL2	27	18	A Out	Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator.
MONEN	28	19	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.
VREF	12	7	A I/O	Bandgap Voltage Reference Output (all devices). DAC Voltage Reference Input (F021/3 only).
VREFA		8	A In	ADC0 and ADC1 Voltage Reference Input.
VREF0	16		A In	ADC0 Voltage Reference Input.
VREF1	17		A In	ADC1 Voltage Reference Input.
VREFD	15		A In	DAC Voltage Reference Input.

Table 4.1. Pin Definitions

Name	Pin Numbers		Type	Description
	F020 F022	F021 F023		
AIN0.0	18	9	A In	ADC0 Input Channel 0 (See ADC0 Specification for complete description).
AIN0.1	19	10	A In	ADC0 Input Channel 1 (See ADC0 Specification for complete description).
AIN0.2	20	11	A In	ADC0 Input Channel 2 (See ADC0 Specification for complete description).
AIN0.3	21	12	A In	ADC0 Input Channel 3 (See ADC0 Specification for complete description).
AIN0.4	22	13	A In	ADC0 Input Channel 4 (See ADC0 Specification for complete description).
AIN0.5	23	14	A In	ADC0 Input Channel 5 (See ADC0 Specification for complete description).
AIN0.6	24	15	A In	ADC0 Input Channel 6 (See ADC0 Specification for complete description).
AIN0.7	25	16	A In	ADC0 Input Channel 7 (See ADC0 Specification for complete description).
CP0+	9	4	A In	Comparator 0 Non-Inverting Input.
CP0-	8	3	A In	Comparator 0 Inverting Input.
CP1+	7	2	A In	Comparator 1 Non-Inverting Input.
CP1-	6	1	A In	Comparator 1 Inverting Input.
DAC0	100	64	A Out	Digital to Analog Converter 0 Voltage Output. (See DAC Specification for complete description).
DAC1	99	63	A Out	Digital to Analog Converter 1 Voltage Output. (See DAC Specification for complete description).
P0.0	62	55	D I/O	Port 0.0. See Port Input/Output section for complete description.
P0.1	61	54	D I/O	Port 0.1. See Port Input/Output section for complete description.
P0.2	60	53	D I/O	Port 0.2. See Port Input/Output section for complete description.
P0.3	59	52	D I/O	Port 0.3. See Port Input/Output section for complete description.
P0.4	58	51	D I/O	Port 0.4. See Port Input/Output section for complete description.
ALE/P0.5	57	50	D I/O	ALE Strobe for External Memory Address bus (multiplexed mode) Port 0.5 See Port Input/Output section for complete description.

Table 4.1. Pin Definitions

Name	Pin Numbers		Type	Description
	F020	F021		
	F022	F023		
/RD/P0.6	56	49	D I/O	/RD Strobe for External Memory Address bus Port 0.6 See Port Input/Output section for complete description.
/WR/P0.7	55	48	D I/O	/WR Strobe for External Memory Address bus Port 0.7 See Port Input/Output section for complete description.
AIN1.0/A8/P1.0	36	29	A In D I/O	ADC1 Input Channel 0 (See ADC1 Specification for complete description). Bit 8 External Memory Address bus (Non-multiplexed mode) Port 1.0 See Port Input/Output section for complete description.
AIN1.1/A9/P1.1	35	28	A In D I/O	Port 1.1. See Port Input/Output section for complete description.
AIN1.2/A10/P1.2	34	27	A In D I/O	Port 1.2. See Port Input/Output section for complete description.
AIN1.3/A11/P1.3	33	26	A In D I/O	Port 1.3. See Port Input/Output section for complete description.
AIN1.4/A12/P1.4	32	23	A In D I/O	Port 1.4. See Port Input/Output section for complete description.
AIN1.5/A13/P1.5	31	22	A In D I/O	Port 1.5. See Port Input/Output section for complete description.
AIN1.6/A14/P1.6	30	21	A In D I/O	Port 1.6. See Port Input/Output section for complete description.
AIN1.7/A15/P1.7	29	20	A In D I/O	Port 1.7. See Port Input/Output section for complete description.
A8m/A0/P2.0	46	37	D I/O	Bit 8 External Memory Address bus (Multiplexed mode) Bit 0 External Memory Address bus (Non-multiplexed mode) Port 2.0 See Port Input/Output section for complete description.
A9m/A1/P2.1	45	36	D I/O	Port 2.1. See Port Input/Output section for complete description.
A10m/A2/P2.2	44	35	D I/O	Port 2.2. See Port Input/Output section for complete description.
A11m/A3/P2.3	43	34	D I/O	Port 2.3. See Port Input/Output section for complete description.
A12m/A4/P2.4	42	33	D I/O	Port 2.4. See Port Input/Output section for complete description.
A13m/A5/P2.5	41	32	D I/O	Port 2.5. See Port Input/Output section for complete description.

A14m/A6/P2.6	40	31	D I/O	Port 2.6. See Port Input/Output section for complete description.
A15m/A7/P2.7	39	30	D I/O	Port 2.7. See Port Input/Output section for complete description.
AD0/D0/P3.0	54	470 Tc(54)TjETEMC/Span <<03		

Table 4.1. Pin Definitions

Name	Pin Numbers		Type	Description
	F020	F021		
	F022	F023		
A11/P5.3	85		D I/O	Port 5.3. See Port Input/Output section for complete description.
A12/P5.4	84		D I/O	Port 5.4. See Port Input/Output section for complete description.
A13/P5.5	83		D I/O	Port 5.5. See Port Input/Output section for complete description.
A14/P5.6	82		D I/O	Port 5.6. See Port Input/Output section for complete description.
A15/P5.7	81		D I/O	Port 5.7. See Port Input/Output section for complete description.
A8m/A0/P6.0	80		D I/O	Bit 8 External Memory Address bus (Multiplexed mode) Bit 0 External Memory Address bus (Non-multiplexed mode) Port 6.0 See Port Input/Output section for complete description.
A9m/A1/P6.1	79		D I/O	Port 6.1. See Port Input/Output section for complete description.
A10m/A2/P6.2	78		D I/O	Port 6.2. See Port Input/Output section for complete description.
A11m/A3/P6.3	77		D I/O	Port 6.3. See Port Input/Output section for complete description.
A12m/A4/P6.4	76		D I/O	Port 6.4. See Port Input/Output section for complete description.
A13m/A5/P6.5	75		D I/O	Port 6.5. See Port Input/Output section for complete description.
A14m/A6/P6.6	74		D I/O	Port 6.6. See Port Input/Output section for complete description.
A15m/A7/P6.7	73		D I/O	Port 6.7. See Port Input/Output section for complete description.
AD0/D0/P7.0	72		D I/O	Bit 0 External Memory Address/Data bus (Multiplexed mode) Bit 0 External Memory Data bus (Non-multiplexed mode) Port 7.0 See Port Input/Output section for complete description.
AD1/D1/P7.1	71		D I/O	Port 7.1. See Port Input/Output section for complete description.
AD2/D2/P7.2	70		D I/O	Port 7.2. See Port Input/Output section for complete description.
AD3/D3/P7.3	69		D I/O	Port 7.3. See Port Input/Output section for complete description.
AD4/D4/P7.4	68		D I/O	Port 7.4. See Port Input/Output section for complete description.
AD5/D5/P7.5	67		D I/O	Port 7.5. See Port Input/Output section for complete description.
AD6/D6/P7.6	66		D I/O	Port 7.6. See Port Input/Output section for complete description.
AD7/D7/P7.7	65		D I/O	Port 7.7. See Port Input/Output section for complete description.

Figure 4.1. TQFP-100 Pinout Diagram

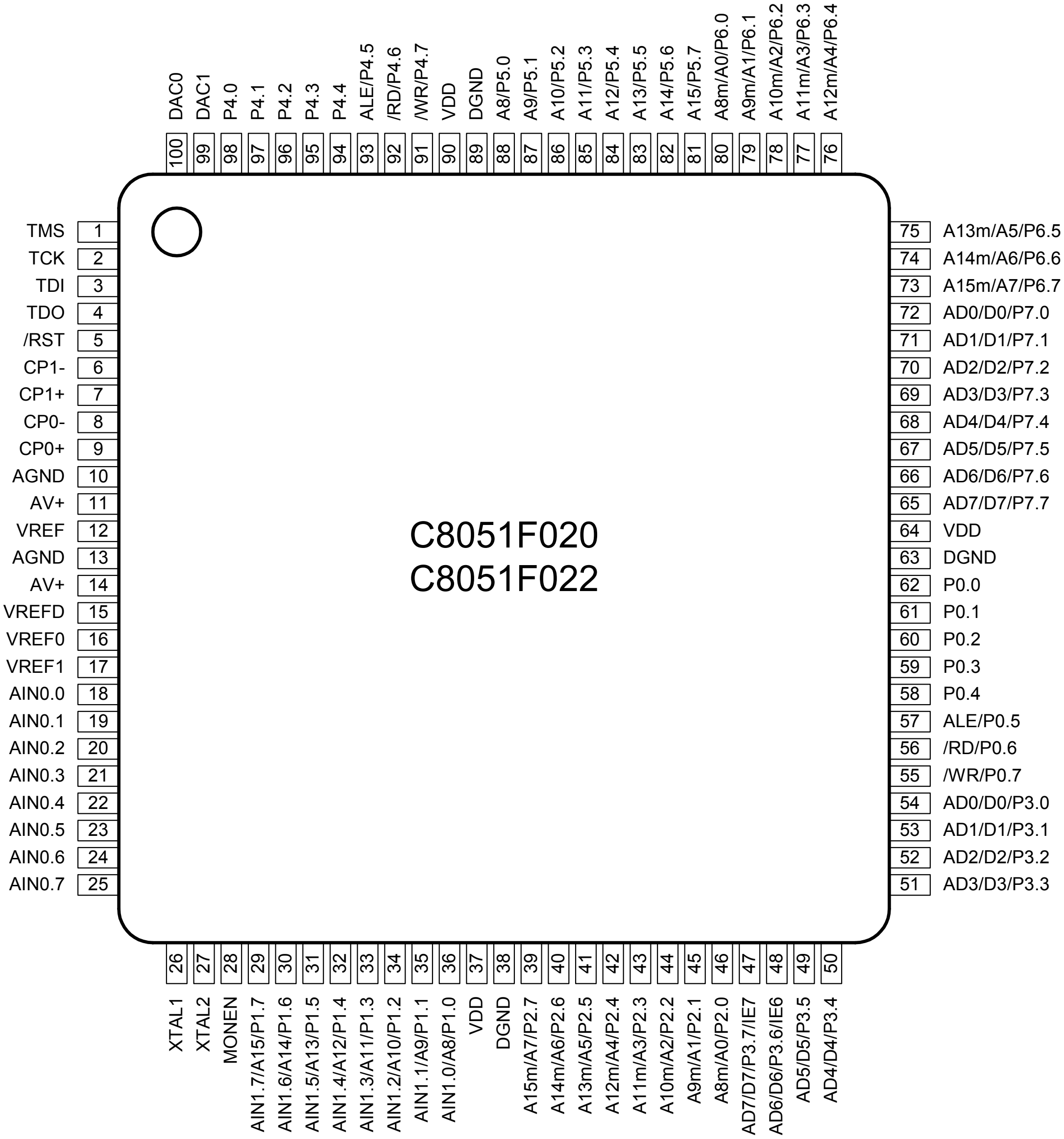
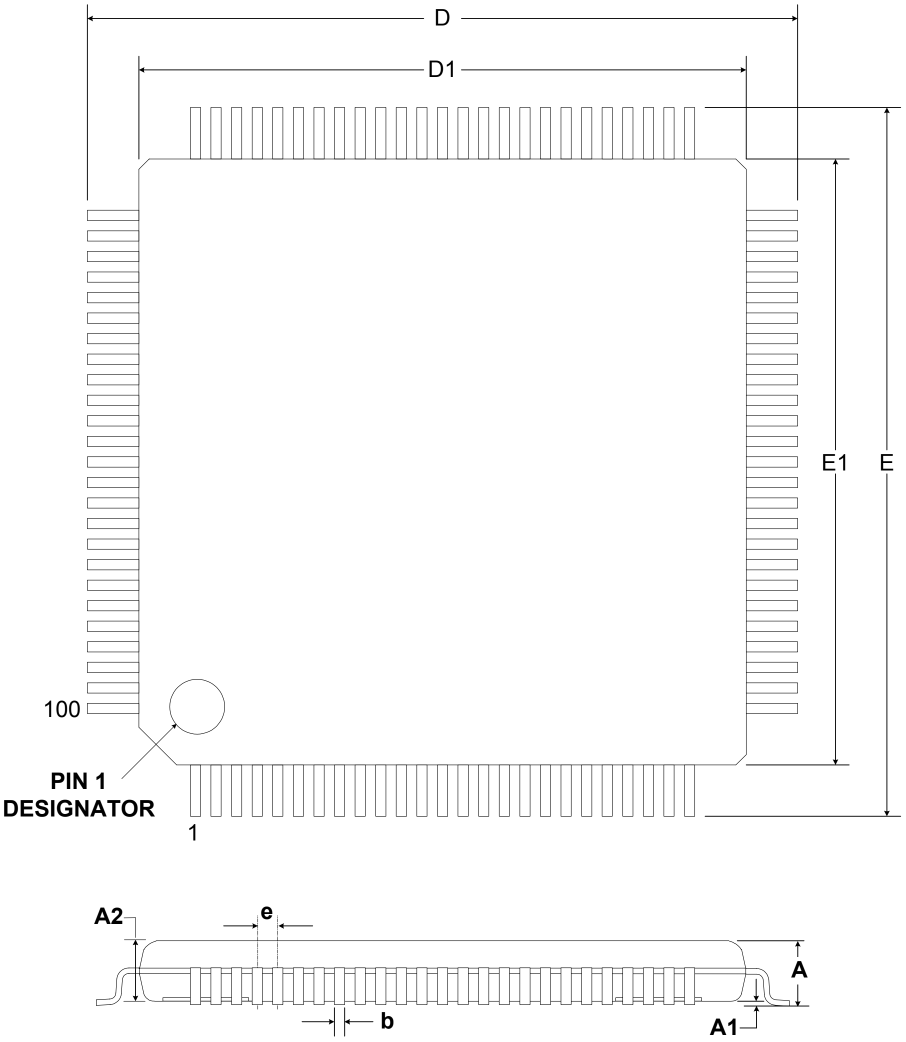

C8051F020
C8051F022
1 TMS
2 TCK
3 TDI
4 TDO
5 /RST
6 CP1-
7 CP1+
8 CP0-
9 CP0+
10 AGND
11 AV+
12 VREF
13 AGND
14 AV+
15 VREFD
16 VREF0
17 VREF1
18 AIN0.0
19 AIN0.1
20 AIN0.2
21 AIN0.3
22 AIN0.4
23 AIN0.5
24 AIN0.6
25 AIN0.7
26 XTAL1
27 XTAL2
28 MONEN
29 AIN1.7/A15/P1.7
30 AIN1.6/A14/P1.6
31 AIN1.5/A13/P1.5
32 AIN1.4/A12/P1.4
33 AIN1.3/A11/P1.3
34 AIN1.2/A10/P1.2
35 AIN1.1/A9/P1.1
36 AIN1.0/A8/P1.0
37 VDD
38 DGND
39 A15m/A7/P2.7
40 A14m/A6/P2.6
41 A13m/A5/P2.5
42 A12m/A4/P2.4
43 A11m/A3/P2.3
44 A10m/A2/P2.2
45 A9m/A1/P2.1
46 A8m/A0/P2.0
47 AD7/D7/P3.7/IE7
48 AD6/D6/P3.6/IE6
49 AD5/D5/P3.5
50 AD4/D4/P3.4
51 AD3/D3/P3.3
52 AD2/D2/P3.2
53 AD1/D1/P3.1
54 AD0/D0/P3.0
55 /WR/P0.7
56 /RD/P0.6
57 ALE/P0.5
58 P0.4
59 P0.3
60 P0.2
61 P0.1
62 P0.0
63 DGND
64 VDD
65 AD7/D7/P7.7
66 AD6/D6/P7.6
67 AD5/D5/P7.5
68 AD4/D4/P7.4
69 AD3/D3/P7.3
70 AD2/D2/P7.2
71 AD1/D1/P7.1
72 AD0/D0/P7.0
73 A15m/A7/P6.7
74 A14m/A6/P6.6
75 A13m/A5/P6.5
76 A12m/A4/P6.4
77 A11m/A3/P6.3
78 A10m/A2/P6.2
79 A9m/A1/P6.1
80 A8m/A0/P6.0
81 A15/P5.7
82 A14/P5.6
83 A13/P5.5
84 A12/P5.4
85 A11/P5.3
86 A10/P5.2
87 A9/P5.1
88 A8/P5.0
89 DGND
90 VDD
91 /WR/P4.7
92 /RD/P4.6
93 ALE/P4.5
94 P4.4
95 P4.3
96 P4.2
97 P4.1
98 P4.0
99 DAC1
100 DAC0

Figure 4.2. TQFP-100 Package Drawing


D
D1
E1
E
100
PIN 1
DESIGNATOR
1
A2
e
A
b
A1

	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	-

Figure 4.3. TQFP-64 Pinout Diagram

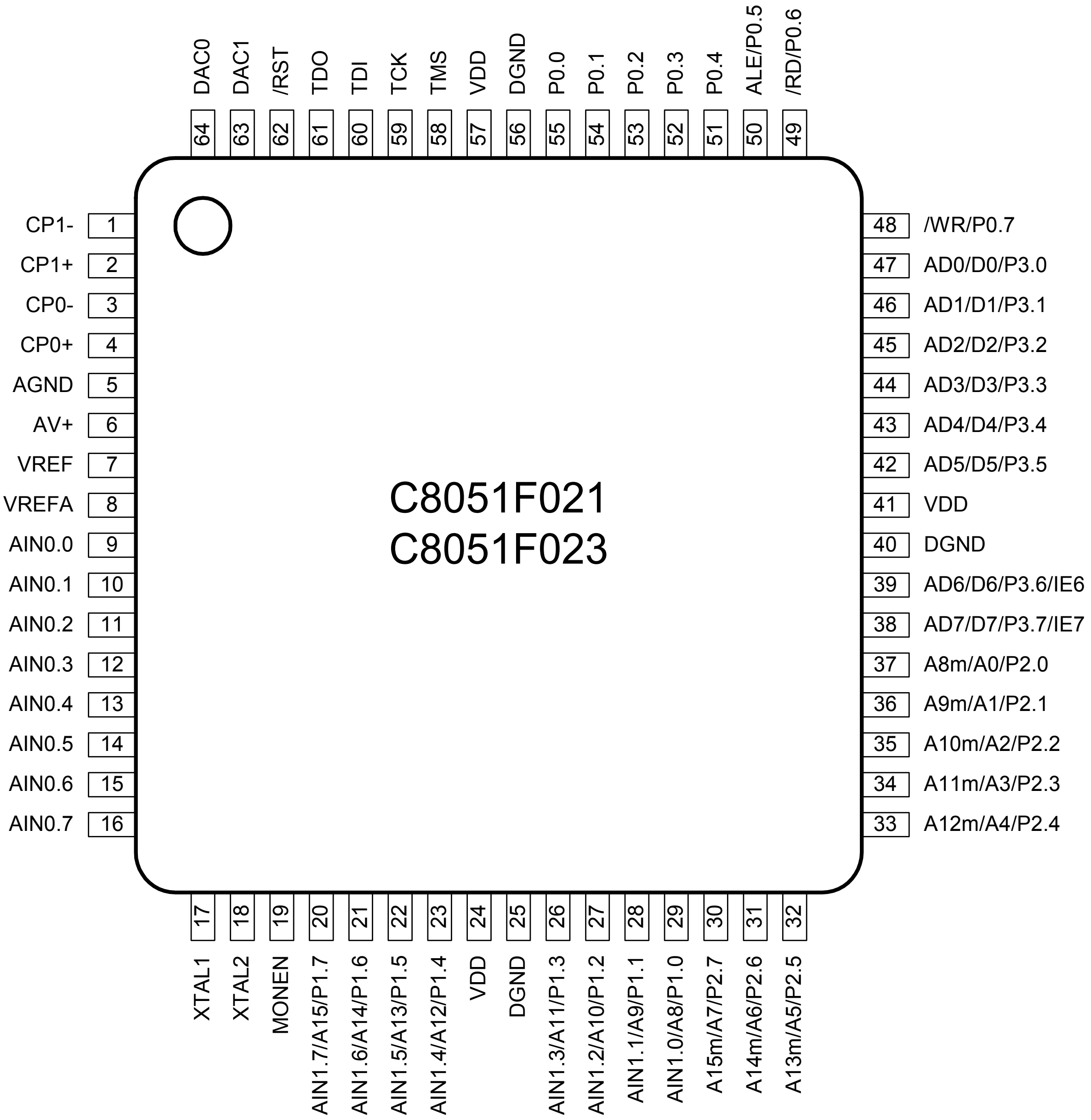
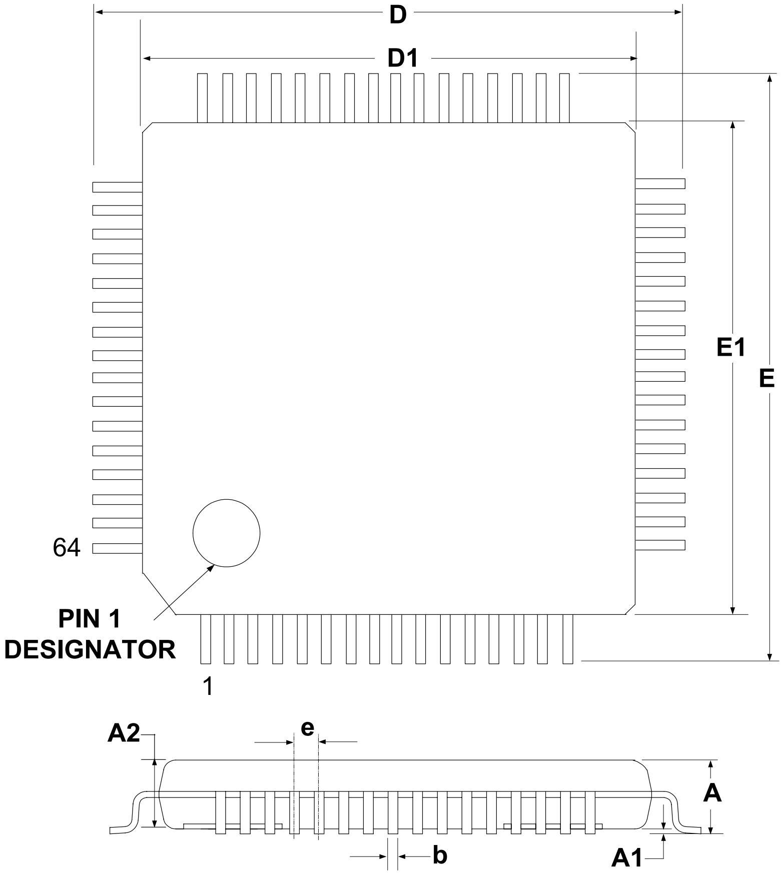

C8051F021
C8051F023
1 CP1-
2 CP1+
3 CP0-
4 CP0+
5 AGND
6 AV+
7 VREF
8 VREFA
9 AIN0.0
10 AIN0.1
11 AIN0.2
12 AIN0.3
13 AIN0.4
14 AIN0.5
15 AIN0.6
16 AIN0.7
17 XTAL1
18 XTAL2
19 MONEN
20 AIN1.7/A15/P1.7
21 AIN1.6/A14/P1.6
22 AIN1.5/A13/P1.5
23 AIN1.4/A12/P1.4
24 VDD
25 DGND
26 AIN1.3/A11/P1.3
27 AIN1.2/A10/P1.2
28 AIN1.1/A9/P1.1
29 AIN1.0/A8/P1.0
30 A15m/A7/P2.7
31 A14m/A6/P2.6
32 A13m/A5/P2.5
33 A12m/A4/P2.4
34 A11m/A3/P2.3
35 A10m/A2/P2.2
36 A9m/A1/P2.1
37 A8m/A0/P2.0
38 AD7/D7/P3.7/IE7
39 AD6/D6/P3.6/IE6
40 DGND
41 VDD
42 AD5/D5/P3.5
43 AD4/D4/P3.4
44 AD3/D3/P3.3
45 AD2/D2/P3.2
46 AD1/D1/P3.1
47 AD0/D0/P3.0
48 /WR/P0.7
49 /RD/P0.6
50 ALE/P0.5
51 P0.4
52 P0.3
53 P0.2
54 P0.1
55 P0.0
56 DGND
57 VDD
58 TMS
59 TCK
60 TDI
61 TDO
62 /RST
63 DAC1
64 DAC0

Figure 4.4. TQFP-64 Package Drawing


D
D1
E1
E
64
PIN 1
DESIGNATOR
1
A2
e
A
b
A1

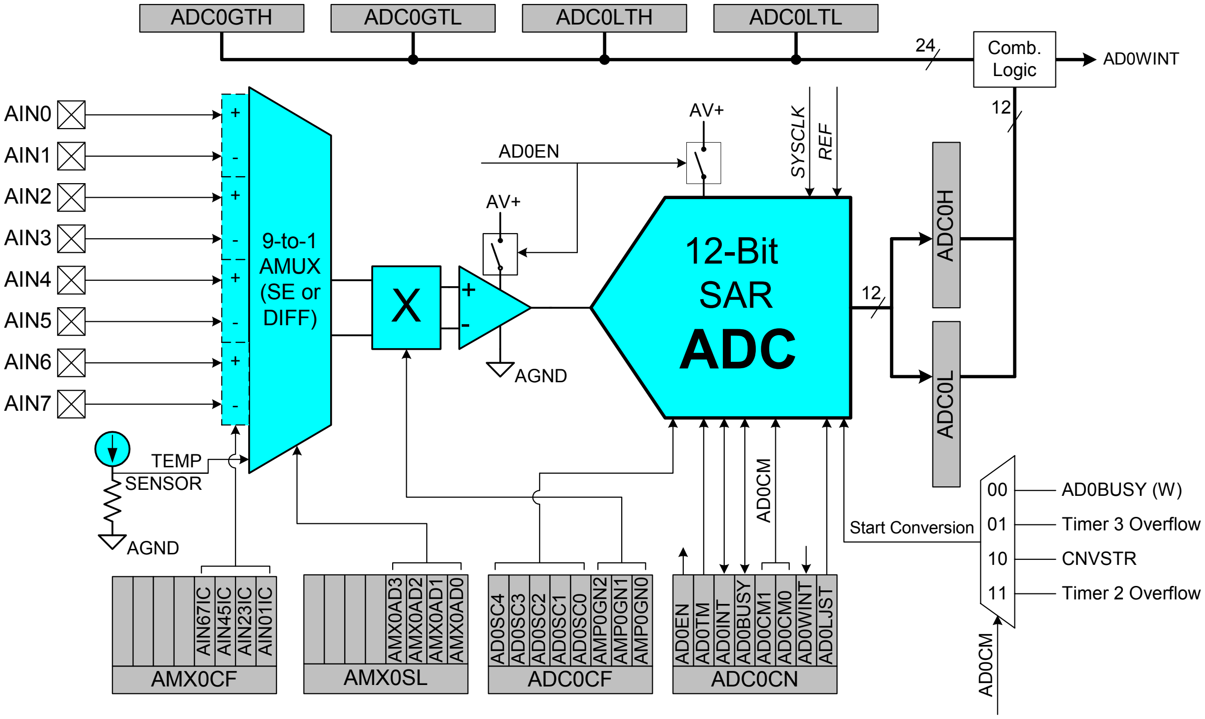
	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	-	1.05
b	0.17	0.22	0.27
D	-	12.00	-
D1	-	10.00	-
e	-	0.50	-
E	-	12.00	-
E1	-	10.00	-

Notes

5. ADC0 (12-BIT ADC, C8051F020/1 ONLY)

The ADC0 subsystem for the C8051F020/1 consists of a 9-channel, configurable analog multiplexer (AMUX0), a programmable gain amplifier (PGA0), and a 100 ksps, 12-bit successive-approximation-register ADC with integrated track-and-hold and Programmable Window Detector (see block diagram in Figure 5.1). The AMUX0, PGA0, Data Conversion Modes, and Window Detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. The voltage reference used by ADC0 is selected as described in **Section “9. VOLTAGE REFERENCE (C8051F020/2)” on page 91** for C8051F020/2 devices, or **Section “10. VOLTAGE REFERENCE (C8051F021/3)” on page 93** for C8051F021/3 devices. The ADC0 subsystem (ADC0, track-and-hold and PGA0) is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

Figure 5.1. 12-Bit ADC0 Functional Block Diagram



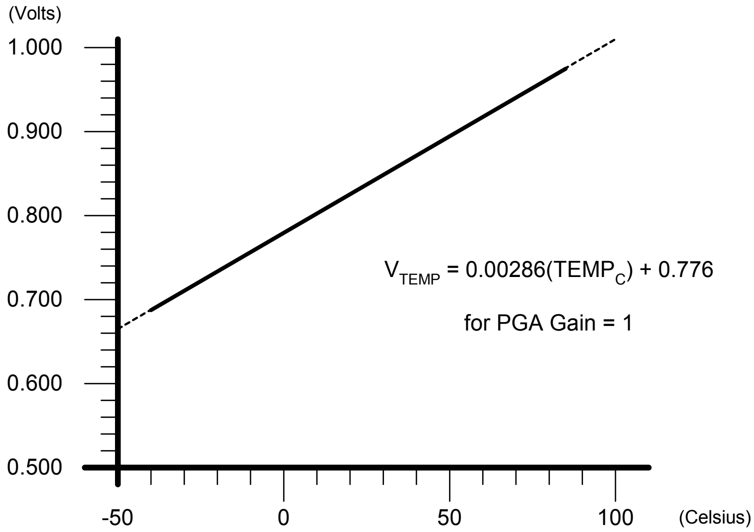
ADC0GTH
ADC0GTL
ADC0LTH
ADC0LTL
24
Comb. Logic
AD0WINT
12
AIN0
AIN1
AIN2
AIN3
AIN4
AIN5
AIN6
AIN7
+
-
9-to-1 AMUX (SE or DIFF)
X
AV+
AD0EN
SYSCLK
REF
12-Bit SAR ADC
ADC0H
ADC0L
12
AGND
TEMP SENSOR
AGND
AD0CM
Start Conversion
00 AD0BUSY (W)
01 Timer 3 Overflow
10 CNVSTR
11 Timer 2 Overflow
AD0CM
AIN67IC
AIN45IC
AIN23IC
AIN01IC
AMX0CF
AMX0AD3
AMX0AD2
AMX0AD1
AMX0AD0
AMX0SL
AD0SC4
AD0SC3
AD0SC2
AD0SC1
AD0SC0
AMP0GN2
AMP0GN1
AMP0GN0
ADC0CF
AD0EN
AD0TM
AD0INT
AD0BUSY
AD0CM1
AD0CM0
AD0WINT
AD0LJST
ADC0CN

5.1. Analog Multiplexer and PGA

Eight of the AMUX channels are available for external measurements while the ninth channel is internally connected to an on-chip temperature sensor (temperature transfer function is shown in Figure 5.2). AMUX input pairs can be programmed to operate in either differential or single-ended mode. This allows the user to select the best measurement technique for each input channel, and even accommodates mode changes "on-the-fly". The AMUX defaults to all single-ended inputs upon reset. There are two registers associated with the AMUX: the Channel Selection register AMX0SL (Figure 5.6), and the Configuration register AMX0CF (Figure 5.7). The table in Figure 5.6 shows AMUX functionality by channel, for each possible configuration. The PGA amplifies the AMUX output signal by an amount determined by the states of the AMP0GN2-0 bits in the ADC0 Configuration register, ADC0CF (Figure 5.7). The PGA can be software-programmed for gains of 0.5, 2, 4, 8 or 16. Gain defaults to unity on reset.

The Temperature Sensor transfer function is shown in Figure 5.2. The output voltage (V_{TEMP}) is the PGA input when the Temperature Sensor is selected by bits AMX0AD3-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings.

Figure 5.2. Temperature Sensor Transfer Function



5.2. ADC Modes of Operation

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

5.2.1. Starting a Conversion

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

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

The AD0BUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the AD0INT interrupt flag (ADC0CN.5). Converted data is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 5.11) depending on the programmed state of the AD0LJST bit in the ADC0CN register.

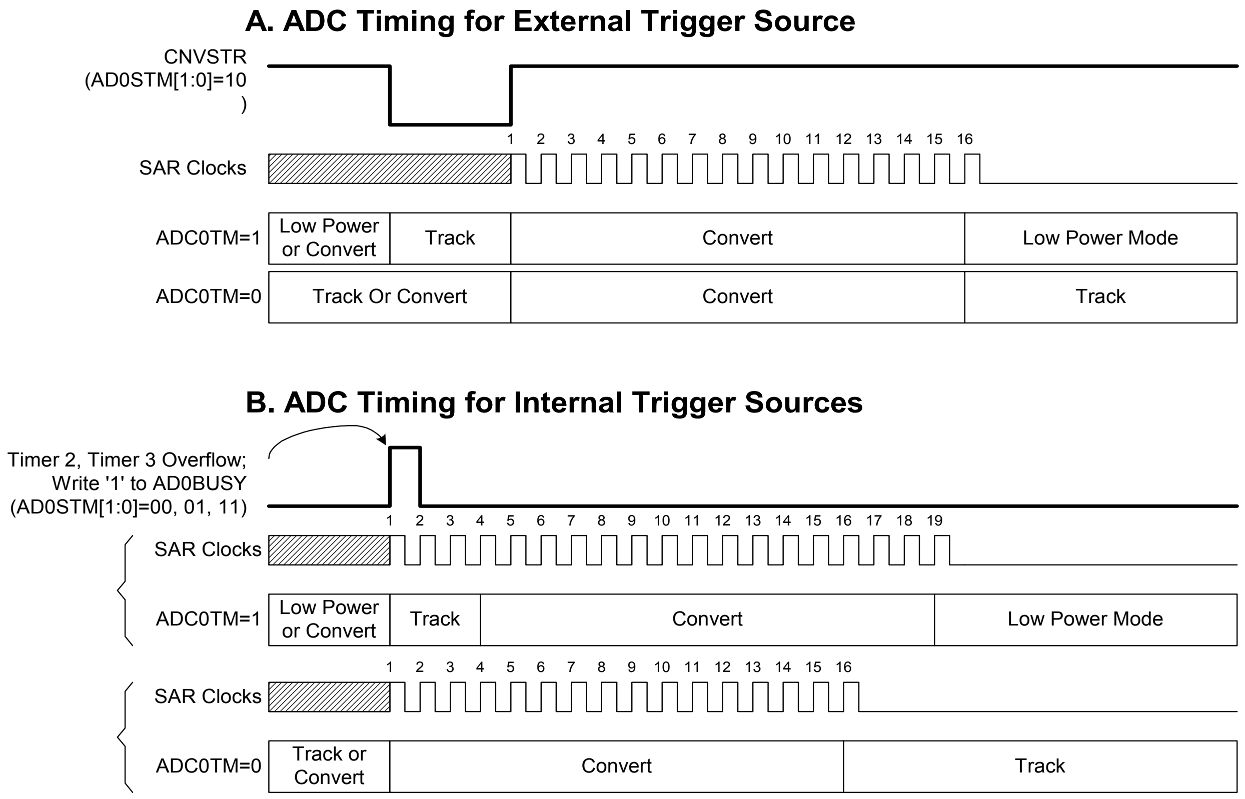
When initiating conversions by writing a ‘1’ to AD0BUSY, the AD0INT bit should be polled to determine when a conversion has completed (ADC0 interrupts may also be used). The recommended polling procedure is shown below.

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

5.2.2. Tracking Modes

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked when a conversion is not in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 5.3). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX or PGA settings are frequently changed, to ensure that settling time requirements are met (see **Section "5.2.3. Settling Time Requirements" on page 46**).

Figure 5.3. 12-Bit ADC Track and Conversion Example Timing


A. ADC Timing for External Trigger Source
CNVSTR
(AD0STM[1:0]=10
)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
SAR Clocks
ADC0TM=1
Low Power or Convert
Track
Convert
Low Power Mode
ADC0TM=0
Track Or Convert
Convert
Track
B. ADC Timing for Internal Trigger Sources
Timer 2, Timer 3 Overflow;
Write '1' to AD0BUSY
(AD0STM[1:0]=00, 01, 11)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
SAR Clocks
ADC0TM=1
Low Power or Convert
Track
Convert
Low Power Mode
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
SAR Clocks
ADC0TM=0
Track or Convert
Convert
Track

5.2.3. Settling Time Requirements

When the ADC0 input configuration is changed (i.e., a different MUX or PGA selection is made), a minimum settling (or tracking) time is required before an accurate conversion can be performed. This settling time is determined by the ADC0 MUX resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 5.4 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required settling time for a given settling accuracy (*SA*) may be approximated by Equation 5.1. When measuring the Temperature Sensor output, R_{TOTAL} reduces to R_{MUX}. Note that in low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the tracking requirements. See Table 5.1 on page 58 for absolute minimum settling/tracking time requirements.

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

Figure 5.4. ADC0 Equivalent Input Circuits

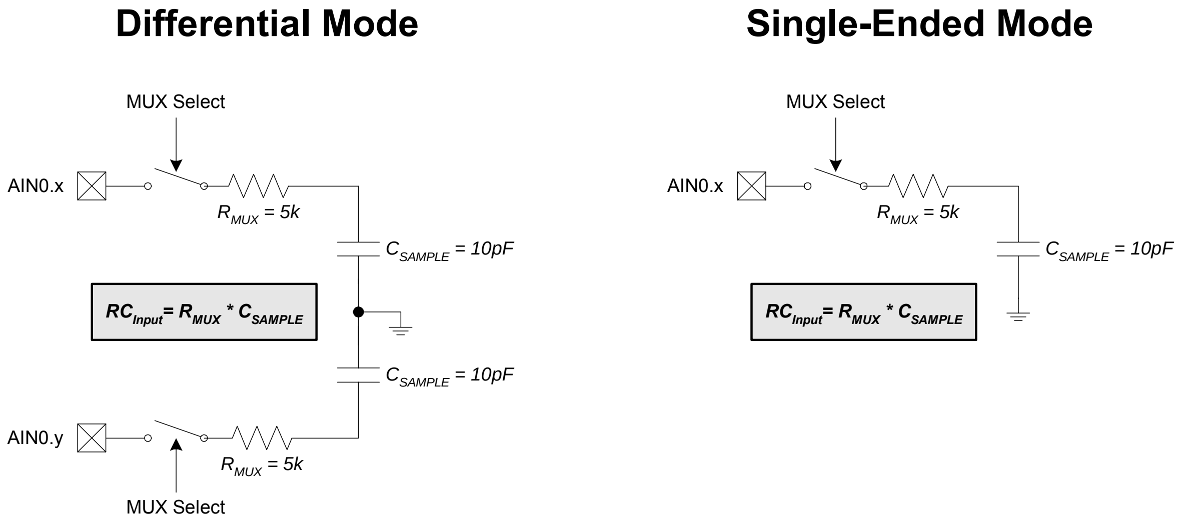

Differential Mode
MUX Select
AIN0.x
$R_{MUX} = 5k$
$C_{SAMPLE} = 10pF$
$RC_{Input} = R_{MUX} * C_{SAMPLE}$
$C_{SAMPLE} = 10pF$
AIN0.y
$R_{MUX} = 5k$
MUX Select
Single-Ended Mode
MUX Select
AIN0.x
$R_{MUX} = 5k$
$C_{SAMPLE} = 10pF$
$RC_{Input} = R_{MUX} * C_{SAMPLE}$

Figure 5.5. AMX0CF: AMUX0 Configuration Register (C8051F020/1)

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	SFR Address: 0xBA

Bits7-4: UNUSED. Read = 0000b; Write = don’t care

Bit3: AIN67IC: AIN6, AIN7 Input Pair Configuration Bit
0: AIN6 and AIN7 are independent single-ended inputs
1: AIN6, AIN7 are (respectively) +, - differential input pair

Bit2: AIN45IC: AIN4, AIN5 Input Pair Configuration Bit
0: AIN4 and AIN5 are independent single-ended inputs
1: AIN4, AIN5 are (respectively) +, - differential input pair

Bit1: AIN23IC: AIN2, AIN3 Input Pair Configuration Bit
0: AIN2 and AIN3 are independent single-ended inputs
1: AIN2, AIN3 are (respectively) +, - differential input pair

Bit0: AIN01IC: AIN0, AIN1 Input Pair Configuration Bit
0: AIN0 and AIN1 are independent single-ended inputs
1: AIN0, AIN1 are (respectively) +, - differential input pair

NOTE: The ADC0 Data Word is in 2’s complement format for channels configured as differential.

Figure 5.6. AMX0SL: AMUX0 Channel Select Register (C8051F020/1)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	AMX0AD3	AMX0AD2	AMX0AD1	AMX0AD0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBB

Bits7-4: UNUSED. Read = 0000b; Write = don't care

Bits3-0: AMX0AD3-0: AMX0 Address Bits
0000-1111b: ADC Inputs selected per chart below

		AMX0AD3-0								
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1xxx**
AMX0CF Bits 3-0	**0000**	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	1000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR

Figure 5.7. ADC0CF: ADC0 Configuration Register (C8051F020/1)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AMP0GN2	AMP0GN1	AMP0GN0	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

Bits7-3: AD0SC4-0: ADC0 SAR Conversion Clock Period Bits
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in AD0SC4-0, and CLK_{SAR0} refers to the desired ADC0 SAR clock. See Table 5.1 on page 58 for SAR clock setting requirements.

$$AD0SC = \frac{SYSCLK}{CLK_{SAR0}} - 1$$

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

Figure 5.8. ADC0CN: ADC0 Control Register (C8051F020/1)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	AD0LJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE8

Bit7:	AD0EN: ADC0 Enable Bit. 0: ADC0 Disabled. ADC0 is in low-power shutdown. 1: ADC0 Enabled. ADC0 is active and ready for data conversions.
Bit6:	AD0TM: ADC Track Mode Bit 0: When the ADC is enabled, tracking is continuous unless a conversion is in process 1: Tracking Defined by ADSTM1-0 bits
Bit5:	AD0INT: ADC0 Conversion Complete Interrupt Flag. This flag must be cleared by software. 0: ADC0 has not completed a data conversion since the last time this flag was cleared. 1: ADC0 has completed a data conversion.
Bit4:	AD0BUSY: ADC0 Busy Bit. Read: 0: ADC0 Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY. 1: ADC0 Conversion is in progress. Write: 0: No Effect. 1: Initiates ADC0 Conversion if AD0STM1-0 = 00b
Bit3-2:	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 CNVSTR. 11: ADC0 conversion initiated on overflow of Timer 2. If AD0TM = 1: 00: Tracking starts with the write of ‘1’ to AD0BUSY and lasts for 3 SAR clocks, followed by conversion. 01: Tracking started by the overflow of Timer 3 and last for 3 SAR clocks, followed by conversion. 10: ADC0 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge. 11: Tracking started by the overflow of Timer 2 and last for 3 SAR clocks, followed by conversion.
Bit1:	AD0WINT: ADC0 Window Compare Interrupt Flag. This bit must be cleared by software. 0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared. 1: ADC0 Window Comparison Data match has occurred.
Bit0:	AD0LJST: ADC0 Left Justify Select. 0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified.

Figure 5.9. ADC0H: ADC0 Data Word MSB Register (C8051F020/1)

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

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

Figure 5.10. ADC0L: ADC0 Data Word LSB Register (C8051F020/1)

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

Bits7-0:	ADC0 Data Word Low-Order Bits. For AD0LJST = 0: Bits 7-0 are the lower 8 bits of the 12-bit ADC0 Data Word. For AD0LJST = 1: Bits 7-4 are the lower 4 bits of the 12-bit ADC0 Data Word. Bits3-0 will always read ‘0’.

Figure 5.11. ADC0 Data Word Example (C8051F020/1)

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

ADC0H[3:0]:ADC0L[7:0], if AD0LJST = 0

(ADC0H[7:4] will be sign-extension of ADC0H.3 for a differential reading, otherwise = 0000b).

ADC0H[7:0]:ADC0L[7:4], if AD0LJST = 1

(ADC0L[3:0] = 0000b).

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

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

Example: ADC0 Data Word Conversion Map, AIN0-AIN1 Differential Input Pair
(AMX0CF = 0x01, AMX0SL = 0x00)

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

For AD0LJST = 0:

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

5.3. ADC0 Programmable Window Detector

The ADC0 Programmable Window Detector continuously compares the ADC0 output to user-programmed limits, and notifies the system when an out-of-bound condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in ADC0CN) can also be used in polled mode. The high and low bytes of the reference words are loaded into the ADC0 Greater-Than and ADC0 Less-Than registers (ADC0GTH, ADC0GTL, ADC0LTH, and ADC0LTL). Reference comparisons are shown starting on page 54. 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 5.12. ADC0GTH: ADC0 Greater-Than Data High Byte Register (C8051F020/1)

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

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

Figure 5.13. ADC0GTL: ADC0 Greater-Than Data Low Byte Register (C8051F020/1)

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

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

Figure 5.14. ADC0LTH: ADC0 Less-Than Data High Byte Register (C8051F020/1)

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

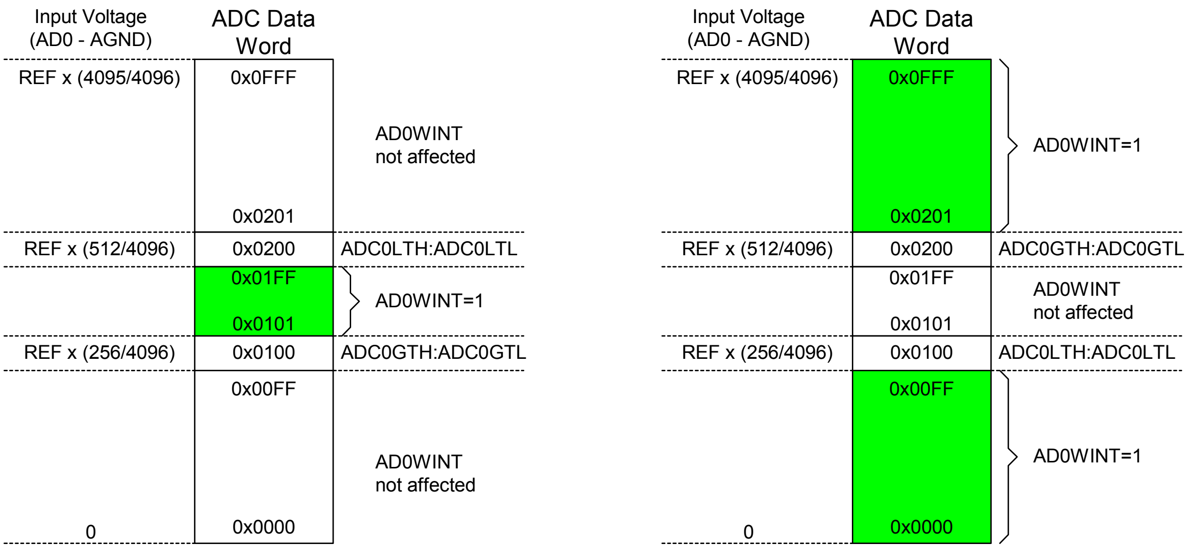
Bits7-0: High byte of ADC0 Less-Than Data Word.

