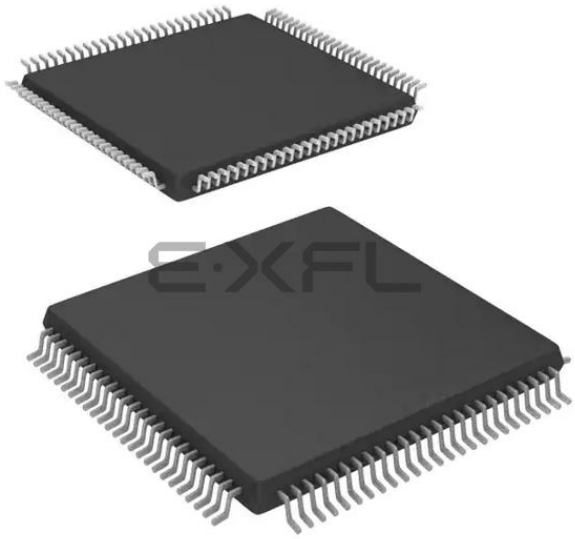


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	64
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x8b, 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f022-gqr

Notes

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

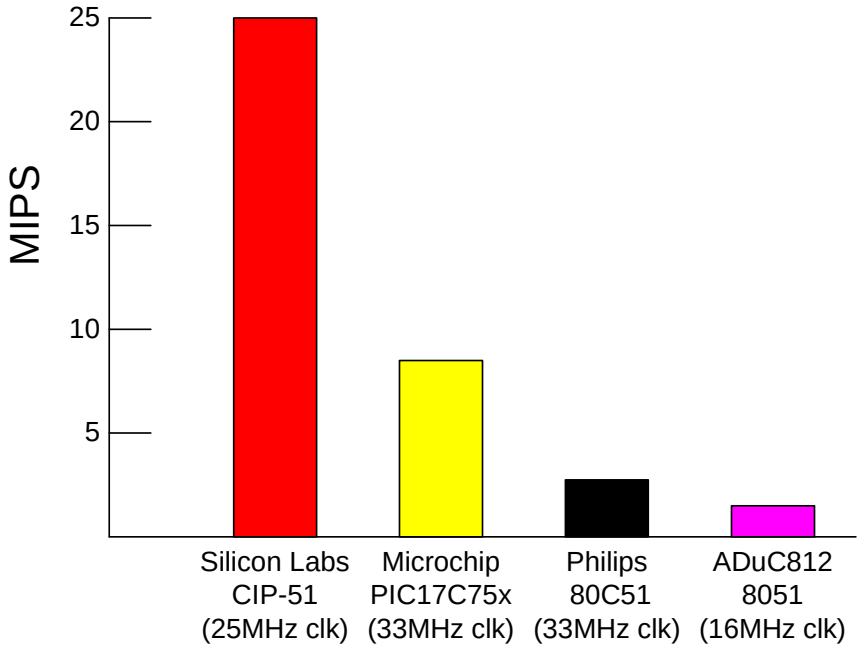


Figure 5.8. ADC0CN: ADC0 Control Register (C8051F020/1)

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 5.11. 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.

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

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 6.1 on page 74 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 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.

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**.

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.

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 17.10. 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 (bit addressable)	SFR Address: 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 17.23. 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.

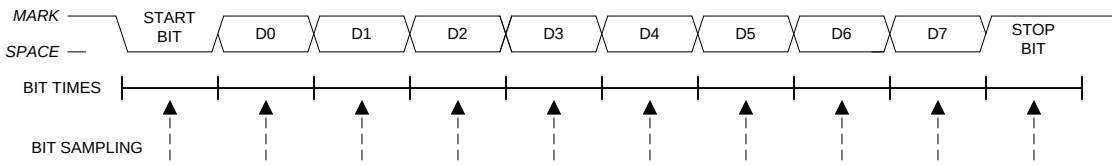
20.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

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**.

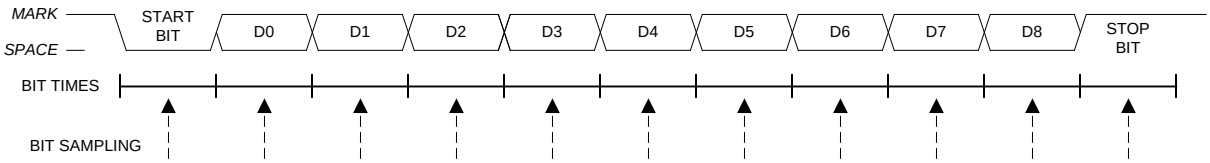
If the above conditions are satisfied, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is 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