Figure 5.15. ADC0LTL: ADC0 Less-Than Data Low Byte Register (C8051F020/1)

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

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

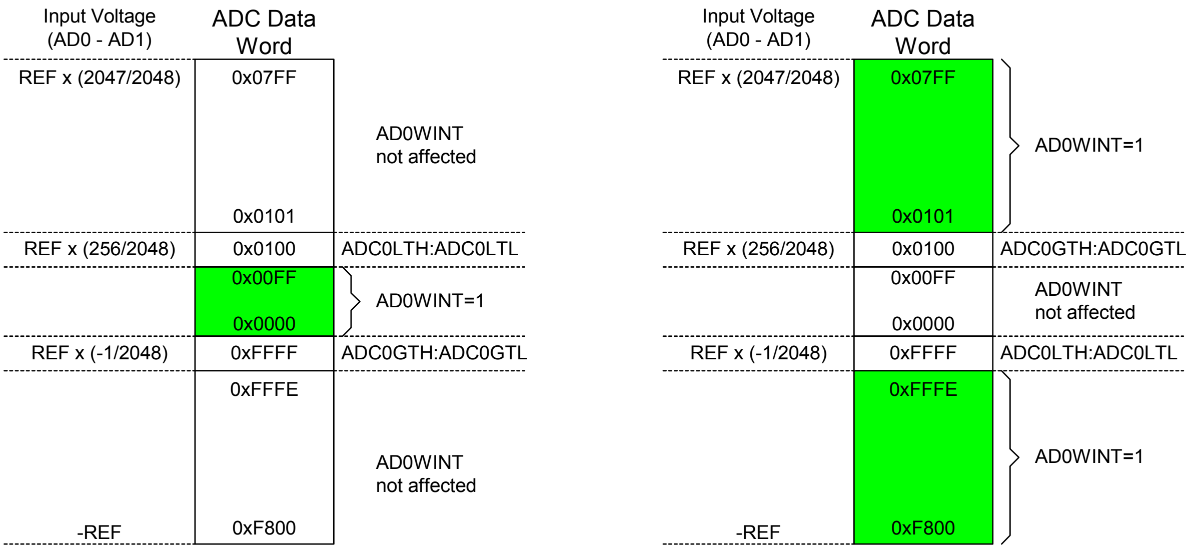
Figure 5.16. 12-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data


Input Voltage
(AD0 - AGND)
ADC Data
Word
REF x (4095/4096)
0x0FFF
AD0WINT
not affected
0x0201
REF x (512/4096)
0x0200
ADC0LTH:ADC0LTL
0x01FF
AD0WINT=1
0x0101
REF x (256/4096)
0x0100
ADC0GTH:ADC0GTL
0x00FF
AD0WINT
not affected
0
0x0000
Input Voltage
(AD0 - AGND)
ADC Data
Word
REF x (4095/4096)
0x0FFF
AD0WINT=1
0x0201
REF x (512/4096)
0x0200
ADC0GTH:ADC0GTL
0x01FF
AD0WINT
not affected
0x0101
REF x (256/4096)
0x0100
ADC0LTH:ADC0LTL
0x00FF
AD0WINT=1
0
0x0000

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

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

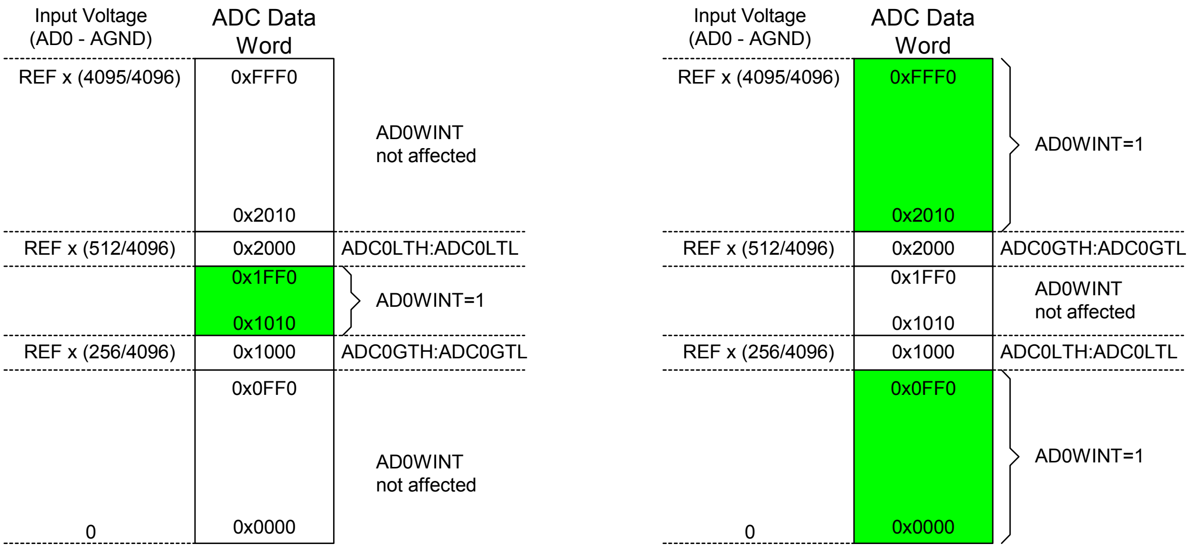
Figure 5.17. 12-Bit ADC0 Window Interrupt Example: Right Justified Differential Data


Input Voltage
(AD0 - AD1)
ADC Data Word
REF x (2047/2048)
0x07FF
AD0WINT not affected
0x0101
REF x (256/2048)
0x0100
ADC0LTH:ADC0LTL
0x00FF
AD0WINT=1
0x0000
REF x (-1/2048)
0xFFFF
ADC0GTH:ADC0GTL
0xFFFE
AD0WINT not affected
-REF
0xF800
Input Voltage
(AD0 - AD1)
ADC Data Word
REF x (2047/2048)
0x07FF
AD0WINT=1
0x0101
REF x (256/2048)
0x0100
ADC0GTH:ADC0GTL
0x00FF
AD0WINT not affected
0x0000
REF x (-1/2048)
0xFFFF
ADC0LTH:ADC0LTL
0xFFFE
AD0WINT=1
-REF
0xF800

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

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

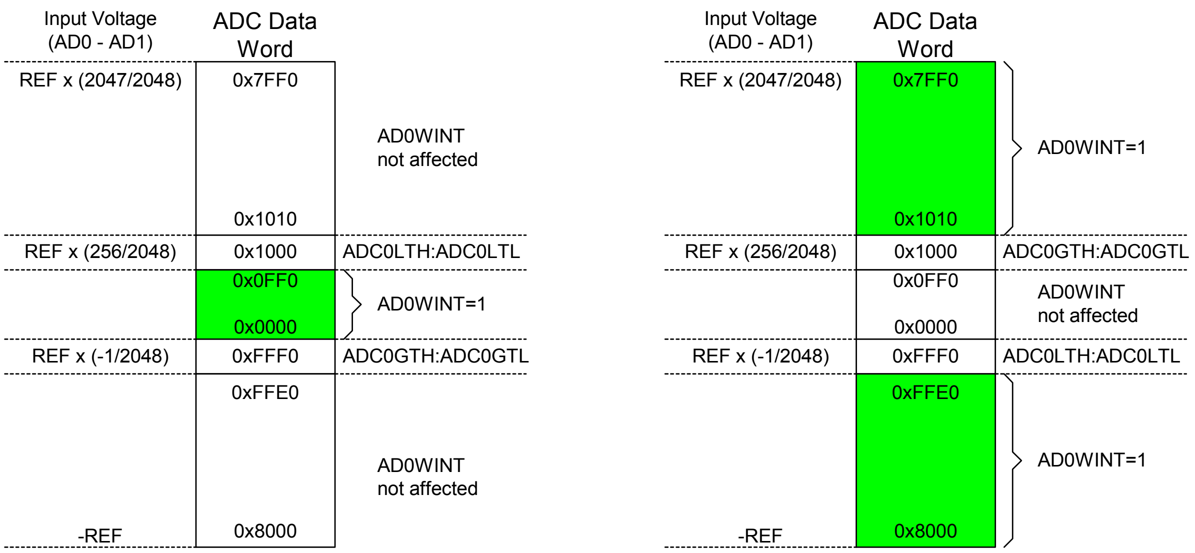
Figure 5.18. 12-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data


Input Voltage (AD0 - AGND)
ADC Data Word
REF x (4095/4096)
0xFFF0
AD0WINT not affected
0x2010
REF x (512/4096)
0x2000
ADC0LTH:ADC0LTL
0x1FF0
AD0WINT=1
0x1010
REF x (256/4096)
0x1000
ADC0GTH:ADC0GTL
0x0FF0
AD0WINT not affected
0
0x0000
Input Voltage (AD0 - AGND)
ADC Data Word
REF x (4095/4096)
0xFFF0
AD0WINT=1
0x2010
REF x (512/4096)
0x2000
ADC0GTH:ADC0GTL
0x1FF0
AD0WINT not affected
0x1010
REF x (256/4096)
0x1000
ADC0LTH:ADC0LTL
0x0FF0
AD0WINT=1
0
0x0000

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

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

Figure 5.19. 12-Bit ADC0 Window Interrupt Example: Left Justified Differential Data


Input Voltage (AD0 - AD1)
ADC Data Word
REF x (2047/2048)
0x7FF0
ADOWINT not affected
0x1010
REF x (256/2048)
0x1000
ADC0LTH:ADC0LTL
0x0FF0
ADOWINT=1
0x0000
REF x (-1/2048)
0xFFF0
ADC0GTH:ADC0GTL
0xFFE0
ADOWINT not affected
-REF
0x8000
Input Voltage (AD0 - AD1)
ADC Data Word
REF x (2047/2048)
0x7FF0
ADOWINT=1
0x1010
REF x (256/2048)
0x1000
ADC0GTH:ADC0GTL
0x0FF0
ADOWINT not affected
0x0000
REF x (-1/2048)
0xFFF0
ADC0LTH:ADC0LTL
0xFFE0
ADOWINT=1
-REF
0x8000

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

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

Table 5.1. 12-Bit ADC0 Electrical Characteristics (C8051F020/1)

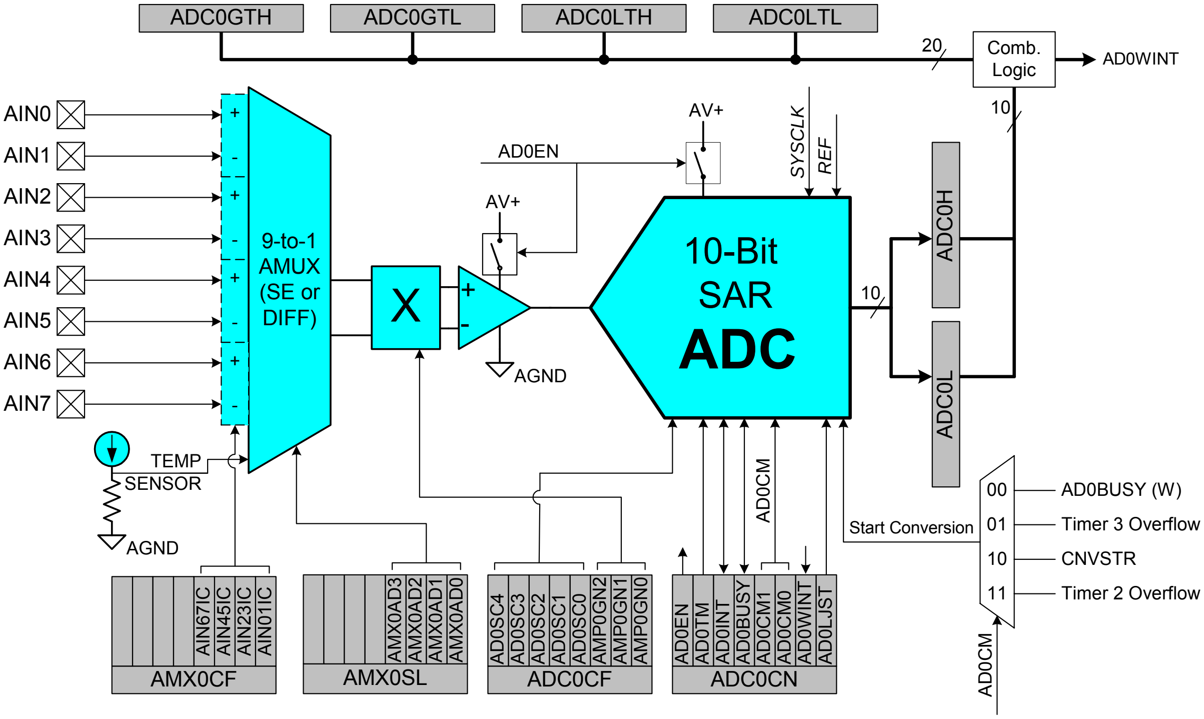
VDD = 3.0V, AV+ = 3.0V, VREF = 2.40V (REFBE=0), PGA Gain = 1, -40°C to +85°C unless otherwise specified

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
DC ACCURACY					
Resolution			12		bits
Integral Nonlinearity				±1	LSB
Differential Nonlinearity	Guaranteed Monotonic			±1	LSB
Offset Error			-3±1		LSB
Full Scale Error	Differential mode		-7±3		LSB
Offset Temperature Coefficient			±0.25		ppm/°C
DYNAMIC PERFORMANCE (10 kHz sine-wave input, 0 to 1 dB below Full Scale, 100 ksps					
Signal-to-Noise Plus Distortion		66			dB
Total Harmonic Distortion	Up to the 5th harmonic		-75		dB
Spurious-Free Dynamic Range			80		dB
CONVERSION RATE					
SAR Clock Frequency				2.5	MHz
Conversion Time in SAR Clocks		16			clocks
Track/Hold Acquisition Time		1.5			µs
Throughput Rate				100	ksps
ANALOG INPUTS					
Input Voltage Range	Single-ended operation	0		VREF	V
***Common-mode Voltage Range**	Differential operation	AGND		AV+	V
Input Capacitance			10		pF
TEMPERATURE SENSOR					
Nonlinearity		-1.0		+1.0	°C
Absolute Accuracy			±3		°C
Gain	PGA Gain = 1		2.86		mV/°C
Offset	PGA Gain = 1, Temp = 0°C		0.776		V
POWER SPECIFICATIONS					
Power Supply Current (AV+ supplied to ADC)	Operating Mode, 100 ksps		450	900	µA
Power Supply Rejection			±0.3		mV/V

6. ADC0 (10-BIT ADC, C8051F022/3 ONLY)

The ADC0 subsystem for the C8051F022/3 consists of a 9-channel, configurable analog multiplexer (AMUX0), a programmable gain amplifier (PGA0), and a 100 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and Programmable Window Detector (see block diagram in Figure 6.1). The AMUX0, PGA0, Data Conversion Modes, and Window Detector are all configurable under software control via the Special Function Registers shown in Figure 6.1. The voltage reference used by ADC0 is selected as described in **Section “9. VOLTAGE REFERENCE (C8051F020/2)” on page 91** for C8051F020/2 devices, or **Section “10. VOLTAGE REFERENCE (C8051F021/3)” on page 93** for C8051F021/3 devices. The ADC0 subsystem (ADC0, track-and-hold and PGA0) is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

Figure 6.1. 10-Bit ADC0 Functional Block Diagram

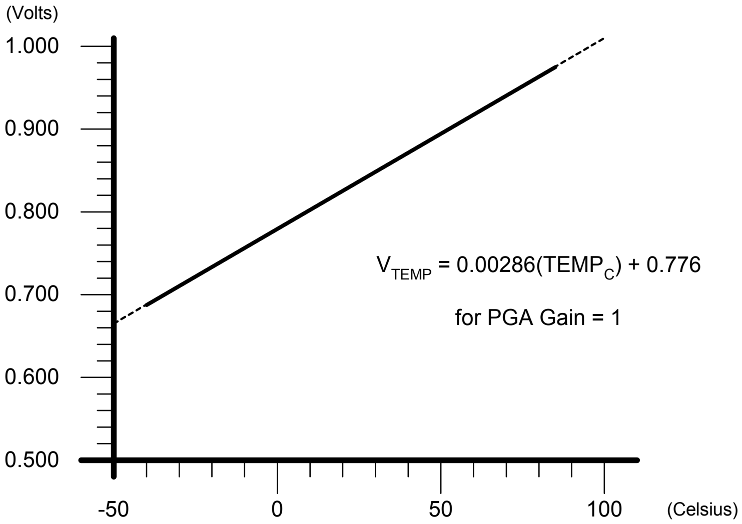

ADC0GTH
ADC0GTL
ADC0LTH
ADC0LTL
20
Comb. Logic
AD0WINT
10
AIN0
AIN1
AIN2
AIN3
AIN4
AIN5
AIN6
AIN7
9-to-1 AMUX (SE or DIFF)
X
AD0EN
AV+
AV+
SYSCLK
REF
10-Bit SAR ADC
AGND
10
ADC0H
ADC0L
TEMP SENSOR
AGND
AD0CM
Start Conversion
00 AD0BUSY (W)
01 Timer 3 Overflow
10 CNVSTR
11 Timer 2 Overflow
AD0CM
AIN67IC
AIN45IC
AIN23IC
AIN01IC
AMX0CF
AMX0AD3
AMX0AD2
AMX0AD1
AMX0AD0
AMX0SL
AD0SC4
AD0SC3
AD0SC2
AD0SC1
AD0SC0
AMP0GN2
AMP0GN1
AMP0GN0
ADC0CF
AD0EN
AD0TM
AD0INT
AD0BUSY
AD0CM1
AD0CM0
AD0WINT
AD0LJST
ADC0CN

6.1. Analog Multiplexer and PGA

Eight of the AMUX channels are available for external measurements while the ninth channel is internally connected to an on-chip temperature sensor (temperature transfer function is shown in Figure 6.2). AMUX input pairs can be programmed to operate in either differential or single-ended mode. This allows the user to select the best measurement technique for each input channel, and even accommodates mode changes "on-the-fly". The AMUX defaults to all single-ended inputs upon reset. There are two registers associated with the AMUX: the Channel Selection register AMX0SL (Figure 6.6), and the Configuration register AMX0CF (Figure 6.7). The table in Figure 6.6 shows AMUX functionality by channel, for each possible configuration. The PGA amplifies the AMUX output signal by an amount determined by the states of the AMP0GN2-0 bits in the ADC0 Configuration register, ADC0CF (Figure 6.7). The PGA can be software-programmed for gains of 0.5, 2, 4, 8 or 16. Gain defaults to unity on reset.

The Temperature Sensor transfer function is shown in Figure 6.2. The output voltage (V_{TEMP}) is the PGA input when the Temperature Sensor is selected by bits AMX0AD3-0 in register AMX0SL; this voltage will be amplified by the PGA according to the user-programmed PGA settings.

Figure 6.2. Temperature Sensor Transfer Function



6.2. ADC Modes of Operation

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

6.2.1. Starting a Conversion

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

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

The AD0BUSY bit is set to logic 1 during conversion and restored to logic 0 when conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the AD0INT interrupt flag (ADC0CN.5). Converted data is available in the ADC0 data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 6.11) depending on the programmed state of the AD0LJST bit in the ADC0CN register.

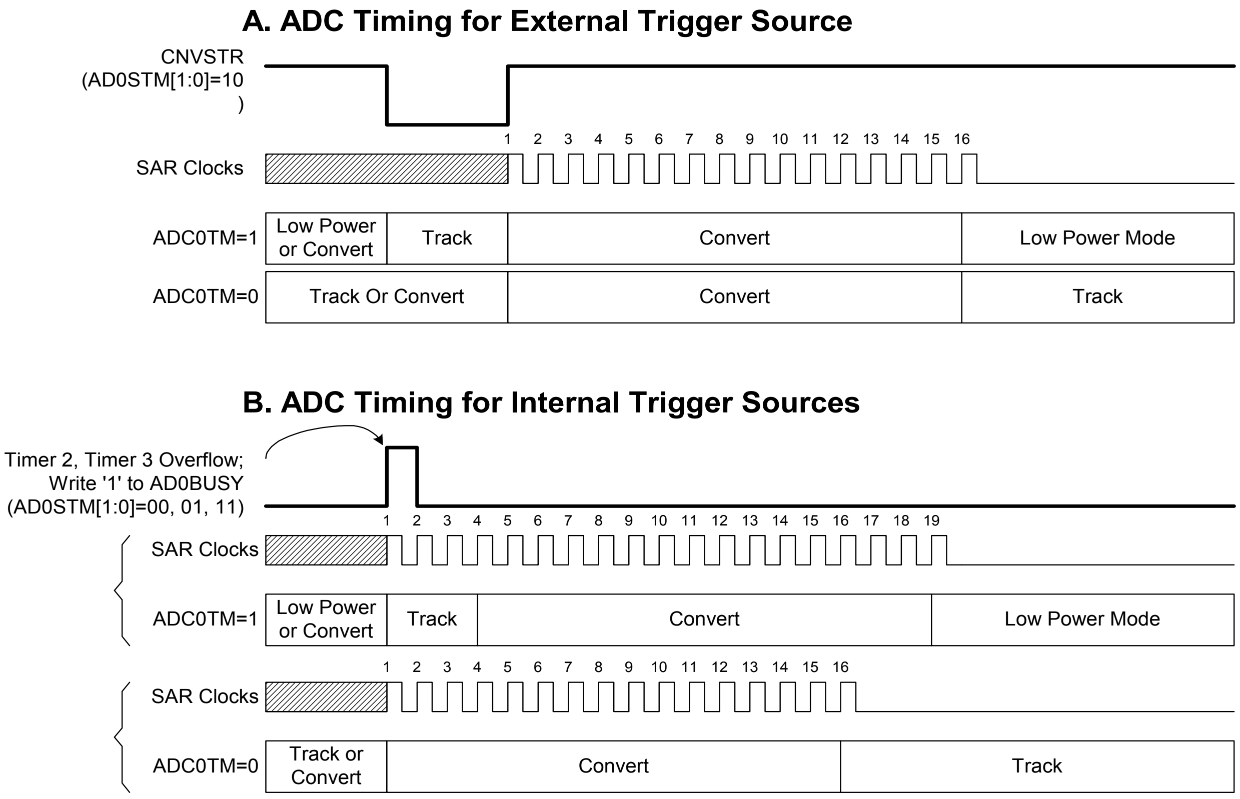
When initiating conversions by writing a ‘1’ to AD0BUSY, the AD0INT bit should be polled to determine when a conversion has completed (ADC0 interrupts may also be used). The recommended polling procedure is shown below.

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

6.2.2. Tracking Modes

The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked when a conversion is not in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 6.3). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX or PGA settings are frequently changed, to ensure that settling time requirements are met (see **Section "6.2.3. Settling Time Requirements" on page 62**).

Figure 6.3. 10-Bit ADC Track and Conversion Example Timing


A. ADC Timing for External Trigger Source
CNVSTR
(AD0STM[1:0]=10
)
SAR Clocks
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
ADC0TM=1
Low Power or Convert
Track
Convert
Low Power Mode
ADC0TM=0
Track Or Convert
Convert
Track
B. ADC Timing for Internal Trigger Sources
Timer 2, Timer 3 Overflow;
Write '1' to AD0BUSY
(AD0STM[1:0]=00, 01, 11)
SAR Clocks
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
ADC0TM=1
Low Power or Convert
Track
Convert
Low Power Mode
SAR Clocks
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
ADC0TM=0
Track or Convert
Convert
Track

6.2.3. Settling Time Requirements

When the ADC0 input configuration is changed (i.e., a different MUX or PGA selection is made), a minimum settling (or tracking) time is required before an accurate conversion can be performed. This settling time is determined by the ADC0 MUX resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 6.4 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required settling time for a given settling accuracy (SA) may be approximated by Equation 6.1. When measuring the Temperature Sensor output, R_{TOTAL} reduces to R_{MUX}. 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 settling time requirements. See Table 6.1 on page 74 for minimum settling/tracking time requirements.

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

Figure 6.4. ADC0 Equivalent Input Circuits

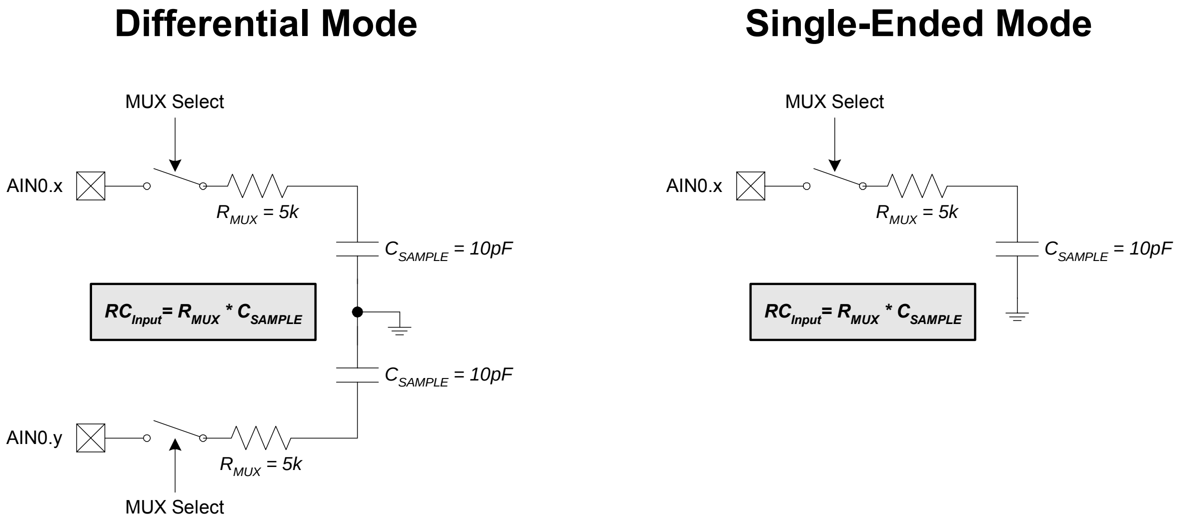

Differential Mode
MUX Select
AIN0.x
R_{MUX} = 5k
C_{SAMPLE} = 10pF
$RC_{Input} = R_{MUX} * C_{SAMPLE}$
C_{SAMPLE} = 10pF
AIN0.y
R_{MUX} = 5k
MUX Select
Single-Ended Mode
MUX Select
AIN0.x
R_{MUX} = 5k
C_{SAMPLE} = 10pF
$RC_{Input} = R_{MUX} * C_{SAMPLE}$

Figure 6.5. AMX0CF: AMUX0 Configuration Register (C8051F022/3)

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	SFR Address: 0xBA

Bits7-4: UNUSED. Read = 0000b; Write = don’t care

Bit3: AIN67IC: AIN6, AIN7 Input Pair Configuration Bit
0: AIN6 and AIN7 are independent single-ended inputs
1: AIN6, AIN7 are (respectively) +, - differential input pair

Bit2: AIN45IC: AIN4, AIN5 Input Pair Configuration Bit
0: AIN4 and AIN5 are independent single-ended inputs
1: AIN4, AIN5 are (respectively) +, - differential input pair

Bit1: AIN23IC: AIN2, AIN3 Input Pair Configuration Bit
0: AIN2 and AIN3 are independent single-ended inputs
1: AIN2, AIN3 are (respectively) +, - differential input pair

Bit0: AIN01IC: AIN0, AIN1 Input Pair Configuration Bit
0: AIN0 and AIN1 are independent single-ended inputs
1: AIN0, AIN1 are (respectively) +, - differential input pair

NOTE: The ADC0 Data Word is in 2’s complement format for channels configured as differential.

Figure 6.6. AMX0SL: AMUX0 Channel Select Register (C8051F022/3)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	AMX0AD3	AMX0AD2	AMX0AD1	AMX0AD0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBB

Bits7-4: UNUSED. Read = 0000b; Write = don’t care

Bits3-0: AMX0AD3-0: AMX0 Address Bits
0000-1111b: ADC Inputs selected per chart below

		AMX0AD3-0								
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1xxx**
AMX0CF Bits 3-0	**0000**	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
	0100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	0111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		AIN6	AIN7	TEMP SENSOR
	1000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1001	+(AIN0) -(AIN1)		AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1010	AIN0	AIN1	+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1011	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		AIN4	AIN5	+(AIN6) -(AIN7)		TEMP SENSOR
	1100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1101	+(AIN0) -(AIN1)		AIN2	AIN3	+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1110	AIN0	AIN1	+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR
	1111	+(AIN0) -(AIN1)		+(AIN2) -(AIN3)		+(AIN4) -(AIN5)		+(AIN6) -(AIN7)		TEMP SENSOR

Figure 6.7. ADC0CF: ADC0 Configuration Register (C8051F022/3)

Bits7-3: AD0SC4-0: ADC0 SAR Conversion Clock Period Bits

SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in AD0SC4-0, and CLK_{SAR0} refers to the desired ADC0 SAR clock. See Table 6.1 on page 74 for SAR clock setting requirements.

Figure 6.8. ADC0CN: ADC0 Control Register (C8051F022/3)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	AD0LJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE8

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

Bit6: AD0TM: ADC Track Mode Bit
0: When the ADC is enabled, tracking is continuous unless a conversion is in process
1: Tracking Defined by ADSTM1-0 bits

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

Bit4: AD0BUSY: ADC0 Busy Bit.
Read:
0: ADC0 Conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.
1: ADC0 Conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0STM1-0 = 00b

Bit3-2: 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 CNVSTR.
11: ADC0 conversion initiated on overflow of Timer 2.
If AD0TM = 1:
00: Tracking starts with the write of ‘1’ to AD0BUSY and lasts for 3 SAR clocks, followed by conversion.
01: Tracking started by the overflow of Timer 3 and last for 3 SAR clocks, followed by conversion.
10: ADC0 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge.
11: Tracking started by the overflow of Timer 2 and last for 3 SAR clocks, followed by conversion.

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

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

Figure 6.9. ADC0H: ADC0 Data Word MSB Register (C8051F022/3)

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

Bits7-0: ADC Data Word High-Order Bits.
For ADLJST = 0: Bits 7-2 are the sign extension of Bit1. Bits 1-0 are the upper 2 bits of the 10-bit ADC Data Word.
For ADLJST = 1: Bits 7-0 are the most-significant bits of the 10-bit ADC Data Word.

Figure 6.10. ADC0L: ADC0 Data Word LSB Register (C8051F022/3)

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

Bits7-0: ADC Data Word Low-Order Bits.
For ADLJST = 0: Bits 7-0 are the lower 8 bits of the 10-bit ADC Data Word.
For ADLJST = 1: Bits 7-6 are the lower 2 bits of the 10-bit ADC Data Word. Bits 5-0 will always read ‘0’.

Figure 6.11. ADC0 Data Word Example (C8051F022/3)

10-bit ADC Data Word appears in the ADC Data Word Registers as follows:

ADC0H[1:0]:ADC0L[7:0], if ADLJST = 0

(ADC0H[7:2] will be sign-extension of ADC0H.1 for a differential reading, otherwise = 000000b).

ADC0H[7:0]:ADC0L[7:6], if ADLJST = 1

(ADC0L[5:0] = 000000b).

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

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

Example: ADC Data Word Conversion Map, AIN0-AIN1 Differential Input Pair
(AMX0CF = 0x01, AMX0SL = 0x00)

AIN0-AGND (Volts)	**ADC0H:ADC0L** (ADLJST = 0)	**ADC0H:ADC0L** (ADLJST = 1)
VREF * (511/512)	0x01FF	0x7FC0
VREF / 2	0x0100	0x4000
VREF * (1/512)	0x0001	0x0040
0	0x0000	0x0000
-VREF * (1/512)	0xFFFF (-1)	0xFFC0
-VREF / 2	0xFF00 (-256)	0xC000
-VREF	0xFE00 (-512)	0x8000

ADLJST = 0:

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

6.3. ADC0 Programmable Window Detector

The ADC0 Programmable Window Detector continuously compares the ADC0 output to user-programmed limits, and notifies the system when an out-of-bound condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in ADC0CN) can also be used in polled mode. The high and low bytes of the reference words are loaded into the ADC0 Greater-Than and ADC0 Less-Than registers (ADC0GTH, ADC0GTL, ADC0LTH, and ADC0LTL). Reference comparisons are shown starting on page 70. 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 6.12. ADC0GTH: ADC0 Greater-Than Data High Byte Register (C8051F022/3)

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

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

Figure 6.13. ADC0GTL: ADC0 Greater-Than Data Low Byte Register (C8051F022/3)

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

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

Figure 6.14. ADC0LTH: ADC0 Less-Than Data High Byte Register (C8051F022/3)

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

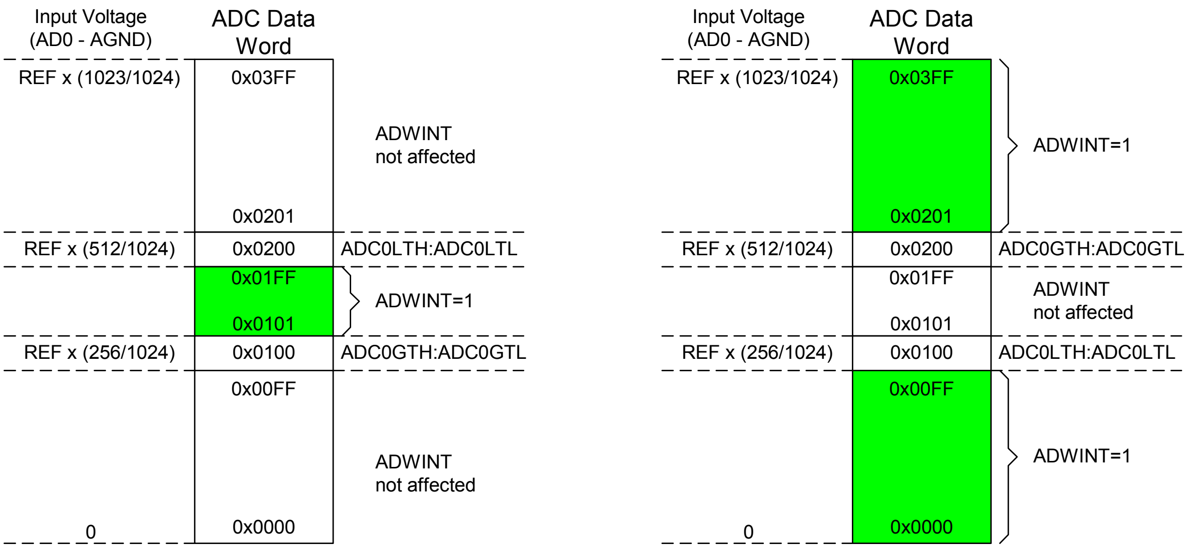
Bits7-0: High byte of ADC0 Less-Than Data Word.

Figure 6.15. ADC0LTL: ADC0 Less-Than Data Low Byte Register (C8051F022/3)

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

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

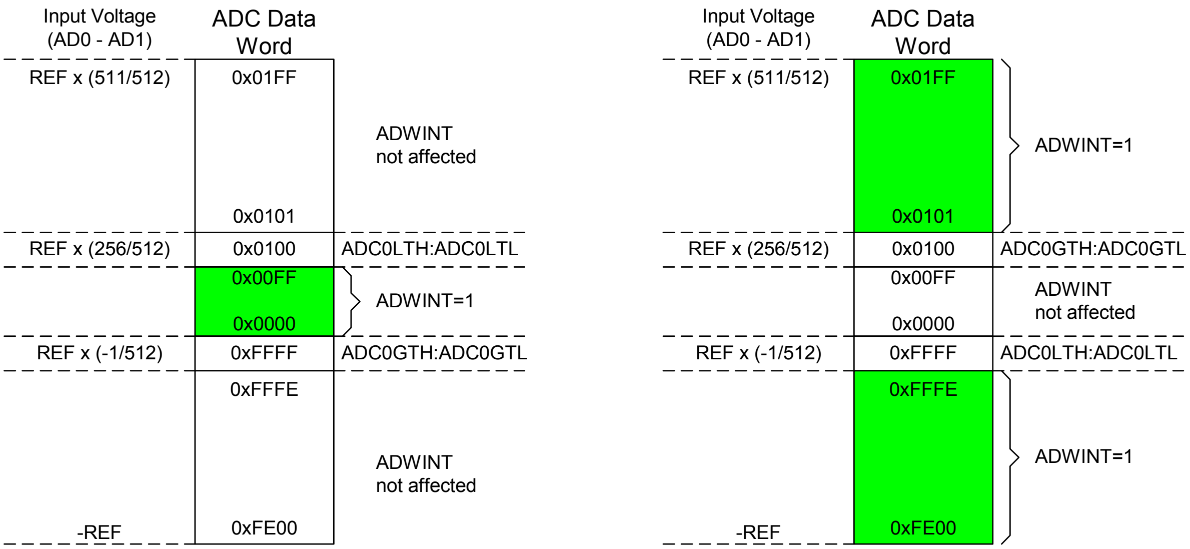
Figure 6.16. 10-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data


Input Voltage
(AD0 - AGND)
ADC Data Word
REF x (1023/1024)
0x03FF
ADWINT not affected
0x0201
REF x (512/1024)
0x0200
ADC0LTH:ADC0LTL
0x01FF
ADWINT=1
0x0101
REF x (256/1024)
0x0100
ADC0GTH:ADC0GTL
0x00FF
ADWINT not affected
0
0x0000
Input Voltage
(AD0 - AGND)
ADC Data Word
REF x (1023/1024)
0x03FF
ADWINT=1
0x0201
REF x (512/1024)
0x0200
ADC0GTH:ADC0GTL
0x01FF
ADWINT not affected
0x0101
REF x (256/1024)
0x0100
ADC0LTH:ADC0LTL
0x00FF
ADWINT=1
0
0x0000

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

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

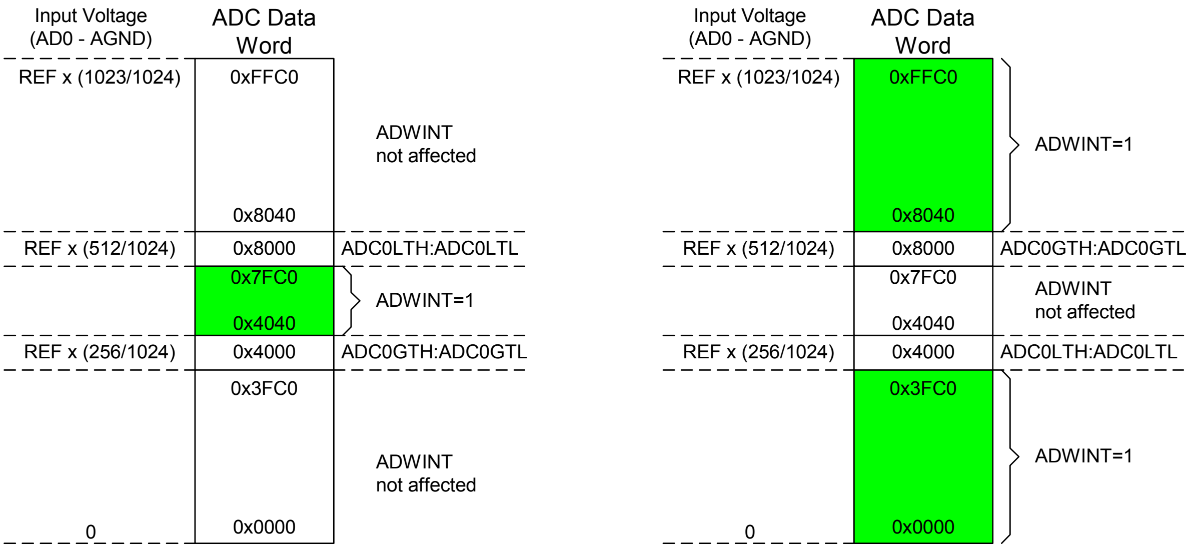
Figure 6.17. 10-Bit ADC0 Window Interrupt Example: Right Justified Differential Data


Input Voltage (AD0 - AD1)
ADC Data Word
REF x (511/512)
0x01FF
ADWINT not affected
0x0101
REF x (256/512)
0x0100
ADC0LTH:ADC0LTL
0x00FF
ADWINT=1
0x0000
REF x (-1/512)
0xFFFF
ADC0GTH:ADC0GTL
0xFFFE
ADWINT not affected
-REF
0xFE00
Input Voltage (AD0 - AD1)
ADC Data Word
REF x (511/512)
0x01FF
ADWINT=1
0x0101
REF x (256/512)
0x0100
ADC0GTH:ADC0GTL
0x00FF
ADWINT not affected
0x0000
REF x (-1/512)
0xFFFF
ADC0LTH:ADC0LTL
0xFFFE
ADWINT=1
-REF
0xFE00

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

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

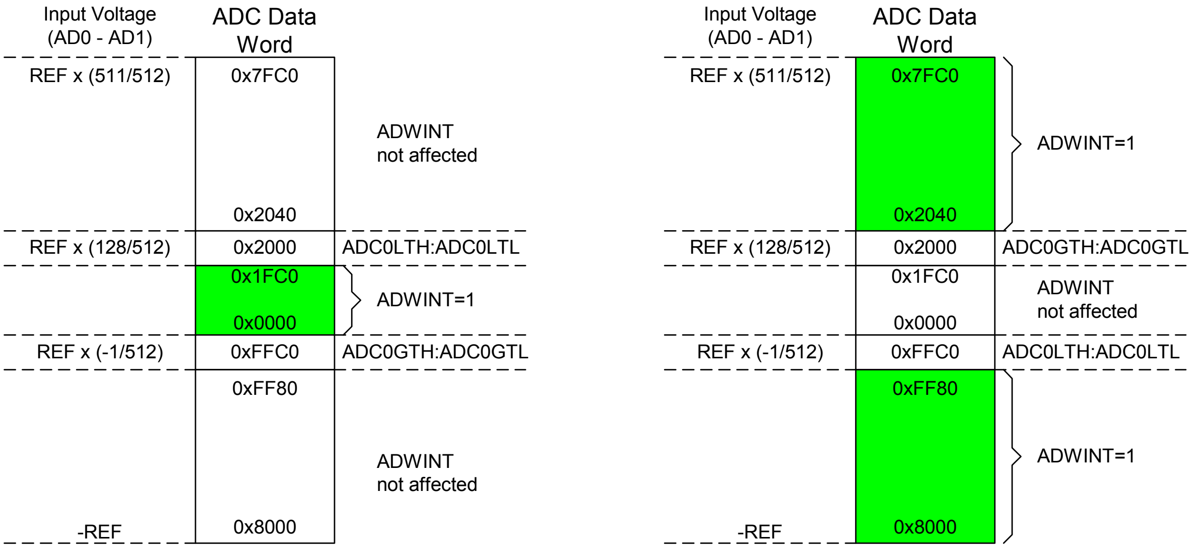
Figure 6.18. 10-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data


Input Voltage (AD0 - AGND)
ADC Data Word
REF x (1023/1024)
0xFFC0
ADWINT not affected
0x8040
REF x (512/1024)
0x8000
ADC0LTH:ADC0LTL
0x7FC0
ADWINT=1
0x4040
REF x (256/1024)
0x4000
ADC0GTH:ADC0GTL
0x3FC0
ADWINT not affected
0
0x0000
Input Voltage (AD0 - AGND)
ADC Data Word
REF x (1023/1024)
0xFFC0
ADWINT=1
0x8040
REF x (512/1024)
0x8000
ADC0GTH:ADC0GTL
0x7FC0
ADWINT not affected
0x4040
REF x (256/1024)
0x4000
ADC0LTH:ADC0LTL
0x3FC0
ADWINT=1
0
0x0000

Given:
AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 1,
ADC0LTH:ADC0LTL = 0x8000,
ADC0GTH:ADC0GTL = 0x4000.
An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x8000 and > 0x4000.

Given:
AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 1,
ADC0LTH:ADC0LTL = 0x4000,
ADC0GTH:ADC0GTL = 0x8000.
An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x4000 or > 0x8000.

Figure 6.19. 10-Bit ADC0 Window Interrupt Example: Left Justified Differential Data


Input Voltage (AD0 - AD1)
ADC Data Word
REF x (511/512)
0x7FC0
ADWINT not affected
0x2040
REF x (128/512)
0x2000
ADC0LTH:ADC0LTL
0x1FC0
ADWINT=1
0x0000
REF x (-1/512)
0xFFC0
ADC0GTH:ADC0GTL
0xFF80
ADWINT not affected
-REF
0x8000
Input Voltage (AD0 - AD1)
ADC Data Word
REF x (511/512)
0x7FC0
ADWINT=1
0x2040
REF x (128/512)
0x2000
ADC0GTH:ADC0GTL
0x1FC0
ADWINT not affected
0x0000
REF x (-1/512)
0xFFC0
ADC0LTH:ADC0LTL
0xFF80
ADWINT=1
-REF
0x8000

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

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

Table 6.1. 10-Bit ADC0 Electrical Characteristics (C8051F022/3)

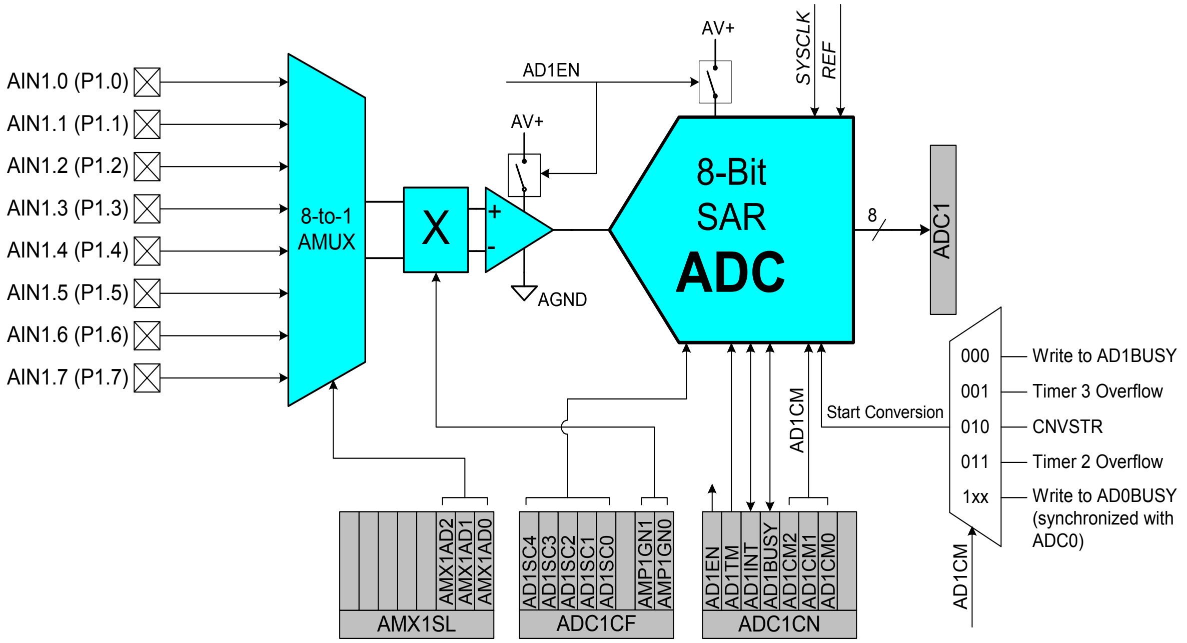
VDD = 3.0V, AV+ = 3.0V, VREF = 2.40V (REFBE=0), PGA Gain = 1, -40°C to +85°C unless otherwise specified

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
DC ACCURACY					
Resolution			10		bits
Integral Nonlinearity				±1	LSB
Differential Nonlinearity	Guaranteed Monotonic			±1	LSB
Offset Error			±0.5		LSB
Full Scale Error	Differential mode		-1.5±0.5		LSB
Offset Temperature Coefficient			±0.25		ppm/°C
DYNAMIC PERFORMANCE (10 kHz sine-wave input, 0 to 1 dB below Full Scale, 100 ksps					
Signal-to-Noise Plus Distortion		59			dB
Total Harmonic Distortion	Up to the 5th harmonic		-70		dB
Spurious-Free Dynamic Range			80		dB
CONVERSION RATE					
SAR Clock Frequency				2.5	MHz
Conversion Time in SAR Clocks		16			clocks
Track/Hold Acquisition Time		1.5			µs
Throughput Rate				100	ksps
ANALOG INPUTS					
Input Voltage Range	Single-ended operation	0		VREF	V
***Common-mode Voltage Range**	Differential operation	AGND		AV+	V
Input Capacitance			10		pF
TEMPERATURE SENSOR					
Nonlinearity		-1.0		+1.0	°C
Absolute Accuracy			±3		°C
Gain	PGA Gain = 1		2.86		mV/°C
Offset	PGA Gain = 1, Temp = 0°C		0.776		V
POWER SPECIFICATIONS					
Power Supply Current (AV+ supplied to ADC)	Operating Mode, 100 ksps		450	900	µA
Power Supply Rejection			±0.3		mV/V

7. ADC1 (8-BIT ADC)

The ADC1 subsystem for the C8051F020/1/2/3 consists of an 8-channel, configurable analog multiplexer (AMUX1), a programmable gain amplifier (PGA1), and a 500 ksps, 8-bit successive-approximation-register ADC with integrated track-and-hold (see block diagram in Figure 7.1). The AMUX1, PGA1, and Data Conversion Modes, are all configurable under software control via the Special Function Registers shown in Figure 7.1. The ADC1 subsystem (8-bit ADC, track-and-hold and PGA) is enabled only when the AD1EN bit in the ADC1 Control register (ADC1CN) is set to logic 1. The ADC1 subsystem is in low power shutdown when this bit is logic 0. The voltage reference used by ADC1 is selected as described in **Section "9. VOLTAGE REFERENCE (C8051F020/2)" on page 91** for C8051F020/2 devices, or **Section "10. VOLTAGE REFERENCE (C8051F021/3)" on page 93** for C8051F021/3 devices.

Figure 7.1. ADC1 Functional Block Diagram


AIN1.0 (P1.0)
AIN1.1 (P1.1)
AIN1.2 (P1.2)
AIN1.3 (P1.3)
AIN1.4 (P1.4)
AIN1.5 (P1.5)
AIN1.6 (P1.6)
AIN1.7 (P1.7)
8-to-1 AMUX
X
AD1EN
AV+
AV+
+
-
AGND
SYSCLK
REF
8-Bit SAR ADC
8
ADC1
000
001
010
011
1xx
Write to AD1BUSY
Timer 3 Overflow
CNVSTR
Timer 2 Overflow
Write to AD0BUSY (synchronized with ADC0)
Start Conversion
AD1CM
AD1CM
AMX1AD2
AMX1AD1
AMX1AD0
AMX1SL
AD1SC4
AD1SC3
AD1SC2
AD1SC1
AD1SC0
AMP1GN1
AMP1GN0
ADC1CF
AD1EN
AD1TM
AD1INT
AD1BUSY
AD1CM2
AD1CM1
AD1CM0
ADC1CN

7.1. Analog Multiplexer and PGA

Eight ADC1 channels are available for measurement, as selected by the AMX1SL register (see Figure 7.5). The PGA amplifies the ADC1 output signal by an amount determined by the states of the AMP1GN2-0 bits in the ADC1 Configuration register, ADC1CF (Figure 7.4). The PGA can be software-programmed for gains of 0.5, 1, 2, or 4. Gain defaults to 0.5 on reset.

Important Note: AIN1 pins also function as Port 1 I/O pins, and must be configured as analog inputs when used as ADC1 inputs. To configure an AIN1 pin for analog input, set to '0' the corresponding bit in register P1MDIN. Port 1 pins selected as analog inputs are skipped by the Digital I/O Crossbar. See **Section "17.1.6. Configuring Port 1 Pins as Analog Inputs (AIN1.[7:0])" on page 165** for more information on configuring the AIN1 pins.

7.2. ADC1 Modes of Operation

ADC1 has a maximum conversion speed of 500 ksps. The ADC1 conversion clock (SAR1 clock) is a divided version of the system clock, determined by the AD1SC bits in the ADC1CF register (system clock divided by (AD1SC + 1) for $0 \leq \text{AD1SC} \leq 31$). The maximum ADC1 conversion clock is 6 MHz.

7.2.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC1 Start of Conversion Mode bits (AD1CM2-0) in register ADC1CN. 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, CNVSTR;
4. A Timer 2 overflow (i.e. timed continuous conversions);
5. Writing a ‘1’ to the AD0BUSY of register ADC0CN (initiate conversion of ADC1 and ADC0 with a single software command).

During conversion, the AD1BUSY bit is set to logic 1 and restored to 0 when conversion is complete. The falling edge of AD1BUSY triggers an interrupt (when enabled) and sets the interrupt flag in ADC1CN. Converted data is available in the ADC1 data word, ADC1.

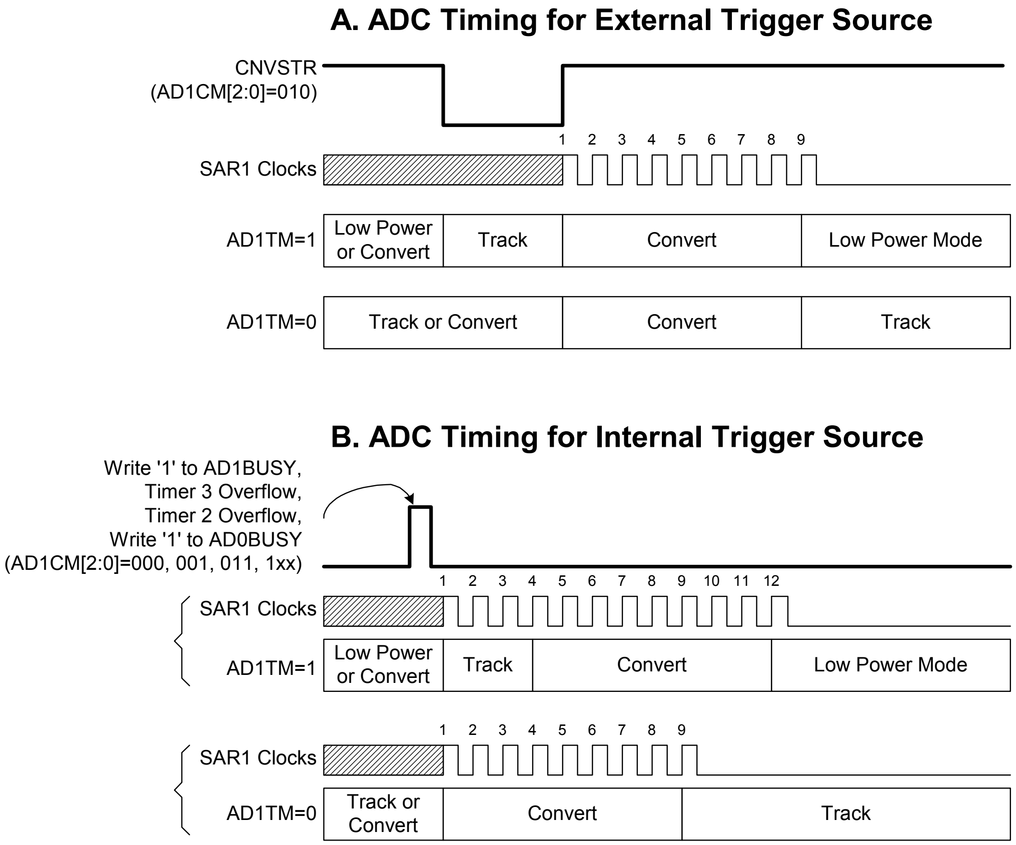
When a conversion is initiated by writing a ‘1’ to AD1BUSY, it is recommended to poll AD1INT to determine when the conversion is complete. The recommended procedure is:

Step 1. Write a ‘0’ to AD1INT;
Step 2. Write a ‘1’ to AD1BUSY;
Step 3. Poll AD1INT for ‘1’;
Step 4. Process ADC1 data.

7.2.2. Tracking Modes

The AD1TM bit in register ADC1CN controls the ADC1 track-and-hold mode. In its default state, the ADC1 input is continuously tracked, except when a conversion is in progress. When the AD1TM bit is logic 1, ADC1 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC1 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 7.2). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power Track-and-Hold mode is also useful when AMUX or PGA settings are frequently changed, due to the settling time requirements described in **Section “7.2.3. Settling Time Requirements” on page 78**.

Figure 7.2. ADC1 Track and Conversion Example Timing


A. ADC Timing for External Trigger Source

B. ADC Timing for Internal Trigger Source
CNVSTR
(AD1CM[2:0]=010)
1 2 3 4 5 6 7 8 9
SAR1 Clocks
AD1TM=1
Low Power or Convert
Track
Convert
Low Power Mode
AD1TM=0
Track or Convert
Convert
Track
Write '1' to AD1BUSY,
Timer 3 Overflow,
Timer 2 Overflow,
Write '1' to AD0BUSY
(AD1CM[2:0]=000, 001, 011, 1xx)
1 2 3 4 5 6 7 8 9 10 11 12
SAR1 Clocks
AD1TM=1
Low Power or Convert
Track
Convert
Low Power Mode
1 2 3 4 5 6 7 8 9
SAR1 Clocks
AD1TM=0
Track or Convert
Convert
Track

7.2.3. Settling Time Requirements

When the ADC1 input configuration is changed (i.e., a different MUX or PGA selection), a minimum settling (or tracking) time is required before an accurate conversion can be performed. This settling time is determined by the ADC1 MUX resistance, the ADC1 sampling capacitance, any external source resistance, and the accuracy required for the conversion. Figure 7.3 shows the equivalent ADC1 input circuit. The required ADC1 settling time for a given settling accuracy (SA) may be approximated by Equation 7.1. Note that in low-power tracking mode, three SAR1 clocks are used for tracking at the start of every conversion. For most applications, these three SAR1 clocks will meet the tracking requirements. See Table 7.1 for absolute minimum settling time requirements.

Equation 7.1. ADC1 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 tracking time in seconds
R_{TOTAL} is the sum of the ADC1 MUX resistance and any external source resistance.
n is the ADC resolution in bits (8).

Figure 7.3. ADC1 Equivalent Input Circuit

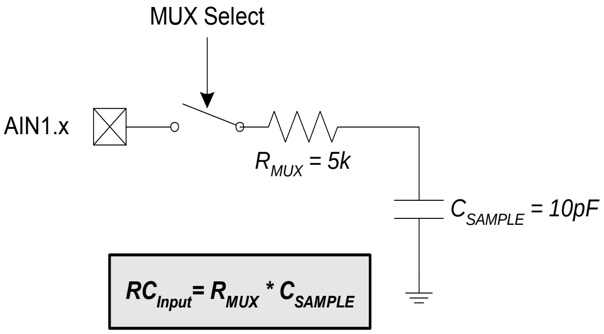


Figure 7.4. ADC1CF: ADC1 Configuration Register (C8051F020/1/2/3)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD1SC4	AD1SC3	AD1SC2	AD1SC1	AD1SC0	-	AMP1GN1	AMP1GN0	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xAB

Bits7-3: AD1SC4-0: ADC1 SAR Conversion Clock Period Bits
SAR Conversion clock is derived from system clock by the following equation, where *AD1SC* refers to the 5-bit value held in AD1SC4-0. SAR conversion clock requirements are given in Table 7.1.

$$AD1SC = \frac{SYSCLK}{CLK_{SAR1}} - 1$$

Bit2: UNUSED. Read = 0b. Write = don’t care.
Bits1-0: AMP1GN1-0: ADC1 Internal Amplifier Gain (PGA)
00: Gain = 0.5
01: Gain = 1
10: Gain = 2
11: Gain = 4

Figure 7.5. AMX1SL: AMUX1 Channel Select Register (C8051F020/1/2/3)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	AMX1AD2	AMX1AD1	AMX1AD0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xAC

Bits7-3: UNUSED. Read = 00000b; Write = don’t care
Bits2-0: AMX1AD2-0: AMX1 Address Bits
000-111b: ADC1 Inputs selected as follows:
000: AIN1.0 selected
001: AIN1.1 selected
010: AIN1.2 selected
011: AIN1.3 selected
100: AIN1.4 selected
101: AIN1.5 selected
110: AIN1.6 selected
111: AIN1.7 selected

Figure 7.6. ADC1CN: ADC1 Control Register (C8051F020/1/2/3)

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	SFR Address: 0xAA

Bit7: AD1EN: ADC1 Enable Bit.
0: ADC1 Disabled. ADC1 is in low-power shutdown.
1: ADC1 Enabled. ADC1 is active and ready for data conversions.

Bit6: AD1TM: ADC1 Track Mode Bit.
0: Normal Track Mode: When ADC1 is enabled, tracking is continuous unless a conversion is in process.
1: Low-power Track Mode: Tracking Defined by AD1STM2-0 bits (see below).

Bit5: 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.

Bit4: 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 AD1STM2-0 = 000b

Bit3-1: AD1CM2-0: ADC1 Start of Conversion Mode Select.
AD1TM = 0:
000: ADC1 conversion initiated on every write of ‘1’ to AD1BUSY.
001: ADC1 conversion initiated on overflow of Timer 3.
010: ADC1 conversion initiated on rising edge of external CNVSTR.
011: ADC1 conversion initiated on overflow of Timer 2.
1xx: ADC1 conversion initiated on write of ‘1’ to AD0BUSY (synchronized with ADC0 software-commanded conversions).
AD1TM = 1:
000: Tracking initiated on write of ‘1’ to AD1BUSY and lasts 3 SAR1 clocks, followed by conversion.
001: Tracking initiated on overflow of Timer 3 and lasts 3 SAR1 clocks, followed by conversion.
010: ADC1 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge.
011: Tracking initiated on overflow of Timer 2 and lasts 3 SAR1 clocks, followed by conversion.
1xx: Tracking initiated on write of ‘1’ to AD0BUSY and lasts 3 SAR1 clocks, followed by conversion.

Bit0: UNUSED. Read = 0b. Write = don’t care.

Figure 7.7. ADC1: ADC1 Data Word 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

Bits7-0: ADC1 Data Word.

Figure 7.8. ADC1 Data Word Example

8-bit ADC Data Word appears in the ADC1 Data Word Register as follows:

Example: ADC1 Data Word Conversion Map, AIN1.0 Input
(AMX1SL = 0x00)

AIN1.0-AGND (Volts)	ADC1
VREF * (255/256)	0xFF
VREF / 2	0x80
VREF * (127/256)	0x7F
0	0x00

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

Table 7.1. ADC1 Electrical Characteristics

VDD = 3.0 V, AV+ = 3.0 V, VREF1 = 2.40 V (REFBE=0), PGA1 = 1, -40°C to +85°C unless otherwise specified

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
DC ACCURACY					
Resolution			8		bits
Integral Nonlinearity				±1	LSB
Differential Nonlinearity	Guaranteed Monotonic			±1	LSB
Offset Error			0.5±0.3		LSB
Full Scale Error	Differential mode		-1±0.2		LSB
Offset Temperature Coefficient			TBD		ppm/°C
DYNAMIC PERFORMANCE (10 kHz sine-wave input, 0 to 1 dB below Full Scale, 500 ksps					
Signal-to-Noise Plus Distortion		45	47		dB
Total Harmonic Distortion	Up to the 5th harmonic		-51		dB
Spurious-Free Dynamic Range			52		dB
CONVERSION RATE					
SAR Conversion Clock				6	MHz
Conversion Time in SAR Clocks		8			clocks
Track/Hold Acquisition Time		300			ns
Throughput Rate				500	ksps
ANALOG INPUTS					
Input Voltage Range		0		VREF	V
Input Capacitance			10		pF
POWER SPECIFICATIONS					
Power Supply Current (AV+ supplied to ADC1)	Operating Mode, 500 ksps		420	900	µA
Power Supply Rejection			±0.3		mV/V

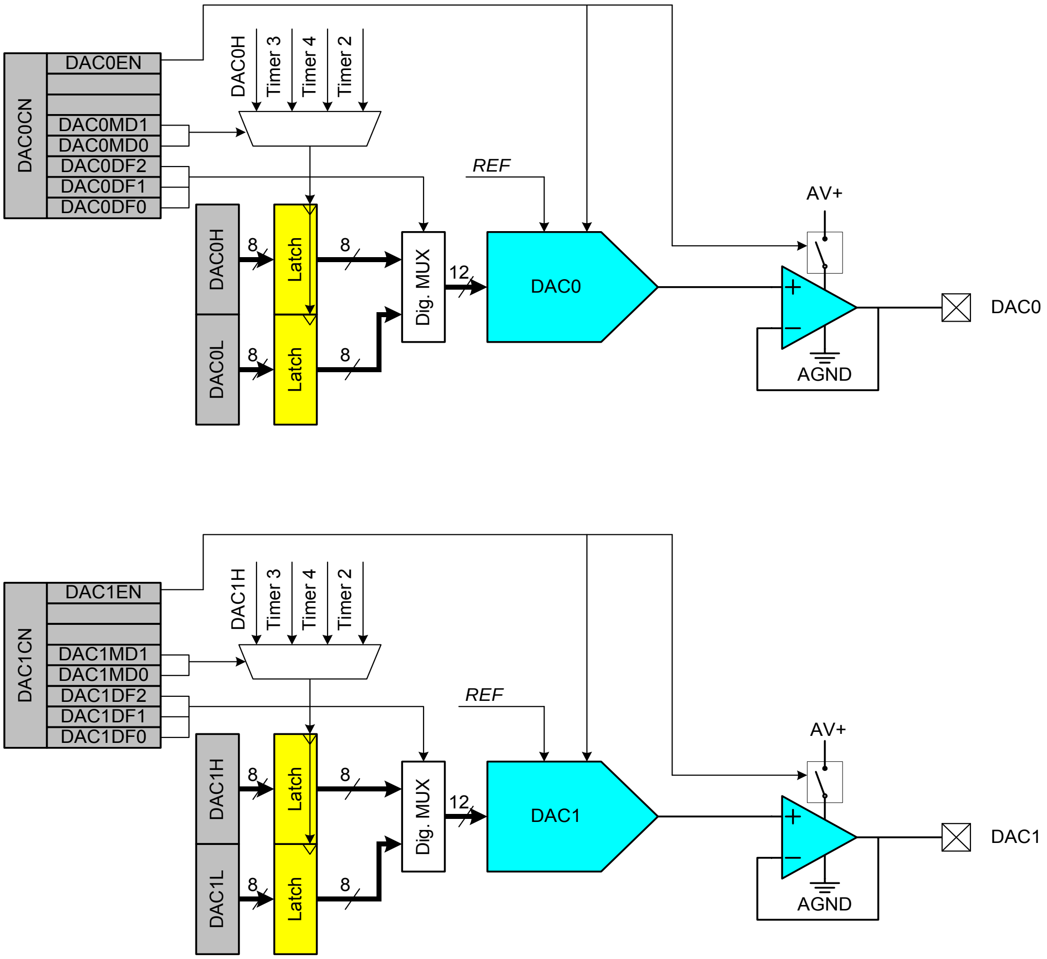
8. DACS, 12-BIT VOLTAGE MODE

Each C8051F020/1/2/3 device includes two on-chip 12-bit voltage-mode Digital-to-Analog Converters (DACs). Each DAC has an output swing of 0V 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 (C8051F020/2 devices) or the VREF pin (C8051F021/3 devices). Note that the VREF pin on C8051F021/3 devices may be driven by the internal voltage reference or an external source. If the internal voltage reference is used it must be enabled in order for the DAC outputs to be valid. See **Section “9. VOLTAGE REFERENCE (C8051F020/2)” on page 91** or **Section “10. VOLTAGE REFERENCE (C8051F021/3)” on page 93** for more information on configuring the voltage reference for the DACs.

8.1. DAC Output Scheduling

Each DAC features a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. The following examples are written in terms of DAC0, but DAC1 operation is identical. Note that reads from DAC0L return pre-latch data, meaning the value read is the same as the last value written to this register, not the value at the DAC0L latch. Reads from DAC0H always return the value at the DAC0H latch.

Figure 8.1. DAC Functional Block Diagram


DAC0CN
DAC0EN
DAC0MD1
DAC0MD0
DAC0DF2
DAC0DF1
DAC0DF0
DAC0H
Timer 3
Timer 4
Timer 2
REF
DAC0H
DAC0L
Latch
Latch
8
8
8
8
Dig. MUX
12
DAC0
AV+
AGND
DAC0
DAC1CN
DAC1EN
DAC1MD1
DAC1MD0
DAC1DF2
DAC1DF1
DAC1DF0
DAC1H
Timer 3
Timer 4
Timer 2
REF
DAC1H
DAC1L
Latch
Latch
8
8
8
8
Dig. MUX
12
DAC1
AV+
AGND
DAC1

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’s 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

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD3

Bits7-0: DAC0 Data Word Most Significant Byte.

Figure 8.3. DAC0L: DAC0 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:
								0xD2

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

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

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: 0xD6

Bits7-0: DAC1 Data Word Most Significant Byte.

Figure 8.6. DAC1L: DAC1 Low Byte Register

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: 0xD7

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			±2		LSB
Differential Nonlinearity				±1	LSB
Output Noise	No Output Filter 100 kHz Output Filter 10 kHz Output Filter		250 128 41		µVrms
Offset Error	Data Word = 0x014		±3	±30	mV
Offset Tempco			6		ppm/°C
Gain Error			±20	±60	mV
Gain-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		110	400	µA

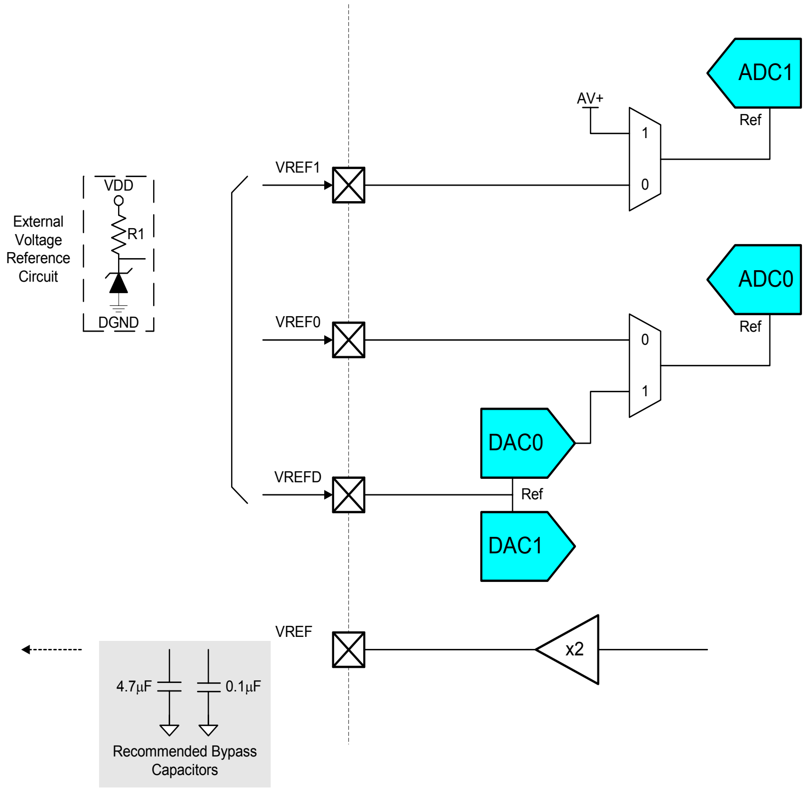
Notes

9. VOLTAGE REFERENCE (C8051F020/2)

The voltage reference circuit offers full flexibility in operating the ADC and DAC modules. Three voltage reference input pins allow each ADC and the two DACs to reference an external voltage reference or the on-chip voltage reference output. ADC0 may also reference the DAC0 output internally, and ADC1 may reference the analog power supply voltage, via the VREF multiplexers shown in Figure 9.1.

The internal voltage reference circuit consists of a 1.2 V, 15 ppm/°C (typical) bandgap voltage reference generator and a gain-of-two output buffer amplifier. The internal reference may be routed via the VREF pin to external system components or to the voltage reference input pins shown in Figure 9.1. Bypass capacitors of 0.1 µF and 4.7 µF are recommended from the VREF pin to AGND, as shown in Figure 9.1. See Table 9.1 for voltage reference specifications.

The Reference Control Register, REF0CN (defined in Figure 9.2) enables/disables the internal reference generator and selects the reference inputs for ADC0 and ADC1. The BIASE bit in REF0CN enables the on-board reference generator while the REFBE bit enables the gain-of-two buffer amplifier which drives the VREF pin. When disabled, the supply current drawn by the bandgap and buffer amplifier falls to less than 1 µA (typical) and the output of the buffer amplifier enters a high impedance state. If the internal bandgap is used as the reference voltage generator, BIASE and REFBE must both be set to logic 1. If the internal reference is not used, REFBE may be set to logic 0. Note that the BIASE bit must be set to logic 1 if either DAC or ADC is used, regardless of whether the voltage reference is derived from the on-chip reference or supplied by an off-chip source. If neither the ADC nor the DAC are being used, both of these bits can be set to logic 0 to conserve power. Bits AD0VRS and AD1VRS select the ADC0 and ADC1 voltage reference sources, respectively. The electrical specifications for the Voltage Reference circuit are given in Table 9.1.



The temperature sensor connects to the highest order input of the ADC0 input multiplexer (see **Section "5.1. Analog Multiplexer and PGA" on page 43** for C8051F020/1 devices, or **Section "6.1. Analog Multiplexer and PGA" on page 59** for C8051F022/3 devices). The TEMPE bit within REF0CN 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 undefined data.

Figure 9.2. REF0CN: Reference Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	AD0VRS	AD1VRS	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD1

Bits7-5: UNUSED. Read = 000b; Write = don't care.
Bit4: AD0VRS: ADC0 Voltage Reference Select
0: ADC0 voltage reference from VREF0 pin.
1: ADC0 voltage reference from DAC0 output.
Bit3: AD1VRS: ADC1 Voltage Reference Select
0: ADC1 voltage reference from VREF1 pin.
1: ADC1 voltage reference from AV+.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC or DAC).
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°C 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 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	4.7µF tantalum, 0.1µF ceramic bypass		2		ms
VREF Turn-on Time 2	0.1µF 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

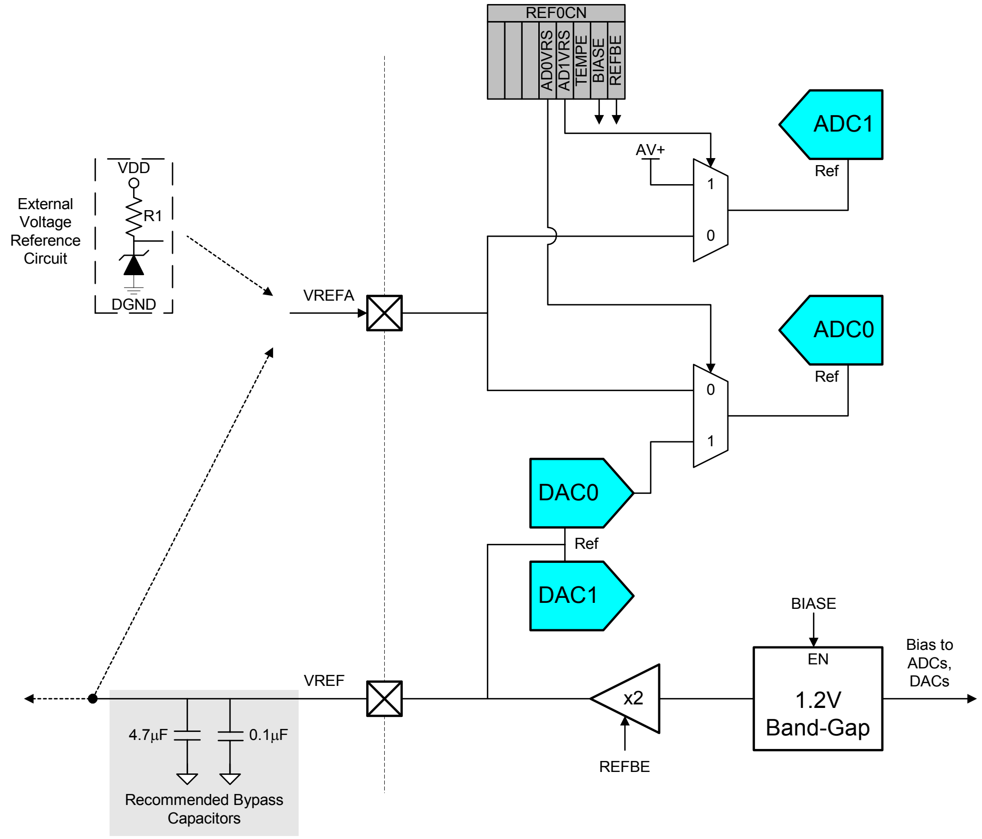
10. VOLTAGE REFERENCE (C8051F021/3)

The internal voltage reference circuit consists of a 1.2 V, 15 ppm/°C (typical) bandgap voltage reference generator and a gain-of-two output buffer amplifier. The internal reference may be routed via the VREF pin to external system components or to the VREFA input pin shown in Figure 10.1. Bypass capacitors of 0.1 µF and 4.7 µF are recommended from the VREF pin to AGND, as shown in Figure 10.1. See Table 10.1 for voltage reference specifications.

The VREFA pin provides a voltage reference input for ADC0 and ADC1. ADC0 may also reference the DAC0 output internally, and ADC1 may reference the analog power supply voltage, via the VREF multiplexers shown in Figure 10.1.

The Reference Control Register, REF0CN (defined in Figure 10.2) enables/disables the internal reference generator and selects the reference inputs for ADC0 and ADC1. The BIASE bit in REF0CN enables the on-board reference generator while the REFBE bit enables the gain-of-two buffer amplifier which drives the VREF pin. When disabled, the supply current drawn by the bandgap and buffer amplifier falls to less than 1 µA (typical) and the output of the buffer amplifier enters a high impedance state. If the internal bandgap is used as the reference voltage generator, BIASE and REFBE must both be set to 1 (this includes any time a DAC is used). 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 either ADC is used, regardless of whether the voltage reference is derived from the on-chip reference or supplied by an off-chip source. If neither the ADC nor the DAC are being used, both of these bits can be set to logic 0 to conserve power. Bits AD0VRS and AD1VRS select the ADC0 and ADC1 voltage reference sources, respectively. The electrical specifications for the Voltage Reference are given in Table 10.1.

Figure 10.1. Voltage Reference Functional Block Diagram


REF0CN
AD0VRS
AD1VRS
TEMPE
BIASE
REFBE
ADC1
Ref
AV+
1
0
External Voltage Reference Circuit
VDD
R1
DGND
VREFA
ADC0
Ref
0
1
DAC0
Ref
DAC1
BIASE
EN
1.2V Band-Gap
Bias to ADCs, DACs
VREF
x2
REFBE
4.7µF
0.1µF
Recommended Bypass Capacitors

The temperature sensor connects to the highest order input of the ADC0 input multiplexer (see **Section "5.1. Analog Multiplexer and PGA" on page 43** for C8051F020/1 devices, or **Section "6.1. Analog Multiplexer and PGA" on page 59** for C8051F022/3 devices). The TEMPE bit within REF0CN 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 undefined data.

Figure 10.2. REF0CN: Reference Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	AD0VRS	AD1VRS	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD1

Bits7-5: UNUSED. Read = 000b; Write = don't care.
Bit4: AD0VRS: ADC0 Voltage Reference Select
0: ADC0 voltage reference from VREFA pin.
1: ADC0 voltage reference from DAC0 output.
Bit3: AD1VRS: ADC1 Voltage Reference Select
0: ADC1 voltage reference from VREFA pin.
1: ADC1 voltage reference from AV+.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC or DAC).
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°C 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 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	4.7µF tantalum, 0.1µF ceramic bypass		2		ms
VREF Turn-on Time 2	0.1µF 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. COMPARATORS

Each MCU includes two on-board voltage comparators as shown in Figure 11.1. The inputs of each Comparator are available at the package pins. The output of each comparator is optionally available at the package pins via the I/O crossbar. When assigned to package pins, each comparator output can be programmed to operate in open drain or push-pull modes. See Section "17. PORT INPUT/OUTPUT" on page 161 for Crossbar and port initialization details.

The hysteresis of each comparator is software-programmable via its respective Comparator control register (CPT0CN and CPT1CN for Comparator0 and Comparator1, respectively). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage. The output of the comparator can be polled in software, or can be used as an interrupt source. Each comparator can be individually enabled or disabled (shutdown). When disabled, the comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, its interrupt capability is suspended and its supply current falls to less than 1 µA. Comparator inputs can be externally driven from -0.25 V to (AV+) + 0.25 V without damage or upset.

The Comparator0 hysteresis is programmed using bits 3-0 in the Comparator0 Control Register CPT0CN (shown in Figure 11.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits; In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits. See Table 11.1 on page 99 for hysteresis level specifications.

Comparator interrupts can be generated on rising-edge and/or falling-edge output transitions. (For interrupt enable and priority control, see Section "12.3. Interrupt Handler" on page 116). The CP0FIF flag is set upon a Comparator0 falling-edge interrupt, and the CP0RIF flag is set upon the Comparator0 rising-edge interrupt. Once set, these bits remain set until cleared by software. The Output State of Comparator0 can be obtained at any time by reading the CP0OUT bit. Comparator0 is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit

Figure 11.1. Comparator Functional Block Diagram

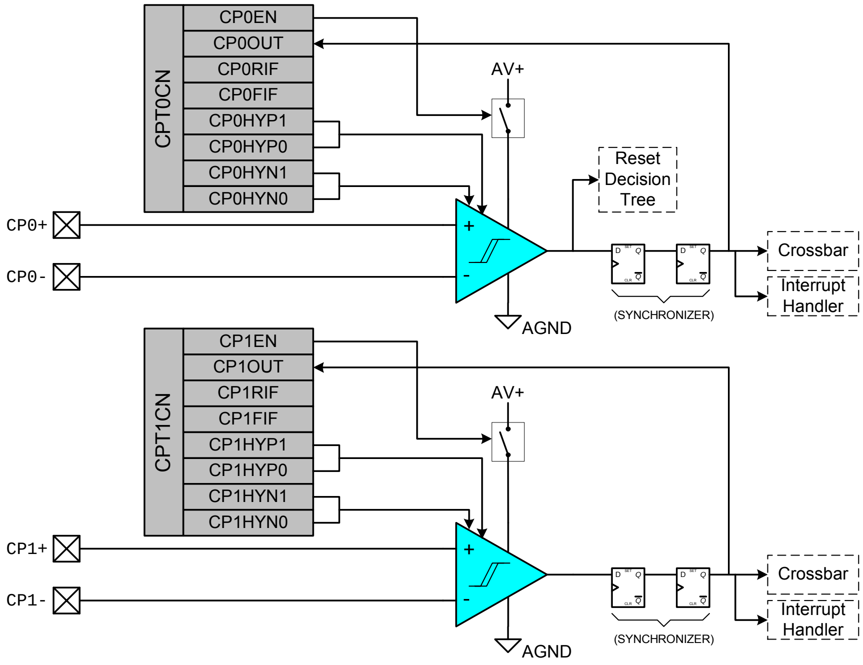
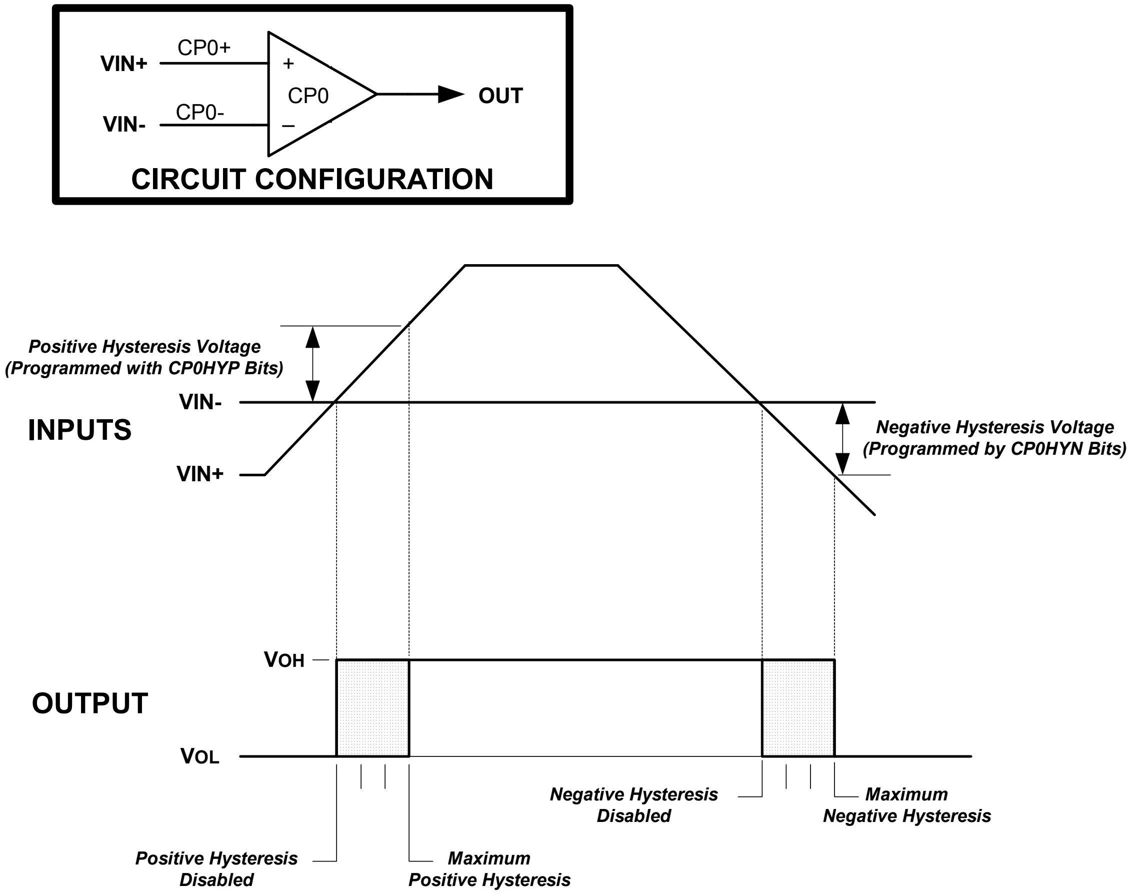

CPT0CN
CP0EN
CP0OUT
CP0RIF
CP0FIF
CP0HYP1
CP0HYP0
CP0HYN1
CP0HYN0
AV+
Reset
Decision
Tree
CP0+
CP0-
+
-
AGND
(SYNCHRONIZER)
Crossbar
Interrupt
Handler
CPT1CN
CP1EN
CP1OUT
CP1RIF
CP1FIF
CP1HYP1
CP1HYP0
CP1HYN1
CP1HYN0
AV+
CP1+
CP1-
+
-
AGND
(SYNCHRONIZER)
Crossbar
Interrupt
Handler

Figure 11.2. Comparator Hysteresis Plot


VIN+
CP0+
VIN-
CP0-
+
CP0
–
OUT
CIRCUIT CONFIGURATION
Positive Hysteresis Voltage
(Programmed with CP0HYP Bits)
VIN-
INPUTS
VIN+
Negative Hysteresis Voltage
(Programmed by CP0HYN Bits)
VOH
OUTPUT
VOL
Negative Hysteresis
Disabled
Maximum
Negative Hysteresis
Positive Hysteresis
Disabled
Maximum
Positive Hysteresis

to logic 0. Comparator0 can also be programmed as a reset source; for details, see Section “13.6. Comparator0 Reset” on page 129.

The operation of Comparator1 is identical to that of Comparator0, though Comparator1 may not be configured as a reset source. Comparator1 is controlled by the CPT1CN Register (Figure 11.4). The complete electrical specifications for the Comparators are given in Table 11.1.

Figure 11.3. CPT0CN: Comparator0 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP1	CP0HYP0	CP0HYN1	CP0HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9E

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

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

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

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

Bits3-2: CP0HYP1-0: Comparator0 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 2 mV.
10: Positive Hysteresis = 4 mV.
11: Positive Hysteresis = 10 mV.

Bits1-0: CP0HYN1-0: Comparator0 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 2 mV.
10: Negative Hysteresis = 4 mV.
11: Negative Hysteresis = 10 mV.

Figure 11.4. CPT1CN: Comparator1 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

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

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

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

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

Bits3-2: CP1HYP1-0: Comparator1 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 2 mV.
10: Positive Hysteresis = 4 mV.
11: Positive Hysteresis = 10 mV.

Bits1-0: CP1HYN1-0: Comparator1 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 2 mV.
10: Negative Hysteresis = 4 mV.
11: Negative Hysteresis = 10 mV.

Table 11.1. Comparator Electrical Characteristics

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

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Response Time 1	CP+ - CP- = 100 mV		4		µs
Response Time 2	CP+ - CP- = 10 mV		12		µs
Common-Mode Rejection Ratio			1.5	4	mV/V
Positive Hysteresis 1	CPnHYP1-0 = 00		0	1	mV
Positive Hysteresis 2	CPnHYP1-0 = 01	2	4.5	7	mV
Positive Hysteresis 3	CPnHYP1-0 = 10	4	9	13	mV
Positive Hysteresis 4	CPnHYP1-0 = 11	10	17	25	mV
Negative Hysteresis 1	CPnHYN1-0 = 00		0	1	mV
Negative Hysteresis 2	CPnHYN1-0 = 01	2	4.5	7	mV
Negative Hysteresis 3	CPnHYN1-0 = 10	4	9	13	mV
Negative Hysteresis 4	CPnHYN1-0 = 11	10	17	25	mV
Inverting or Non-Inverting Input Voltage Range		-0.25		(AV+) + 0.25	V
Input Capacitance			7		pF
Input Bias Current		-5	0.001	+5	nA
Input Offset Voltage		-10		+10	mV
POWER SUPPLY					
Power-up Time	CPnEN from 0 to 1		20		µs
Power Supply Rejection			0.1	1	mV/V
Supply Current	Operating Mode (each comparator) at DC		1.5	10	µA

Notes

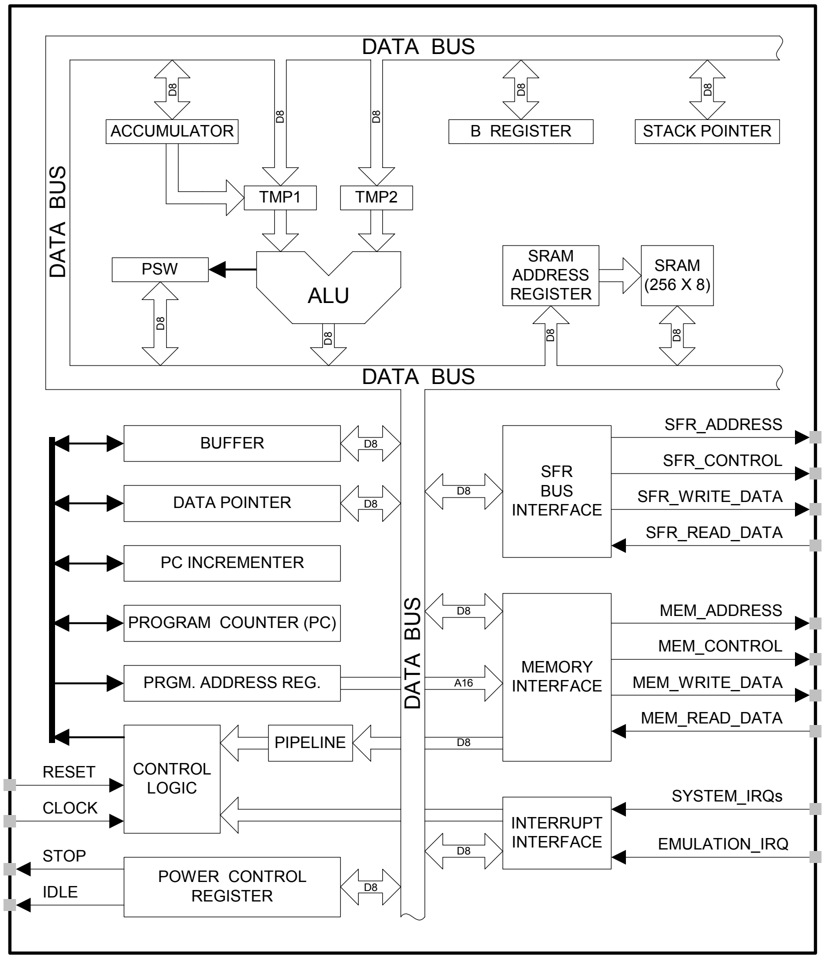
12. CIP-51 MICROCONTROLLER

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 22**), two full-duplex UARTs (see description in **Section 20** and **Section 21**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (see **Section 12.2.6**), and 8/4 byte-wide I/O Ports (see description in **Section 17**). The CIP-51 also includes on-chip debug hardware (see description in **Section 24**), and interfaces directly with the MCUs' analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 12.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- 256 Bytes of Internal RAM
- 8/4 Byte-Wide I/O Ports
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

Figure 12.1. CIP-51 Block Diagram


DATA BUS
D8
ACCUMULATOR
B REGISTER
STACK POINTER
TMP1
TMP2
PSW
ALU
SRAM ADDRESS REGISTER
SRAM (256 X 8)
DATA BUS
BUFFER
DATA POINTER
PC INCREMENTER
PROGRAM COUNTER (PC)
PRGM. ADDRESS REG.
PIPELINE
CONTROL LOGIC
RESET
CLOCK
STOP
IDLE
POWER CONTROL REGISTER
A16
SFR BUS INTERFACE
SFR_ADDRESS
SFR_CONTROL
SFR_WRITE_DATA
SFR_READ_DATA
MEMORY INTERFACE
MEM_ADDRESS
MEM_CONTROL
MEM_WRITE_DATA
MEM_READ_DATA
INTERRUPT INTERFACE
SYSTEM_IRQs
EMULATION_IRQ

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

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

12.1. 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.

12.1.1. 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 12.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

12.1.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip XRAM, and accessing on-chip program FLASH memory. The FLASH access feature provides a mechanism for user software to update program code and use the program memory space for non-volatile data storage (see **Section “15. FLASH**

MEMORY” on page 139). The External Memory Interface provides a fast access to off-chip XRAM (or memory-mapped peripherals) via the MOVX instruction. Refer to **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for details.

Table 12.1. 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 borro9(m)1.8(A wit)-4.2(h)1.1(bo10 0 9.96 510.4799 500.34 Tm46.		

Table 12.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
DATA TRANSFER			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
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

Table 12.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
PROGRAM BRANCHING			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
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.

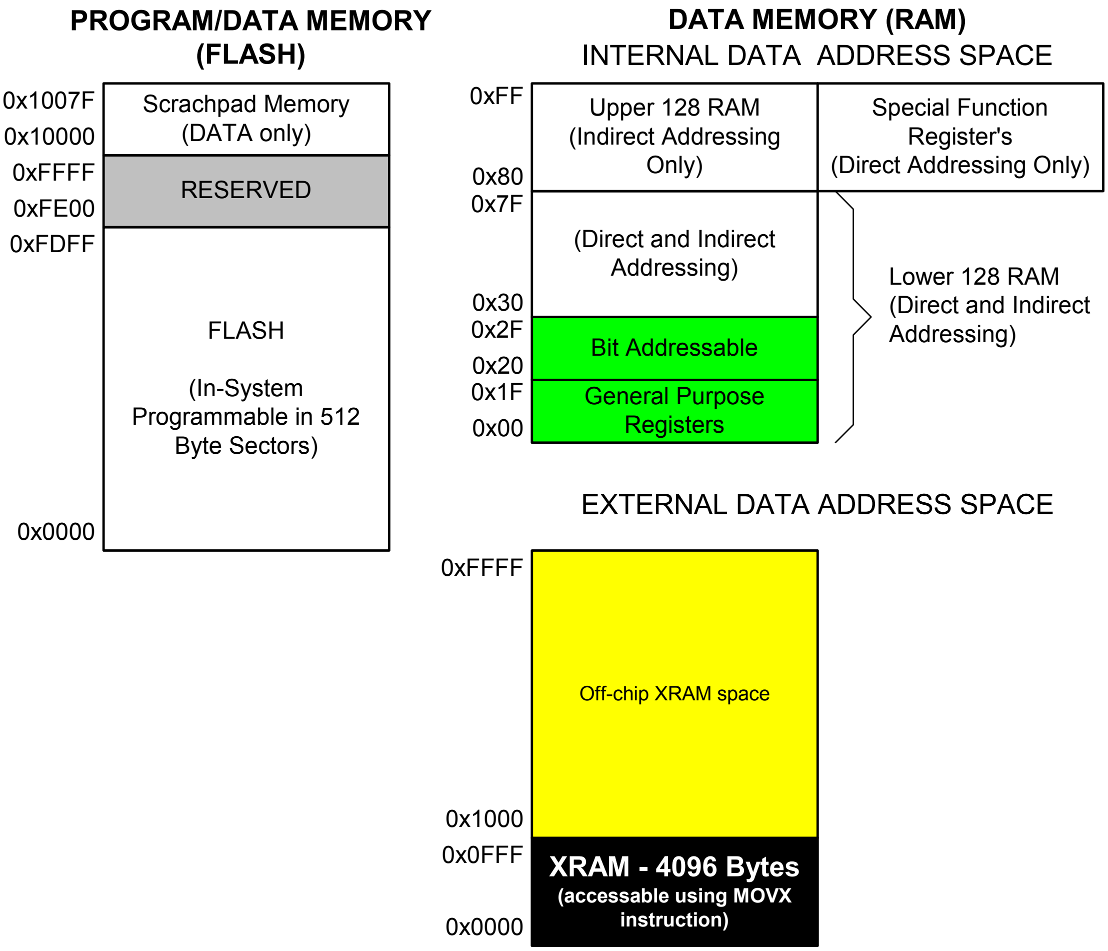
addr16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space.

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

12.2. 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 64k bytes of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 12.2.

Figure 12.2. Memory Map


PROGRAM/DATA MEMORY (FLASH)
0x1007F
0x10000
Scratchpad Memory (DATA only)
0xFFFF
0xFE00
RESERVED
0xFDFF
FLASH
(In-System Programmable in 512 Byte Sectors)
0x0000
DATA MEMORY (RAM)
INTERNAL DATA ADDRESS SPACE
0xFF
Upper 128 RAM (Indirect Addressing Only)
Special Function Register's (Direct Addressing Only)
0x80
0x7F
(Direct and Indirect Addressing)
0x30
0x2F
Bit Addressable
0x20
0x1F
General Purpose Registers
0x00
Lower 128 RAM (Direct and Indirect Addressing)
EXTERNAL DATA ADDRESS SPACE
0xFFFF
Off-chip XRAM space
0x1000
0x0FFF
XRAM - 4096 Bytes
(accessable using MOVX instruction)
0x0000

12.2.1. Program Memory

The CIP-51 has a 64k byte program memory space. The MCU implements 65536 bytes of this program memory space as in-system re-programmed FLASH memory, organized in a contiguous block from addresses 0x0000 to 0xFFFF. Note: 512 bytes (0xEE00 to 0xFFFF) of this memory are reserved for factory use and are not available for user program storage.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section “15. FLASH MEMORY” on page 139** for further details.

12.2.2. 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 (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 12.2 illustrates the data memory organization of the CIP-51.

12.2.3. 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 12.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

12.2.4. Bit Addressable Locations

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

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

```
MOV    C, 22.3h
```

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

12.2.5. 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. 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.

12.2.6. 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 12.2 lists the SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, 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 12.3, for a detailed description of each register.

Table 12.2. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0H	PCA0CPH0	PCA0CPH1	PCA0CPH2	PCA0CPH3	PCA0CPH4	WDTCN
F0	B	SCON1	SBUF1	SADDR1	TL4	TH4	EIP1	EIP2
E8	ADC0CN	PCA0L	PCA0CPL0	PCA0CPL1	PCA0CPL2	PCA0CPL3	PCA0CPL4	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	RCAP4L	RCAP4H	EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	
D0	PSW	REF0CN	DAC0L	DAC0H	DAC0CN	DAC1L	DAC1H	DAC1CN
C8	T2CON	T4CON	RCAP2L	RCAP2H	TL2	TH2		SMB0CR
C0	SMB0CN	SMB0STA	SMB0DAT	SMB0ADR	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH
B8	IP	SADEN0	AMX0CF	AMX0SL	ADC0CF	P1MDIN	ADC0L	ADC0H
B0	P3	OSCXCN	OSCICN			P74OUT†	FLSCL	FLACL
A8	IE	SADDR0	ADC1CN	ADC1CF	AMX1SL	P3IF	SADEN1	EMI0CN
A0	P2	EMI0TC		EMI0CF	P0MDOUT	P1MDOUT	P2MDOUT	P3MDOUT
98	SCON0	SBUF0	SPI0CFG	SPI0DAT	ADC1	SPI0CKR	CPT0CN	CPT1CN
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	P7†	
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	P4†	P5†	P6†	PCON
	0(8) (bit addressable)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Table 12.3. Special Function Registers

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

Register	Address	Description	Page No.
ACC	0xE0	Accumulator	page 115
ADC0CF	0xBC	ADC0 Configuration	page 49*, page 65**
ADC0CN	0xE8	ADC0 Control	page 50*, page 66**
ADC0GTH	0xC5	ADC0 Greater-Than High	page 53*, page 69**
ADC0GTL	0xC4	ADC0 Greater-Than Low	page 53*, page 69**
ADC0H	0xBF	ADC0 Data Word High	page 51*, page 67**
ADC0L	0xBE	ADC0 Data Word Low	page 51*, page 67**

Table 12.3. Special Function Registers

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

Register	Address	Description	Page No.
ADC0LTH	0xC7	ADC0 Less-Than High	page 53*, page 69**
ADC0LTL	0xC6	ADC0 Less-Than Low	page 53*, page 69**
ADC1CF	0xAB	ADC1 Analog Multiplexer Configuration	page 79
ADC1CN	0xAA	ADC1 Control	page 80
ADC1	0x9C	ADC1 Data Word	page 81
AMX0CF	0xBA	ADC0 Multiplexer Configuration	page 47*, page 63**
AMX0SL	0xBB	ADC0 Multiplexer Channel Select	page 48*, page 64**
AMX1SL	0xAC	ADC1 Analog Multiplexer Channel Select	page 79
B	0xF0	B Register	page 115
CKCON	0x8E	Clock Control	page 226
CPT0CN	0x9E	Comparator 0 Control	page 97
CPT1CN	0x9F	Comparator 1 Control	page 98
DAC0CN	0xD4	DAC0 Control	page 86
DAC0H	0xD3	DAC0 High	page 85
DAC0L	0xD2	DAC0 Low	page 85
DAC1CN	0xD7	DAC1 Control	page 88
DAC1H	0xD6	DAC1 High Byte	page 87
DAC1L	0xD5	DAC1 Low Byte	page 87
DPH	0x83	Data Pointer High	page 113
DPL	0x82	Data Pointer Low	page 113
EIE1	0xE6	Extended Interrupt Enable 1	page 121
EIE2	0xE7	Extended Interrupt Enable 2	page 122
EIP1	0xF6	External Interrupt Priority 1	page 123
EIP2	0xF7	External Interrupt Priority 2	page 124
EMI0CN	0xAF	External Memory Interface Control	page 147
EMI0CF	0xA3	EMIF Configuration	page 147
EMI0TC	0xA1	EMIF Timing Control	page 152
FLACL	0xB7	FLASH Access Limit	page 142
FLSCL	0xB6	FLASH Scale	page 143
IE	0xA8	Interrupt Enable	page 119
IP	0xB8	Interrupt Priority	page 120
OSCICN	0xB2	Internal Oscillator Control	page 136
OSCXCN	0xB1	External Oscillator Control	page 137
P0	0x80	Port 0 Latch	page 173
P0MDOUT	0xA4	Port 0 Output Mode Configuration	page 173
P1	0x90	Port 1 Latch	page 174
P1MDIN	0xBD	Port 1 Input Mode	page 174
P1MDOUT	0xA5	Port 1 Output Mode Configuration	page 175
P2	0xA0	Port 2 Latch	page 175
P2MDOUT	0xA6	Port 2 Output Mode Configuration	page 175
P3	0xB0	Port 3 Latch	page 176
P3IF	0xAD	Port 3 Interrupt Flags	page 177
P3MDOUT	0xA7	Port 3 Output Mode Configuration	page 176
†P4	0x84	Port 4 Latch	page 180†
†P5	0x85	Port 5 Latch	page 180†

Table 12.3. Special Function Registers

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

Register	Address	Description	Page No.
†P6	0x86	Port 6 Latch	page 181†
†P7	0x96	Port 7 Latch	page 181†
†P74OUT	0xB5	Port 4 through 7 Output Mode	page 179†
PCA0CN	0xD8	PCA Control	page 259
PCA0CPH0	0xFA	PCA Capture 0 High	page 263
PCA0CPH1	0xFB	PCA Capture 1 High	page 263
PCA0CPH2	0xFC	PCA Capture 2 High	page 263
PCA0CPH3	0xFD	PCA Capture 3 High	page 263
PCA0CPH4	0xFE	PCA Capture 4 High	page 263
PCA0CPL0	0xEA	PCA Capture 0 Low	page 263
PCA0CPL1	0xEB	PCA Capture 1 Low	page 263
PCA0CPL2	0xEC	PCA Capture 2 Low	page 263
PCA0CPL3	0xED	PCA Capture 3 Low	page 263
PCA0CPL4	0xEE	PCA Capture 4 Low	page 263
PCA0CPM0	0xDA	PCA Module 0 Mode Register	page 261
PCA0CPM1	0xDB	PCA Module 1 Mode Register	page 261
PCA0CPM2	0xDC	PCA Module 2 Mode Register	page 261
PCA0CPM3	0xDD	PCA Module 3 Mode Register	page 261
PCA0CPM4	0xDE	PCA Module 4 Mode Register	page 261
PCA0H	0xF9	PCA Counter High	page 262
PCA0L	0xE9	PCA Counter Low	page 262
PCA0MD	0xD9	PCA Mode	page 260
PCON	0x87	Power Control	page 126
PSCTL	0x8F	Program Store R/W Control	page 144
PSW	0xD0	Program Status Word	page 114
RCAP2H	0xCB	Timer/Counter 2 Capture High	page 239
RCAP2L	0xCA	Timer/Counter 2 Capture Low	page 239
RCAP4H	0xE5	Timer/Counter 4 Capture High	page 248
RCAP4L	0xE4	Timer/Counter 4 Capture Low	page 248
REF0CN	0xD1	Programmable Voltage Reference Control	page 92†, page 94††
RSTSRC	0xEF	Reset Source Register	page 132
SADDR0	0xA9	UART0 Slave Address	page 214
SADDR1	0xF3	UART1 Slave Address	page 224
SADEN0	0xB9	UART0 Slave Address Enable	page 214
SADEN1	0xAE	UART1 Slave Address Enable	page 224
SBUF0	0x99	UART0 Data Buffer	page 214
SBUF1	0xF2	UART1 Data Buffer	page 224
SCON0	0x98	UART0 Control	page 213
SCON1	0xF1	UART1 Control	page 223
SMB0ADR	0xC3	SMBus Slave Address	page 193
SMB0CN	0xC0	SMBus Control	page 191
SMB0CR	0xCF	SMBus Clock Rate	page 192
SMB0DAT	0xC2	SMBus Data	page 193
SMB0STA	0xC1	SMBus Status	page 194
SP	0x81	Stack Pointer	page 113

Table 12.3. Special Function Registers

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

Register	Address	Description	Page No.
SPI0CFG	0x9A	SPI Configuration	page 201
SPI0CKR	0x9D	SPI Clock Rate Control	page 203
SPI0CN	0xF8	SPI Control	page 202
SPI0DAT	0x9B	SPI Data	page 203
T2CON	0xC8	Timer/Counter 2 Control	page 238
T4CON	0xC9	Timer/Counter 4 Control	page 247
TCON	0x88	Timer/Counter Control	page 231
TH0	0x8C	Timer/Counter 0 High	page 233
TH1	0x8D	Timer/Counter 1 High	page 233
TH2	0xCD	Timer/Counter 2 High	page 239
TH4	0xF5	Timer/Counter 4 High	page 248
TL0	0x8A	Timer/Counter 0 Low	page 233
TL1	0x8B	Timer/Counter 1 Low	page 233
TL2	0xCC	Timer/Counter 2 Low	page 239
TL4	0xF4	Timer/Counter 4 Low	page 248
TMOD	0x89	Timer/Counter Mode	page 232
TMR3CN	0x91	Timer 3 Control	page 241
TMR3H	0x95	Timer 3 High	page 242
TMR3L	0x94	Timer 3 Low	page 242
TMR3RLH	0x93	Timer 3 Reload High	page 242
TMR3RLL	0x92	Timer 3 Reload Low	page 241
WDTCN	0xFF	Watchdog Timer Control	page 131
XBR0	0xE1	Port I/O Crossbar Control 0	page 170
XBR1	0xE2	Port I/O Crossbar Control 1	page 171
XBR2	0xE3	Port I/O Crossbar Control 2	page 172
0x97, 0xA2, 0xB3, 0xB4, 0xCE, 0xDF		Reserved	

* Refers to a register in the C8051F020/1 only.

** Refers to a register in the C8051F022/3 only.

† Refers to a register in the C8051F020/2 only.

†† Refers to a register in the C8051F021/3 only.

12.2.7. 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 12.3. 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

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 12.4. 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

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 12.5. 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

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 12.6. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xD0

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 if the last arithmetic operation resulted in a carry (addition), borrow (subtraction), or overflow (multiply or divide). It is cleared to 0 by all other arithmetic operations.

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 12.7. 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	SFR Address:
							(bit addressable)	0xE0

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

Figure 12.8. 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	SFR Address:
							(bit addressable)	0xF0

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

12.3. 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.

12.3.1. 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 12.4. 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).

12.3.2. External Interrupts

Two of the external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of bits IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flag for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag follows the state of the external interrupt's input pin. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

The remaining 2 external interrupts (External Interrupts 6-7) are edge-sensitive inputs configurable as active-low or active-high. The interrupt-pending flags and configuration bits for these interrupts are in the Port 3 Interrupt Flag Register shown in **Figure “17.19 P3IF: Port3 Interrupt Flag Register” on page 177**.

Table 12.4. 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) TI0 (SCON0.1)	Y		ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow (or EXF2)	0x002B	5	TF2 (T2CON.7)	Y		ET2 (IE.5)	PT2 (IP.5)
Serial Peripheral Interface	0x0033	6	SPIF (SPI0CN.7)	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.2)	Y		EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
Programmable Counter Array	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y		EPCA0 (EIE1.3)	PPCA0 (EIP1.3)
Comparator 0 Falling Edge	0x0053	10	CP0FIF (CPT0CN.4)			ECP0F (EIE1.4)	PCP0F (EIP1.4)
Comparator 0 Rising Edge	0x005B	11	CP0RIF (CPT0CN.5)			ECP0R (EIE1.5)	PCP0R (EIP1.5)
Comparator 1 Falling Edge	0x0063	12	CP1FIF (CPT1CN.4)			ECP1F (EIE1.6)	PCP1F (EIP1.6)
Comparator 1 Rising Edge	0x006B	13	CP1RIF (CPT1CN.5)			ECP1R (EIE1.7)	PCP1F (EIP1.7)
Timer 3 Overflow	0x0073	14	TF3 (TMR3CN.7)			ET3 (EIE2.0)	PT3 (EIP2.0)
ADC0 End of Conversion	0x007B	15	AD0INT (ADC0CN.5)	Y		EADC0 (EIE2.1)	PADC0 (EIP2.1)
Timer 4 Overflow	0x0083	16	TF4 (T4CON.7)			ET4 (EIE2.2)	PT4 (EIP2.2)
ADC1 End of Conversion	0x008B	17	AD1INT (ADC1CN.5)			EADC1 (EIE2.3)	PADC1 (EIP2.3)
External Interrupt 6	0x0093	18	IE6 (P3IF.5)			EX6 (EIE2.4)	PX6 (EIP2.4)
External Interrupt 7	0x009B	19	IE7 (P3IF.6)			EX7 (EIE2.5)	PX7 (EIP2.5)
UART1	0x00A3	20	RI1 (SCON1.0) TI1 (SCON1.1)			ES1	PS1
External Crystal OSC Ready	0x00AB	21	XTLVLD (OSCXCN.7)			EXVLD (EIE2.7)	PXVLD (EIP2.7)

12.3.3. 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 12.4.

12.3.4. 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.

12.3.5. 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 12.9. 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	SFR Address:
							(bit addressable)	0xA8

Bit7: EA: Enable All Interrupts.
This bit globally enables/disables all interrupts. When set to ‘0’, individual interrupt mask settings are overridden.
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 (T2CON.7).

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 (TCON.7).

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 (TCON.5).

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 12.10. 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	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB8

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 priority determined by default priority order. 1: Timer 2 interrupts set to high priority level.
Bit4:	PS0: UART0 Interrupt Priority Control. This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt priority determined by default priority order. 1: UART0 interrupts 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 priority determined by default priority order. 1: Timer 1 interrupts 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 priority determined by default priority order. 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 priority determined by default priority order. 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 priority determined by default priority order. 1: External Interrupt 0 set to high priority level.

Figure 12.11. EIE1: Extended Interrupt Enable 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ECP1R	ECP1F	ECP0R	ECP0F	EPCA0	EWADC0	ESMB0	ESPI0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE6

Bit7: ECP1R: Enable Comparator1 (CP1) Rising Edge Interrupt.
This bit sets the masking of the CP1 interrupt.
0: Disable CP1 Rising Edge interrupt.
1: Enable interrupt requests generated by the CP1RIF flag (CPT1CN.5).
Bit6: ECP1F: Enable Comparator (CP1) Falling Edge Interrupt.
This bit sets the masking of the CP1 interrupt.
0: Disable CP1 Falling Edge interrupt.
1: Enable interrupt requests generated by the CP1FIF flag (CPT1CN.4).
Bit5: ECP0R: Enable Comparator0 (CP0) Rising Edge Interrupt.
This bit sets the masking of the CP0 interrupt.
0: Disable CP0 Rising Edge interrupt.
1: Enable interrupt requests generated by the CP0RIF flag (CPT0CN.5).
Bit4: ECP0F: Enable Comparator0 (CP0) Falling Edge Interrupt.
This bit sets the masking of the CP0 interrupt.
0: Disable CP0 Falling Edge interrupt.
1: Enable interrupt requests generated by the CP0FIF flag (CPT0CN.4).
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 (SMB0CN.3).
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 SPIF flag (SPI0CN.7).

Figure 12.12. EIE2: Extended Interrupt Enable 2

Bit7: EXVLD: Enable External Clock Source Valid (XTLVLD) Interrupt.
This bit sets the masking of the XTLVLD interrupt.
0: Disable XTLVLD interrupt.
1: Enable interrupt requests generated by the XTLVLD flag (OSCXCN.7)
Bit6: ES1: Enable UART1 Interrupt.
This bit sets the masking of the UART1 interrupt.
0: Disable UART1 interrupt.
1: Enable UART1 interrupt.
Bit5: EX7: Enable External Interrupt 7.
This bit sets the masking of External Interrupt 7.
0: Disable External Interrupt 7.
1: Enable interrupt requests generated by the External Interrupt 7 input pin.
Bit4: EX6: Enable External Interrupt 6.
This bit sets the masking of External Interrupt 6.
0: Disable External Interrupt 6.
1: Enable interrupt requests generated by the External Interrupt 6 input pin.
Bit3: EADC1: Enable ADC1 End Of Conversion Interrupt.
This bit sets the masking of the ADC1 End of Conversion interrupt.
0: Disable ADC1 End of Conversion interrupt.
1: Enable interrupt requests generated by the ADC1 End of Conversion Interrupt.
Bit2: ET4: Enable Timer 4 Interrupt
This bit sets the masking of the Timer 4 interrupt.

Figure 12.13. EIP1: Extended Interrupt Priority 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PCP1R	PCP1F	PCP0R	PCP0F	PPCA0	PWADC0	PSMB0	PSPI0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF6

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

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

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

Bit4: PCP0F: Comparator0 (CP0) Falling Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 falling interrupt set to low priority level.
1: CP0 falling 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 12.14. EIP2: Extended Interrupt Priority 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PXVLD	EP1	PX7	PX6	PADC1	PT4	PADC0	PT3	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF7

Bit7: PXVLD: External Clock Source Valid (XTLVLD) Interrupt Priority Control.
This bit sets the priority of the XTLVLD interrupt.
0: XTLVLD interrupt set to low priority level.
1: XTLVLD interrupt set to high priority level.

Bit6: EP1: 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: PX7: External Interrupt 7 Priority Control.
This bit sets the priority of the External Interrupt 7.
0: External Interrupt 7 set to low priority level.
1: External Interrupt 7 set to high priority level.

Bit4: PX6: External Interrupt 6 Priority Control.
This bit sets the priority of the External Interrupt 6.
0: External Interrupt 6 set to low priority level.
1: External Interrupt 6 set to high priority level.

Bit3: PADC1: ADC1 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.
1: ADC1 End of Conversion interrupt set to low 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 low priority.

Bit1: 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.

Bit0: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupts.
0: Timer 3 interrupt priority determined by default priority order.
1: Timer 3 interrupt set to high priority level.

12.4. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the external peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the system clock is stopped. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. Figure 12.15 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

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

12.4.1. 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 “13.8. Watchdog Timer Reset” on page 129** for more information on the use and configuration of the WDT.

12.4.2. 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 oscillator 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 12.15. PCON: Power Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SMOD0	SSTAT0	Reserved	SMOD1	SSTAT1	Reserved	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x87

Bit7:	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.
Bit6:	SSTAT0: UART0 Enhanced Status Mode Select. This bit controls the access mode of the SM20-SM00 bits in register SCON0. 0: Reads/writes of SM20-SM00 access the SM20-SM00 UART0 mode setting. 1: Reads/writes of SM20-SM00 access the Framing Error (FE0), RX Overrun (RXOV0), and TX Collision (TXCOL0) status bits.
Bit5:	Reserved. Read is undefined. Must write 0.
Bit4:	SMOD1: UART1 Baud Rate Doubler Enable. This bit enables/disables the divide-by-two function of the UART1 baud rate logic for configurations described in the UART1 section. 0: UART1 baud rate divide-by-two enabled. 1: UART1 baud rate divide-by-two disabled.
Bit3:	SSTAT1: UART1 Enhanced Status Mode Select. This bit controls the access mode of the SM21-SM01 bits in SCON1. 0: Reads/writes of SM21-SM01 access the SM21-SM01 UART1 mode setting. 1: Reads/writes of SM21-SM01 access the Framing Error (FE1), RX Overrun (RXOV1), and TX Collision (TXCOL1) status bits.
Bit2:	Reserved. Read is undefined. Must write 0.
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.)

13. RESET SOURCES

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External port pins are forced to a known state
- Interrupts and timers are disabled.

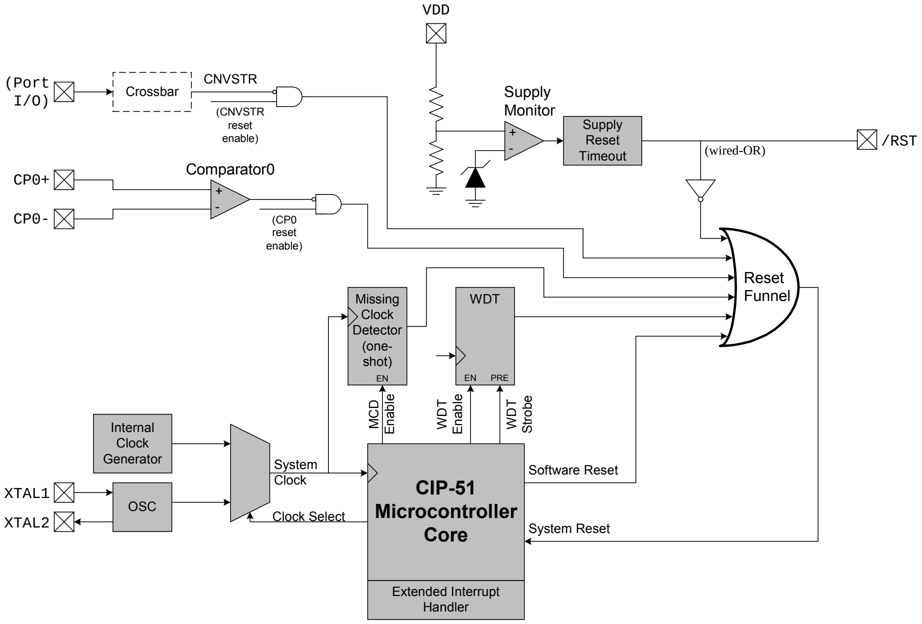
All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost even though the data on the stack 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. Note that weak pull-ups are disabled during the reset, and enabled when the device exits 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 2 MHz. Refer to Section "**14. OSCILLATORS**" on page **135** for information on selecting and configuring the system clock source. The Watchdog Timer is enabled using its longest timeout interval (see Section "**13.8. Watchdog Timer Reset**" on page **129**). 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 CNVSTR signal, software command, Comparator0, Missing Clock Detector, and Watchdog Timer. Each reset source is described in the following sections.

Figure 13.1. Reset Sources


VDD
(Port I/O)
Crossbar
CNVSTR
(CNVSTR reset enable)
Supply Monitor
Supply Reset Timeout
(wired-OR)
/RST
CP0+
CP0-
Comparator0
(CP0 reset enable)
Reset Funnel
Missing Clock Detector (one-shot)
EN
WDT
EN
PRE
MCD Enable
WDT Enable
WDT Strobe
Internal Clock Generator
XTAL1
XTAL2
OSC
System Clock
Clock Select
CIP-51 Microcontroller Core
Software Reset
System Reset
Extended Interrupt Handler

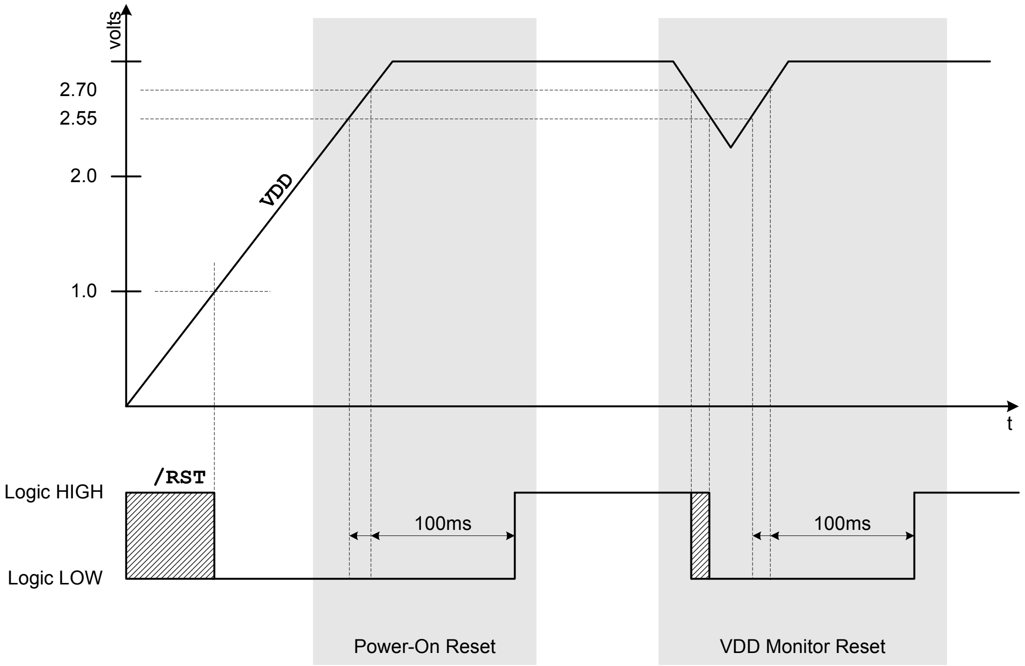
13.1. Power-on Reset

The C8051F020/1/2/3 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 13.2 for timing diagram, and refer to Table 13.1 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.

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.

The VDD monitor function is enabled by tying the MONEN pin directly to VDD. This is the recommended configuration for the MONEN pin.

.



13.2. 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 13.2). 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.

13.3. External Reset

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

13.4. Software Forced Reset

Writing a ‘1’ to the SWRSEF bit forces a Software Reset as described in **Section 13.1**.

13.5. 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 MSCLKE bit in the OSCICN register (see Section “**14. OSCILLATORS**” on page **135**) enables the Missing Clock Detector.

13.6. 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 “**11. COMPARATORS**” on page **95**) prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (CP0+ pin) is less than the inverting input voltage (CP0- pin), the MCU is put into the reset state. After a Comparator0 Reset, the C0RSEF flag (RSTSRC.5) will read ‘1’ signifying Comparator0 as the reset source; otherwise, this bit reads ‘0’. The state of the /RST pin is unaffected by this reset.

13.7. External CNVSTR Pin Reset

The external CNVSTR signal can be configured as a reset input by writing a ‘1’ to the CNVRSEF flag (RSTSRC.6). The CNVSTR signal can appear on any of the P0, P1, P2 or P3 I/O pins as described in Section “**17.1. Ports 0 through 3 and the Priority Crossbar Decoder**” on page **163**. Note that the Crossbar must be configured for the CNVSTR signal to be routed to the appropriate Port I/O. The Crossbar should be configured and enabled before the CNVRSEF is set. When configured as a reset, CNVSTR is active-low and level sensitive. After a CNVSTR reset, the CNVRSEF flag (RSTSRC.6) will read ‘1’ signifying CNVSTR as the reset source; otherwise, this bit reads ‘0’. The state of the /RST pin is unaffected by this reset.

13.8. 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/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 13.3.

13.8.1. Enable/Reset WDT

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

13.8.2. 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.

13.8.3. Disable WDT Lockout

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

13.8.4. 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 2 MHz system clock, this provides an interval range of 0.032 ms to 524 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 13.3. 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

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 13.4. RSTSRC: Reset Source Register

R	R/W	R/W	R/W	R	R	R/W	R	Reset Value
-	CNVRSEF	C0RSEF	SWRSEF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xEF

(Note: Do not use read-modify-write operations on this register.)

Bit7: Reserved.

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

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 prior reset was not Comparator0.
1: Source of prior reset was Comparator0.

Bit4: SWRSF: Software Reset Force and Flag
Write: 0: No Effect.
1: Forces an internal reset. /RST pin is not affected.
Read: 0: Prior reset source was not a write to the SWRSF bit.
1: Prior reset source was a write to the SWRSF bit.

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

Bit2: MCDRSF: Missing Clock Detector Flag
0: Source of prior reset was not Missing Clock Detector timeout.
1: Source of prior reset was Missing Clock Detector timeout.

Bit1: PORSF: Power-On Reset Force and Flag
Write: 0: No effect.
1: Forces a Power-On Reset. /RST is driven low.
Read: 0: Source of prior reset was not POR.
1: Source of prior reset was POR.

Bit0: PINRSF: HW Pin Reset Flag
0: Source of prior reset was not /RST pin.
1: Source of prior reset was /RST pin.

Table 13.1. Reset Electrical Characteristics

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

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
/RST Output High Voltage	I_{OH} = -3 mA	VDD - 0.7			V
/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 Timeout	Time from last system clock to reset initiation	100	220	500	µs

Notes

14. OSCILLATORS

Each MCU includes an internal oscillator and an external oscillator drive circuit, either of which can generate the system clock. The MCUs operate from the internal oscillator after any reset. This internal oscillator can be enabled/disabled and its frequency can be set using the Internal Oscillator Control Register (OSCICN) as shown in Figure 14.1. The internal oscillator's electrical specifications are given in Table 14.1.

Both oscillators are disabled when the /RST pin is held low. The MCUs can run from the internal oscillator permanently, or can switch to the external oscillator if desired using CLKSL bit in the OSCICN Register. The external oscillator requires an external resonator, crystal, capacitor, or RC network connected to the XTAL1/XTAL2 pins (see Table 14.1). The oscillator circuit must be configured for one of these sources in the OSCXCN register. An external CMOS clock can also provide the system clock; in this configuration, the XTAL1 pin is used as the CMOS clock input. The XTAL1 and XTAL2 pins are NOT 5V tolerant.

Figure 14.1. Oscillator Diagram

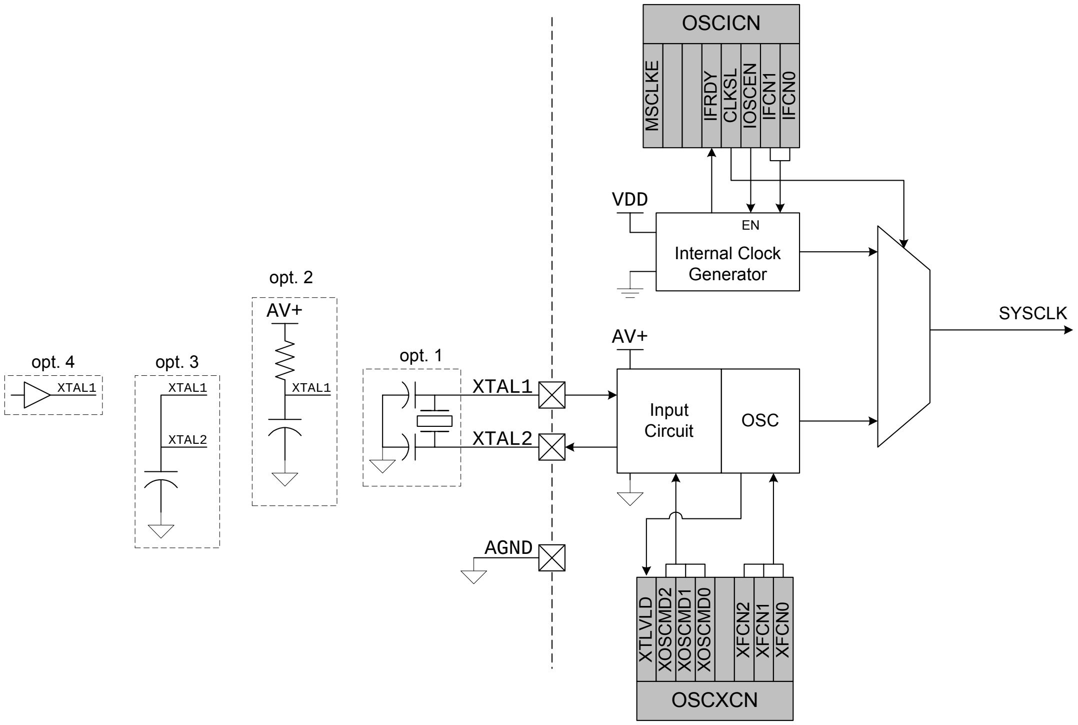

OSCICN
MSCLKE
IFRDY
CLKSL
IOSCEN
IFCN1
IFCN0
VDD
EN
Internal Clock Generator
SYSCLK
opt. 2
AV+
XTAL1
opt. 4
XTAL1
opt. 3
XTAL1
XTAL2
opt. 1
XTAL1
XTAL2
AV+
Input Circuit
OSC
AGND
XTLVLD
XOSCMD2
XOSCMD1
XOSCMD0
XFCN2
XFCN1
XFCN0
OSCXCN

Figure 14.2. OSCICN: Internal Oscillator Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
MSCLKE	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	00010100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

Bit7: MSCLKE: Missing Clock Enable Bit
0: Missing Clock Detector Disabled
1: Missing Clock Detector Enabled; reset triggered if clock is missing for more than 100 µs
Bits6-5: UNUSED. Read = 00b, Write = don't care
Bit4: IFRDY: Internal Oscillator Frequency Ready Flag
0: Internal Oscillator Frequency not running at speed specified by the IFCN bits.
1: Internal Oscillator Frequency running at speed specified by the IFCN bits.
Bit3: CLKSL: System Clock Source Select Bit
0: Uses Internal Oscillator as System Clock.
1: Uses External Oscillator as System Clock.
Bit2: IOSCEN: Internal Oscillator Enable Bit
0: Internal Oscillator Disabled
1: Internal Oscillator Enabled
Bits1-0: IFCN1-0: Internal Oscillator Frequency Control Bits
00: Internal Oscillator typical frequency is 2 MHz.
01: Internal Oscillator typical frequency is 4 MHz.
10: Internal Oscillator typical frequency is 8 MHz.
11: Internal Oscillator typical frequency is 16 MHz.

Table 14.1. Internal Oscillator Electrical Characteristics

VDD = 2.7V to 3.6V; T_a = -40°C to +85°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Internal Oscillator Frequency	OSCICN.[1:0] = 00	1.5	2	2.4	MHz
	OSCICN.[1:0] = 01	3.1	4	4.8	
	OSCICN.[1:0] = 10	6.2	8	9.6	
	OSCICN.[1:0] = 11	12.3	16	19.2	
Internal Oscillator Current Consumption (from VDD)	OSCICN.2 = 1		200		µA

Figure 14.3. OSCXCN: External Oscillator Control Register

R/W	R/W	R/W	R/W	R/W	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: 0xB1

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: Off. XTAL1 pin is grounded internally.
010: System Clock from External CMOS Clock on XTAL1 pin.
011: System Clock from External CMOS Clock on XTAL1 pin divided by 2.
10x: RC/C Oscillator Mode with divide by 2 stage.
110: Crystal Oscillator Mode
111: Crystal Oscillator Mode with divide by 2 stage.
Bit3: RESERVED. Read = undefined, Write = don't care
Bits2-0: XFCN2-0: External Oscillator Frequency Control Bits
000-111:

XFCN	Crystal (XOSCMD = 11x)	RC (XOSCMD = 10x)	C (XOSCMD = 10x)
000	f < 12 kHz	f < 25 kHz	K Factor = 0.44
001	12 kHz < f ≤ 30 kHz	25 kHz < f ≤ 50 kHz	K Factor = 1.4
010	30 kHz < f ≤ 95 kHz	50 kHz < f ≤ 100 kHz	K Factor = 4.4
011	95 kHz < f ≤ 270 kHz	100 kHz < f ≤ 200 kHz	K Factor = 13
100	270 kHz < f ≤ 720 kHz	200 kHz < f ≤ 400 kHz	K Factor = 38
101	720 kHz < f ≤ 2.2 MHz	400 kHz < f ≤ 800 kHz	K Factor = 100
110	2.2 MHz < f ≤ 6.7 MHz	800 kHz < f ≤ 1.6 MHz	K Factor = 420
111	f > 6.7 MHz	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1400

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

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

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

14.1. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be as shown in Figure 14.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in Figure 14.3 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

The Crystal Oscillator Valid Flag (XTLVLD in register OSCXCN) is set to logic 1 by hardware when the external crystal oscillator is running and stable. The XTLVLD detection circuit requires a startup time of at least 1 ms between enabling the oscillator and checking the XTLVLD bit. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

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

Important Note: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device, as should the loading capacitors on the crystal pins. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

14.2. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be as shown in Figure 14.1, Option 2. The capacitor must be no greater than 100 pF; however for small capacitors (less than ~20 pF), the total capacitance may be dominated by PWB parasitic capacitance. 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:

$$f = 1.23(10^3) / RC = 1.23 (10^3) / [246 * 50] = 0.1 \text{ MHz} = 100 \text{ kHz}$$

$$\text{XFCN} \geq \log_2 (f / 25 \text{ kHz})$$
$$\text{XFCN} \geq \log_2 (100 \text{ kHz} / 25 \text{ kHz}) = \log_2 (4)$$
$$\text{XFCN} \geq 2, \text{ or code } 010\text{b}$$

14.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be as shown in Figure 14.1, Option 3. The capacitor must be no greater than 100 pF; however for small capacitors (less than ~20 pF), the total capacitance may be dominated by PWB parasitic capacitance. 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 90 kHz is desired, select the K Factor from the table in Figure 14.3 as KF = 13:

$$f = 13 / 150 = 0.087 \text{ MHz, or } 87 \text{ kHz}$$

Therefore, the XFCN value to use in this example is 011b.

15. FLASH MEMORY

The C8051F020/1/2/3 family includes 64k + 128 bytes of on-chip, reprogrammable FLASH memory for program code and non-volatile data storage. The FLASH memory can be programmed in-system, a single byte at a time, through the JTAG interface or by software. 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. Refer to Table 15.1 for the electrical characteristics of the FLASH memory.

15.1. Programming The FLASH Memory

The simplest means of programming the FLASH memory is through the JTAG interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the JTAG commands to program FLASH memory, see **Section “24.2. Flash Programming Commands” on page 268**.

The FLASH memory can be programmed by software using a MOVX write instruction, with the address and data byte to be programmed provided as normal operands. Before writing to FLASH memory using a MOVX write, 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 XRAM. 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 read instruction. MOVX reads are always directed to XRAM, regardless of the state of PSWE.

To ensure the integrity of FLASH contents, it is strongly recommended that the on-chip VDD monitor be enabled by tying the MONEN pin to VDD in any system which includes code that writes to or erases FLASH memory from software.

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 64k byte 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 by 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 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. Interrupts that are posted during a FLASH write or erase operation are held pending until the FLASH operation has completed, at which time they are serviced by the CPU in priority order.

Table 15.1. FLASH Electrical Characteristics

VDD = 2.7V to 3.6V; T_a = -40°C to +85°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Endurance		20k	100k		Erase/Write
Erase Cycle Time		10	12	14	ms
Write Cycle Time		40	50	60	µs

15.2. 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 read 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 64k byte FLASH memory; its address ranges from 0x00 to 0x7F (see Figure 15.1). 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 permitted.

15.3. 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 modify 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.

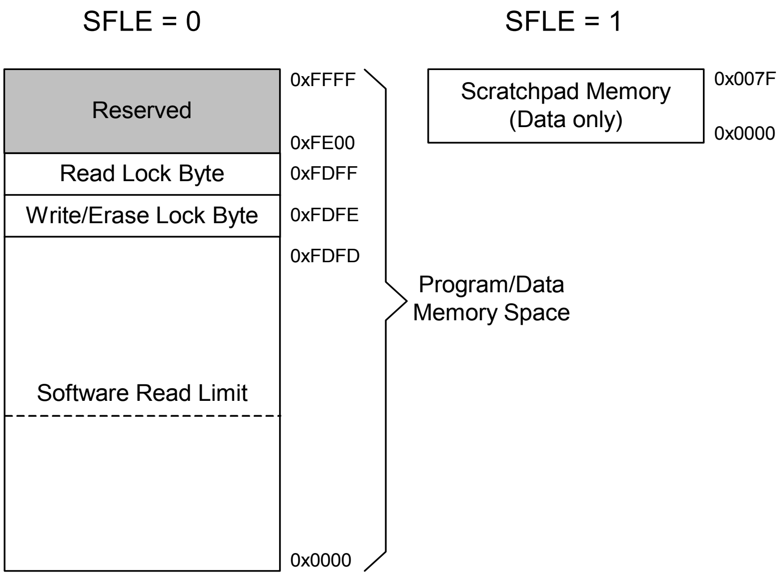
A set of security lock bytes stored at 0xFDFE and 0xFDFF protect the FLASH program memory from being read or altered across the JTAG interface. Each bit in a security lock-byte protects one 8k-byte block of memory. Clearing a bit to logic 0 in a Read Lock Byte prevents the corresponding block of FLASH 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 128-byte scratchpad sector is locked only when all other sectors are locked.

The Read Lock Byte is at location 0xFDFF. The Write/Erase Lock Byte is located at 0xFDFE. Figure 15.1 shows the location and bit definitions of the security bytes. 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 port.

Figure 15.1. FLASH Program Memory Map and Security Bytes

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

Bit	Memory Block
7	0xE000 - 0xFDFD
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 highest block is locked, the security bytes may be written but not erased.

FLASH access Limit Register (FLACL)

The content of this register is used as the high byte of the 16-bit software read limit address. This 16-bit read limit address value is calculated as 0xNN00 where NN is replaced by content of this register on reset. Software 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. Any attempts to read locations below this limit will return the value 0x00.

The lock bits can always be read and cleared 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: The only means of removing a lock once set is to erase the entire program memory space by performing a JTAG erase operation (i.e. cannot be done in user firmware). Addressing either security byte while performing a JTAG erase operation will automatically initiate erasure of the entire program memory space (except for the reserved area). This erasure can only be performed via JTAG. If a non-security byte in the 0xFBFF-0xFDFF page is addressed during the JTAG erasure, only that page (including the security bytes) will be erased.**

The FLASH Access Limit security feature (see Figure 15.1) protects proprietary program code and data from being read by software running on the C8051F020/1/2/3. 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 Software Read Limit (SRL) 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 SRL address, and the second is a lower partition consisting of all the program memory locations starting at 0x0000 up to (but excluding) the SRL 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 SRL address is specified using the contents of the FLASH Access Register. The 16-bit SRL address is calculated as 0xNN00, where NN is the contents of the SRL Security Register. Thus, the SRL 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 SRL security byte is 0x00, thereby setting the SRL address to 0x0000 and allowing read access to all locations in program memory space by default.

Figure 15.2. 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:
								0xB7

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.

Figure 15.3. 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: 0xB6

Bit7: 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.

Bit6: FRAE: FLASH Read Always Enable
0: FLASH reads per One-Shot Timer.
1: FLASH always in read mode.

Bits5-1: RESERVED. Read = 00000b. Must Write 00000b.

Bit0: FLWE: FLASH Read/Write Enable
This bit must be set to allow FLASH writes from user software.
0: FLASH writes disabled.
1: FLASH writes enabled.

Figure 15.4. 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: 0x8F

Bits7-3: UNUSED. Read = 00000b, Write = don't care.

Bit2: SFLE: Scratchpad FLASH Memory Access Enable.
When this bit is set, FLASH 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 unpredictable results.
0: FLASH access from user software directed to the 64k byte Program/Data FLASH sector.
1: FLASH access from user software directed to the 128 byte Scratchpad sector.

Bit1: 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.
0: FLASH program memory erasure disabled.
1: FLASH program memory erasure enabled.

Bit0: PSWE: Program Store Write Enable.
Setting this bit allows writing a byte of data to the FLASH program memory using the MOVX instruction. The location must be erased before writing data.
0: Write to FLASH program memory disabled.
1: Write to FLASH program memory enabled.

16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM

The C8051F020/1/2/3 MCUs include 4k bytes of on-chip RAM mapped into the external data memory space (XRAM), as well as an External Data Memory Interface which can be used to access off-chip memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in Figure 16.1). Note: the MOVX instruction can also be used for writing to the FLASH memory. See **Section “15. FLASH MEMORY” on page 139** for details. The MOVX instruction accesses XRAM by default. The EMIF can be configured to appear on the lower I/O ports (P0-P3) or the upper I/O ports (P4-P7).

16.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read 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.

16.1.1. 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.

16.1.2. 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
```

16.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of four steps:

1. Select EMIF on Low Ports (P3, P2, P1, and P0) or High Ports (P7, P6, P5, and P4).
2. Select Multiplexed mode or Non-multiplexed mode.
3. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
4. Set up timing to interface with off-chip memory or peripherals.
5. Select the desired output mode for the associated Ports (registers PnMDOUT, P74OUT).

Each of these four steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in Figure 16.2.

16.3. Port Selection and Configuration

The External Memory Interface can appear on Ports 3, 2, 1, and 0 (C8051F020/1/2/3 devices) or on Ports 7, 6, 5, and 4 (C8051F020/2 devices only), depending on the state of the PRTSEL bit (EMI0CF.5). If the lower Ports are selected, the EMIFLE bit (XBR2.1) must be set to a ‘1’ so that the Crossbar will skip over P0.7 (/WR), P0.6 (/RD), and if multiplexed mode is selected P0.5 (ALE). For more information about the configuring the Crossbar, see **Section “17. PORT INPUT/OUTPUT” on page 161**.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar (on Ports 3, 2, 1, and 0). See **Section “17. PORT INPUT/OUTPUT” on page 161** for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to ‘park’ the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. See **Section “17. PORT INPUT/OUTPUT” on page 161** for more information about Port output mode configuration.

Figure 16.1. 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: 0xAF

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 16.2. 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

Bits7-6: Unused. Read = 00b. Write = don't care.
Bit5: PRTSEL: EMIF Port Select.
0: EMIF active on P0-P3.
1: EMIF active on P4-P7.
Bit4: EMD2: EMIF Multiplex Mode Select.
0: EMIF operates in multiplexed address/data mode.
1: EMIF operates in non-multiplexed mode (separate address and data pins).
Bits3-2: EMD1-0: EMIF Operating Mode Select.
These bits control the operating mode of the External Memory Interface.
00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space.
01: Split Mode without Bank Select: Accesses below the 4k boundary are directed on-chip. Accesses above the 4k 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 4k boundary are directed on-chip. Accesses above the 4k 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.

16.4. 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.

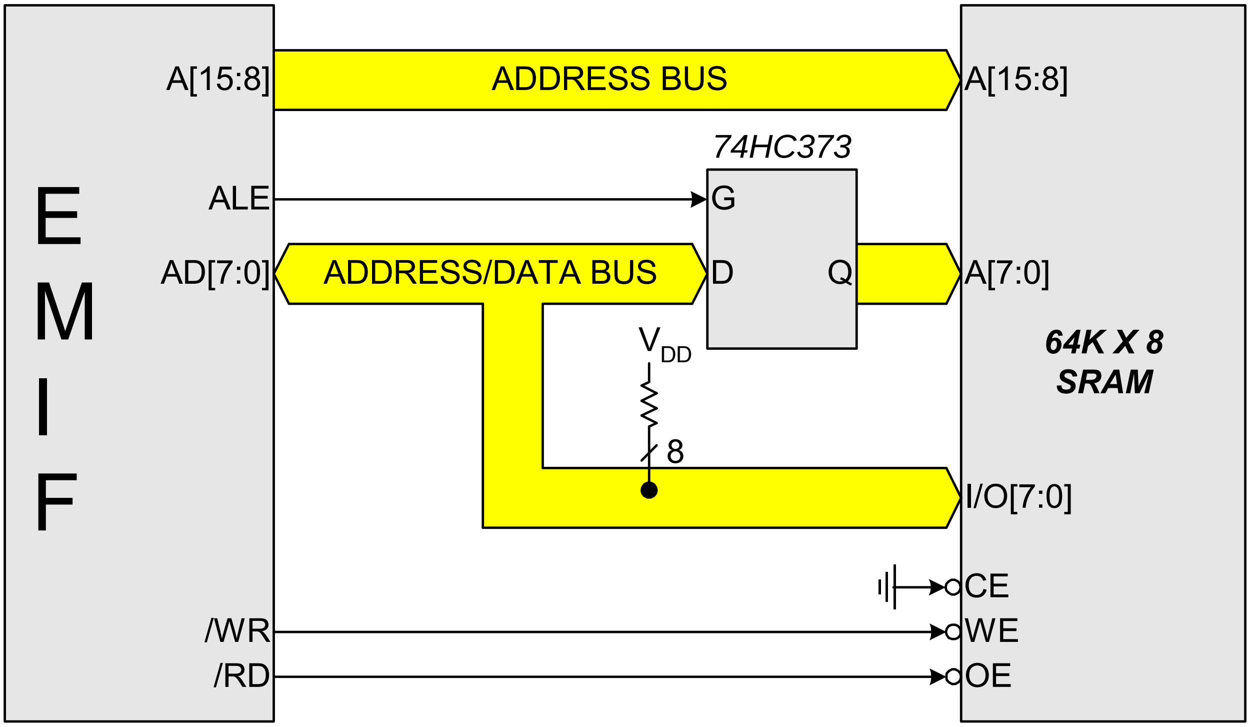
16.4.1. Multiplexed Configuration

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 16.3.

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 "16.6.2. Multiplexed Mode" on page 156** for more information.

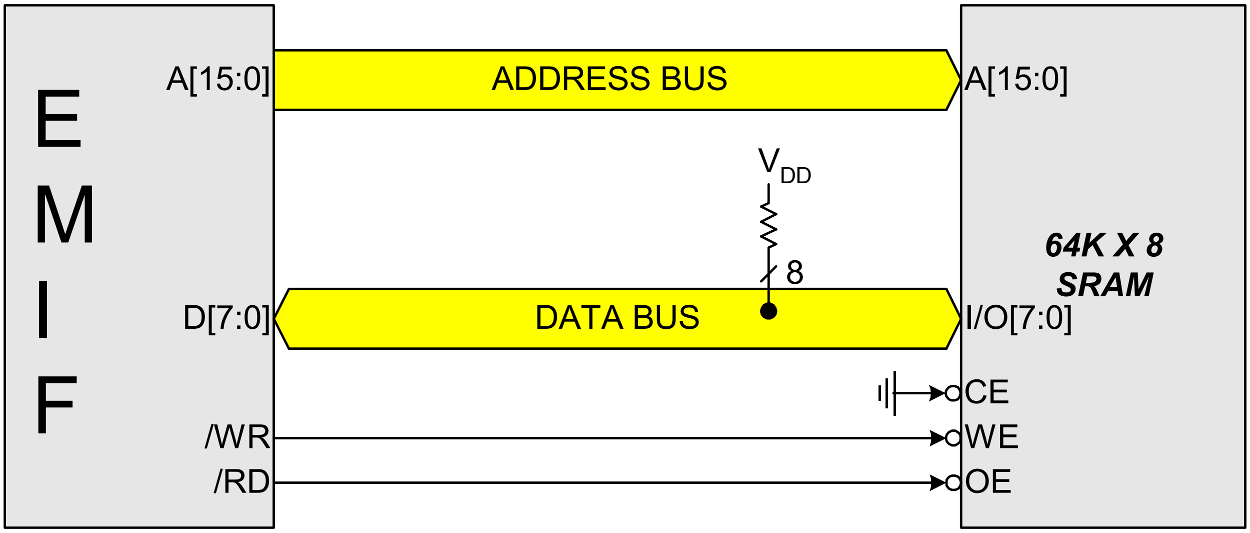
Figure 16.3. Multiplexed Configuration Example



16.4.2. 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 16.4. See **Section "16.6.1. Non-multiplexed Mode" on page 153** for more information about Non-multiplexed operation.

Figure 16.4. Non-multiplexed Configuration Example


E
M
I
F
A[15:0]
ADDRESS BUS
A[15:0]
V_{DD}
8
64K X 8
SRAM
D[7:0]
DATA BUS
I/O[7:0]
CE
/WR
WE
/RD
OE

16.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 16.5, based on the EMIF Mode bits in the EMI0CF register (Figure 16.2). These modes are summarized below. More information about the different modes can be found in **Section “ .” on page 152**.

16.5.1. 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 4k 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.

16.5.2. 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 4k boundary will access on-chip XRAM space.
- Effective addresses beyond the 4k 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 lower 8-bits of the Address Bus A[7:0] are driven as defined by R0 or R1. 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

16.5.3. 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 4k boundary will access on-chip XRAM space.
- Effective addresses beyond the 4k 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.

16.5.4. 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 4k 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.

16.6. 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 16.6, 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 of an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 SYSCLKs for /ALE, 1 for /RD or /WR + 4 SYSCLKs). The programmable setup and hold times default to the maximum delay settings after a reset.

Table 16.1 lists the AC parameters for the External Memory Interface, and Figure 16.7 through Figure 16.11 show the timing diagrams for the different External Memory Interface modes and MOVX operations

.

Figure 16.6. EMI0TC: External Memory Timing Control

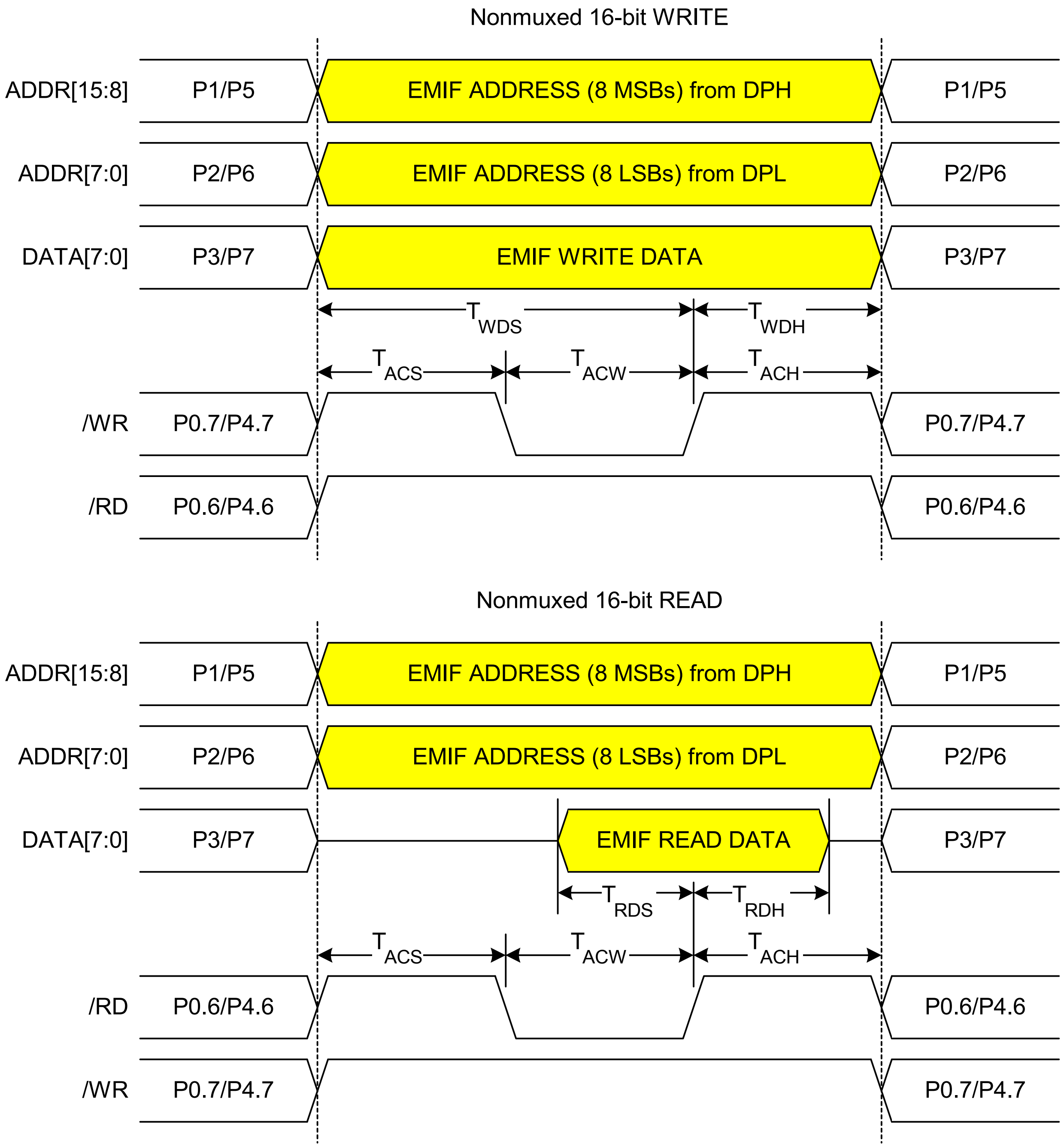
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EAS1	EAS0	EWR3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xA1

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.

16.6.1. Non-multiplexed Mode

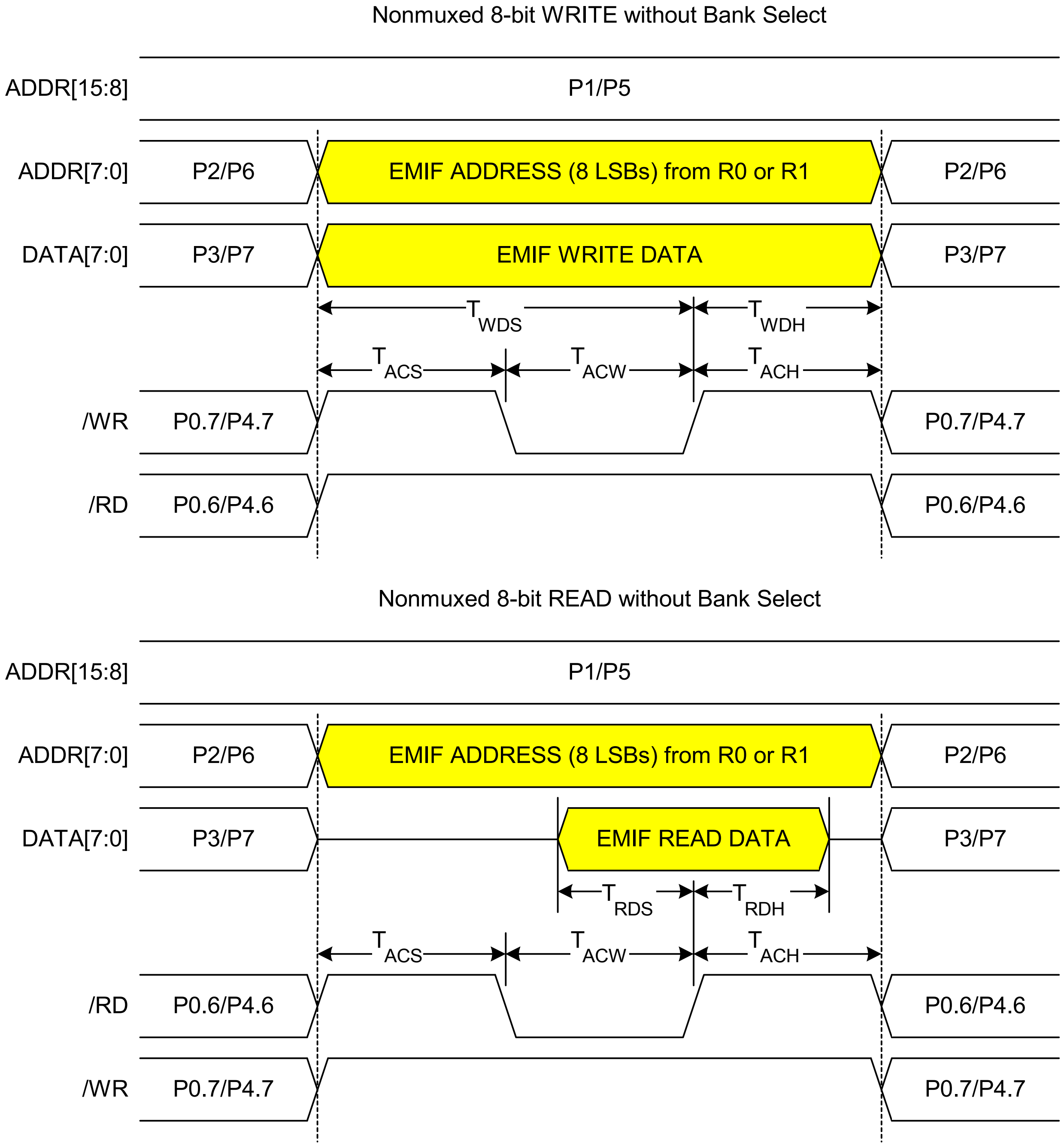
16.6.1.1. 16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’.

Figure 16.7. Non-multiplexed 16-bit MOVX Timing


Nonmuxed 16-bit WRITE
ADDR[15:8]
P1/P5
EMIF ADDRESS (8 MSBs) from DPH
P1/P5
ADDR[7:0]
P2/P6
EMIF ADDRESS (8 LSBs) from DPL
P2/P6
DATA[7:0]
P3/P7
EMIF WRITE DATA
P3/P7
T_WDS
T_WDH
T_ACS
T_ACW
T_ACH
/WR
P0.7/P4.7
P0.7/P4.7
/RD
P0.6/P4.6
P0.6/P4.6
Nonmuxed 16-bit READ
ADDR[15:8]
P1/P5
EMIF ADDRESS (8 MSBs) from DPH
P1/P5
ADDR[7:0]
P2/P6
EMIF ADDRESS (8 LSBs) from DPL
P2/P6
DATA[7:0]
P3/P7
EMIF READ DATA
P3/P7
T_RDS
T_RDH
T_ACS
T_ACW
T_ACH
/RD
P0.6/P4.6
P0.6/P4.6
/WR
P0.7/P4.7
P0.7/P4.7

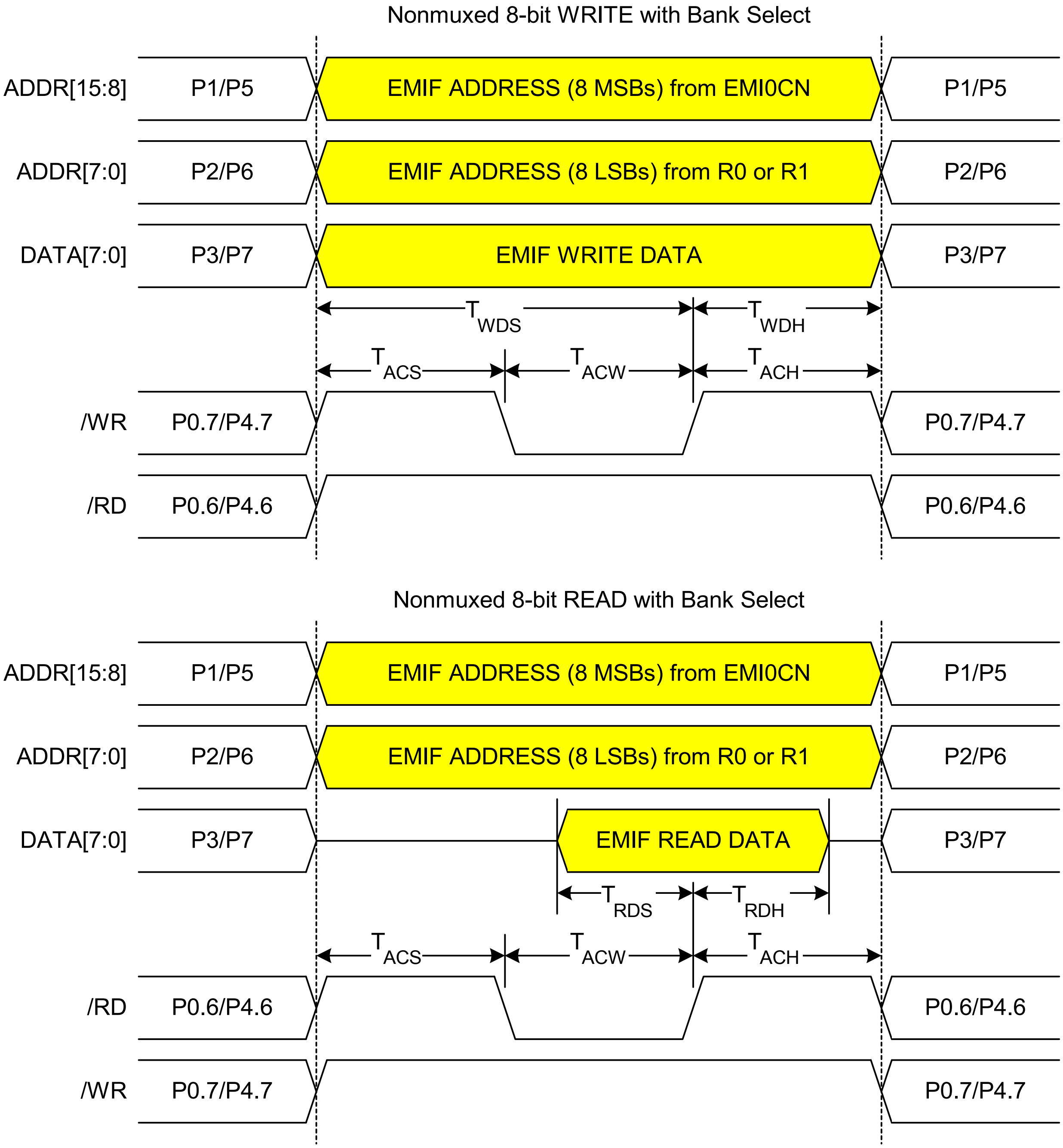
16.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘101’ or ‘111’.

Figure 16.8. Non-multiplexed 8-bit MOVX without Bank Select Timing


Nonmuxed 8-bit WRITE without Bank Select
ADDR[15:8]
P1/P5
ADDR[7:0]
P2/P6
EMIF ADDRESS (8 LSBs) from R0 or R1
P2/P6
DATA[7:0]
P3/P7
EMIF WRITE DATA
P3/P7
T_WDS
T_WDH
T_ACS
T_ACW
T_ACH
/WR
P0.7/P4.7
P0.7/P4.7
/RD
P0.6/P4.6
P0.6/P4.6
Nonmuxed 8-bit READ without Bank Select
ADDR[15:8]
P1/P5
ADDR[7:0]
P2/P6
EMIF ADDRESS (8 LSBs) from R0 or R1
P2/P6
DATA[7:0]
P3/P7
EMIF READ DATA
P3/P7
T_RDS
T_RDH
T_ACS
T_ACW
T_ACH
/RD
P0.6/P4.6
P0.6/P4.6
/WR
P0.7/P4.7
P0.7/P4.7

16.6.1.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = '110'.

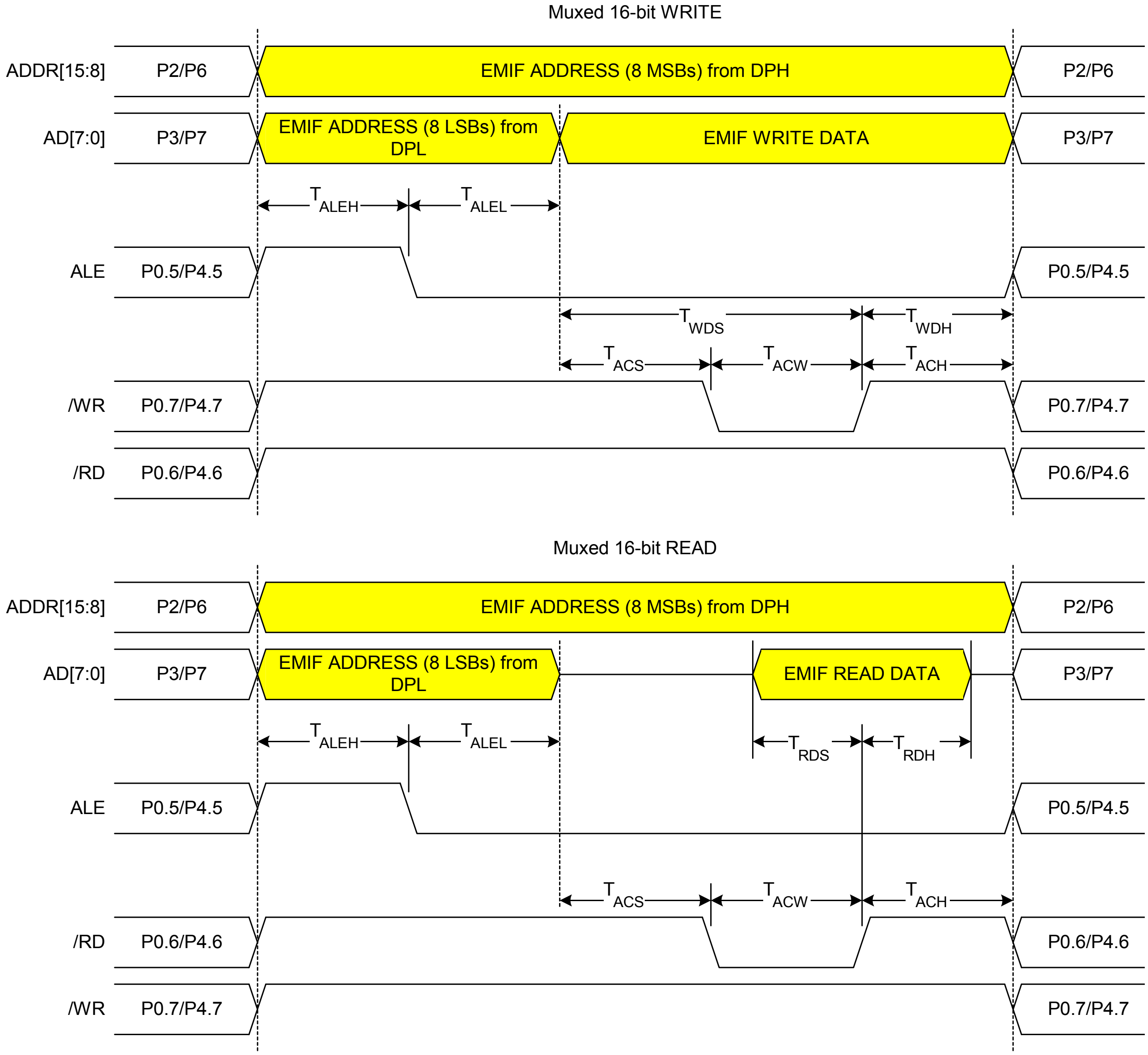
Figure 16.9. Non-multiplexed 8-bit MOVX with Bank Select Timing


Nonmuxed 8-bit WRITE with Bank Select
ADDR[15:8]
P1/P5
EMIF ADDRESS (8 MSBs) from EMI0CN
P1/P5
ADDR[7:0]
P2/P6
EMIF ADDRESS (8 LSBs) from R0 or R1
P2/P6
DATA[7:0]
P3/P7
EMIF WRITE DATA
P3/P7
T_WDS
T_WDH
T_ACS
T_ACW
T_ACH
/WR
P0.7/P4.7
P0.7/P4.7
/RD
P0.6/P4.6
P0.6/P4.6
Nonmuxed 8-bit READ with Bank Select
ADDR[15:8]
P1/P5
EMIF ADDRESS (8 MSBs) from EMI0CN
P1/P5
ADDR[7:0]
P2/P6
EMIF ADDRESS (8 LSBs) from R0 or R1
P2/P6
DATA[7:0]
P3/P7
EMIF READ DATA
P3/P7
T_RDS
T_RDH
T_ACS
T_ACW
T_ACH
/RD
P0.6/P4.6
P0.6/P4.6
/WR
P0.7/P4.7
P0.7/P4.7

16.6.2. Multiplexed Mode

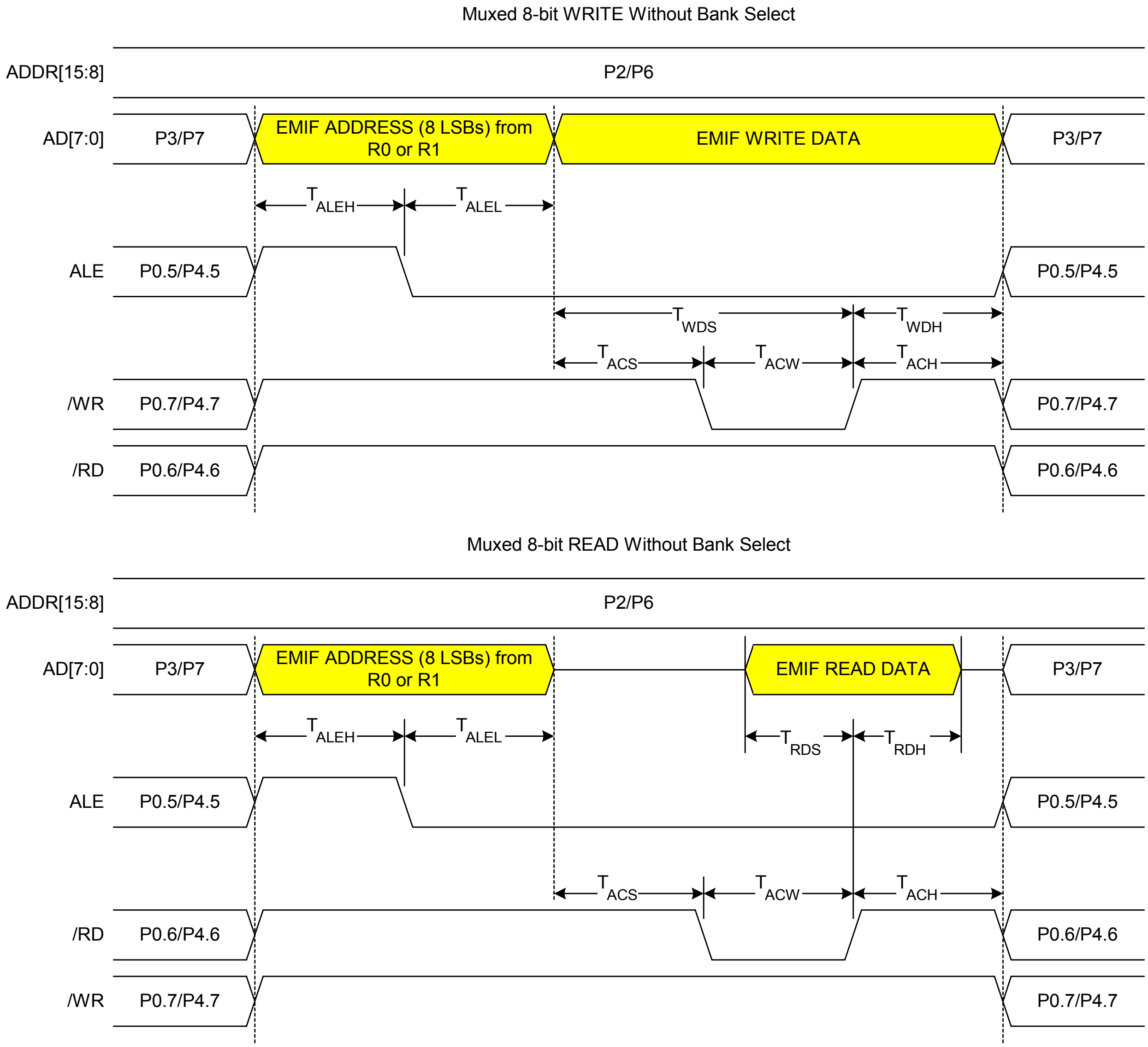
16.6.2.1. 16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

Figure 16.10. Multiplexed 16-bit MOVX Timing


Muxed 16-bit WRITE
ADDR[15:8]
P2/P6
EMIF ADDRESS (8 MSBs) from DPH
P2/P6
AD[7:0]
P3/P7
EMIF ADDRESS (8 LSBs) from DPL
EMIF WRITE DATA
P3/P7
T_ALEH
T_ALEL
ALE
P0.5/P4.5
P0.5/P4.5
T_WDS
T_WDH
T_ACS
T_ACW
T_ACH
/WR
P0.7/P4.7
P0.7/P4.7
/RD
P0.6/P4.6
P0.6/P4.6
Muxed 16-bit READ
ADDR[15:8]
P2/P6
EMIF ADDRESS (8 MSBs) from DPH
P2/P6
AD[7:0]
P3/P7
EMIF ADDRESS (8 LSBs) from DPL
EMIF READ DATA
P3/P7
T_ALEH
T_ALEL
T_RDS
T_RDH
ALE
P0.5/P4.5
P0.5/P4.5
T_ACS
T_ACW
T_ACH
/RD
P0.6/P4.6
P0.6/P4.6
/WR
P0.7/P4.7
P0.7/P4.7

16.6.2.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

Figure 16.11. Multiplexed 8-bit MOVX without Bank Select Timing


Muxed 8-bit WRITE Without Bank Select
ADDR[15:8]
P2/P6
AD[7:0]
P3/P7
EMIF ADDRESS (8 LSBs) from R0 or R1
EMIF WRITE DATA
P3/P7
T ALEH
T ALEL
ALE
P0.5/P4.5
P0.5/P4.5
T WDS
T WDH
T ACS
T ACW
T ACH
/WR
P0.7/P4.7
P0.7/P4.7
/RD
P0.6/P4.6
P0.6/P4.6
Muxed 8-bit READ Without Bank Select
ADDR[15:8]
P2/P6
AD[7:0]
P3/P7
EMIF ADDRESS (8 LSBs) from R0 or R1
EMIF READ DATA
P3/P7
T ALEH
T ALEL
T RDS
T RDH
ALE
P0.5/P4.5
P0.5/P4.5
T ACS
T ACW
T ACH
/RD
P0.6/P4.6
P0.6/P4.6
/WR
P0.7/P4.7
P0.7/P4.7

[redacted]

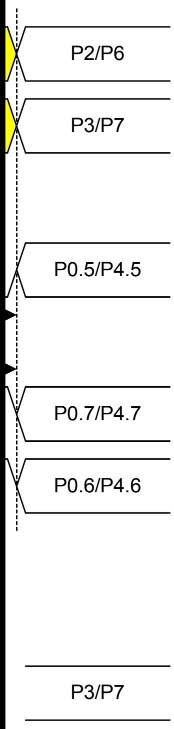

P2/P6
P3/P7
P0.5/P4.5
P0.7/P4.7
P0.6/P4.6
P3/P7

Table 16.1. 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

Notes

17. PORT INPUT/OUTPUT

The C8051F020/1/2/3 are fully integrated mixed-signal System on a Chip MCUs with 64 digital I/O pins (C8051F020/2) or 32 digital I/O pins (C8051F021/3), organized as 8-bit Ports. The lower ports: P0, P1, P2, and P3, are both bit- and byte-addressable through their corresponding Port Data registers. The upper ports: P4, P5, P6, and P7 are byte-addressable. All Port pins are 5 V-tolerant, and all support configurable Open-Drain or Push-Pull output modes and weak pull-ups. A block diagram of the Port I/O cell is shown in Figure 17.1. Complete Electrical Specifications for the Port I/O pins are given in Table 16.1.

Figure 17.1. Port I/O Cell Block Diagram

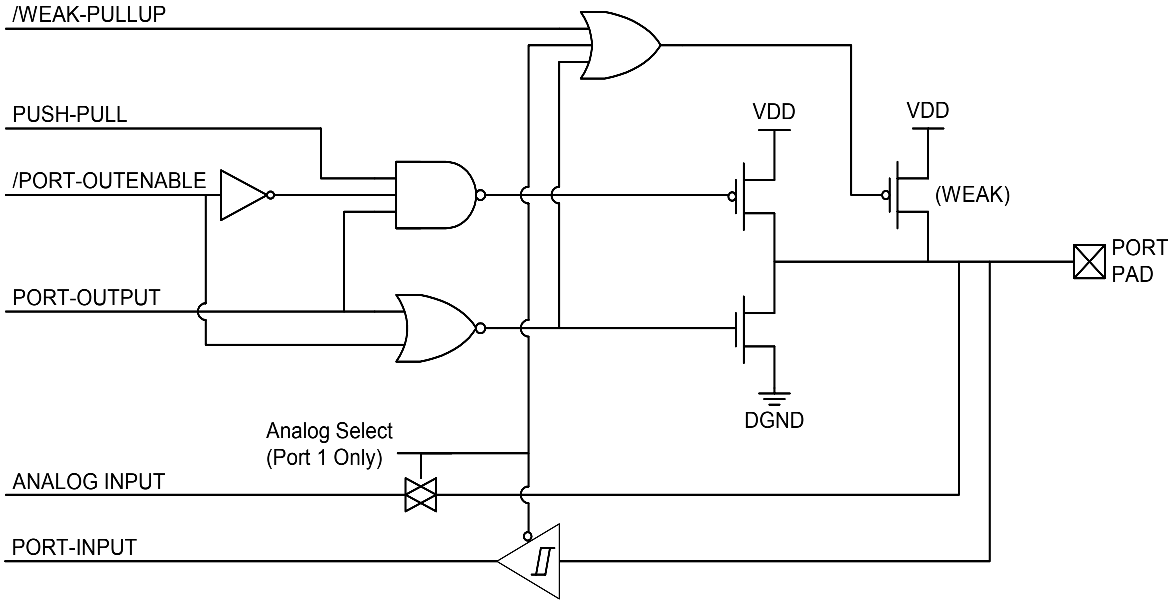


Table 17.1. Port I/O DC Electrical Characteristics

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

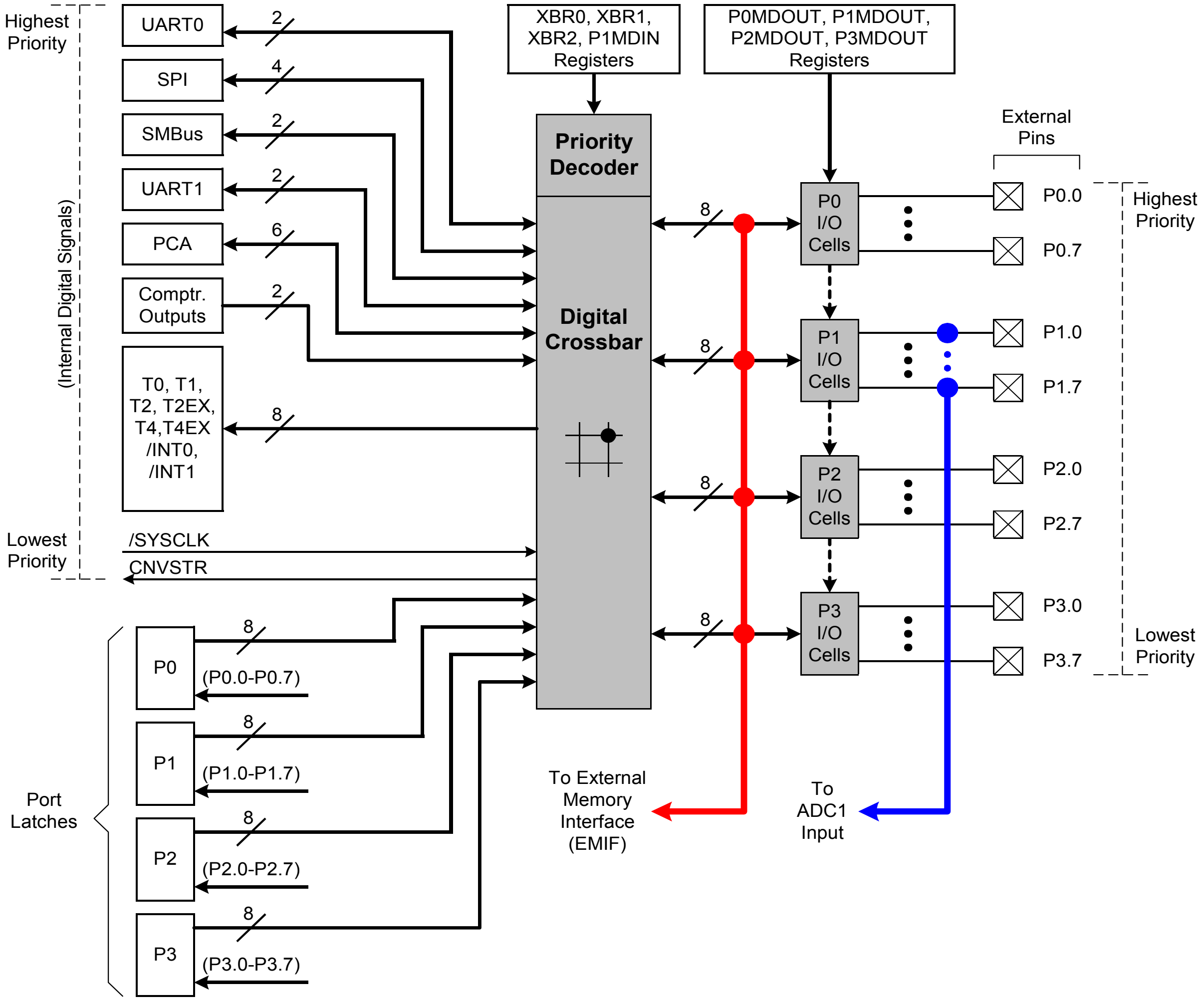
PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Output High Voltage (V_{OH})	I_{OH} = -10 µA, Port I/O Push-Pull I_{OH} = -3 mA, Port I/O Push-Pull I_{OH} = -10 mA, Port I/O Push-Pull	VDD - 0.1 VDD - 0.7	VDD - 0.8		V
Output Low Voltage (V_{OL})	I_{OL} = 10 µA I_{OL} = 8.5 mA I_{OL} = 25 mA		1.0	0.1 0.6	V
Input High Voltage (VIH)		0.7 x VDD			V
Input Low Voltage (VIL)				0.3 x VDD	V
Input Leakage Current	DGND < Port Pin < VDD, Pin Tri-state Weak Pull-up Off Weak Pull-up On		10	± 1	µA
Input Capacitance			5		pF

The C8051F020/1/2/3 devices have a wide array of digital resources which are available through the four lower I/O Ports: P0, P1, P2, and P3. Each of the pins on P0, P1, P2, and P3, can be defined as a General-Purpose I/O (GPIO) pin or can be controlled by a digital peripheral or function (like UART0 or /INT1 for example), as shown in Figure 17.2. The system designer controls which digital functions are assigned pins, limited only by the number of pins available. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read from its associated Data register regardless of whether that pin has been assigned to a digital peripheral or behaves as GPIO. The Port pins on Port 1 can be used as Analog Inputs to ADC1.

An External Memory Interface which is active during the execution of a MOVX instruction whose target address resides in off-chip memory can be active on either the lower Ports or the upper Ports. See **Section "16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM" on page 145** for more information about the External Memory Interface.

The upper Ports (available on C8051F020/2) can be byte-accessed as GPIO pins.

Figure 17.2. Lower Port I/O Functional Block Diagram


Highest Priority
UART0
2
SPI
4
SMBus
2
UART1
2
PCA
6
Comptr. Outputs
2
T0, T1, T2, T2EX, T4,T4EX /INT0, /INT1
8
(Internal Digital Signals)
Lowest Priority
/SYSCLK
CNVSTR
XBR0, XBR1, XBR2, P1MDIN Registers
P0MDOUT, P1MDOUT, P2MDOUT, P3MDOUT Registers
Priority Decoder
Digital Crossbar
8
8
8
8
P0 I/O Cells
P1 I/O Cells
P2 I/O Cells
P3 I/O Cells
External Pins
P0.0
P0.7
P1.0
P1.7
P2.0
P2.7
P3.0
P3.7
Highest Priority
Lowest Priority
Port Latches
P0
8
(P0.0-P0.7)
P1
8
(P1.0-P1.7)
P2
8
(P2.0-P2.7)
P3
8
(P3.0-P3.7)
To External Memory Interface (EMIF)
To ADC1 Input

17.1. Ports 0 through 3 and the Priority Crossbar Decoder

The Priority Crossbar Decoder, or “Crossbar”, allocates and assigns Port pins on Port 0 through Port 3 to the digital peripherals (UARTs, SMBus, PCA, Timers, etc.) on the device using a priority order. The Port pins are allocated in order starting with P0.0 and continue through P3.7 if necessary. The digital peripherals are assigned Port pins in a priority order which is listed in Figure 17.3, with UART0 having the highest priority and CNVSTR having the lowest priority.

17.1.1. Crossbar Pin Assignment and Allocation

The Crossbar assigns Port pins to a peripheral if the corresponding enable bits of the peripheral are set to a logic 1 in the Crossbar configuration registers XBR0, XBR1, and XBR2, shown in Figure 17.7, Figure 17.8, and Figure 17.9. 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 peripheral’s enable bits are not set to a logic 1, then its ports are not accessible at the Port pins of the device. Also note that the Crossbar assigns pins to all associated functions when a serial communication peripheral is selected (i.e. SMBus, SPI, UART). It would be impossible, for example, to assign TX0

Figure 17.3. Priority Crossbar Decode Table (EMIFLE = 0; P1MDIN = 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				●		●																											
SDA	●		●		●		●																										SMB0EN: XBR0.0
SCL		●		●		●		●																									
TX1	●		●		●		●		●																								UART1EN: XBR2.2
RX1		●		●		●		●	●	●																							
CEX0	●		●		●		●		●	●	●																						PCA0ME: XBR0.[5:3]
CEX1		●		●		●		●	●	●	●	●																					
CEX2			●		●		●		●	●	●	●	●																				
CEX3				●		●		●	●	●	●	●	●	●																			
CEX4					●		●		●	●	●	●	●	●	●																		
ECI	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●																	ECI0E: XBR0.6
CP0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●																CP0E: XBR0.7
CP1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●															CP1E: XBR1.0
T0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●														T0E: XBR1.1
/INT0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●													INT0E: XBR1.2
T1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●												T1E: XBR1.3
/INT1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●											INT1E: XBR1.4
T2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										T2E: XBR1.5
T2EX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									T2EXE: XBR1.6
T4	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								T4E: XBR2.3
T4EX	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							T4EXE: XBR2.4
/SYSCLK	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						SYSCKE: XBR1.7
CNVSTR	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					CNVSTE: XBR2.0
						ALE	/RD	/WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN1 Inputs/Non-muxed Addr H								Muxed Addr H/Non-muxed Addr L								Muxed Data/Non-muxed Data								

to a Port pin without assigning RX0 as well. Each combination of enabled peripherals results in a unique device pinout.

All Port pins on Ports 0 through 3 that are not allocated by the Crossbar can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See Figure 17.10, Figure 17.12, Figure 17.15, and Figure 17.17), 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, SET, and the bitwise MOV 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 (XBR2.6) 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.

17.1.2. 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 (XBR2.6) to a logic 1.

The output mode of each port pin can be configured as either Open-Drain or Push-Pull; the default state is Open-Drain. 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 17.11, Figure 17.14, Figure 17.16, and Figure 17.18). 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.

17.1.3. 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.

17.1.4. External Interrupts (IE6 and IE7)

In addition to the external interrupts /INT0 and /INT1, whose Port pins are allocated and assigned by the Crossbar, P3.6 and P3.7 can be configured to generate edge sensitive interrupts; these interrupts are configurable as falling- or rising-edge sensitive using the IE6CF (P3IF.2) and IE7CF (P3IF.3) bits. When an active edge is detected on P3.6 or P3.7, a corresponding External Interrupt flag (IE6 or IE7) will be set to a logic 1 in the P3IF register (See Figure 17.19). If the associated interrupt is enabled, an interrupt will be generated and the CPU will vector to the associated interrupt vector location. See **Section “12.3. Interrupt Handler” on page 116** for more information about interrupts.

17.1.5. 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.

17.1.6. Configuring Port 1 Pins as Analog Inputs (AIN1.[7:0])

The pins on Port 1 can serve as analog inputs to the ADC1 analog MUX. A Port pin is configured as an Analog Input by writing a logic 0 to the associated bit in the P1MDIN register (see Figure 17.13). 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.

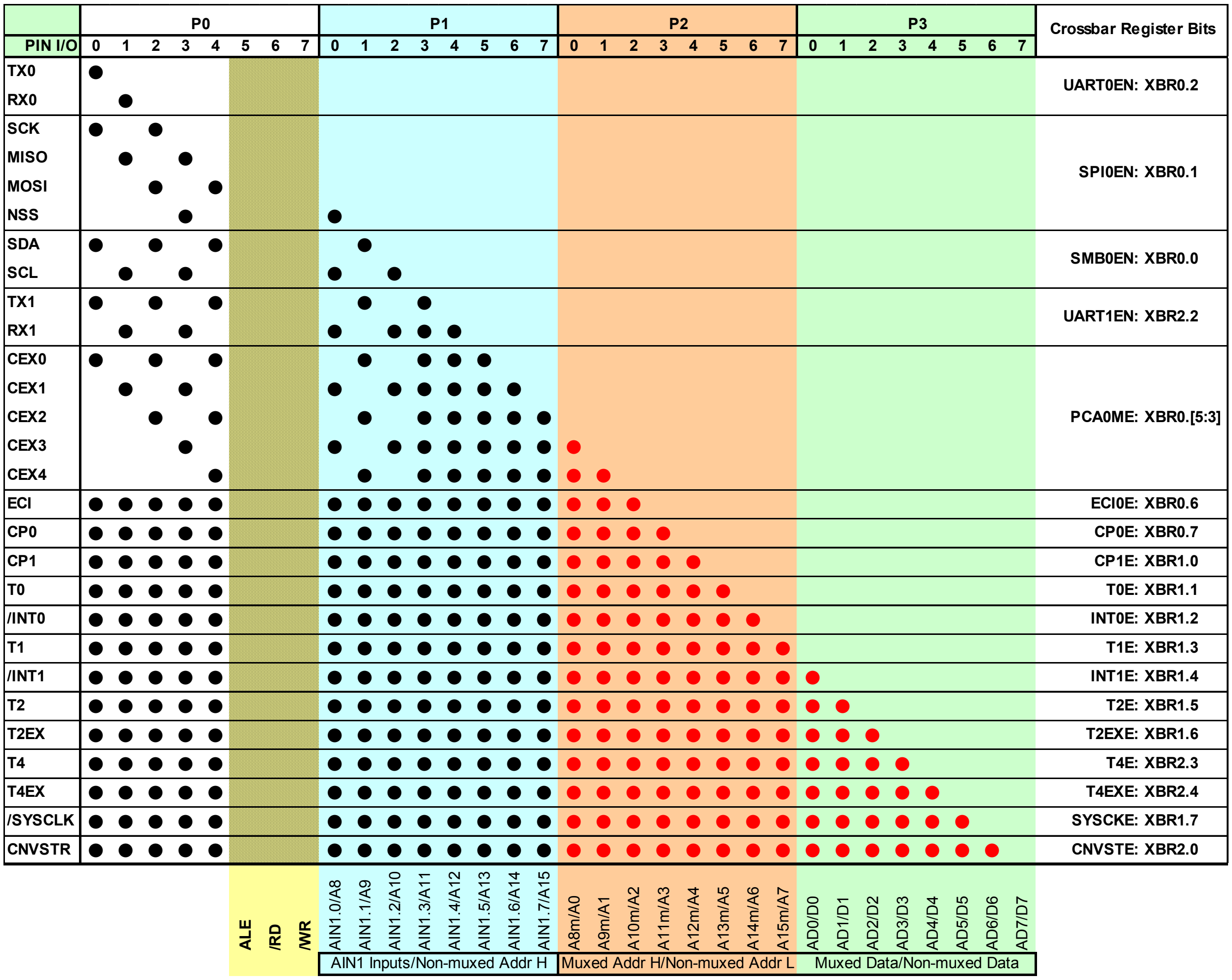
Note that the output drivers on a pin configured as an Analog Input are not explicitly disabled. Therefore, the associated P1MDOUT bits of pins configured as Analog Inputs should explicitly be set to logic 0 (Open-Drain output mode), and the associated Port Data bits should be set to logic 1 (high-impedance). 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 the ADC1 MUX; however, it is strongly recommended. See **Section “7. ADC1 (8-Bit ADC)” on page 75** for more information about ADC1.

17.1.7. External Memory Interface Pin Assignments

If the External Memory Interface (EMIF) is enabled on the Low ports (Ports 0 through 3), EMIFLE (XBR2.1) should be set to a logic 1 so that the Crossbar will not assign peripherals to P0.7 (/WR), P0.6 (/RD), and if the External Memory Interface is in Multiplexed mode, P0.5 (ALE). Figure 17.4 shows an example Crossbar Decode Table with EMIFLE=1 and the EMIF in Multiplexed mode. Figure 17.5 shows an example Crossbar Decode Table with EMIFLE=1 and the EMIF in Non-multiplexed mode.

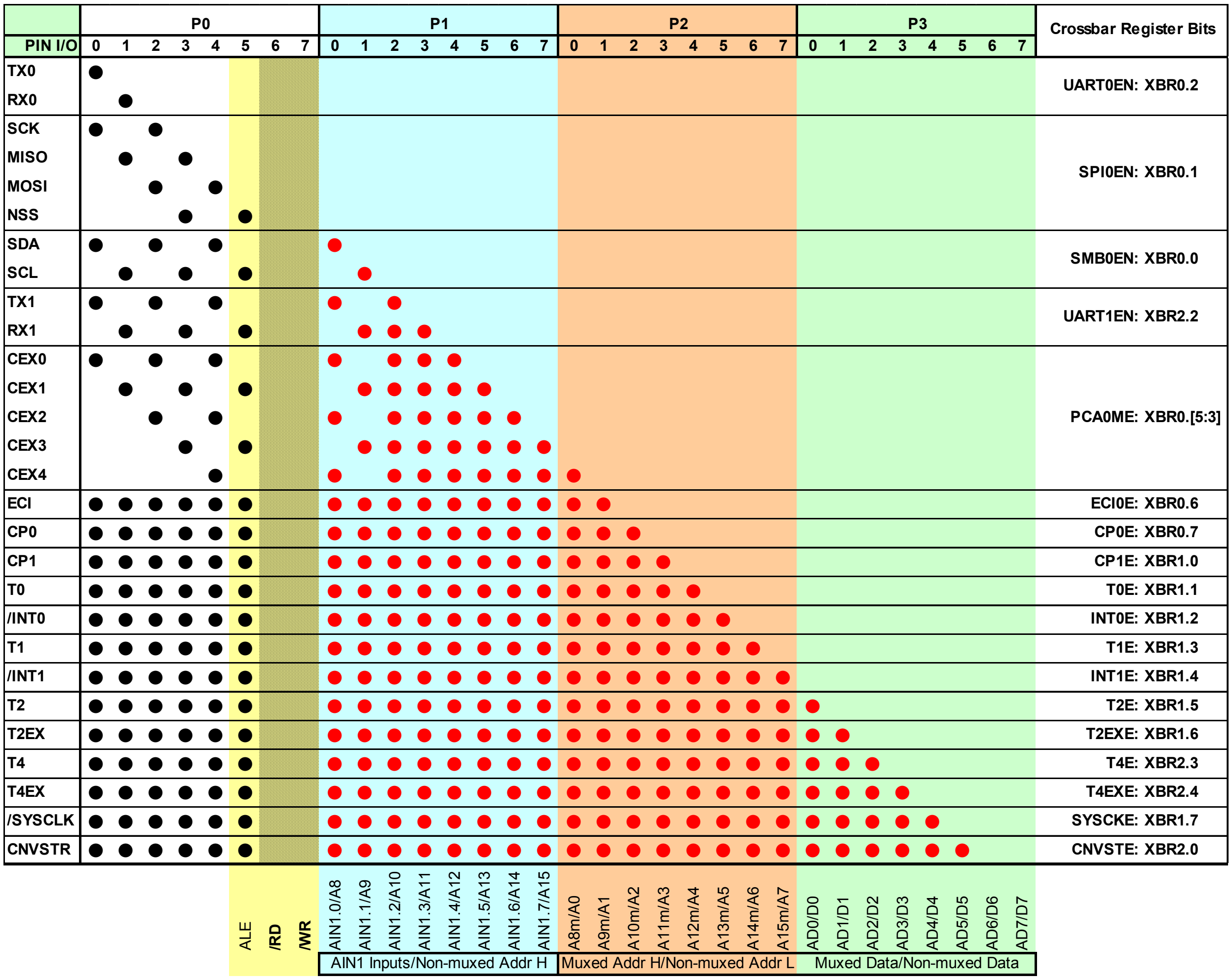
If the External Memory Interface is enabled on the Low ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Crossbar registers or the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus. See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.4. Priority Crossbar Decode Table
EMIFLE = 1; EMIF in Multiplexed Mode; P1MDIN = 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				●					●																								
SDA	●		●		●					●																							SMB0EN: XBR0.0
SCL		●		●					●		●																						
TX1	●		●		●					●		●																					UART1EN: XBR2.2
RX1		●		●					●		●	●	●																				
CEX0	●		●		●					●		●	●	●																			PCA0ME: XBR0.[5:3]
CEX1		●		●					●		●	●	●	●	●																		
CEX2			●		●					●		●	●	●	●	●																	
CEX3				●					●		●	●	●	●	●	●	●																
CEX4					●					●		●	●	●	●	●	●	●															
ECI	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●														ECI0E: XBR0.6
CP0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●													CP0E: XBR0.7
CP1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●												CP1E: XBR1.0
T0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●											T0E: XBR1.1
/INT0	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										INT0E: XBR1.2
T1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									T1E: XBR1.3
/INT1	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								INT1E: XBR1.4
T2	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							T2E: XBR1.5
T2EX	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						T2EXE: XBR1.6
T4	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					T4E: XBR2.3
T4EX	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				T4EXE: XBR2.4
/SYSCLK	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			SYSCKE: XBR1.7
CNVSTR	●	●	●	●	●				●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		CNVSTE: XBR2.0
						ALE	/RD	/WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN1 Inputs/Non-muxed Addr H								Muxed Addr H/Non-muxed Addr L								Muxed Data/Non-muxed Data								

Figure 17.5. Priority Crossbar Decode Table (EMIFLE = 1; EMIF in Non-multiplexed Mode; P1MDIN = 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				●		●																											
SDA	●		●		●				●																								SMB0EN: XBR0.0
SCL		●		●		●				●																							
TX1	●		●		●				●		●																						UART1EN: XBR2.2
RX1		●		●		●				●	●	●																					
CEX0	●		●		●				●		●	●	●																				PCA0ME: XBR0.[5:3]
CEX1		●		●		●				●	●	●	●	●																			
CEX2			●		●				●		●	●	●	●	●																		
CEX3				●		●				●	●	●	●	●	●	●																	
CEX4					●				●		●	●	●	●	●	●	●																
ECI	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●															ECI0E: XBR0.6
CP0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●														CP0E: XBR0.7
CP1	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●													CP1E: XBR1.0
T0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●												T0E: XBR1.1
/INT0	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●											INT0E: XBR1.2
T1	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●										T1E: XBR1.3
/INT1	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									INT1E: XBR1.4
T2	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								T2E: XBR1.5
T2EX	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							T2EXE: XBR1.6
T4	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						T4E: XBR2.3
T4EX	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					T4EXE: XBR2.4
/SYSCLK	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				SYSCKE: XBR1.7
CNVSTR	●	●	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●			CNVSTE: XBR2.0
						ALE	/RD	/WR	AIN1.0/A8	AIN1.1/A9	AIN1.2/A10	AIN1.3/A11	AIN1.4/A12	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN1 Inputs/Non-muxed Addr H								Muxed Addr H/Non-muxed Addr L								Muxed Data/Non-muxed Data								

17.1.8. Crossbar Pin Assignment Example

In this example (Figure 17.6), we configure the Crossbar to allocate Port pins for UART0, the SMBus, UART1, /INT0, and /INT1 (8 pins total). Additionally, we configure the External Memory Interface to operate in Multiplexed mode and to appear on the Low ports. Further, we configure P1.2, P1.3, and P1.4 for Analog Input mode so that the voltages at these pins can be measured by ADC1. The configuration steps are as follows:

1. XBR0, XBR1, and XBR2 are set such that UART0EN = 1, SMB0EN = 1, INT0E = 1, INT1E = 1, and EMIFLE = 1. Thus: XBR0 = 0x05, XBR1 = 0x14, and XBR2 = 0x02.
2. We configure the External Memory Interface to use Multiplexed mode and to appear on the Low ports. PRTSEL = 0, EMD2 = 0.
3. We configure the desired Port 1 pins to Analog Input mode by setting P1MDIN to 0xE3 (P1.4, P1.3, and P1.2 are Analog Inputs, so their associated P1MDIN bits are set to logic 0).
4. We enable the Crossbar by setting XBARE = 1: XBR2 = 0x46.
    - UART0 has the highest priority, so P0.0 is assigned to TX0, and P0.1 is assigned to RX0.
    - The SMBus is next in priority order, so P0.2 is assigned to SDA, and P0.3 is assigned to SCL.
    - UART1 is next in priority order, so P0.4 is assigned to TX1. Because the External Memory Interface is selected on the lower Ports, EMIFLE = 1, which causes the Crossbar to skip P0.6 (/RD) and P0.7 (/WR). Because the External Memory Interface is configured in Multiplexed mode, the Crossbar will also skip P0.5 (ALE). RX1 is assigned to the next non-skipped pin, which in this case is P1.0.
    - /INT0 is next in priority order, so it is assigned to P1.1.
    - P1MDIN is set to 0xE3, which configures P1.2, P1.3, and P1.4 as Analog Inputs, causing the Crossbar to skip these pins.
    - /INT1 is next in priority order, so it is assigned to the next non-skipped pin, which is P1.5.
    - The External Memory Interface will drive Ports 2 and 3 (denoted by red dots in Figure 17.6) during the execution of an off-chip MOVX instruction.
5. We set the UART0 TX pin (TX0, P0.0), UART1 TX pin (TX1, P0.4), ALE, /RD, /WR (P0.[7:3]) outputs to Push-Pull by setting P0MDOUT = 0xF1.
6. We configure the output modes of the EMIF Ports (P2, P3) to Push-Pull by setting P2MDOUT = 0xFF and P3MDOUT = 0xFF.
7. We explicitly disable the output drivers on the 3 Analog Input pins by setting P1MDOUT = 0x00 (configure outputs to Open-Drain) and P1 = 0xFF (a logic 1 selects the high-impedance state).

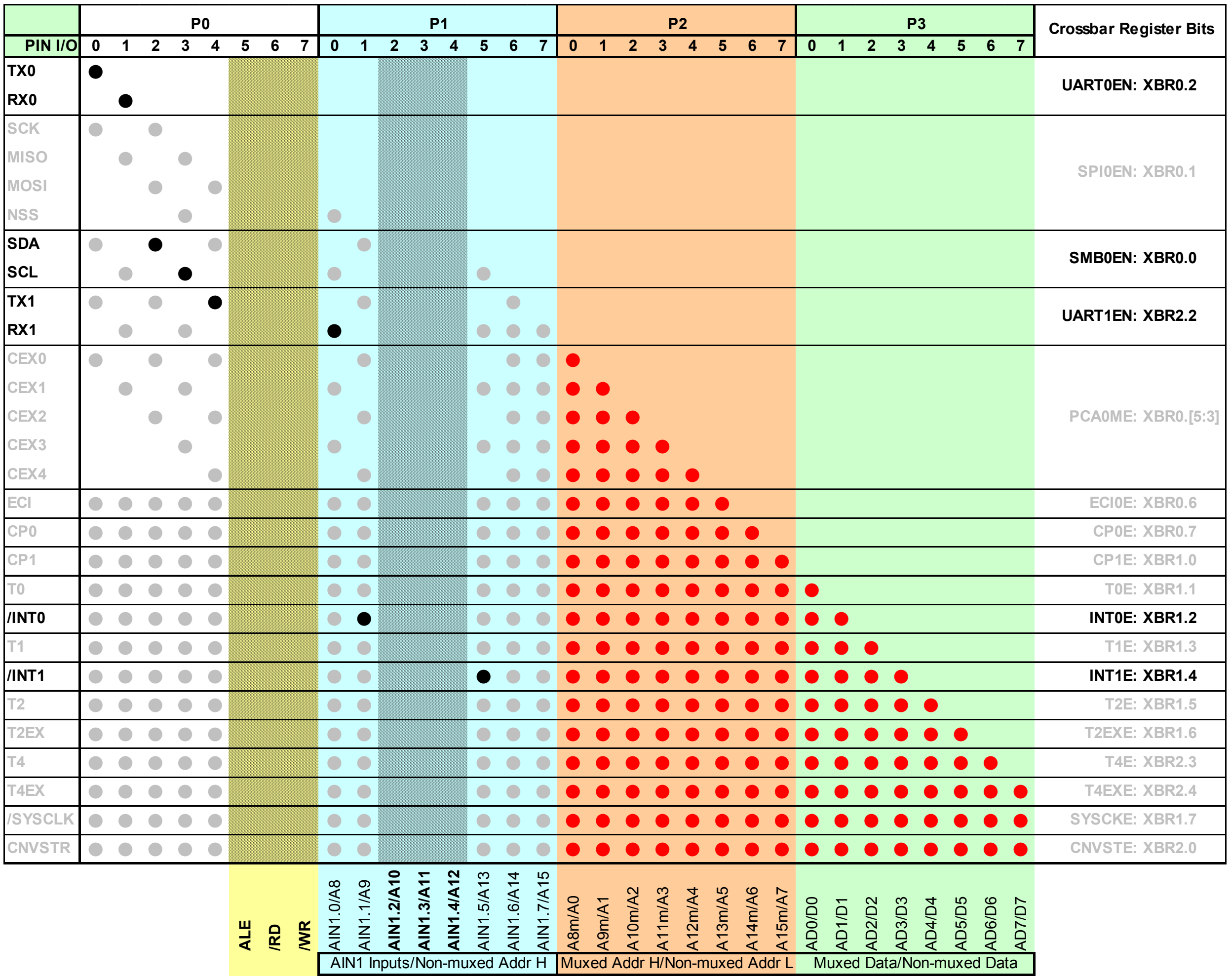


Figure 17.6. Crossbar Example:
(EMIFLE = 1; EMIF in Multiplexed Mode; P1MDIN = 0xE3;
XBR0 = 0x05; XBR1 = 0x14; XBR2 = 0x46)

	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					○					○					○	○	🔴	🔴	🔴	🔴	🔴												
ECI	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴											ECI0E: XBR0.6
CP0	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴										CP0E: XBR0.7
CP1	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴									CP1E: XBR1.0
T0	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴								T0E: XBR1.1
/INT0	○	○	○	○	○				○	●				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴							**INT0E: XBR1.2**
T1	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴						T1E: XBR1.3
/INT1	○	○	○	○	○				○	○				●	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴					**INT1E: XBR1.4**
T2	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴				T2E: XBR1.5
T2EX	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴			T2EXE: XBR1.6
T4	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴		T4E: XBR2.3
T4EX	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	T4EXE: XBR2.4
/SYSCLK	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	SYSCKE: XBR1.7
CNVSTR	○	○	○	○	○				○	○				○	○	○	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	🔴	CNVSTE: XBR2.0
						ALE	**/RD**	**/WR**	AIN1.0/A8	AIN1.1/A9	**AIN1.2/A10**	**AIN1.3/A11**	**AIN1.4/A12**	AIN1.5/A13	AIN1.6/A14	AIN1.7/A15	A8m/A0	A9m/A1	A10m/A2	A11m/A3	A12m/A4	A13m/A5	A14m/A6	A15m/A7	AD0/D0	AD1/D1	AD2/D2	AD3/D3	AD4/D4	AD5/D5	AD6/D6	AD7/D7	
									AIN1 Inputs/Non-muxed Addr H								Muxed Addr H/Non-muxed Addr L								Muxed Data/Non-muxed Data								

Figure 17.7. 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

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: RESERVED
111: RESERVED

Bit2: UART0EN: UART0 I/O Enable Bit.
0: UART0 I/O unavailable at Port pins.
1: UART0 TX routed to P0.0, and RX routed to P0.1.

Bit1: SPI0EN: SPI0 Bus I/O Enable Bit.
0: SPI0 I/O unavailable at Port pins.
1: SPI0 SCK, MISO, MOSI, and NSS routed to 4 Port pins.

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 17.8. 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

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: T0 routed to Port pin.
Bit0: CP1E: CP1 Output Enable Bit.
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

Figure 17.9. 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	EMIFLE	CNVSTE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3

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.

Bit1: EMIFLE: External Memory Interface Low-Port Enable Bit.
0: P0.7, P0.6, and P0.5 functions are determined by the Crossbar or the Port latches.
1: If EMI0CF.4 = ‘0’ (External Memory Interface is in Multiplexed mode)
    P0.7 (/WR), P0.6 (/RD), and P0.5 (ALE) are ‘skipped’ by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.
1: If EMI0CF.4 = ‘1’ (External Memory Interface is in Non-multiplexed mode)
    P0.7 (/WR) and P0.6 (/RD) are ‘skipped’ by the Crossbar and their output states are determined by the Port latches and the External Memory Interface.

Bit0: CNVSTE: External Convert Start Input Enable Bit.
0: CNVSTR unavailable at Port pin.
1: CNVSTR routed to Port pin.

Figure 17.10. 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	SFR Address:
							(bit addressable)	0x80

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.

Note: P0.7 (/WR), P0.6 (/RD), and P0.5 (ALE) can be driven by the External Data Memory Interface. See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information. See also Figure 17.9 for information about configuring the Crossbar for External Memory accesses.

Figure 17.11. 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

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 17.12. 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	SFR Address:
							(bit addressable)	0x90

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.

Notes:

1. P1.[7:0] can be configured as inputs to ADC1 as AIN1.[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 17.13). Note that in analog mode, the output mode of the pin is determined by the Port 1 latch and P1MDOUT (Figure 17.14). See **Section “7. ADC1 (8-Bit ADC)” on page 75** for more information about ADC1.
2. P1.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Non-multiplexed mode). See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.13. 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:
								0xBD

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 17.9).

Figure 17.14. 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

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 17.15. 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	SFR Address:
							(bit addressable)	0xA0

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.

Note: P2.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Multiplexed mode, or as Address[7:0] in Non-multiplexed mode). See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.16. 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

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 17.17. 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	SFR Address:
							(bit addressable)	0xB0

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: P3.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.18. 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

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.

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.

17.2. Ports 4 through 7 (C8051F020/2 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 17.21, Figure 17.22, Figure 17.23, and Figure 17.24), a set of SFRs which are byte-addressable.

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 ANL, ORL, XRL, CPL, INC, 89 -1.20(ti)-6 -1.2

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 the P74OUT register (see Figure 17.20). Each bit in P74OUT controls the output mode of a 4-bit bank of Port pins on Ports 4, 5, 6, and 7. A logic 1 in P74OUT.7 will configure the output modes of 4 most-significant bits of Port 7, P7.[7:4], to Push-Pull; a logic 0 in P74OUT.7 will configure the output modes of P7.[7:4] to Open-Drain.

17.2.3. 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 P74OUT.7 to a logic 0 and P7.7 to a logic 1.

17.2.4. 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 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.

17.2.5. External Memory Interface

If the External Memory Interface (EMIF) is enabled on the High ports (Ports 4 through 7), EMIFLE (XBR2.1) should be set to a logic 0.

If the External Memory Interface is enabled on the High ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus during the MOVX execution. See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.20. P74OUT: Ports 7 - 4 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P7H	P7L	P6H	P6L	P5H	P5L	P4H	P4L	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB5

Bit7: P7H: Port7 Output Mode High Nibble Bit.
0: P7.[7:4] configured as Open-Drain.
1: P7.[7:4] configured as Push-Pull.
Bit6: P7L: Port7 Output Mode Low Nibble Bit.
0: P7.[3:0] configured as Open-Drain.
1: P7.[3:0] configured as Push-Pull.
Bit5: P6H: Port6 Output Mode High Nibble Bit.
0: P6.[7:4] configured as Open-Drain.
1: P6.[7:4] configured as Push-Pull.
Bit4: P6L: Port6 Output Mode Low Nibble Bit.
0: P6.[3:0] configured as Open-Drain.
1: P6.[3:0] configured as Push-Pull.
Bit3: P5H: Port5 Output Mode High Nibble Bit.
0: P5.[7:4] configured as Open-Drain.
1: P5.[7:4] configured as Push-Pull.
Bit2: P5L: Port5 Output Mode Low Nibble Bit.
0: P5.[3:0] configured as Open-Drain.
1: P5.[3:0] configured as Push-Pull.
Bit1: P4H: Port4 Output Mode High Nibble Bit.
0: P4.[7:4] configured as Open-Drain.
1: P4.[7:4] configured as Push-Pull.
Bit0: P4L: Port4 Output Mode Low Nibble Bit.
0: P4.[3:0] configured as Open-Drain.
1: P4.[3:0] configured as Push-Pull.

Figure 17.21. 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	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x84

Bits7-0: P4.[7:0]: Port4 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P74OUT bit = 0). See Figure 17.20.
Read - Returns states of I/O pins.
0: P4.n pin is logic low.
1: P4.n pin is logic high.

Note: P4.7 (/WR), P4.6 (/RD), and P4.5 (ALE) can be driven by the External Data Memory Interface. See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information.

Figure 17.22. 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	SFR Address: 0x85

Bits7-0: P5.[7:0]: Port5 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P74OUT bit = 0). See Figure 17.20.
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 “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.23. 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	SFR Address: 0x86

Bits7-0: P6.[7:0]: Port6 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P74OUT bit = 0). See Figure 17.20.
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 "16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM" on page 145** for more information about the External Memory Interface.

Figure 17.24. 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	SFR Address: 0x96

Bits7-0: P7.[7:0]: Port7 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (Open-Drain if corresponding P74OUT bit = 0). See Figure 17.20.
Read - Returns states of I/O pins.
0: P7.n pin is logic low.
1: P7.n pin is logic high.

Note: P7.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See **Section "16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM" on page 145** for more information about the External Memory Interface.

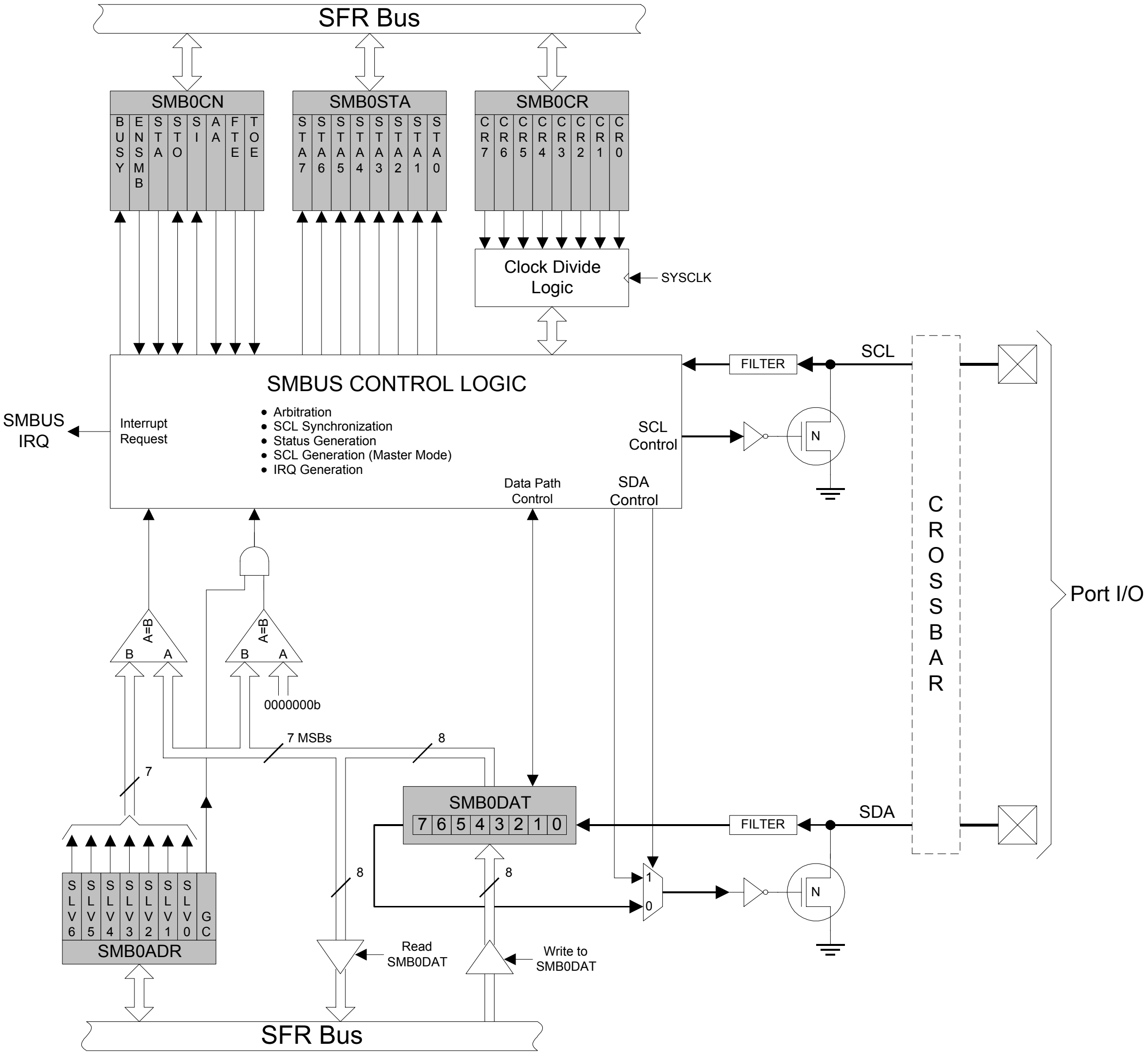
Notes

18. SYSTEM MANAGEMENT BUS / I^{2}C 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 I^{2}C 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. Data can be transferred at up to 1/8th of the system clock if desired (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

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. SMBus0 is controlled by SFRs as described in **Section 18.4 on page 189**.

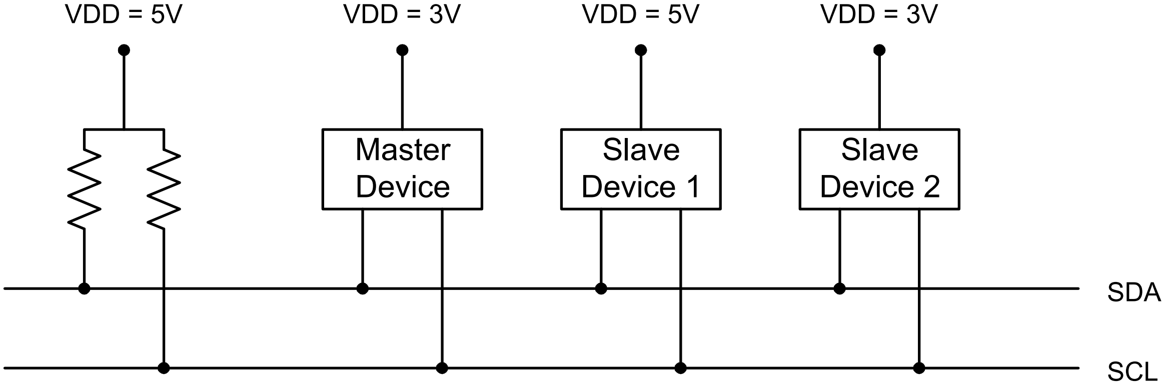
Figure 18.1. SMBus0 Block Diagram



SFR Bus
SMB0CN
BUSY
ENSMB
STA
STO
SI
AA
FTE
TOE
SMB0STA
STA7
STA6
STA5
STA4
STA3
STA2
STA1
STA0
SMB0CR
CR7
CR6
CR5
CR4
CR3
CR2
CR1
CR0
Clock Divide Logic
SYSCLK
SMBUS CONTROL LOGIC
Interrupt Request
• Arbitration
• SCL Synchronization
• Status Generation
• SCL Generation (Master Mode)
• IRQ Generation
SCL Control
Data Path Control
SDA Control
SMBUS IRQ
FILTER
SCL
N
CROSSBAR
Port I/O
A=B
B
A
0000000b
7 MSBs
8
7
SMB0DAT
7 6 5 4 3 2 1 0
FILTER
SDA
1
0
N
8
8
SLV6
SLV5
SLV4
SLV3
SLV2
SLV1
SLV0
GC
SMB0ADR
Read SMB0DAT
Write to SMB0DAT
SFR Bus

Figure 18.2 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 18.2. Typical SMBus Configuration



18.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^{2}C-bus and how to use it (including specifications), Philips Semiconductor.
2. The I^{2}C-Bus Specification -- Version 2.0, Philips Semiconductor.
3. System Management Bus Specification -- Version 1.1, SBS Implementers Forum.

18.2. SMBus Protocol

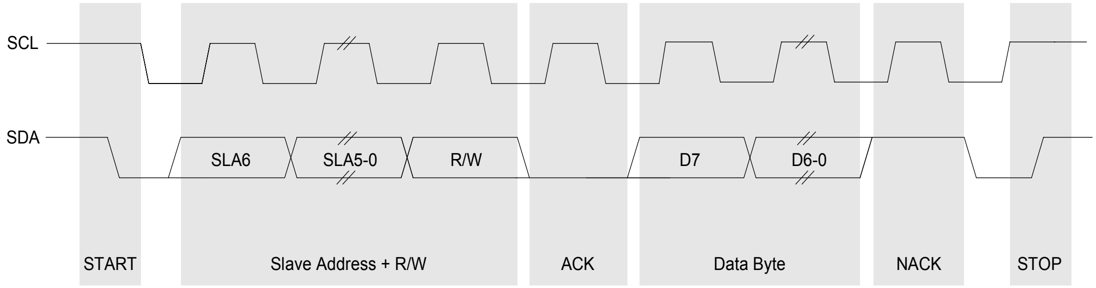
Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses 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 18.3). 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 18.3 illustrates a typical SMBus transaction.

Figure 18.3. SMBus Transaction



18.2.1. 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 18.2.4**). 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.

18.2.2. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I^2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

18.2.3. 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.

18.2.4. 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 a bus free timeout.

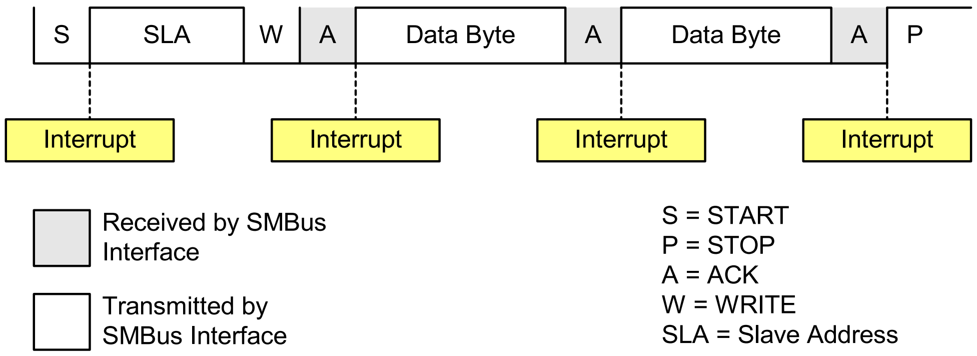
18.3. SMBus Transfer Modes

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 18.1 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.

18.3.1. 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.

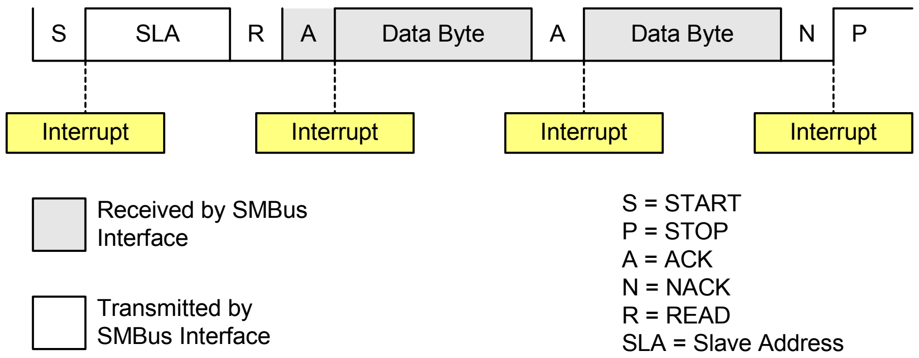
Figure 18.4. Typical Master Transmitter Sequence


S
SLA
W
A
Data Byte
A
Data Byte
A
P
Interrupt
Interrupt
Interrupt
Interrupt
Received by SMBus Interface
Transmitted by SMBus Interface
S = START
P = STOP
A = ACK
W = WRITE
SLA = Slave Address

18.3.2. 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.

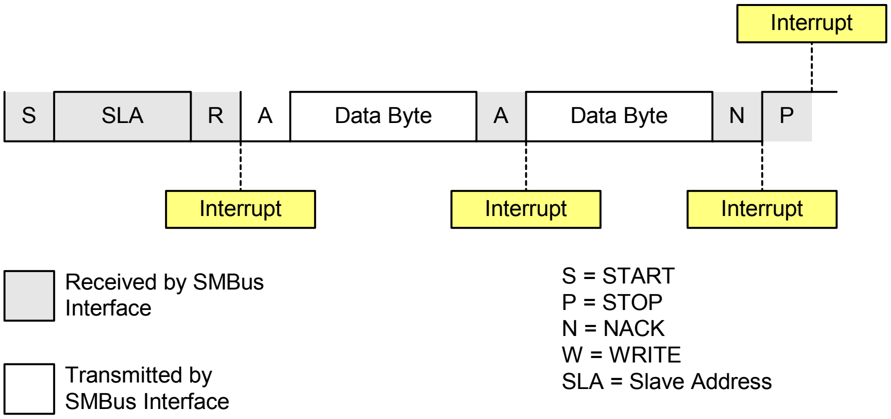
Figure 18.5. Typical Master Receiver Sequence


S
SLA
R
A
Data Byte
A
Data Byte
N
P
Interrupt
Interrupt
Interrupt
Interrupt
Received by SMBus Interface
Transmitted by SMBus Interface
S = START
P = STOP
A = ACK
N = NACK
R = READ
SLA = Slave Address

18.3.3. 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.

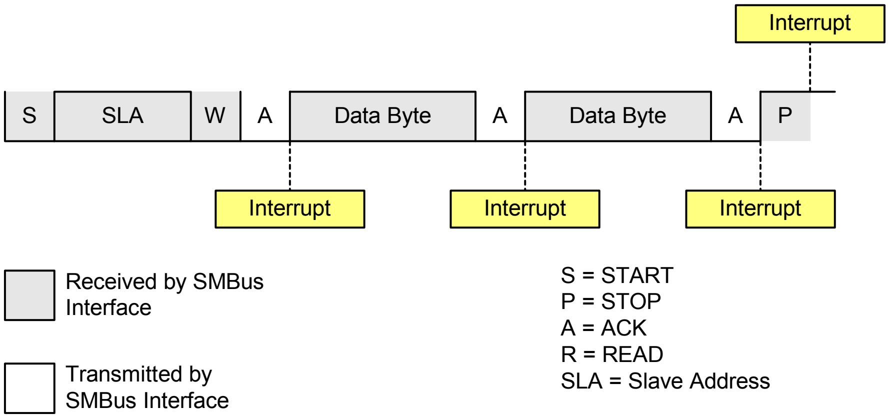
Figure 18.6. Typical Slave Transmitter Sequence


Interrupt
S
SLA
R
A
Data Byte
A
Data Byte
N
P
Interrupt
Interrupt
Interrupt
Received by SMBus Interface
Transmitted by SMBus Interface
S = START
P = STOP
N = NACK
W = WRITE
SLA = Slave Address

18.3.4. 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 18.7. Typical Slave Receiver Sequence


Interrupt
S
SLA
W
A
Data Byte
A
Data Byte
A
P
Interrupt
Interrupt
Interrupt
Received by SMBus Interface
Transmitted by SMBus Interface
S = START
P = STOP
A = ACK
R = READ
SLA = Slave Address

18.4. 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.

18.4.1. 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 cleared to logic 0 by hardware when a STOP condition is detected on the bus.

Setting the ENSMS 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. To ensure proper operation, the STO bit should be explicitly cleared to ‘0’ before setting the STA bit to ‘1’.

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 18.9, SMBus0 Clock Rate Register).

Figure 18.8. 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	SFR Address:
							(bit addressable)	0xC0

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. To ensure proper operation, the STO bit should be explicitly cleared to ‘0’ before setting the STA bit to ‘1’.

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.

18.4.2. Clock Rate Register

Figure 18.9. 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

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:

$$SMB0CR < ((288 - 0.85 \cdot SYSCLK) / 1.125)$$

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:

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

18.4.3. 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 is cleared to 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 18.10. 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

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 in/out and the CPU should not attempt to access this register.

18.4.4. 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 18.11. 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

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.

18.4.5. 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 18.12. 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	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC1

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 18.1. 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 2) Set STO, OR 3) Clear STO then set STA for repeated START.
	0x30	Data byte transmitted. NACK received.	1) Retry transfer OR 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 18.1. SMB0STA Status Codes and States

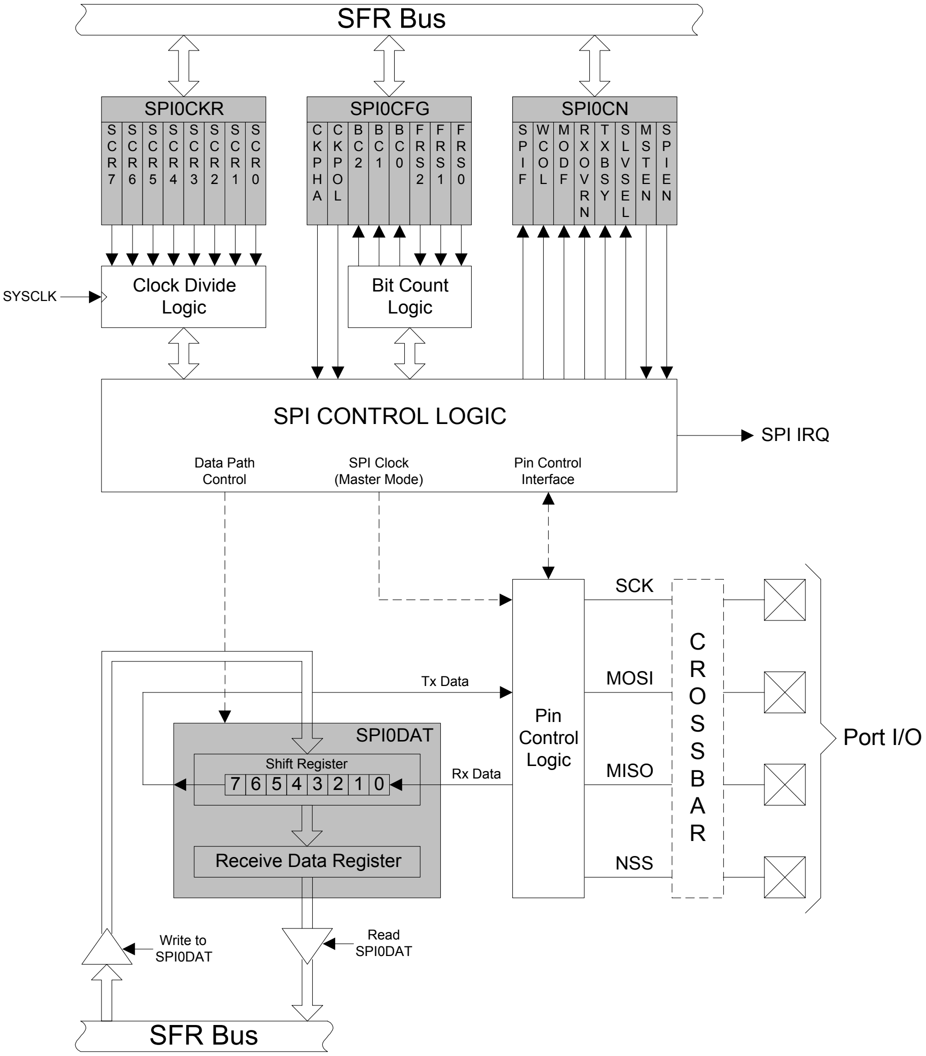
Mode	Status Code	SMBus State	Typical Action
Slave Receiver	0x60	Own slave address + W received. ACK transmitted.	Wait for data.
	0x68	Arbitration lost in sending SLA + R/W as master. Own address + W received. ACK transmitted.	Save current data for retry when bus is free. Wait for data.
	0x70	General 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.

19. SERIAL PERIPHERAL INTERFACE BUS (SPI0)

The Serial Peripheral Interface (SPI0) provides access to a four-wire, full-duplex, serial bus. SPI0 may operate as a master or a slave, and supports the connection of multiple slaves and masters on the same bus. A slave-select input (NSS) is included in the SPI0 interface to select SPI0 as a slave; additional general purpose port I/O can be used as slave-select outputs when SPI0 is operating as a master. Collision detection is provided when two or more masters attempt a data transfer at the same time. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency.

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, and the serial input data synchronously with the system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less that 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 system clock.

Figure 19.1. SPI Block Diagram


SFR Bus
SPI0CKR
SCR7
SCR6
SCR5
SCR4
SCR3
SCR2
SCR1
SCR0
SPI0CFG
CKPHA
CKPOL
BC2
BC1
BC0
FRS2
FRS1
FRS0
SPI0CN
SPIF
WCOL
MODF
RXOVRN
TXBSY
SLVSEL
MSTEN
SPIEN
SYSCLK
Clock Divide Logic
Bit Count Logic
SPI CONTROL LOGIC
SPI IRQ
Data Path Control
SPI Clock (Master Mode)
Pin Control Interface
SCK
MOSI
MISO
NSS
CROSSBAR
Port I/O
Tx Data
Rx Data
Pin Control Logic
SPI0DAT
Shift Register
7 6 5 4 3 2 1 0
Receive Data Register
Write to SPI0DAT
Read SPI0DAT
SFR Bus

19.1. Signal Descriptions

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

19.1.1. 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.

19.1.2. 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. A SPI slave places the MISO pin in a high-impedance state when the slave is not selected.

19.1.3. Serial Clock (SCK)

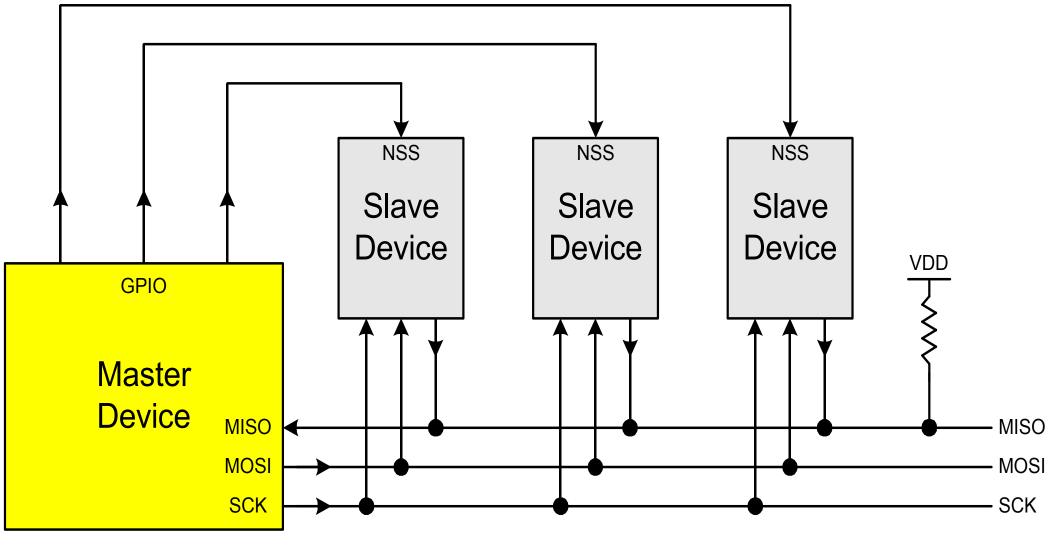
The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master.

19.1.4. Slave Select (NSS)

The slave select (NSS) signal is an input used to select SPI0 as a slave, or to disable SPI0 as a master. Note that the NSS signal is always an input to SPI0; with SPI0 operating as a master, slave select signals must be output via general purpose port I/O pins. See Figure 19.2 for a typical configuration; see Section "**17.1. Ports 0 through 3 and the Priority Crossbar Decoder**" on page **163** for general purpose port configuration.

The NSS signal must be low to initiate a transfer with SPI0 as a slave; SPI0 will exit slave mode when NSS is released high. Note that received data is not latched into the receive buffer until NSS is high. For multiple-byte transfers, NSS must be released high for at least 4 system clocks following each byte that is received by the SPI0 slave.

Figure 19.2. Typical SPI Interconnection

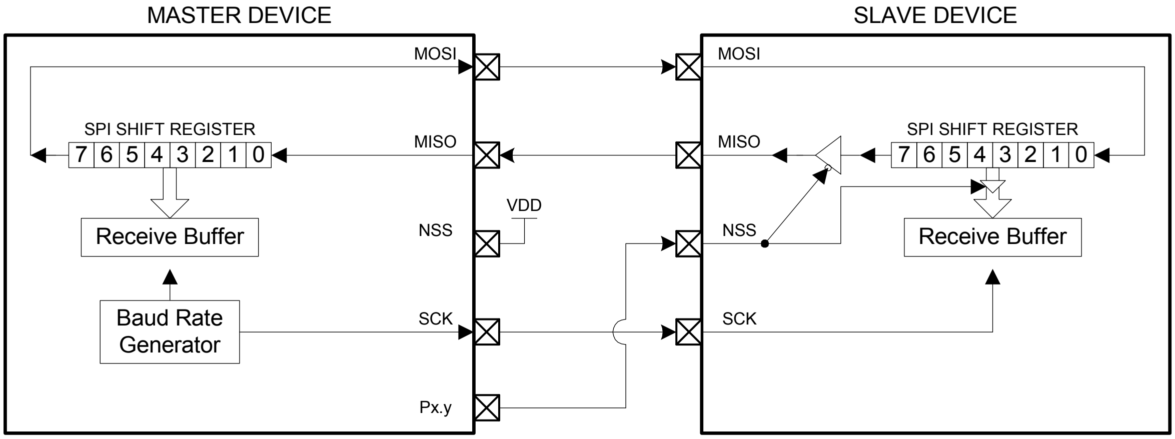


19.2. SPI0 Operation

Only a SPI master device can initiate a data transfer. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.1). Writing a byte of data to the SPI0 data register (SPI0DAT) when in Master Mode starts a data transfer. 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. The SPI0 master can be configured to shift in/out from one to eight bits in a transfer operation in order to accommodate slave devices with different word lengths. The SPIFRS bits in the SP0I Configuration Register (SPI0CFG.[2:0]) are used to select the number of bits to shift in/out in a transfer operation.

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. The data byte received from the slave replaces the data in the master's data register. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data transfer in both directions is synchronized with the serial clock generated by the master. Figure 19.3 illustrates the full-duplex operation of an SPI master and an addressed slave.

Figure 19.3. Full Duplex Operation



When SPI0 is enabled and not configured as a master, it will operate as an SPI slave. Another SPI device acting as a master will initiate a transfer by driving the NSS input signal low. The master then shifts data out of the shift register on the MOSI pin using the its serial clock. The SPIF flag is set to logic 1 when the NSS signal goes high, indicating the end of a data transfer. Note that following a rising edge on NSS, the receive buffer will always contain the last 8 bits of data in the slave shift register. The slave can load its shift register for the next data transfer by writing to the SPI0 data register. The slave must make the write to the data register at least one SPI serial clock cycle before the master starts the next transmission. Otherwise, the byte of data already in the slave's shift register will be transferred. 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.

The SPI0 data register is double buffered on reads, but not on writes. If a write to SPI0DAT is attempted during a data transfer, the WCOL flag (SPI0CN.6) will be set to logic 1 and the write will be ignored. The current data transfer will continue uninterrupted. A read of the SPI0 data register by the system controller actually reads the receive buffer. The receive overrun flag (RXOVRN in register SPI0CN) is set anytime a SPI0 slave detects a rising edge on NSS while the receive buffer still holds unread data from a previous transfer. The new data is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte causing the overrun is lost.

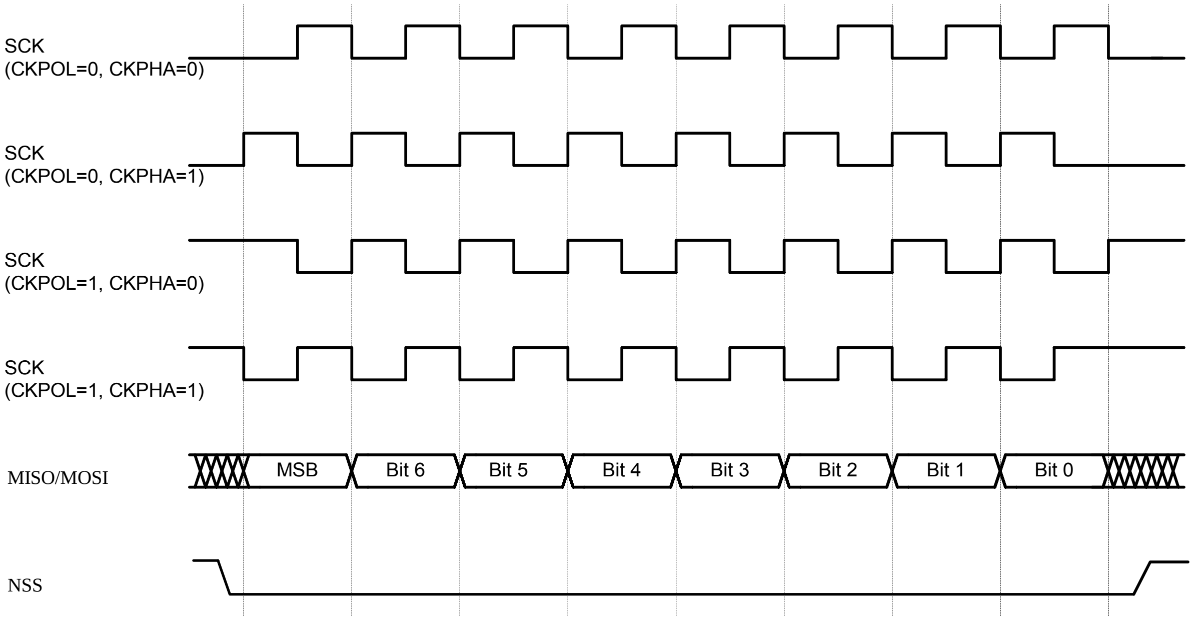
Multiple masters may reside on the same bus. A Mode Fault flag (MODF, SPI0CN.5) is set to logic 1 when SPI0 is configured as a master (MSTEN = 1) and its slave select signal NSS is pulled low. When the Mode Fault flag is set, the MSTEN and SPIEN bits of the SPI control register are cleared by hardware, thereby placing the SPI0 module in an "off-line" state. In a multiple-master environment, the system controller should check the state of the SLVSEL flag (SPI0CN.2) to ensure the bus is free before setting the MSTEN bit and initiating a data transfer.

19.3. Serial Clock Timing

As shown in Figure 19.4, 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.7) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.6) 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. Note: SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) while changing the clock phase and polarity.

The SPI0 Clock Rate Register (SPI0CKR) as shown in Figure 19.7 controls the master mode serial clock frequency. This register is ignored when operating in slave mode.

Figure 19.4. Data/Clock Timing Diagram



19.4. 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 section.

Figure 19.5. SPI0CFG: SPI0 Configuration Register

R/W	R/W	R	R	R	R/W	R/W	R/W	Reset Value
CKPHA	CKPOL	BC2	BC1	BC0	SPIFRS2	SPIFRS1	SPIFRS0	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9A

Bit7: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data sampled on first edge of SCK period.
1: Data sampled on second edge of SCK period.

Bit6: 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.

Bits5-3: BC2-BC0: SPI0 Bit Count.
Indicates which of the up to 8 bits of the SPI0 word have been transmitted.

BC2-BC0			BIT Transmitted
0	0	0	Bit 0 (LSB)
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7 (MSB)

Bits2-0: SPIFRS2-SPIFRS0: SPI0 Frame Size.
These three bits determine the number of bits to shift in/out of the SPI0 shift register during a data transfer in master mode. They are ignored in slave mode.

SPIFRS			Bits Shifted
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Figure 19.6. SPI0CN: SPI0 Control Register

R/W	R/W	R/W	R/W	R	R	R/W	R/W	Reset Value
SPIF	WCOL	MODF	RXOVRN	TXBSY	SLVSEL	MSTEN	SPIEN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xF8

Bit7: 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.

Bit6: 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. 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.

Bit5: 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 and MSTEN = 1). 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.

Bit4: RXOVRN: Receive Overrun Flag.
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. 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.

Bit3: TXBSY: Transmit Busy Flag.
This bit is set to logic 1 by hardware while a master mode transfer is in progress. It is cleared by hardware at the end of the transfer.

Bit2: SLVSEL: Slave Selected Flag.
This bit is set to logic 1 whenever the NSS pin is low indicating it is enabled as a slave. It is cleared to logic 0 when NSS is high (slave disabled).

Bit1: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit0: SPIEN: SPI0 Enable.
This bit enables/disables the SPI.
0: SPI disabled.

Figure 19.7. 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

Bits7-0: 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 down version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CR* is the 8-bit value held in the SPI0CR 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 19.8. 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

Bits7-0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data immediately into the shift register and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

Notes

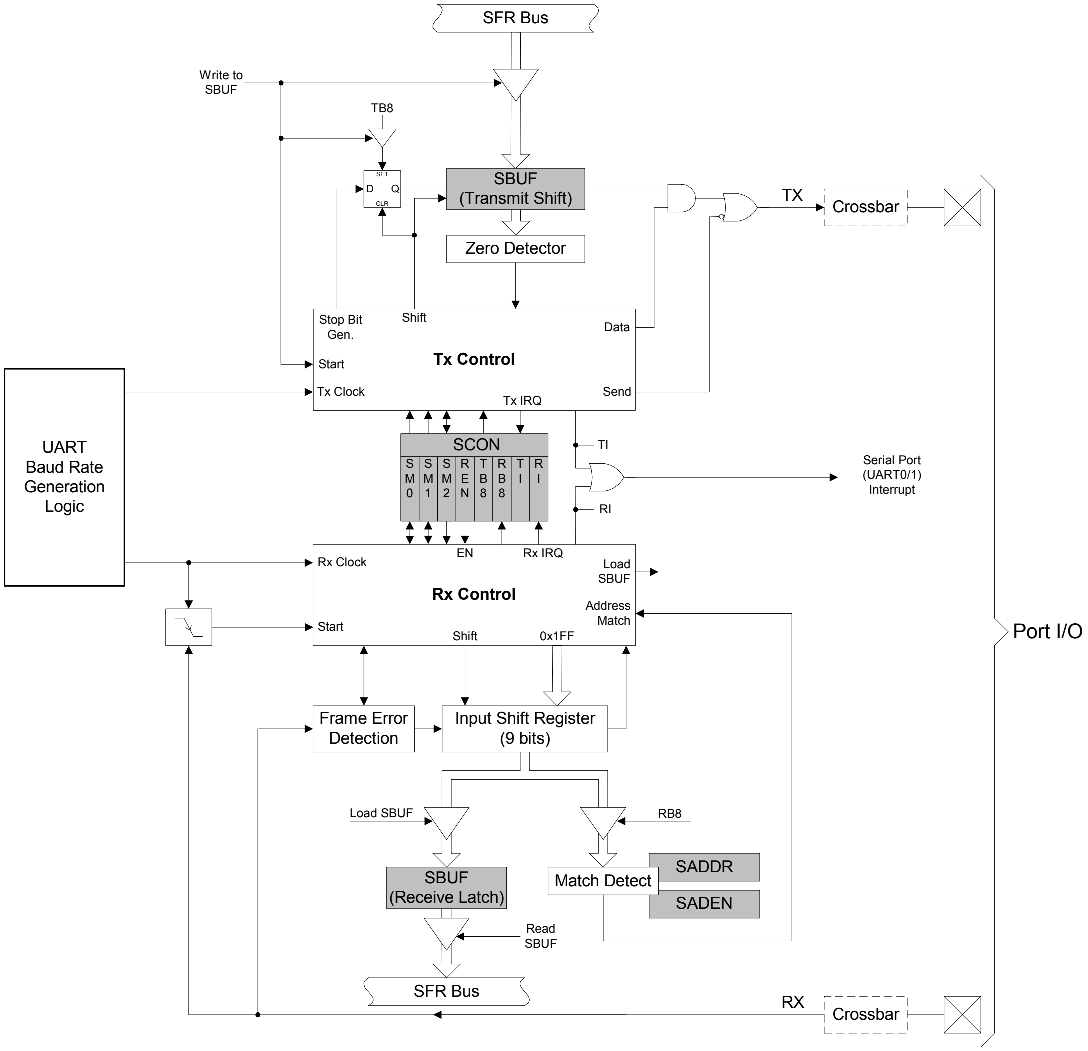
20. UART0

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 previous received byte is 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. Reads access the Receive register and writes access the Transmit register automatically.

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 20.1. UART0 Block Diagram


SFR Bus
Write to SBUF
TB8
SET
D
Q
CLR
SBUF (Transmit Shift)
Zero Detector
TX
Crossbar
Stop Bit Gen.
Shift
Data
Start
Tx Control
Tx Clock
Send
Tx IRQ
UART Baud Rate Generation Logic
SCON
S M 0
S M 1
S M 2
R E N
T B 8
R B 8
T I
R I
TI
Serial Port (UART0/1) Interrupt
RI
Rx Clock
EN
Rx IRQ
Load SBUF
Rx Control
Address Match
Start
Shift
0x1FF
Port I/O
Frame Error Detection
Input Shift Register (9 bits)
Load SBUF
RB8
SBUF (Receive Latch)
Match Detect
SADDR
SADEN
Read SBUF
SFR Bus
RX
Crossbar

20.1. 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 20.1.

Table 20.1. UART0 Modes

Mode	Synchronization	Baud Clock	Data Bits	Start/Stop Bits
0	Synchronous	SYSCLK / 12	8	None
1	Asynchronous	Timer 1 or 2 Overflow	8	1 Start, 1 Stop
2	Asynchronous	SYSCLK / 32 or SYSCLK / 64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1 or 2 Overflow	9	1 Start, 1 Stop

20.1.1. 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 20.2).

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 20.3), 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 20.2. UART0 Mode 0 Interconnect

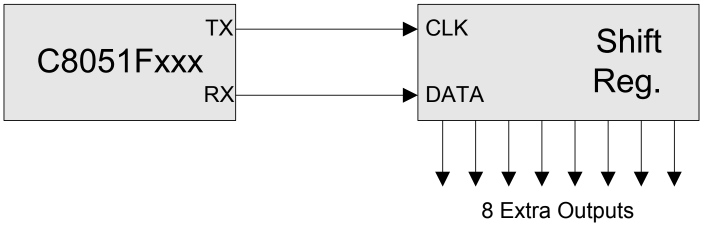
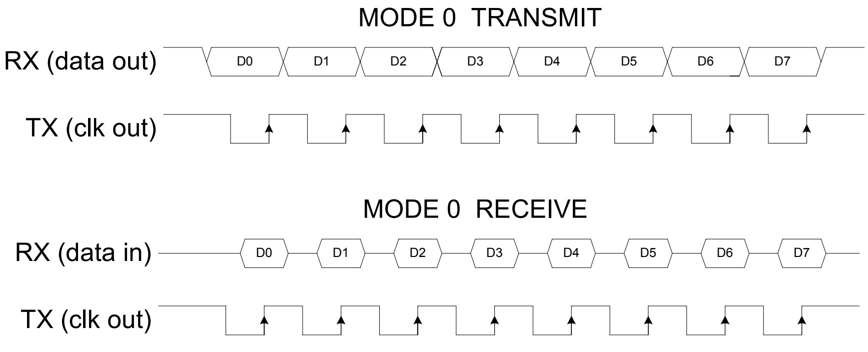


Figure 20.3. UART0 Mode 0 Timing Diagram



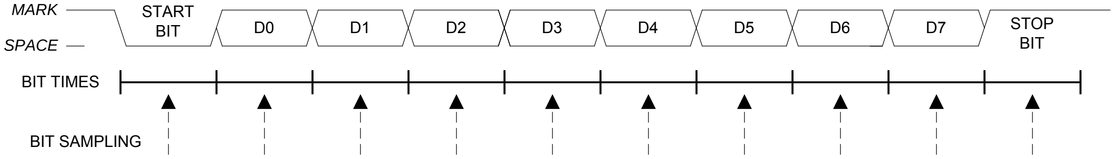
20.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

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 are stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

Figure 20.4. UART0 Mode 1 Timing Diagram



The baud rate generated in Mode 1 is a function of timer overflow, shown in Equation 20.1 and Equation 20.2. UART0 can use Timer 1 operating in *8-Bit Auto-Reload Mode*, or Timer 2 operating in *Baud Rate Generator 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) - to zero) a clock is sent to the baud rate logic.

Timer 2 is selected as TX and/or RX baud clock source by setting the TCLK0 (T2CON.4) and/or RCLK0 (T2CON.5) bits, respectively (see **Section “22. TIMERS” on page 225** for complete timer configuration details). When either TCLK0 or RCLK0 is set to logic 1, Timer 2 is forced into *Baud Rate Generator Mode*, with SYSCLK / 2 as its clock source. If TCLK0 and/or RCLK0 is logic 0, Timer 1 acts as the baud clock source for the TX and/or RX circuits, respectively.

The Mode 1 baud rate equations are shown below, where T1M is the Timer 1 Clock Select bit (register CKCON), TH1 is the 8-bit reload register for Timer 1, SMOD0 is the UART0 baud rate doubler (register PCON) and [RCAP2H , RCAP2L] is the 16-bit reload register for Timer 2.

Equation 20.1. Mode 1 Baud Rate using Timer 1

$$BaudRate = \left(\frac{2^{SMOD0}}{32}\right) \times \left(\frac{SYSCLK \times 12^{(T1M-1)}}{(256 - TH1)}\right)$$

Equation 20.2. Mode 1 Baud Rate using Timer 2

$$BaudRate = \frac{SYSCLK}{32 \times (65536 - [RCAP2H, RCAP2L])}$$

20.1.3. 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 "20.2. Multiprocessor Communications" on page 210**). 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 20.2**.

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 is set.

The baud rate in Mode 2 is either SYSCLK / 32 or SYSCLK / 64, depending on the value of the SMOD0 bit in register PCON.

Equation 20.3. Mode 2 Baud Rate

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

Figure 20.5. UART Modes 2 and 3 Timing Diagram

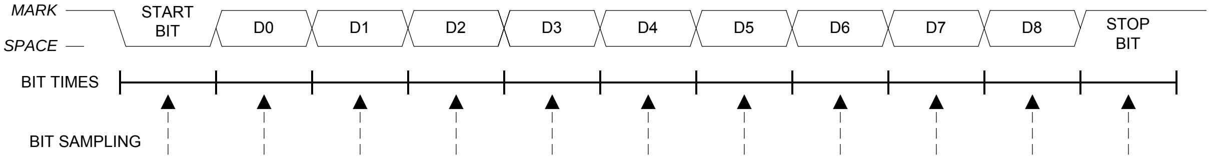
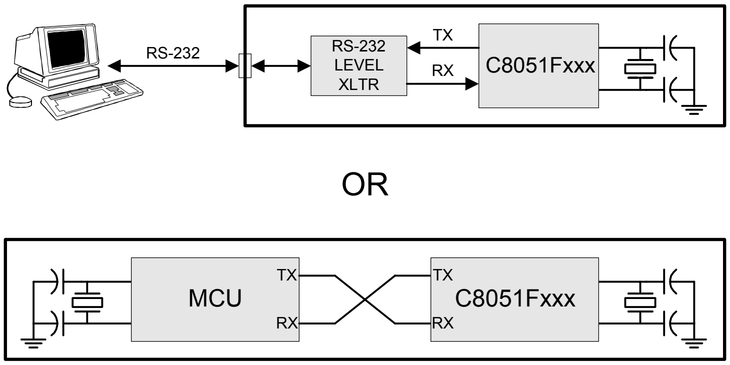


Figure 20.6. UART Modes 1, 2, and 3 Interconnect Diagram


RS-232
RS-232 LEVEL XLTR
TX
RX
C8051Fxxx
OR
MCU
TX
RX
TX
RX
C8051Fxxx

20.1.4. 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 overflows, as defined by Equation 20.1 and Equation 20.2. Multiprocessor communications and hardware address recognition are supported, as described in **Section 20.2**.

20.2. 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. A master processor begins a transfer with an address byte to select one or more target slave devices. 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.

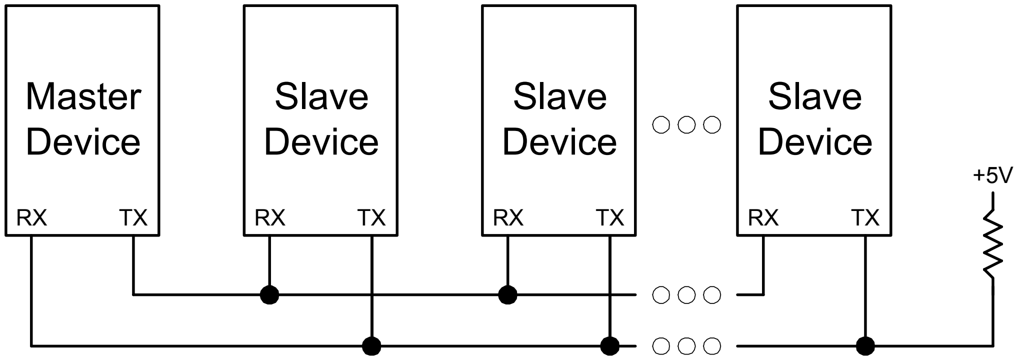
The UART0 address is configured via two SFRs: SADDR0 (Serial Address) and SADEN0 (Serial Address Enable). SADEN0 sets the bit mask for the address held in SADDR0: bits set to logic 1 in SADEN0 correspond to bits in SADDR0 that are checked against the received address byte; bits set to logic 0 in SADEN0 correspond to “don’t care” bits in SADDR0.

Example 1		Example 2		Example 3	
SADDR0	= 00110101	SADDR0	= 00110101	SADDR0	= 00110101
SADEN0	= 00001111	SADEN0	= 11110011	SADEN0	= 11000000
UART0 Address	= xxxx0101	UART0 Address	= 0011xx01	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 should 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 should reset 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.

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).

Figure 20.7. UART Multi-Processor Mode Interconnect Diagram



20.3. Frame and Transmission Error Detection

Frame error detection is available in the following modes when the SSTAT0 bit in register PCON is set to logic 1. Note: The SSTAT0 bit must be logic 1 to access any of the status bits (FE0, RXOVR0, and TXCOL0). To access the UART0 Mode Select bits (SM00, SM10, and SM20), the SSTAT0 bit must be logic 0.

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 also functions as the SM20 bit when the SSTAT0 bit in register PCON is logic 0.

Modes 1, 2, and 3:
The Receive Overrun bit (RXOVR0 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 RXOVR0 bit also functions as the SM10 bit when the SSTAT0 bit in register PCON is logic 0.

The Frame Error bit (FE0 in register SCON0) reads ‘1’ if an invalid (low) STOP bit is detected. Note that the FE0 bit also functions as the SM00 bit when the SSTAT0 bit in register PCON is logic 0.

Table 20.2. Oscillator Frequencies for Standard Baud Rates

Oscillator frequency (MHz)	Divide Factor	Timer 1 Load Value*	Resulting Baud Rate (Hz)**
25.0	434	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	320	0xEC	76800
24.576	848	0xCB	28800 (28921)
24.0	208	0XF3	115200 (115384)
24.0	833	0xCC	28800 (28846)
23.592	205	0xF3	115200 (113423)
23.592	819	0xCD	28800 (28911)
22.1184	192	0xF4	115200
22.1184	768	0xD0	28800
18.432	160	0xF6	115200
18.432	640	0xD8	28800
16.5888	144	0xF7	115200
16.5888	576	0xDC	28800
14.7456	128	0xF8	115200
14.7456	512	0xE0	28800
12.9024	112	0xF9	115200
12.9024	448	0xE4	28800
11.0592	96	0xFA	115200
11.0592	348	0xE8	28800
9.216	80	0xFB	115200
9.216	320	0xEC	28800
7.3728	64	0xFC	115200
7.3728	256	0xF0	28800
5.5296	48	0xFD	115200
5.5296	192	0xF4	28800
3.6864	32	0xFE	115200
3.6864	128	0xF8	28800
1.8432	16	0xFF	115200
1.8432	64	0xFC	28800

* Assumes SMOD0=1 and T1M=1.

** Numbers in parenthesis show the actual baud rate.

Figure 20.8. SCON0: UART0 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SM00/FE0	SM10/RXOV0	SM20/TXCOL0	REN0	TB80	RB80	TI0	RI0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x98

Bits7-6: The function of these bits is determined by the SSTAT0 bit in register PCON.
If SSTAT0 is logic 1, these bits are UART0 status indicators as described in **Section 20.3**.
If SSTAT0 is logic 0, these bits select the Serial Port Operation Mode as shown below.
SM00-SM10: Serial Port Operation Mode:

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

Bit5: SM20: Multiprocessor Communication Enable.
If SSTAT0 is logic 1, this bit is a UART0 status indicator as described in **Section 20.3**.
If SSTAT0 is logic 0, 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.
Modes 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 20.9. SBUF0: UART0 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

Bits7-0: SBUF0.[7:0]: UART0 Buffer Bits 7-0 (MSB-LSB)

This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 is what initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Figure 20.10. SADDR0: UART0 Slave Address 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:
								0xA9

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 20.11. SADEN0: UART0 Slave Address Enable 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:
								0xB9

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.

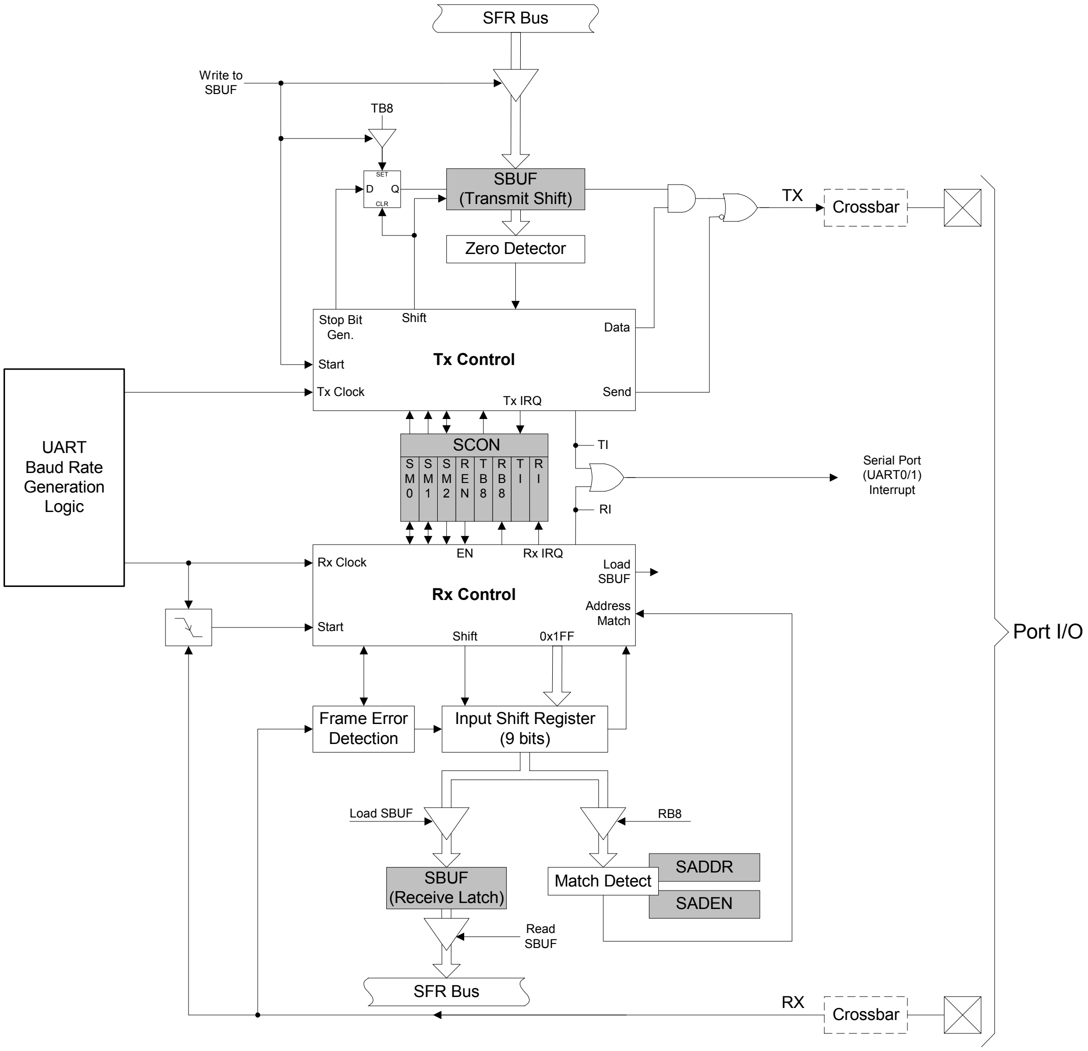
21. UART1

UART1 is an enhanced serial port with frame error detection and address recognition hardware. UART1 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 UART1 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 previous received byte is read.

UART1 is accessed via its associated SFRs, Serial Control (SCON1) and Serial Data Buffer (SBUF1). The single SBUF1 location provides access to both transmit and receive registers. Reads access the Receive register and writes access the Transmit register automatically.

UART1 may be operated in polled or interrupt mode. UART1 has two sources of interrupts: a Transmit Interrupt flag, TI1 (SCON1.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI1 (SCON1.0) set when reception of a data byte is complete. UART1 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 UART1 interrupt (transmit complete or receive complete).

Figure 21.1. UART1 Block Diagram


SFR Bus
Write to SBUF
TB8
SET
D
Q
CLR
SBUF (Transmit Shift)
Zero Detector
TX
Crossbar
Stop Bit Gen.
Shift
Data
Start
Tx Control
Tx Clock
Send
Tx IRQ
UART Baud Rate Generation Logic
SCON
S M 0
S M 1
S M 2
R E N
T B 8
R B 8
T I
R I
TI
RI
Serial Port (UART0/1) Interrupt
Rx Clock
EN
Rx IRQ
Load SBUF
Rx Control
Address Match
Start
Shift
0x1FF
Port I/O
Frame Error Detection
Input Shift Register (9 bits)
Load SBUF
RB8
SBUF (Receive Latch)
Match Detect
SADDR
SADEN
Read SBUF
SFR Bus
RX
Crossbar

21.1. UART1 Operational Modes

UART1 provides four operating modes (one synchronous and three asynchronous) selected by setting configuration bits in the SCON1 register. These four modes offer different baud rates and communication protocols. The four modes are summarized in Table 21.1.

Table 21.1. UART1 Modes

Mode	Synchronization	Baud Clock	Data Bits	Start/Stop Bits
0	Synchronous	SYSCLK / 12	8	None
1	Asynchronous	Timer 1 or 4 Overflow	8	1 Start, 1 Stop
2	Asynchronous	SYSCLK / 32 or SYSCLK / 64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1 or 4 Overflow	9	1 Start, 1 Stop

21.1.1. Mode 0: Synchronous Mode

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX1 pin. The TX1 pin provides the shift clock for both transmit and receive. The MCU must be the master since it generates the shift clock for transmission in both directions (see the interconnect diagram in Figure 21.2).

Data transmission begins when an instruction writes a data byte to the SBUF1 register. Eight data bits are transferred LSB first (see the timing diagram in Figure 21.3), and the TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the eighth bit time. Data reception begins when the REN1 Receive Enable bit (SCON1.4) is set to logic 1 and the RI1 Receive Interrupt Flag (SCON1.0) is cleared. One cycle after the eighth bit is shifted in, the RI1 flag is set and reception stops until software clears the RI1 bit. An interrupt will occur if enabled when either TI1 or RI1 are set.

The Mode 0 baud rate is SYSCLK / 12. RX1 is forced to open-drain in Mode 0, and an external pull-up will typically be required.

Figure 21.2. UART1 Mode 0 Interconnect

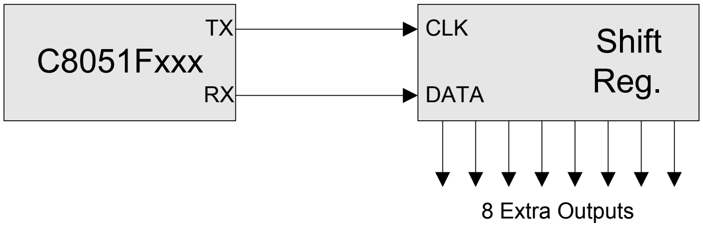
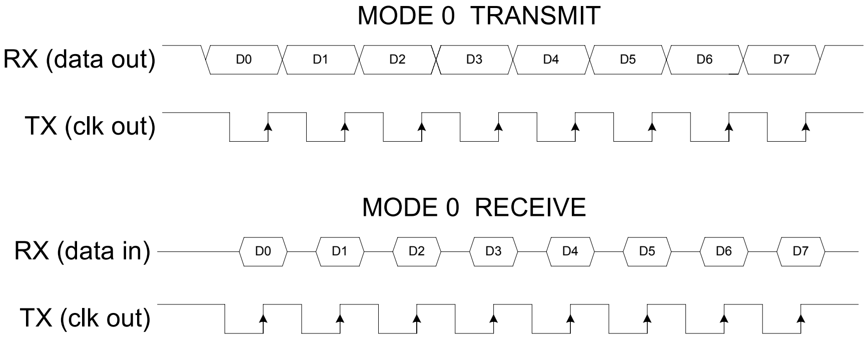


Figure 21.3. UART1 Mode 0 Timing Diagram



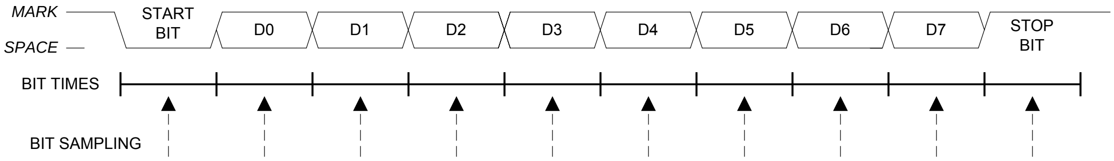
21.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

Mode 1 provides standard asynchronous, full duplex communication using a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted from the 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 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 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 SM21 is logic 1, the stop bit must be logic 1.

If these conditions are met, the eight bits of data are 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 21.4. UART1 Mode 1 Timing Diagram



The baud rate generated in Mode 1 is a function of timer overflow, shown in Equation 21.1 and Equation 21.2. UART1 can use Timer 1 operating in *8-Bit Auto-Reload Mode*, or Timer 4 operating in *Baud Rate Generator 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 4) - to zero) a clock is sent to the baud rate logic.

Timer 4 is selected as TX and/or RX baud clock source by setting the TCLK1 (T4CON.4) and/or RCLK1 (T4CON.5) bits, respectively (see **Section "22. TIMERS" on page 225** for complete timer configuration details). When either TCLK1 or RCLK1 is set to logic 1, Timer 4 is forced into *Baud Rate Generator Mode*, with SYSCLK / 2 as its clock source. If TCLK1 and/or RCLK1 is logic 0, Timer 1 acts as the baud clock source for the TX and/or RX circuits, respectively.

The Mode 1 baud rate equations are shown below, where T1M is the Timer 1 Clock Select bit (register CKCON), TH1 is the 8-bit reload register for Timer 1, SMOD1 is the UART1 baud rate doubler (register PCON), and [RCAP4H , RCAP4L] is the 16-bit reload register for Timer 4.

Equation 21.1. Mode 1 Baud Rate using Timer 1

$$BaudRate = \left(\frac{2^{SMOD1}}{32}\right) \times \left(\frac{SYSCLK \times 12^{(T1M - 1)}}{(256 - TH1)}\right)$$

Equation 21.2. Mode 1 Baud Rate using Timer 4

$$BaudRate = \frac{SYSCLK}{[32 \times (65536 - [RCAP4H, RCAP4L])]}$$

21.1.3. Mode 2: 9-Bit UART, Fixed Baud Rate

Mode 2 provides asynchronous, full-duplex communication using a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. Mode 2 supports multiprocessor communications and hardware address recognition (see **Section "21.2. Multiprocessor Communications" on page 220**). On transmit, the ninth data bit is determined by the value in TB81 (SCON1.3). 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 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 logic 1. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if RI1 is logic 0 and one of the following requirements are met:

1. SM21 is logic 0
2. SM21 is logic 1, the received 9th bit is logic 1, and the received address matches the UART1 address as described in **Section 21.2**.

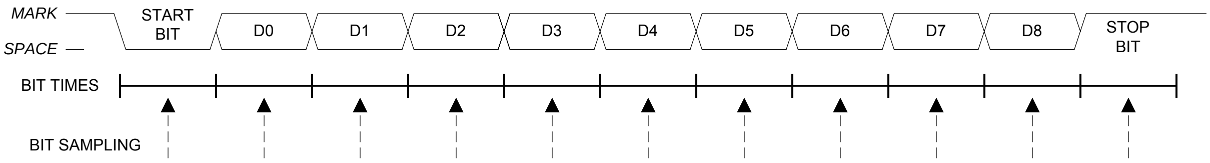
If the above conditions are satisfied, the eight bits of data are stored in SBUF1, the ninth bit is stored in RB81 and the RI1 flag is set. If these conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set. An interrupt will occur if enabled when either TI1 or RI1 is set.

The baud rate in Mode 2 is either SYSCLK / 32 or SYSCLK / 64, depending on the value of the SMOD1 bit in register PCON.

Equation 21.3. Mode 2 Baud Rate

$$BaudRate = 2^{SMOD1} \times \left(\frac{SYSCLK}{64}\right)$$

Figure 21.5. UART Modes 2 and 3 Timing Diagram



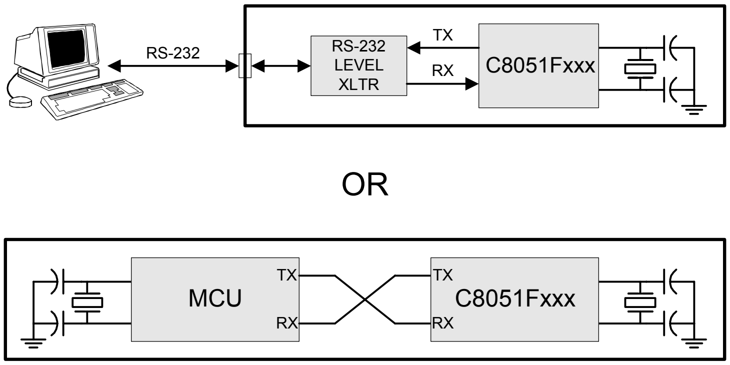

RS-232
RS-232 LEVEL XLTR
TX
RX
C8051Fxxx
OR
MCU
TX
RX
TX
RX
C8051Fxxx

Figure 21.6. UART Modes 1, 2, and 3 Interconnect Diagram

21.1.4. Mode 3: 9-Bit UART, Variable Baud Rate

Mode 3 uses the Mode 2 transmission protocol with the Mode 1 baud rate generation. Mode 3 operation transmits 11 bits: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The baud rate is derived from Timer 1 or Timer 4 overflows, as defined by Equation 21.1 and Equation 21.2. Multiprocessor communications and hardware address recognition are supported, as described in **Section 21.2**.

21.2. Multiprocessor Communications

Modes 2 and 3 support multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit and the built-in UART1 address recognition hardware. A master processor begins a transfer with an address byte to select one or more target slave devices. 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.

The UART1 address is configured via two SFRs: SADDR1 (Serial Address) and SADEN1 (Serial Address Enable). SADEN1 sets the bit mask for the address held in SADDR1: bits set to logic 1 in SADEN1 correspond to bits in SADDR1 that are checked against the received address byte; bits set to logic 0 in SADEN1 correspond to “don’t care” bits in SADDR1.

Example 1

SADDR1 = 00110101
SADEN1 = 00001111
UART1 Address = xxxx0101

Example 2

SADDR1 = 00110101
SADEN1 = 11110011
UART1 Address = 0011xx01

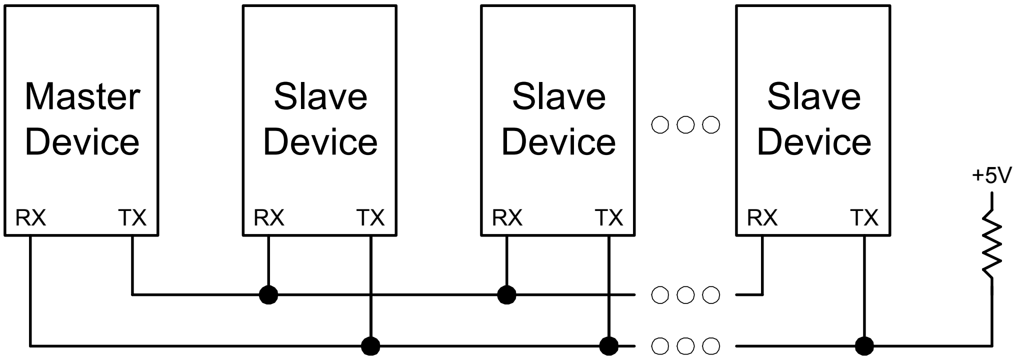
Example 3

SADDR1 = 00110101
SADEN1 = 11000000
UART1 Address = 00xxxxxx

Setting the SM21 bit (SCON1.5) configures UART1 such that when a stop bit is received, UART1 will generate an interrupt only if the ninth bit is logic 1 (RB81 = 1) and the received data byte matches the UART1 slave address. Following the received address interrupt, the slave should clear its SM21 bit to enable interrupts on the reception of the following data byte(s). Once the entire message is received, the addressed slave should reset its SM21 bit to ignore all transmissions until it receives the next address byte. While SM21 is logic 1, UART1 ignores all bytes that do not match the UART1 address and include a ninth bit that is logic 1.

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 SADDR1 and SADEN1, 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).

Figure 21.7. UART Multi-Processor Mode Interconnect Diagram



21.3. Frame and Transmission Error Detection

Frame error detection is available in the following modes when the SSTAT1 bit in register PCON is set to logic 1. Note: The SSTAT1 bit must be logic 1 to access any of the status bits (FE1, RXOVR1, and TXCOL1). To access the UART1 Mode Select bits (SM01, SM11, and SM21), the SSTAT1 bit must be logic 0.

All Modes:
The Transmit Collision bit (TXCOL1 bit in register SCON1) reads ‘1’ if user software writes data to the SBUF1 register while a transmit is in progress. Note that the TXCOL1 bit also functions as the SM21 bit when the SSTAT1 bit in register PCON is logic 0.

Modes 1, 2, and 3:
The Receive Overrun bit (RXOVR1 in register SCON1) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. Note that the RXOVR1 bit also functions as the SM11 bit when the SSTAT1 bit in register PCON is logic 0.

The Frame Error bit (FE1 in register SCON1) reads ‘1’ if an invalid (low) STOP bit is detected. Note that the FE1 bit also functions as the SM01 bit when the SSTAT1 bit in register PCON is logic 0.

Table 21.2. Oscillator Frequencies for Standard Baud Rates

Oscillator frequency (MHz)	Divide Factor	Timer 1 Load Value*	Resulting Baud Rate (Hz)**
25.0	434	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	320	0xEC	76800
24.576	848	0xCB	28800 (28921)
24.0	208	0XF3	115200 (115384)
24.0	833	0xCC	28800 (28846)
23.592	205	0xF3	115200 (113423)
23.592	819	0xCD	28800 (28911)
22.1184	192	0xF4	115200
22.1184	768	0xD0	28800
18.432	160	0xF6	115200
18.432	640	0xD8	28800
16.5888	144	0xF7	115200
16.5888	576	0xDC	28800
14.7456	128	0xF8	115200
14.7456	512	0xE0	28800
12.9024	112	0xF9	115200
12.9024	448	0xE4	28800
11.0592	96	0xFA	115200
11.0592	348	0xE8	28800
9.216	80	0xFB	115200
9.216	320	0xEC	28800
7.3728	64	0xFC	115200
7.3728	256	0xF0	28800
5.5296	48	0xFD	115200
5.5296	192	0xF4	28800
3.6864	32	0xFE	115200
3.6864	128	0xFC	

Figure 21.8. SCON1: UART1 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SM01/FE1	SM11/RXOV1	SM21/TXCOL1	REN1	TB81	RB81	TI1	RI1	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF1

Bits7-6: The function of these bits is determined by the SSTAT1 bit in register PCON.
If SSTAT1 is logic 1, these bits are UART1 status indicators as described in Section 21.3.
If SSTAT1 is logic 0, these bits select the Serial Port Operation Mode as shown below.
SM01-SM11: Serial Port Operation Mode:

SM01	SM11	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

Bit5: SM21: Multiprocessor Communication Enable.
If SSTAT1 is logic 1, this bit is a UART1 status indicator as described in Section 21.3.
If SSTAT1 is logic 0, 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: RI1 will only be activated if stop bit is logic level 1.
Modes 2 and 3: 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 and the received address matches the UART1 address or the broadcast address.

Bit4: REN1: Receive Enable.
This bit enables/disables the UART1 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 Modes 2 and 3. It is not used in Modes 0 and 1. Set or cleared by software as required.

Bit2: RB81: Ninth Receive Bit.
The bit is assigned the logic level of the ninth bit received in Modes 2 and 3. In Mode 1, if SM21 is logic 0, RB81 is assigned the logic level of the received stop bit. RB8 is not used in Mode 0.

Bit1: TI1: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART1 (after the 8th bit in Mode 0, or at the beginning of the stop bit in other modes). 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 by hardware when a byte of data has been received by UART1 (as selected by the SM21 bit). 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.

Figure 21.9. SBUF1: UART1 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: 0xF2

Bits7-0: SBUF1.[7:0]: UART1 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.

Figure 21.10. SADDR1: UART1 Slave Address 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: 0xF3

Bits7-0: SADDR1.[7:0]: UART1 Slave Address
The contents of this register are used to define the UART1 slave address. Register SADEN1 is a bit mask to determine which bits of SADDR1 are checked against a received address: corresponding bits set to logic 1 in SADEN1 are checked; corresponding bits set to logic 0 are “don’t cares”.

Figure 21.11. SADEN1: UART1 Slave Address Enable 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: 0xAE

Bits7-0: SADEN1.[7:0]: UART1 Slave Address Enable
Bits in this register enable corresponding bits in register SADDR1 to determine the UART1 slave address.
0: Corresponding bit in SADDR1 is a “don’t care”.
1: Corresponding bit in SADDR1 is checked against a received address.

22. TIMERS

The C8051F020/1/2/3 devices contain 5 counter/timers: three are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timers for use with the ADCs, SMBus, UART1, or for general purpose use. These can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 offers additional capabilities not available in Timers 0 and 1. Timer 3 is similar to Timer 2, but without the capture or Baud Rate Generator modes. Timer 4 is identical to Timer 2, and can supply baud-rate generation capabilities to UART1.

Timer 0 and Timer 1:	**Timer 2:**	**Timer 3:**	**Timer 4**
13-bit counter/timer	16-bit counter/timer with auto-reload	16-bit timer with auto-reload	16-bit counter/timer with auto-reload
16-bit counter/timer	16-bit counter/timer with capture		16-bit counter/timer with capture
8-bit counter/timer with auto-reload	Baud rate generator for UART0		Baud rate generator for UART1
Two 8-bit counter/timers (Timer 0 only)			

When functioning as a timer, the counter/timer registers are incremented on each clock tick. Clock ticks are derived from the system clock divided by either one or twelve as specified by the Timer Clock Select bits (T4M-T0M) in CKCON, shown in Figure 22.1. The twelve-clocks-per-tick option provides compatibility with the older generation of the 8051 family. Applications that require a faster timer can use the one-clock-per-tick option.

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 level for at least two full system clock cycles to ensure the level is sampled.

Figure 22.1. CKCON: Clock Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	T4M	T2M	T1M	T0M	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8E

Bit7: UNUSED. Read = 0b, Write = don’t care.

Bit6: T4M: Timer 4 Clock Select.
This bit controls the division of the system clock supplied to Timer 4. This bit is ignored when the timer is in baud rate generator mode or counter mode (i.e. C/T4 = 1).
0: Timer 4 uses the system clock divided by 12.
1: Timer 4 uses the system clock.

Bit5: T2M: Timer 2 Clock Select.
This bit controls the division of the system clock supplied to Timer 2. This bit is ignored when the timer is in baud rate generator mode or counter mode (i.e. C/T2 = 1).
0: Timer 2 uses the system clock divided by 12.
1: Timer 2 uses the system clock.

Bit4: T1M: Timer 1 Clock Select.
This bit controls the division of the system clock supplied to Timer 1.
0: Timer 1 uses the system clock divided by 12.
1: Timer 1 uses the system clock.

Bit3: T0M: Timer 0 Clock Select.
This bit controls the division of the system clock supplied to Counter/Timer 0.
0: Counter/Timer uses the system clock divided by 12.
1: Counter/Timer uses the system clock.

Bits2-0: Reserved. Read = 000b, Must Write = 000.

22.1. Timer 0 and Timer 1

Timer 0 and Timer 1 are accessed and controlled through SFRs. Each counter/timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control (TCON) register is used to enable Timer 0 and Timer 1 as well as indicate their status. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits M1-M0 in the Counter/Timer Mode (TMOD) register. Each timer can be configured independently. Following is a detailed description of each operating mode.

22.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as a 13-bit counter/timer in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. Clearing C/T selects the system clock as the input for the timer. When C/T0 is set to logic 1, high-to-low transitions at the selected input pin (T0) increment the timer register. (Refer to **Section "17.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 163** for information on selecting and configuring external I/O pins for digital peripherals.)

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is 0 or the input signal /INT0 is logic-level one. Setting GATE0 to logic 1 allows the timer to be controlled by the external input signal /INT0, facilitating pulse width measurements.

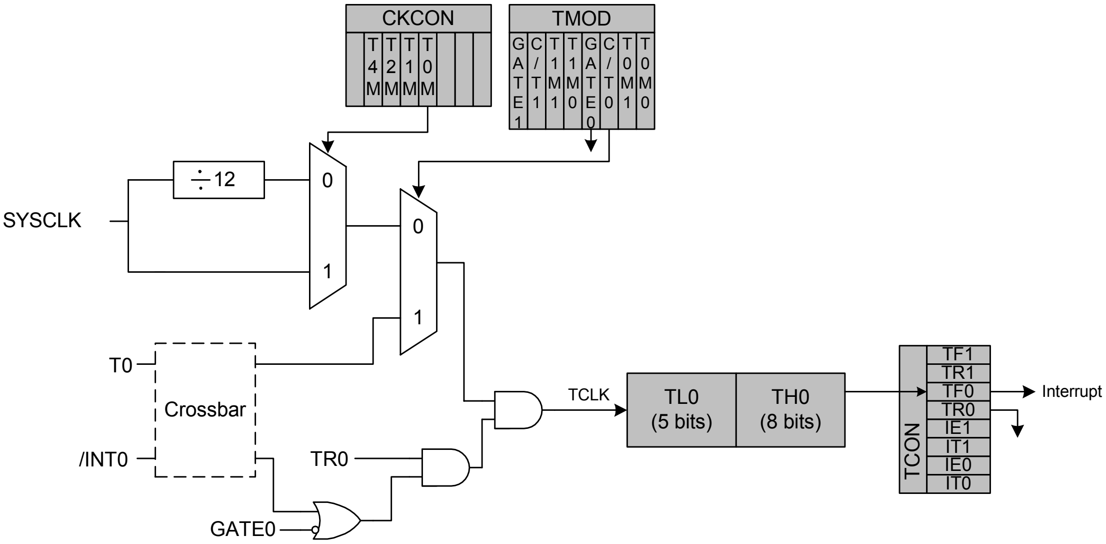
TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

X = Don't Care

Setting TR0 does not reset the timer register. The timer register should be initialized to the desired value before enabling the timer.

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.

Figure 22.2. T0 Mode 0 Block Diagram


CKCON
T4M
T2M
T1M
T0M
TMOD
GATE1
C/T1
T1M1
T1M0
GATE0
C/T0
T0M1
T0M0
÷12
SYSCLK
0
1
0
1
T0
Crossbar
/INT0
TR0
GATE0
TCLK
TL0
(5 bits)
TH0
(8 bits)
TCON
TF1
TR1
TF0
TR0
IE1
IT1
IE0
IT0
Interrupt

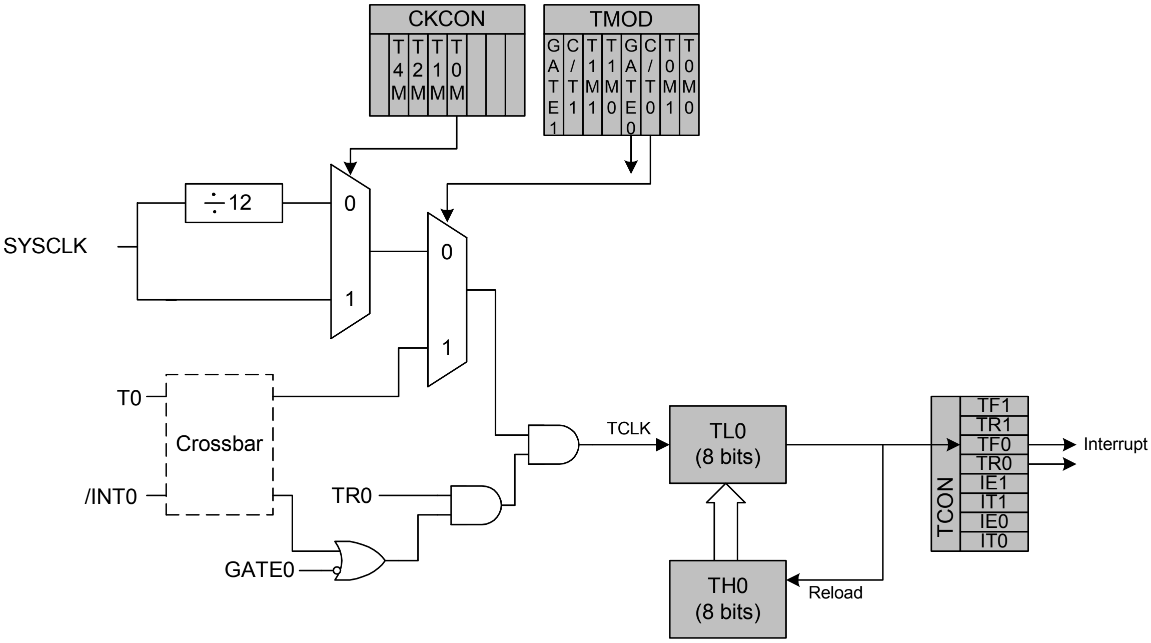
22.1.2. Mode 1: 16-bit Counter/Timer

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.

22.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. The TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter value in TL0 is reloaded from TH0. If 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.

Figure 22.3. T0 Mode 2 (8-bit Auto-Reload) Block Diagram


CKCON
T4M
T2M
T1M
T0M
TMOD
GATE1
C/T1
T1M1
T1M0
GATE0
C/T0
T0M1
T0M0
SYSCLK
÷12
0
1
0
1
T0
/INT0
Crossbar
TR0
GATE0
TCLK
TL0
(8 bits)
TH0
(8 bits)
Reload
TCON
TF1
TR1
TF0
TR0
IE1
IT1
IE0
IT0
Interrupt

22.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

Timer 0 and Timer 1 behave differently 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. It can use either the system clock or an external input signal as its timebase. The timer in the TH0 register is restricted to a timer function sourced by the system 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, so with Timer 0 in Mode 3, Timer 1 can be turned off and on by switching it into and out of its Mode 3. When Timer 0 is 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 the baud clock for UART0 and/or UART1. Refer to **Section "20. UART0" on page 205** and **Section "21. UART1" on page 215** for information on configuring Timer 1 for baud rate generation.

Figure 22.4. T0 Mode 3 (Two 8-bit Timers) Block Diagram

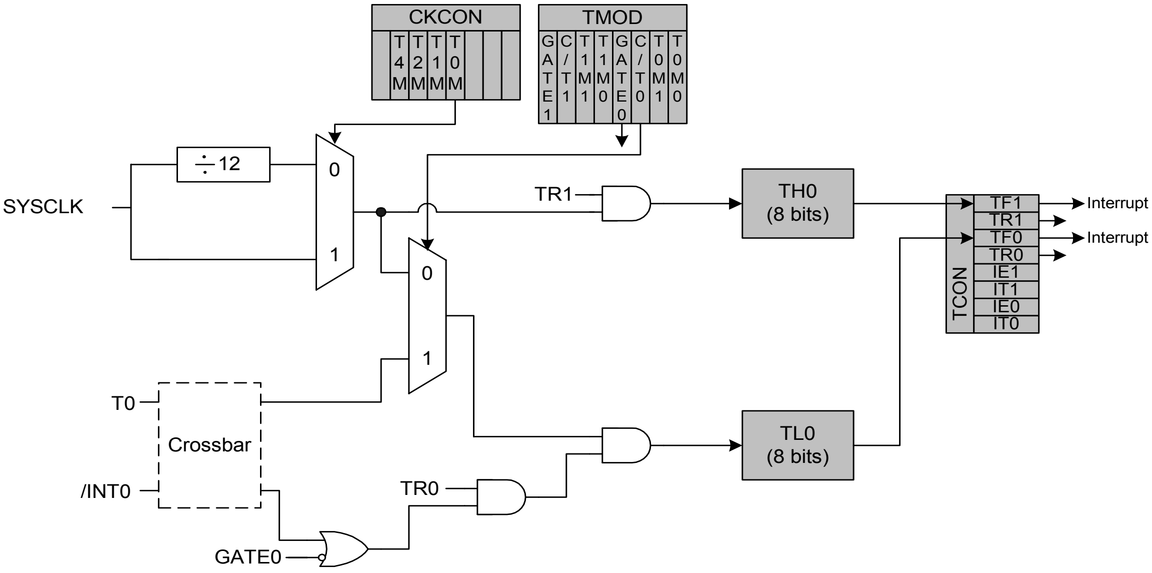

CKCON
T4M
T2M
T1M
T0M
TMOD
GATE1
C/T1
T1M1
T1M0
GATE0
C/T0
T0M1
T0M0
SYSCLK
÷12
0
1
TR1
TH0 (8 bits)
TF1
TR1
TF0
TR0
IE1
IT1
IE0
IT0
TCON
Interrupt
Interrupt
0
1
T0
Crossbar
/INT0
TR0
GATE0
TL0 (8 bits)

Figure 22.5. TCON: Timer Control Register

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.

Figure 22.6. 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

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 22.7. TL0: Timer 0 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8A

Bits 7-0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

Figure 22.8. TL1: Timer 1 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8B

Bits 7-0: TH0: Timer 0 High Byte.
The TH0 register is the low byte of the 16-bit Timer 1.

Figure 22.9. TH0 Timer 0 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8C

Bits 7-0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

Figure 22.10. TH1: Timer 1 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8D

Bits 7-0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

22.1. Timer 2

Timer 2 is a 16-bit counter/timer formed by the two 8-bit SFRs: TL2 (low byte) and TH2 (high byte). As with Timers 0 and 1, Timer 2 can use either the system clock or transitions on an external input pin (T2) as its clock source. The Counter/Timer Select bit C/T2 bit (T2CON.1) selects the clock source for Timer 2. Clearing C/T2 selects the system clock as the input for the timer (divided by either one or twelve as specified by the Timer Clock Select bit T2M in CKCON). When C/T2 is set to 1, high-to-low transitions at the T2 input pin increment the counter/timer register. (Refer to **Section "17.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 163** for information on selecting and configuring external I/O pins for digital peripherals.) Timer 2 can also be used to start an ADC Data Conversion.

Timer 2 offers capabilities not found in Timer 0 and Timer 1. It operates in one of three modes: 16-bit Counter/Timer with Capture, 16-bit Counter/Timer with Auto-Reload or Baud Rate Generator Mode. Timer 2's operating mode is selected by setting configuration bits in the Timer 2 Control register (T2CON). Below is a summary of the Timer 2 operating modes and the T2CON bits used to configure the counter/timer. Detailed descriptions of each mode follow.

RCLK0	TCLK0	CP/RL2	TR2	Mode
0	0	1	1	16-bit Counter/Timer with Capture
0	0	0	1	16-bit Counter/Timer with Auto-Reload
0	1	X	1	Baud Rate Generator for UART0
1	0	X	1	Baud Rate Generator for UART0
1	1	X	1	Baud Rate Generator for UART0
X	X	X	0	Off

22.1.1. Mode 0: 16-bit Counter/Timer with Capture

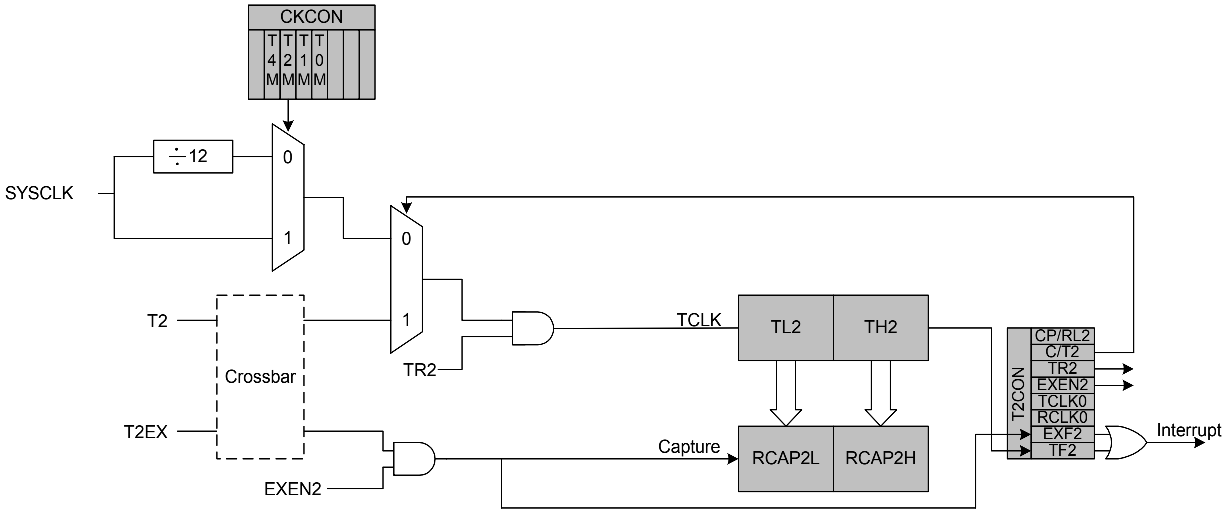
In this mode, Timer 2 operates as a 16-bit counter/timer with capture facility. A high-to-low transition on the T2EX input pin causes the following to occur:

1. The 16-bit value in Timer 2 (TH2, TL2) is loaded into the capture registers (RCAP2H, RCAP2L).
2. The Timer 2 External Flag (EXF2) is set to ‘1’.
3. A Timer 2 interrupt is generated if enabled.

Timer 2 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the T2 input pin as its clock source when operating in Capture mode. Clearing the C/T2 bit (T2CON.1) selects the system clock as the input for the timer (divided by one or twelve as specified by the Timer Clock Select bit T2M in CKCON). When C/T2 is set to logic 1, a high-to-low transition at the T2 input pin increments the counter/timer register. As the 16-bit counter/timer register increments and overflows from 0xFFFF to 0x0000, the TF2 timer overflow flag (T2CON.7) is set and an interrupt will occur if the interrupt is enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RL2 (T2CON.0) and the Timer 2 Run Control bit TR2 (T2CON.2) to logic 1. The Timer 2 External Enable EXEN2 (T2CON.3) must also be set to logic 1 to enable a capture. If EXEN2 is cleared, transitions on T2EX will be ignored.

Figure 22.11. T2 Mode 0 Block Diagram

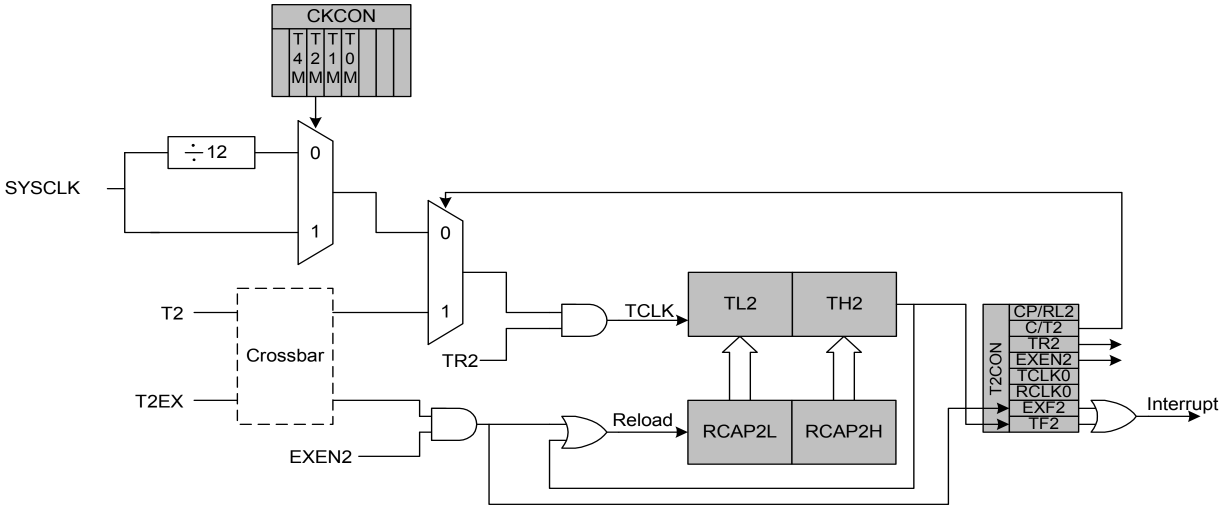

CKCON
T4M
T2M
T1M
T0M
÷12
SYSCLK
0
1
0
1
T2
Crossbar
T2EX
TR2
EXEN2
TCLK
TL2
TH2
Capture
RCAP2L
RCAP2H
T2CON
CP/RL2
C/T2
TR2
EXEN2
TCLK0
RCLK0
EXF2
TF2
Interrupt

22.1.2. Mode 1: 16-bit Counter/Timer with Auto-Reload

The Counter/Timer with Auto-Reload mode sets the TF2 timer overflow flag when the counter/timer register overflows from 0xFFFF to 0x0000. An interrupt is generated if enabled. On overflow, the 16-bit value held in the two capture registers (RCAP2H, RCAP2L) is automatically loaded into the counter/timer register and the timer is restarted.

Counter/Timer with Auto-Reload mode is selected by clearing the CP/RL2 bit. Setting TR2 to logic 1 enables and starts the timer. Timer 2 can use either the system clock or transitions on an external input pin (T2) as its clock source, as specified by the C/T2 bit. If EXEN2 is set to logic 1, a high-to-low transition on T2EX will also cause a Timer 2 reload, and a Timer 2 interrupt if enabled. If EXEN2 is logic 0, transitions on T2EX will be ignored.

Figure 22.12. T2 Mode 1 Block Diagram


CKCON
T4M
T2M
T1M
T0M
÷12
0
1
SYSCLK
0
1
T2
Crossbar
T2EX
TR2
EXEN2
TCLK
TL2
TH2
Reload
RCAP2L
RCAP2H
T2CON
CP/RL2
C/T2
TR2
EXEN2
TCLK0
RCLK0
EXF2
TF2
Interrupt

22.1.3. Mode 2: Baud Rate Generator

Timer 2 can be used as a baud rate generator for UART0 when UART0 is operated in modes 1 or 3 (refer to **Section "20.1. UART0 Operational Modes" on page 206** for more information on the UART0 operational modes). In Baud Rate Generator mode, Timer 2 works similarly to the auto-reload mode. On overflow, the 16-bit value held in the two capture registers (RCAP2H, RCAP2L) is automatically loaded into the counter/timer register. However, the TF2 overflow flag is not set and no interrupt is generated. Instead, the overflow event is used as the input to the UART's shift clock. Timer 2 overflows can be selected to generate baud rates for transmit and/or receive independently.

The Baud Rate Generator mode is selected by setting RCLK0 (T2CON.5) and/or TCLK0 (T2CON.2) to '1'. When RCLK0 or TCLK0 is set to logic 1, Timer 2 operates in the auto-reload mode regardless of the state of the CP/RL2 bit. Note that in Baud Rate Generator mode, the Timer 2 timebase is the system clock divided by two. When selected as the UART0 baud clock source, Timer 2 defines the UART0 baud rate as follows:

$$Baud\ Rate = SYSCLK / ((65536 - [RCAP2H, RCAP2L]) * 32)$$

If a different time base is required, setting the C/T2 bit to logic 1 will allow the timebase to be derived from the external input pin T2. In this case, the baud rate for the UART is calculated as:

$$Baud\ Rate = F_{CLK} / ((65536 - [RCAP2H, RCAP2L]) * 16)$$

Where F_{CLK} is the frequency of the signal (TCLK) supplied to Timer 2 and [RCAP2H, RCAP2L] is the 16-bit value held in the capture registers.

As explained above, in Baud Rate Generator mode, Timer 2 does not set the TF2 overflow flag and therefore cannot generate an interrupt. However, if EXEN2 is set to logic 1, a high-to-low transition on the T2EX input pin will set the EXF2 flag and a Timer 2 interrupt will occur if enabled. Therefore, the T2EX input may be used as an additional external interrupt source.

Figure 22.13. T2 Mode 2 Block Diagram

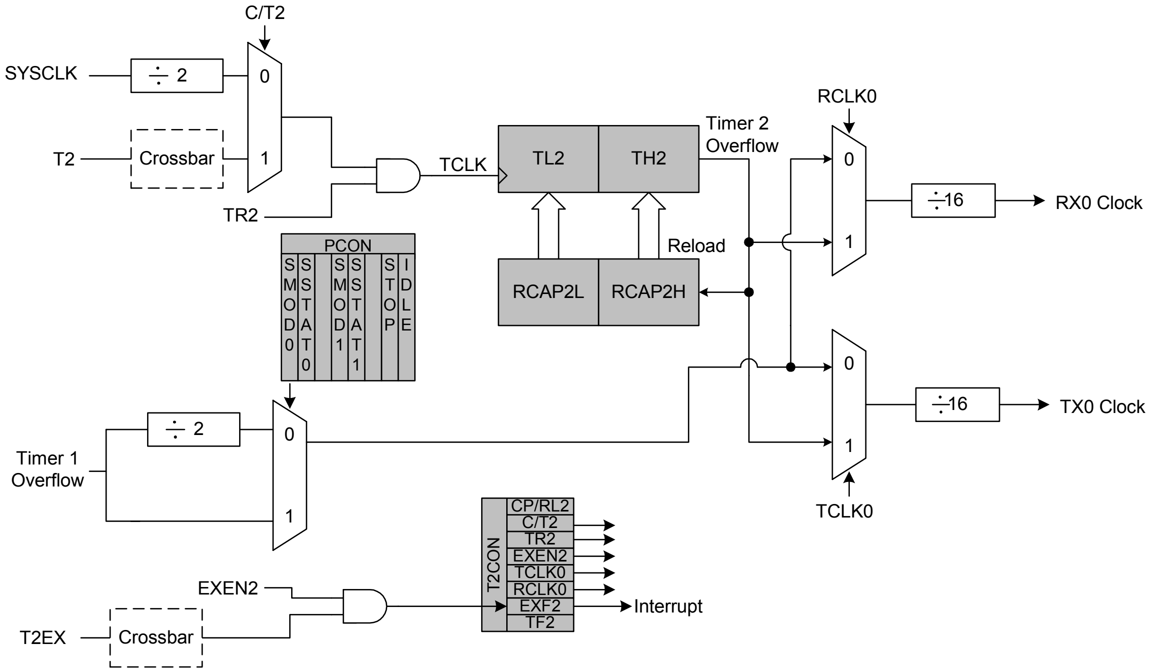

C/T2
SYSCLK
÷ 2
0
1
T2
Crossbar
TR2
TCLK
TL2
TH2
Timer 2 Overflow
RCLK0
0
1
÷16
RX0 Clock
Reload
RCAP2L
RCAP2H
PCON
SMOD0
SSTAT0
SMOD1
SSTAT1
STOP
IDLE
Timer 1 Overflow
÷ 2
0
1
0
1
÷16
TX0 Clock
TCLK0
T2CON
CP/RL2
C/T2
TR2
EXEN2
TCLK0
RCLK0
EXF2
TF2
Interrupt
EXEN2
T2EX
Crossbar

Figure 22.14. T2CON: Timer 2 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF2	EXF2	RCLK0	TCLK0	EXEN2	TR2	C/T2	CP/RL2	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xC8

Bit7: TF2: Timer 2 Overflow Flag.
Set by hardware when Timer 2 overflows. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software. TF2 will not be set when RCLK0 and/or TCLK0 are logic 1.

Bit6: EXF2: Timer 2 External Flag.
Set by hardware when either a capture or reload is caused by a high-to-low transition on the T2EX input pin and EXEN2 is logic 1. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 Interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit5: RCLK0: Receive Clock Flag for UART0.
Selects which timer is used for the UART0 receive clock in modes 1 or 3.
0: Timer 1 overflows used for receive clock.
1: Timer 2 overflows used for receive clock.

Bit4: TCLK0: Transmit Clock Flag for UART0.
Selects which timer is used for the UART0 transmit clock in modes 1 or 3.
0: Timer 1 overflows used for transmit clock.
1: Timer 2 overflows used for transmit clock.

Bit3: EXEN2: Timer 2 External Enable.
Enables high-to-low transitions on T2EX to trigger captures or reloads when Timer 2 is not operating in Baud Rate Generator mode.
0: High-to-low transitions on T2EX ignored.
1: High-to-low transitions on T2EX cause a capture or reload.

Bit2: TR2: Timer 2 Run Control.
This bit enables/disables Timer 2.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit1: C/T2: Counter/Timer Select.
0: Timer Function: Timer 2 incremented by clock defined by T2M (CKCON.5).
1: Counter Function: Timer 2 incremented by high-to-low transitions on external input pin (T2).

Bit0: CP/RL2: Capture/Reload Select.
This bit selects whether Timer 2 functions in capture or auto-reload mode. EXEN2 must be logic 1 for high-to-low transitions on T2EX to be recognized and used to trigger captures or reloads. If RCLK0 or TCLK0 is set, this bit is ignored and Timer 2 will function in auto-reload mode.
0: Auto-reload on Timer 2 overflow or high-to-low transition at T2EX (EXEN2 = 1).
1: Capture on high-to-low transition at T2EX (EXEN2 = 1).

Figure 22.15. RCAP2L: Timer 2 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: 0xCA

Bits 7-0: RCAP2L: Timer 2 Capture Register Low Byte.
The RCAP2L register captures the low byte of Timer 2 when Timer 2 is configured in capture mode. When Timer 2 is configured in auto-reload mode, it holds the low byte of the reload value.

Figure 22.16. RCAP2H: Timer 2 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: 0xCB

Bits 7-0: RCAP2H: Timer 2 Capture Register High Byte.
The RCAP2H register captures the high byte of Timer 2 when Timer 2 is configured in capture mode. When Timer 2 is configured in auto-reload mode, it holds the high byte of the reload value.

Figure 22.17. TL2: Timer 2 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: 0xCC

Bits 7-0: TL2: Timer 2 Low Byte.
The TL2 register contains the low byte of the 16-bit Timer 2.

Figure 22.18. TH2 Timer 2 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: 0xCD

Bits 7-0: TH2: Timer 2 High Byte.
The TH2 register contains the high byte of the 16-bit Timer 2.

22.2. Timer 3

Timer 3 is a 16-bit timer formed by the two 8-bit SFRs, TMR3L (low byte) and TMR3H (high byte). Timer 3 may be clocked by the external oscillator source (divided by eight) or the system clock (divided by either one or twelve as specified by the Timer 3 Clock Select bit T3M in the Timer 3 Control Register TMR3CN). Timer 3 is always configured as an auto-reload timer, with the reload value held in the TMR3RLL (low byte) and TMR3RLH (high byte) registers.

The Timer 3 external clock source feature offers a real-time clock (RTC) mode. When bit T3XCLK (TMR3CN.0) is set to logic 1, Timer 3 is clocked by the external oscillator input (divided by 8) regardless of the system clock selection. This split clock domain allows Timer 3 to be clocked by a precision external source while the system clock is derived from the high-speed internal oscillator. When T3XCLK is logic 0, the Timer 3 clock source is specified by bit T3M (TMR3CN.1).

Timer 3 can also be used to start an ADC Data Conversion, for SMBus timing (see **Section “18. SYSTEM MANAGEMENT BUS / I2C BUS (SMBUS0)” on page 183**), or as a general-purpose timer. Timer 3 does not have a counter mode.

Figure 22.19. Timer 3 Block Diagram

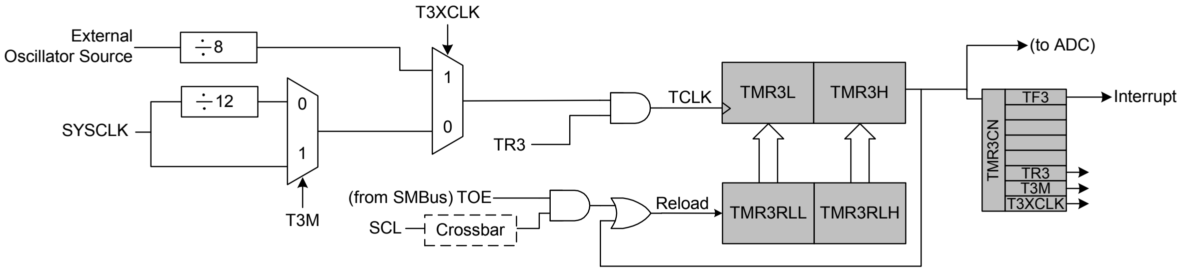

External
Oscillator Source
÷8
T3XCLK
1
0
÷12
0
1
SYSCLK
T3M
TCLK
TR3
TMR3L
TMR3H
(to ADC)
TF3
Interrupt
TMR3CN
TR3
T3M
T3XCLK
(from SMBus) TOE
SCL
Crossbar
Reload
TMR3RLL
TMR3RLH

Figure 22.20. TMR3CN: Timer 3 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF3	-	-	-	-	TR3	T3M	T3XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x91

Bit7: TF3: Timer3 Overflow Flag.
Set by hardware when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 Interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bits6-3: UNUSED. Read = 0000b, Write = don't care.

Bit2: TR3: Timer 3 Run Control.
This bit enables/disables Timer 3.
0: Timer 3 disabled.
1: Timer 3 enabled.

Bit1: T3M: Timer 3 Clock Select.
This bit controls the division of the system clock supplied to Counter/Timer 3.
0: Counter/Timer 3 uses the system clock divided by 12.
1: Counter/Timer 3 uses the system clock.

Bit0: T3XCLK: Timer 3 External Clock Select
This bit selects the external oscillator input divided by 8 as the Timer 3 clock source. When T3XCLK is logic 1, bit T3M (TMR3CN.1) is ignored.
0: Timer 3 clock source defined by bit T3M (TMR3CN.1).
1: Timer 3 clock source is the external oscillator input divided by 8.

Figure 22.21. TMR3RLL: Timer 3 Reload 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: 0x92

Bits 7-0: TMR3RLL: Timer 3 Reload Register Low Byte.
Timer 3 is configured as an auto-reload timer. This register holds the low byte of the reload value.

Figure 22.22. TMR3RLH: Timer 3 Reload 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:
								0x93

Bits 7-0: TMR3RLH: Timer 3 Reload Register High Byte.
Timer 3 is configured as an auto-reload timer. This register holds the high byte of the reload value.

Figure 22.23. TMR3L: Timer 3 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:
								0x94

Bits 7-0: TMR3L: Timer 3 Low Byte.
The TMR3L register is the low byte of Timer 3.

Figure 22.24. TMR3H: Timer 3 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:
								0x95

Bits 7-0: TMR3H: Timer 3 High Byte.
The TMR3H register is the high byte of Timer 3.

22.3. Timer 4

Timer 4 is a 16-bit counter/timer formed by the two 8-bit SFRs: TL4 (low byte) and TH4 (high byte). As with Timers 0 and 1, Timer 4 can use either the system clock or transitions on an external input pin (T4) as its clock source. The Counter/Timer Select bit C/T4 bit (T4CON.1) selects the clock source for Timer 4. Clearing C/T4 selects the system clock as the input for the timer (divided by either one or twelve as specified by the Timer Clock Select bit T4M in CKCON). When C/T4 is set to 1, high-to-low transitions at the T4 input pin increment the counter/timer register. Refer to **Section "17.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 163** for information on selecting and configuring external I/O pins for digital peripherals.

Timer 4 offers capabilities not found in Timer 0 and Timer 1. It operates in one of three modes: 16-bit Counter/Timer with Capture, 16-bit Counter/Timer with Auto-Reload or Baud Rate Generator Mode. Timer 4's operating mode is selected by setting configuration bits in the Timer 4 Control register (T4CON). Below is a summary of the Timer 4 operating modes and the T4CON bits used to configure the counter/timer. Detailed descriptions of each mode follow.

RCLK1	TCLK1	CP/RL4	TR4	Mode
0	0	1	1	16-bit Counter/Timer with Capture
0	0	0	1	16-bit Counter/Timer with Auto-Reload
0	1	X	1	Baud Rate Generator for UART1
1	0	X	1	Baud Rate Generator for UART1
1	1	X	1	Baud Rate Generator for UART1
X	X	X	0	Off

22.3.1. Mode 0: 16-bit Counter/Timer with Capture

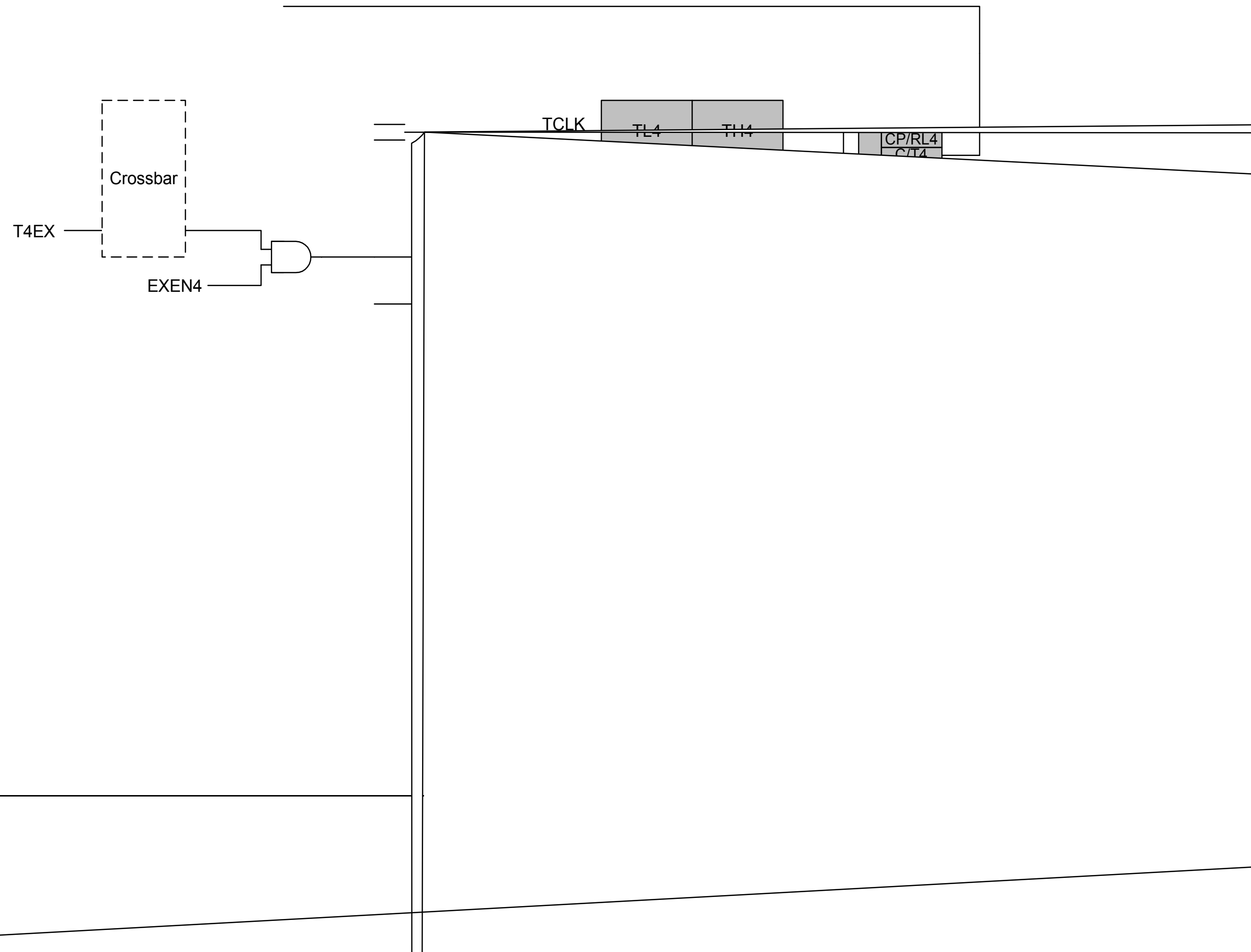
In this mode, Timer 4 operates as a 16-bit counter/timer with capture facility. A high-to-low transition on the T4EX input pin causes the following to occur:

1. The 16-bit value in Timer 4 (TH4, TL4) is loaded into the capture registers (RCAP4H, RCAP4L).
2. The Timer 4 External Flag (EXF2) is set to ‘1’.
3. A Timer 4 interrupt is generated if enabled.

Timer 4 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the T4 input pin as its clock source when operating in Capture mode. Clearing the C/T4 bit (T4CON.1) selects the system clock as the input for the timer (divided by one or twelve as specified by the Timer Clock Select bit T4M in CKCON). When C/T4 is set to logic 1, a high-to-low transition at the T4 input pin increments the counter/timer register. As the 16-bit counter/timer register increments and overflows from 0xFFFF to 0x0000, the TF4 timer overflow flag (T4CON.7) is set and an interrupt will occur if the interrupt is enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RL4 (T4CON.0) and the Timer 4 Run Control bit TR4 (T4CON.2) to logic 1. The Timer 4 External Enable EXEN4 (T4CON.3) must also be set to logic 1 to enable a capture. If EXEN4 is cleared, transitions on T4EX will be ignored.

Figure 22.25. T4 Mode 0 Block Diagram

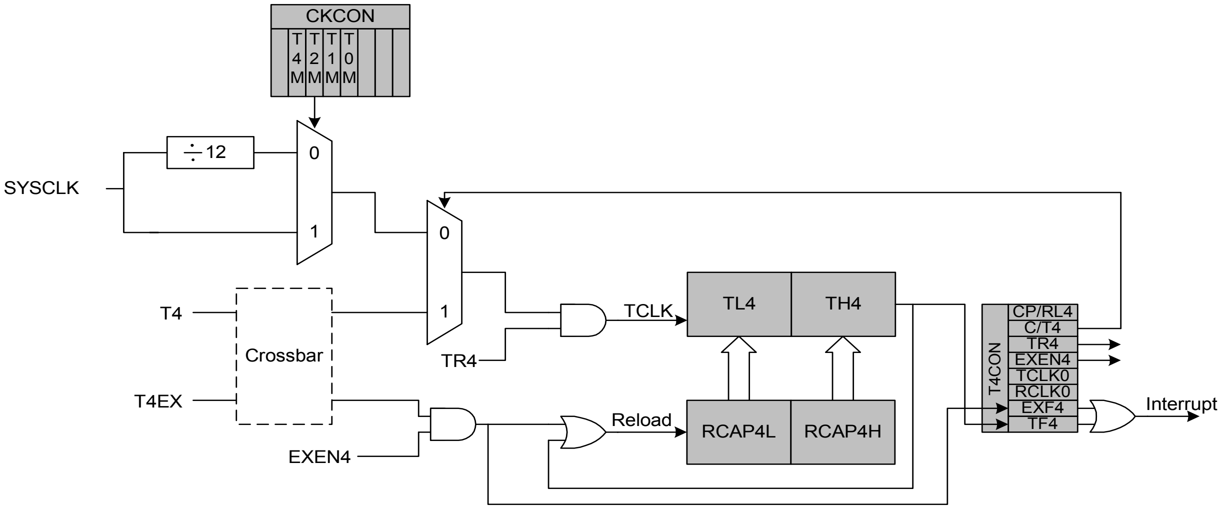


22.3.2. Mode 1: 16-bit Counter/Timer with Auto-Reload

The Counter/Timer with Auto-Reload mode sets the TF4 timer overflow flag when the counter/timer register overflows from 0xFFFF to 0x0000. An interrupt is generated if enabled. On overflow, the 16-bit value held in the two capture registers (RCAP4H, RCAP4L) is automatically loaded into the counter/timer register and the timer is restarted.

Counter/Timer with Auto-Reload mode is selected by clearing the CP/RL4 bit. Setting TR4 to logic 1 enables and starts the timer. Timer 4 can use either the system clock or transitions on an external input pin (T2) as its clock source, as specified by the C/T4 bit. If EXEN4 is set to logic 1, a high-to-low transition on T4EX will also cause a Timer 4 reload, and a Timer 4 interrupt if enabled. If EXEN4 is logic 0, transitions on T4EX will be ignored.

Figure 22.26. T4 Mode 1 Block Diagram


CKCON
T4M
T2M
T1M
T0M
÷12
SYSCLK
0
1
0
1
T4
T4EX
Crossbar
EXEN4
TR4
TCLK
TL4
TH4
Reload
RCAP4L
RCAP4H
T4CON
CP/RL4
C/T4
TR4
EXEN4
TCLK0
RCLK0
EXF4
TF4
Interrupt

22.3.3. Mode 2: Baud Rate Generator

Timer 4 can be used as a baud rate generator for UART1 when UART1 is operated in modes 1 or 3 (refer to **Section "21.1. UART1 Operational Modes" on page 216** for more information on the UART1 operational modes). In Baud Rate Generator mode, Timer 4 works similarly to the auto-reload mode. On overflow, the 16-bit value held in the two capture registers (RCAP4H, RCAP4L) is automatically loaded into the counter/timer register. However, the TF4 overflow flag is not set and no interrupt is generated. Instead, the overflow event is used as the input to the UART's shift clock. Timer 4 overflows can be selected to generate baud rates for transmit and/or receive independently.

The Baud Rate Generator mode is selected by setting RCLK1 (T4CON.5) and/or TCLK1 (T4CON.4) to '1'. When RCLK1 or TCLK1 is set to logic 1, Timer 4 operates in the auto-reload mode regardless of the state of the CP/RL4 bit. Note that in Baud Rate Generator mode, the Timer 4 timebase is the system clock divided by two. When selected as the UART1 baud clock source, Timer 4 defines the UART1 baud rate as follows:

$$Baud\ Rate = SYSCLK / ((65536 - [RCAP4H, RCAP4L]) * 32)$$

If a different time base is required, setting the C/T4 bit to logic 1 will allow the timebase to be derived from the external input pin T4. In this case, the baud rate for the UART is calculated as:

$$Baud\ Rate = F_{CLK} / ((65536 - [RCAP4H, RCAP4L]) * 16)$$

Where F_{CLK} is the frequency of the signal (TCLK) supplied to Timer 4 and [RCAP4H, RCAP4L] is the 16-bit value held in the capture registers.

As explained above, in Baud Rate Generator mode, Timer 4 does not set the TF4 overflow flag and therefore cannot generate an interrupt. However, if EXEN4 is set to logic 1, a high-to-low transition on the T4EX input pin will set the EXF4 flag and a Timer 4 interrupt will occur if enabled. Therefore, the T4EX input may be used as an additional external interrupt source.

Figure 22.27. T4 Mode 2 Block Diagram

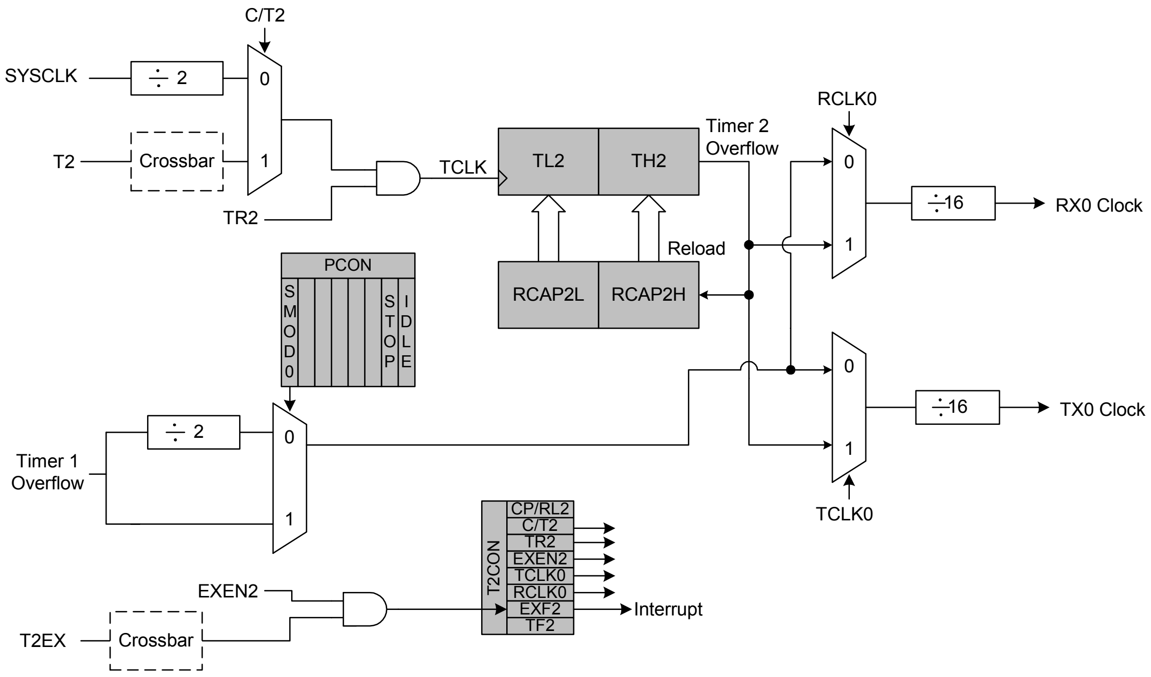

C/T2
SYSCLK
÷ 2
0
1
T2
Crossbar
TR2
TCLK
TL2
TH2
Timer 2 Overflow
RCLK0
Reload
RCAP2L
RCAP2H
÷16
RX0 Clock
PCON
SMOD0
STOP
IDLE
÷16
TX0 Clock
TCLK0
Timer 1 Overflow
÷ 2
T2CON
CP/RL2
C/T2
TR2
EXEN2
TCLK0
RCLK0
EXF2
TF2
Interrupt
EXEN2
T2EX
Crossbar

Figure 22.28. T4CON: Timer 4 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF4	EXF4	RCLK1	TCLK1	EXEN4	TR4	C/T4	CP/RL4	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC9

Bit7: TF4: Timer 4 Overflow Flag.
Set by hardware when Timer 4 overflows. When the Timer 4 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 4 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software. TF4 will not be set when RCLK1 and/or TCLK1 are logic 1.

Bit6: EXF4: Timer 4 External Flag.
Set by hardware when either a capture or reload is caused by a high-to-low transition on the T4EX input pin and EXEN4 is logic 1. When the Timer 4 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 4 Interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit5: RCLK1: Receive Clock Flag for UART1.
Selects which timer is used for the UART1 receive clock in modes 1 or 3.
0: Timer 1 overflows used for receive clock.
1: Timer 4 overflows used for receive clock.

Bit4: TCLK1: Transmit Clock Flag for UART1.
Selects which timer is used for the UART1 transmit clock in modes 1 or 3.
0: Timer 1 overflows used for transmit clock.
1: Timer 4 overflows used for transmit clock.

Bit3: EXEN4: Timer 4 External Enable.
Enables high-to-low transitions on T4EX to trigger captures or reloads when Timer 4 is not operating in Baud Rate Generator mode.
0: High-to-low transitions on T4EX ignored.
1: High-to-low transitions on T4EX cause a capture or reload.

Bit2: TR4: Timer 4 Run Control.
This bit enables/disables Timer 4.
0: Timer 4 disabled.
1: Timer 4 enabled.

Bit1: C/T4: Counter/Timer Select.
0: Timer Function: Timer 4 incremented by clock defined by T4M (CKCON.6).
1: Counter Function: Timer 4 incremented by high-to-low transitions on external input pin (T2).

Bit0: CP/RL4: Capture/Reload Select.
This bit selects whether Timer 4 functions in capture or auto-reload mode. EXEN4 must be logic 1 for high-to-low transitions on T4EX to be recognized and used to trigger captures or reloads. If RCLK1 or TCLK1 is set, this bit is ignored and Timer 4 will function in auto-reload mode.
0: Auto-reload on Timer 4 overflow or high-to-low transition at T4EX (EXEN4 = 1).
1: Capture on high-to-low transition at T4EX (EXEN4 = 1).

Figure 22.29. RCAP4L: Timer 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:
								0xE4

Bits 7-0: RCAP4L: Timer 4 Capture Register Low Byte.
The RCAP4L register captures the low byte of Timer 4 when Timer 4 is configured in capture mode. When Timer 4 is configured in auto-reload mode, it holds the low byte of the reload value.

Figure 22.30. RCAP4H: Timer 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:
								0xE5

Bits 7-0: RCAP4H: Timer 4 Capture Register High Byte.
The RCAP4H register captures the high byte of Timer 4 when Timer 4 is configured in capture mode. When Timer 4 is configured in auto-reload mode, it holds the high byte of the reload value.

Figure 22.31. TL4: Timer 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:
								0xF4

Bits 7-0: TL4: Timer 4 Low Byte.
The TL4 register contains the low byte of the 16-bit Timer 4.

Figure 22.32. TH4 Timer 4 High Byte

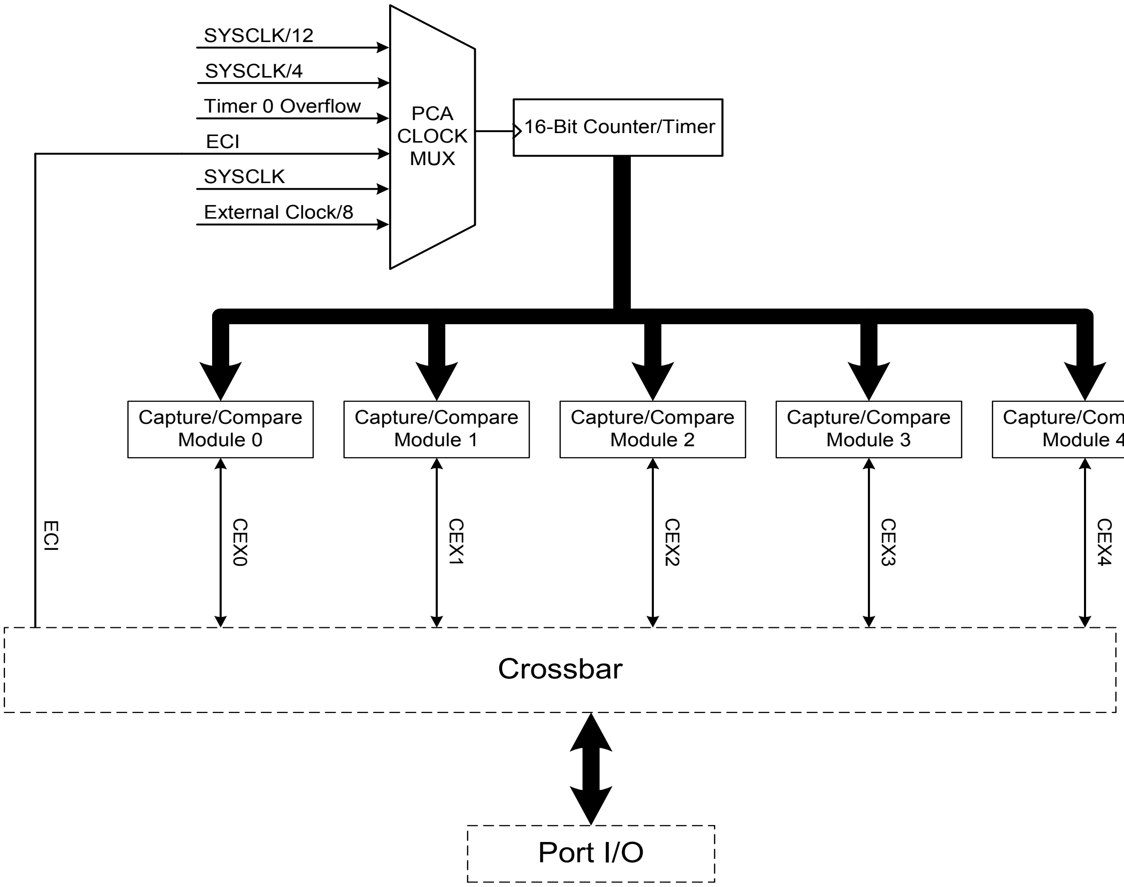
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xF5

Bits 7-0: TH4: Timer 4 High Byte.
The TH4 register contains the high byte of the 16-bit Timer 4.

23. 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 five 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 "17.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 163**). The counter/timer is driven by a programmable timebase that can select between six inputs as its source: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI line. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each is described in **Section 23.2**). The PCA is configured and controlled through the system controller's Special Function Registers. The basic PCA block diagram is shown in Figure 23.1.

Figure 23.1. PCA Block Diagram


SYSCLK/12
SYSCLK/4
Timer 0 Overflow
ECI
SYSCLK
External Clock/8
PCA CLOCK MUX
16-Bit Counter/Timer
Capture/Compare Module 0
Capture/Compare Module 1
Capture/Compare Module 2
Capture/Compare Module 3
Capture/Compare Module 4
ECI
CEX0
CEX1
CEX2
CEX3
CEX4
Crossbar
Port I/O

23.1. 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 23.1. **Note that in ‘External oscillator source divided by 8’ mode, the external oscillator source is synchronized with the system clock, and must have a frequency less than or equal to the system clock.**

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.

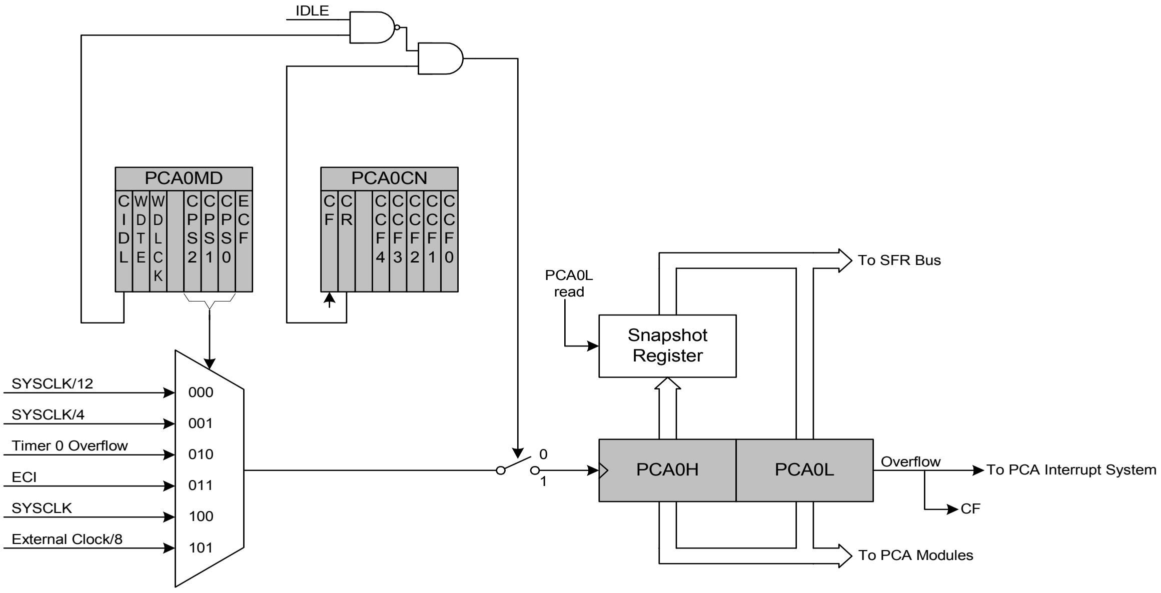
Table 23.1. PCA Timebase Input Options

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†

†External oscillator source divided by 8 is synchronized with the system clock.

‡The minimum high or low time for the ECI input signal is at least 2 system clock cycles.

Figure 23.2. PCA Counter/Timer Block Diagram


IDLE
PCA0MD
CIDL
WDTE
WDLCK
CPS2
CPS1
CPS0
ECF
PCA0CN
CF
CR
CCF4
CCF3
CCF2
CCF1
CCF0
SYSCLK/12
SYSCLK/4
Timer 0 Overflow
ECI
SYSCLK
External Clock/8
000
001
010
011
100
101
0
1
PCA0L read
Snapshot Register
To SFR Bus
PCA0H
PCA0L
Overflow
To PCA Interrupt System
CF
To PCA Modules

Important Note About the PCA0CN Register: If the main PCA counter (PCA0H : PCA0L) overflows during the execution phase of a read-modify-write instruction (bit-wise SETB or CLR, ANL, ORL, XRL) that targets the PCA0CN register, the CF (Counter Overflow) bit will not be set. The following steps should be taken when performing a bit-wise operation on the PCA0CN register:

Step 1. Disable global interrupts (EA = 0).
Step 2. Read PCA0L. This will latch the value of PCA0H.
Step 3. Read PCA0H, saving the value.
Step 4. Execute the bit-wise operation on CCFn (for example, CLR CCF0, or CCF0 = 0;).
Step 5. Read PCA0L.
Step 6. Read PCA0H, saving the value.
Step 7. If the value of PCA0H read in Step 3 is 0xFF and the value for PCA0H read in Step 6 is 0x00, then manually set the CF bit in software (for example, SETB CF, or CF = 1;).
Step 8. Re-enable interrupts (EA = 1).

23.2. 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.

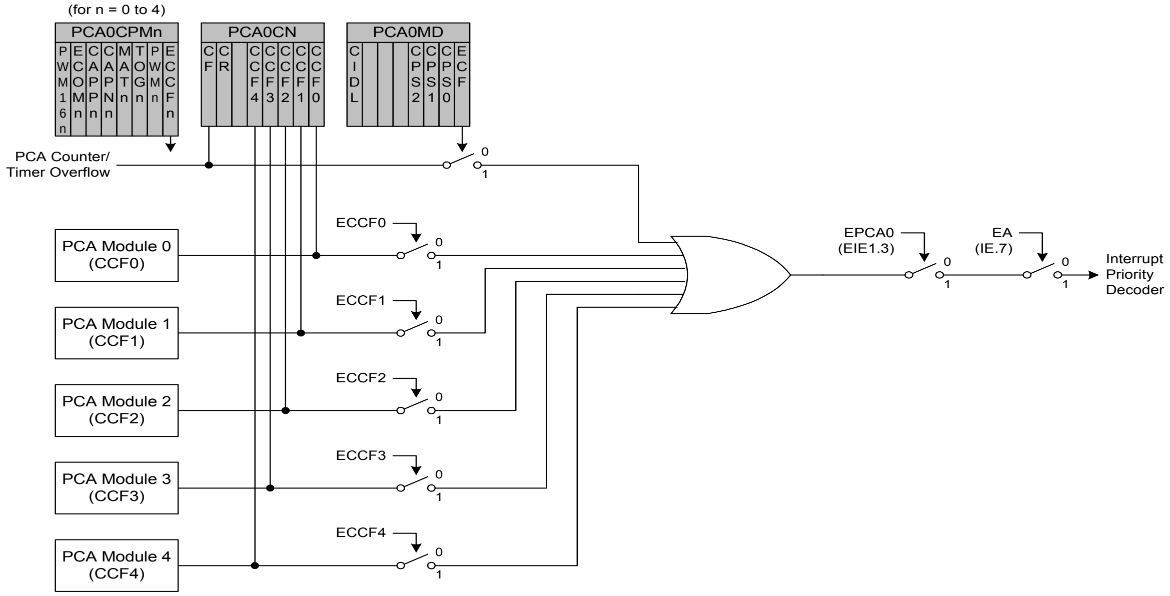
Table 23.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA0 capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic 1. See Figure 23.3 for details on the PCA interrupt configuration.

Table 23.2. 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	X	1	1	X	Frequency Output
0	1	0	0	X	0	1	X	8-Bit Pulse Width Modulator
1	1	0	0	X	0	1	X	16-Bit Pulse Width Modulator

X = Don't Care

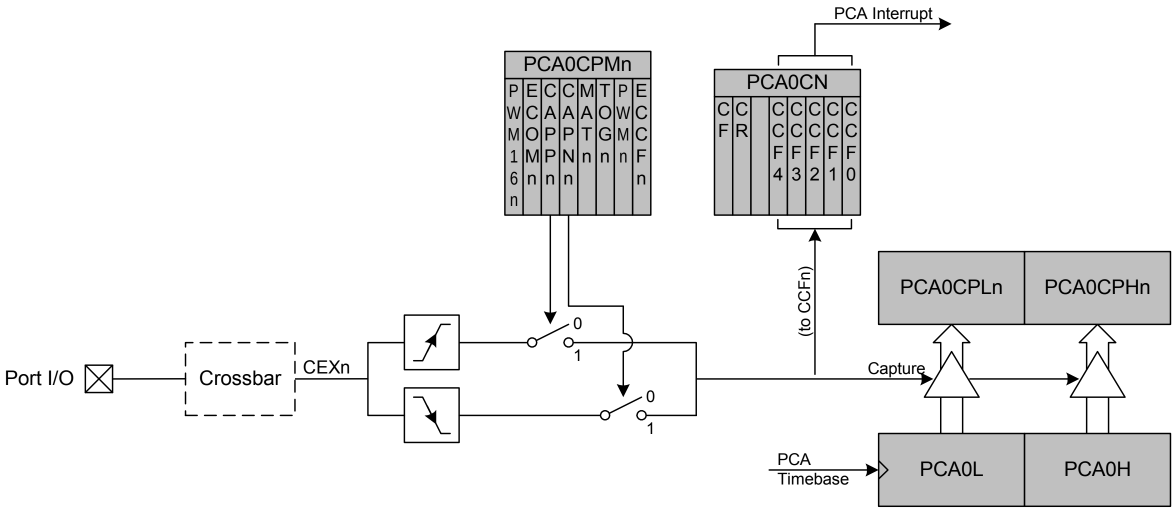
Figure 23.3. PCA Interrupt Block Diagram


(for n = 0 to 4)
PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
PCA0CN
CF
CR
CCF4
CCF3
CCF2
CCF1
CCF0
PCA0MD
CIDL
CPS2
CPS1
CPS0
ECF
PCA Counter/
Timer Overflow
0
1
PCA Module 0
(CCF0)
ECCF0
PCA Module 1
(CCF1)
ECCF1
PCA Module 2
(CCF2)
ECCF2
PCA Module 3
(CCF3)
ECCF3
PCA Module 4
(CCF4)
ECCF4
EPCA0
(EIE1.3)
EA
(IE.7)
Interrupt
Priority
Decoder

23.2.1. 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 23.4. PCA Capture Mode Diagram


PCA Interrupt
PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
PCA0CN
CF
CR
CCF4
CCF3
CCF2
CCF1
CCF0
PCA0CPLn
PCA0CPHn
(to CCFn)
0
1
0
1
Port I/O
Crossbar
CEXn
Capture
PCA Timebase
PCA0L
PCA0H

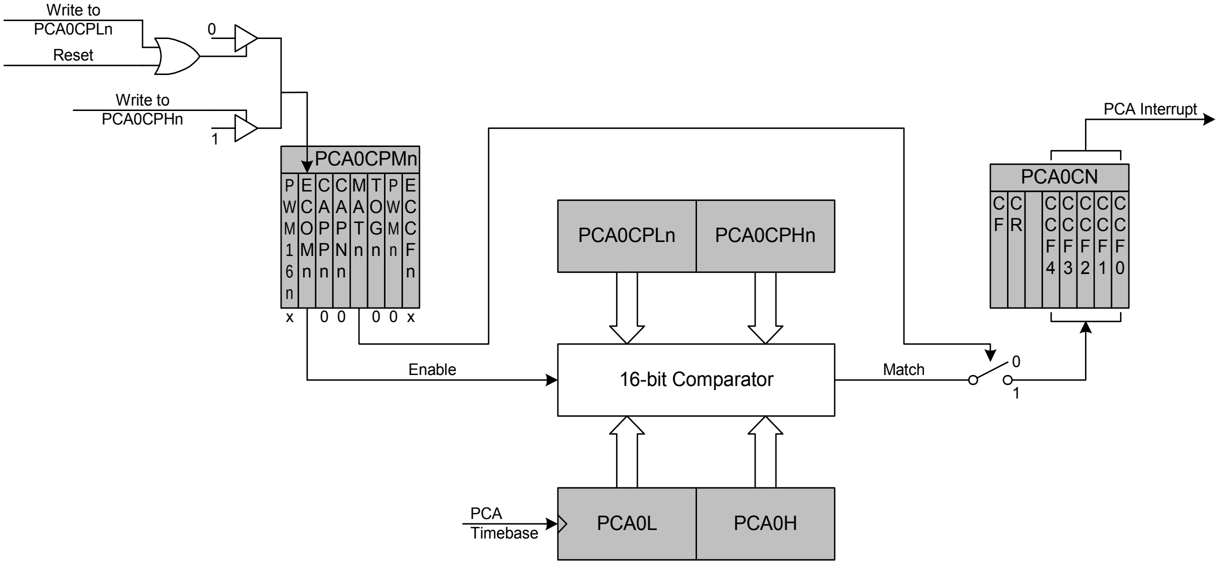
Note: The CEXn input signal must remain high or low for at least 2 system clock cycles in order to be valid.

23.2.2. 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 23.5. PCA Software Timer Mode Diagram



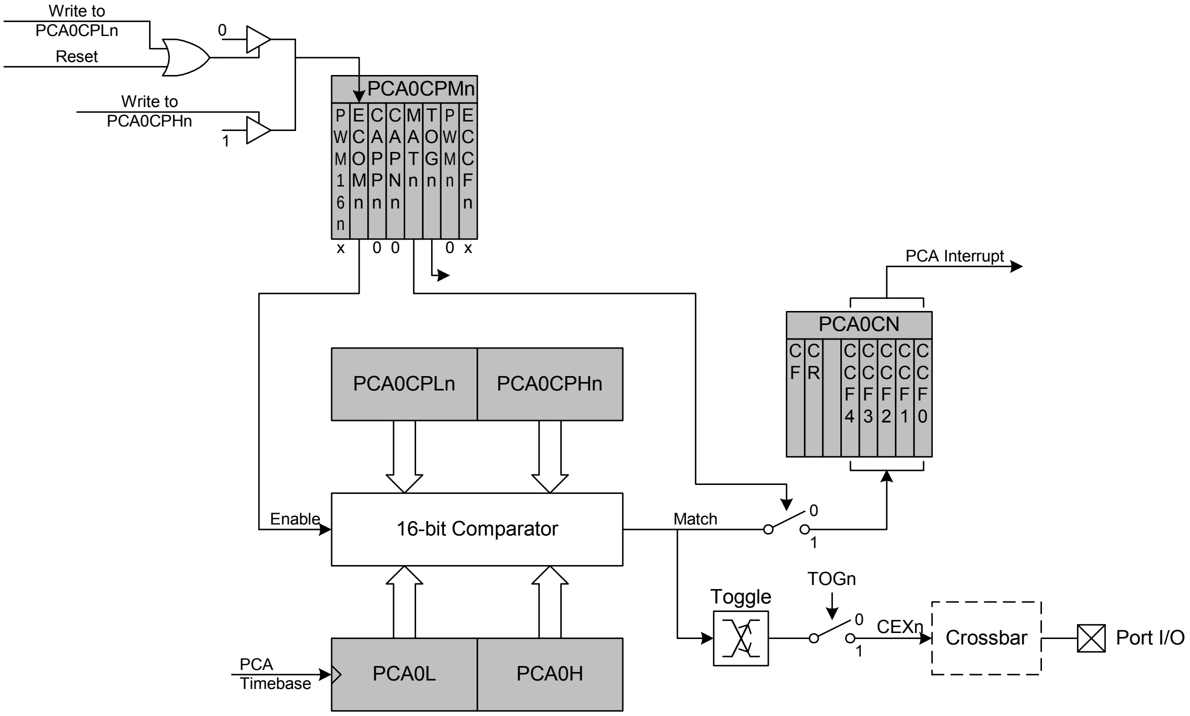
Write to PCA0CPLn
Reset
0
Write to PCA0CPHn
1
PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
x 0 0 0 0 x
PCA0CPLn
PCA0CPHn
Enable
16-bit Comparator
Match
0
1
PCA Interrupt
PCA0CN
CF
CR
CCF4
CCF3
CCF2
CCF1
CCF0
PCA Timebase
PCA0L
PCA0H

23.2.3. 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 23.6. PCA High Speed Output Mode Diagram



Write to PCA0CPLn
Reset
0
Write to PCA0CPHn
1
PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
x 0 0 0 x
PCA Interrupt
PCA0CN
CF
CR
CCF4
CCF3
CCF2
CCF1
CCF0
PCA0CPLn
PCA0CPHn
Enable
16-bit Comparator
Match
0
1
Toggle
TOGn
0
1
CEXn
Crossbar
Port I/O
PCA Timebase
PCA0L
PCA0H

23.2.4. 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 23.1.

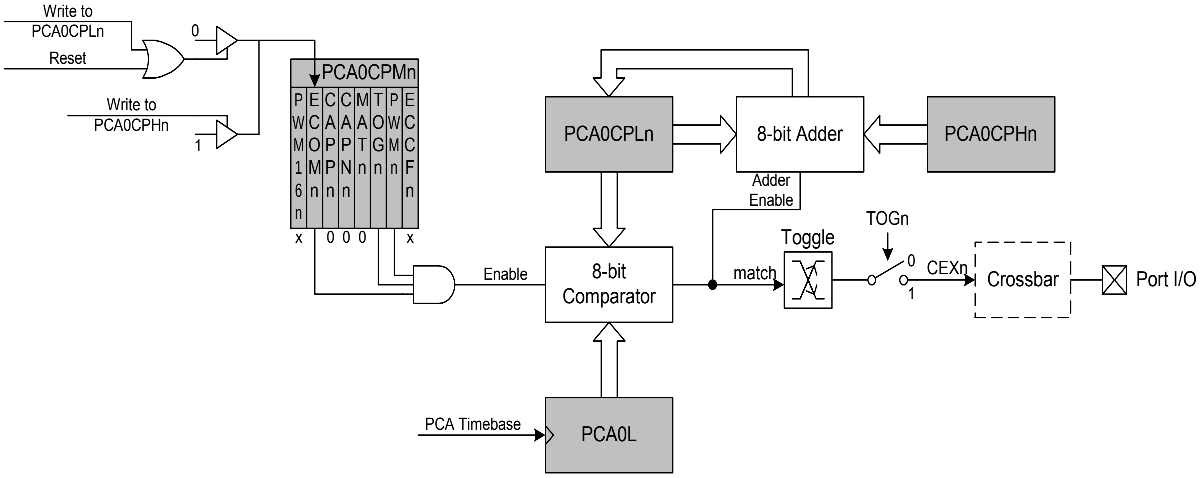
Equation 23.1. Square Wave Frequency Output

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

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.

Figure 23.7. PCA Frequency Output Mode


Write to PCA0CPLn
Reset
0
Write to PCA0CPHn
1
PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
x 0 0 0 x
Enable
8-bit Comparator
PCA0CPLn
8-bit Adder
PCA0CPHn
Adder Enable
Toggle
TOGn
match
0
1
CEXn
Crossbar
Port I/O
PCA Timebase
PCA0L

23.2.5. 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 asserted high. When the count value in PCA0L overflows, the CEXn output will be asserted low (see Figure 23.8). 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 23.2.

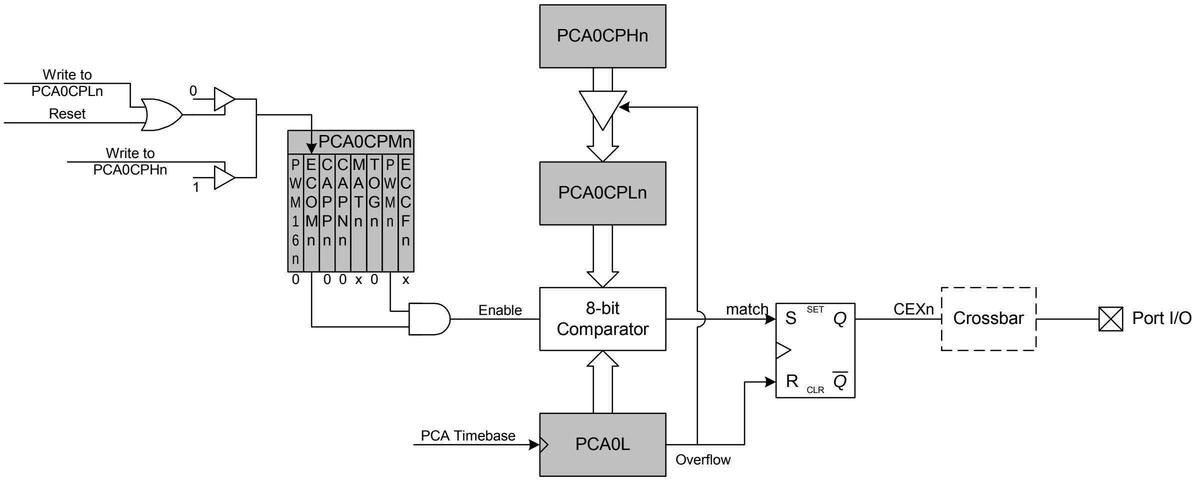
Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to ‘0’; writing to PCA0CPHn sets ECOMn to ‘1’.

Equation 23.2. 8-Bit PWM Duty Cycle

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Using Equation 23.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to ‘0’.

Figure 23.8. PCA 8-Bit PWM Mode Diagram


Write to PCA0CPLn
Reset
0
Write to PCA0CPHn
1
PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
0
0 0 x 0
x
Enable
PCA0CPHn
PCA0CPLn
8-bit Comparator
PCA Timebase
PCA0L
Overflow
match
S
SET
Q
R
CLR
Q
CEXn
Crossbar
Port I/O

23.2.6. 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 23.3.

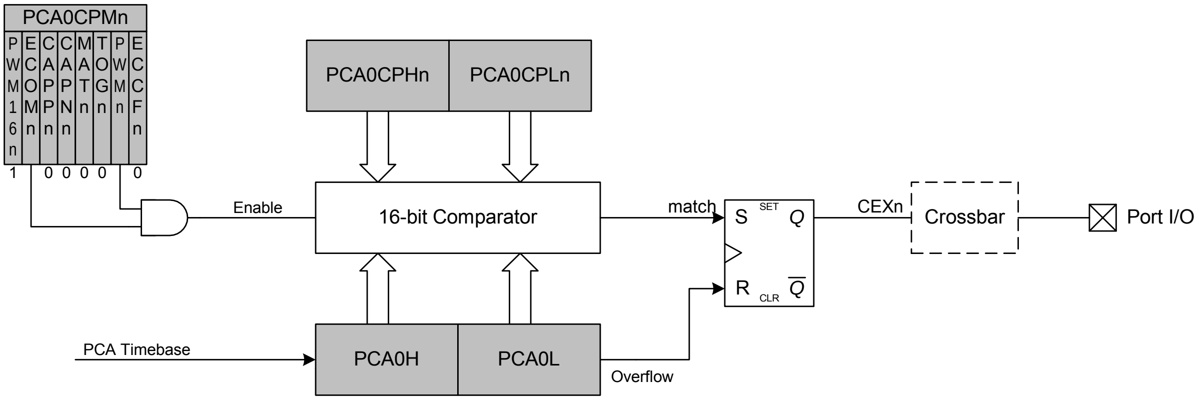
Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to ‘0’; writing to PCA0CPHn sets ECOMn to ‘1’

Equation 23.3. 16-Bit PWM Duty Cycle

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

Using Equation 23.3, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to ‘0’.

Figure 23.9. PCA 16-Bit PWM Mode


PCA0CPMn
PWM16n
ECOMn
CAPPn
CAPNn
MATn
TOGn
PWMn
ECCFn
1
0 0 0 0
0
Enable
PCA0CPHn
PCA0CPLn
16-bit Comparator
match
S
SET
Q
R
CLR
Q̄
CEXn
Crossbar
Port I/O
PCA Timebase
PCA0H
PCA0L
Overflow

23.3. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of PCA0.

Figure 23.10. PCA0CN: PCA Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xD8

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. See “Important Note About the PCA0CN Register” on page 251.

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: UNUSED. Read = 0b, Write = don't care.

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 23.11. PCA0MD: PCA0 Mode Register

Figure 23.12. 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: 0xDA-0xDE

PCA0CPMn Address: PCA0CPM0 = 0xDA (n = 0)
PCA0CPM1 = 0xDB (n = 1)
PCA0CPM2 = 0xDC (n = 2)
PCA0CPM3 = 0xDD (n = 3)
PCA0CPM4 = 0xDE (n = 4)

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 23.13. 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:
								0xE9

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 23.14. 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:
								0xF9

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 23.15. 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: 0xEA - 0xEE

PCA0CPLn Address: PCA0CPL0 = 0xEA (n = 0)
PCA0CPL1 = 0xEB (n = 1)
PCA0CPL2 = 0xEC (n = 2)
PCA0CPL3 = 0xED (n = 3)
PCA0CPL4 = 0xEE (n = 4)

Bits7-0: PCA0CPLn: PCA0 Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

Figure 23.16. 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: 0xFA - 0xFE

PCA0CPHn Address: PCA0CPH0 = 0xFA (n = 0)
PCA0CPH1 = 0xFB (n = 1)
PCA0CPH2 = 0xFC (n = 2)
PCA0CPH3 = 0xFD (n = 3)
PCA0CPH4 = 0xFE (n = 4)

Bits7-0: PCA0CPHn: PCA0 Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

Notes

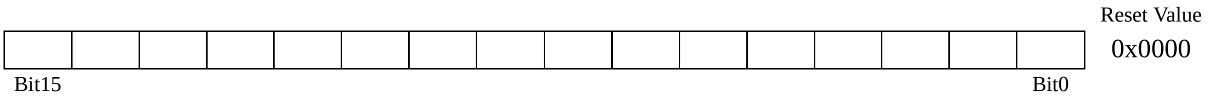
24. JTAG (IEEE 1149.1)

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 seven instructions shown in Figure 24.1 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 24.1. 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

24.1. Boundary Scan

The DR in the Boundary Scan path is an 134-bit shift register. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature via the EXTEST and SAMPLE commands.

Table 24.1. Boundary Data Register Bit Definitions

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

Bit	Action	Target
0	Capture	Reset Enable from MCU (C8051F021/3 devices)
	Update	Reset Enable to /RST pin (C8051F021/3 devices)
1	Capture	Reset input from /RST pin (C8051F021/3 devices)
	Update	Reset output to /RST pin (C8051F021/3 devices)
2	Capture	Reset Enable from MCU (C8051F020/2 devices)
	Update	Reset Enable to /RST pin (C8051F020/2 devices)
3	Capture	Reset input from /RST pin (C8051F020/2 devices)
	Update	Reset output to /RST pin (C8051F020/2 devices)
4	Capture	External Clock from XTAL1 pin
	Update	Not used
5	Capture	Weak pullup enable from MCU
	Update	Weak pullup enable to Port Pins
6, 8, 10, 12, 14, 16, 18, 20	Capture	P0.n output enable from MCU (e.g. Bit6=P0.0, Bit8=P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit6=P0.0oe, Bit8=P0.1oe, etc.)
7, 9, 11, 13, 15, 17, 19, 21	Capture	P0.n input from pin (e.g. Bit7=P0.0, Bit9=P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit7=P0.0, Bit9=P0.1, etc.)
22, 24, 26, 28, 30, 32, 34, 36	Capture	P1.n output enable from MCU
	Update	P1.n output enable to pin
23, 25, 27, 29, 31, 33, 35, 37	Capture	P1.n input from pin
	Update	P1.n output to pin
38, 40, 42, 44, 46, 48, 50, 52	Capture	P2.n output enable from MCU
	Update	P2.n output enable to pin
39, 41, 43, 45, 47, 49, 51, 53	Capture	P2.n input from pin
	Update	P2.n output to pin
54, 56, 58, 60, 62, 64, 66, 68	Capture	P3.n output enable from MCU
	Update	P3.n output enable to pin
55, 57, 59, 61, 63, 65, 67, 69	Capture	P3.n input from pin
	Update	P3.n output to pin
70, 72, 74, 76, 78, 80, 82, 84	Capture	P4.n output enable from MCU
	Update	P4.n output enable to pin
71, 73, 75, 77, 79, 81, 83, 85	Capture	P4.n input from pin
	Update	P4.n output to pin
86, 88, 90, 92, 94, 96, 98, 100	Capture	P5.n output enable from MCU
	Update	P5.n output enable to pin
87, 89, 91, 93, 95, 97, 99, 101	Capture	P5.n input from pin
	Update	P5.n output to pin
102, 104, 106, 108, 110, 112, 114, 116	Capture	P6.n output enable from MCU
	Update	P6.n output enable to pin
103, 105, 107, 109, 111, 113, 115, 117	Capture	P6.n input from pin
	Update	P6.n output to pin

Table 24.1. Boundary Data Register Bit Definitions

Bit	Action	Target
118, 120, 122, 124, 126, 128, 130, 132	Capture	P7.n output enable from MCU
	Update	P7.n output enable to pin
119, 121, 123, 125, 127, 129, 131, 133	Capture	P7.n input from pin
	Update	P7.n output to pin

24.1.1. 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.

24.1.2. SAMPLE Instruction

The SAMPLE instruction is accessed via the IR. The Boundary DR provides observability and presetting of the scan-path latches.

24.1.3. BYPASS Instruction

The BYPASS instruction is accessed via the IR. It provides access to the standard JTAG Bypass data register.

24.1.4. IDCODE Instruction

The IDCODE instruction is accessed via the IR. It provides access to the 32-bit Device ID register.

Figure 24.2. DEVICEID: JTAG Device ID Register

				Reset Value
Version	Part Number	Manufacturer ID	1	0xn0003243
Bit31 Bit28	Bit27 Bit12	Bit11 Bit1	Bit0	

Version = 0000b

Part Number = 0000 0000 0000 0011b (C8051F020/1/2/3)

Manufacturer ID = 0010 0100 001b (Silicon Labs)

24.2. Flash Programming Commands

The Flash memory can be programmed directly over the JTAG interface using the Flash Control, Flash Data, Flash Address, and Flash Scale registers. These Indirect Data Registers are accessed via the JTAG Instruction Register. Read and write operations on indirect data registers are performed by first setting the appropriate DR address in the IR register. Each read or write is then initiated by writing the appropriate Indirect Operation Code (IndOpCode) to the selected data register. Incoming commands to this register have the following format:

19:18	17:0
IndOpCode	WriteData

IndOpCode: These bit set the operation to perform according to the following table:

IndOpCode	**Operation**
0x	Poll
10	Read
11	Write

The Poll operation is used to check the Busy bit as described below. Although a Capture-DR is performed, no Update-DR is allowed for the Poll operation. Since updates are disabled, polling can be accomplished by shifting in/ out a single bit.

The Read operation initiates a read from the register addressed by the DRAddress. Reads can be initiated by shifting only 2 bits into the indirect register. After the read operation is initiated, polling of the Busy bit must be performed to determine when the operation is complete.

The write operation initiates a write of WriteData to the register addressed by DRAddress. Registers of any width up to 18 bits can be written. If the register to be written contains fewer than 18 bits, the data in WriteData should be left-justified, i.e. its MSB should occupy bit 17 above. This allows shorter registers to be written in fewer JTAG clock cycles. For example, an 8-bit register could be written by shifting only 10 bits. After a Write is initiated, the Busy bit should be polled to determine when the next operation can be initiated. The contents of the Instruction Register should not be altered while either a read or write operation is busy.

Outgoing data from the indirect Data Register has the following format:

19	18:1	0
0	ReadData	Busy

The Busy bit indicates that the current operation is not complete. It goes high when an operation is initiated and returns low when complete. Read and Write commands are ignored while Busy is high. In fact, if polling for Busy to be low will be followed by another read or write operation, JTAG writes of the next operation can be made while checking for Busy to be low. They will be ignored until Busy is read low, at which time the new operation will initiate. This bit is placed ate bit 0 to allow polling by single-bit shifts. When waiting for a Read to complete and Busy is 0, the following 18 bits can be shifted out to obtain the resulting data. ReadData is always right-justified. This allows registers shorter than 18 bits to be read using a reduced number of shifts. For example, the results from a byte-read requires 9 bit shifts (Busy + 8 bits).

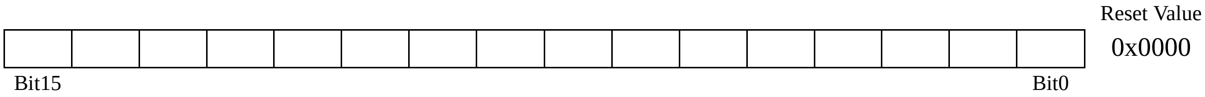
Figure 24.3. FLASHCON: JTAG Flash Control Register

This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

Bit7: SFLE: Scratchpad FLASH Memory Access Enable.
When this bit is set, FLASH reads and writes are directed to the 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 unpredictable results.
0: FLASH access directed to the 64k byte Program/Data FLASH sector.
1: FLASH access directed to the 128 byte 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 FASHDAT 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

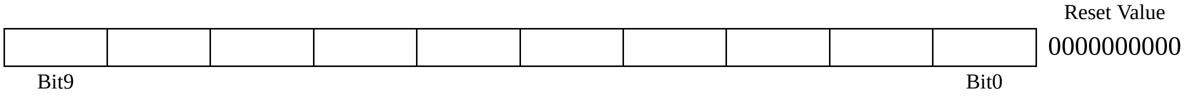
Figure 24.4. FLASHADR: JTAG Flash Address Register



This register holds the address for all JTAG Flash read, write, and erase operations. This register autoincrements after each read or write, regardless of whether the operation succeeded or failed.

Bits15-0: Flash Operation 16-bit Address.

Figure 24.5. 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

24.3. 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 C8051F020DK 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 C8051F020 family. Each kit includes an Integrated Development Environment (IDE) which has a debugger and integrated 8051 assembler. The kit also includes an RS-232 to JTAG interface module referred to as the Serial Adapter, a target application board with a C8051F020 installed, RS-232 and JTAG cables, and wall-mount power supply.

Contact Information

Silicon Laboratories Inc.

4635 Boston Lane
Austin, TX 78735
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: productinfo@silabs.com
Internet: www.silabs.com

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holder.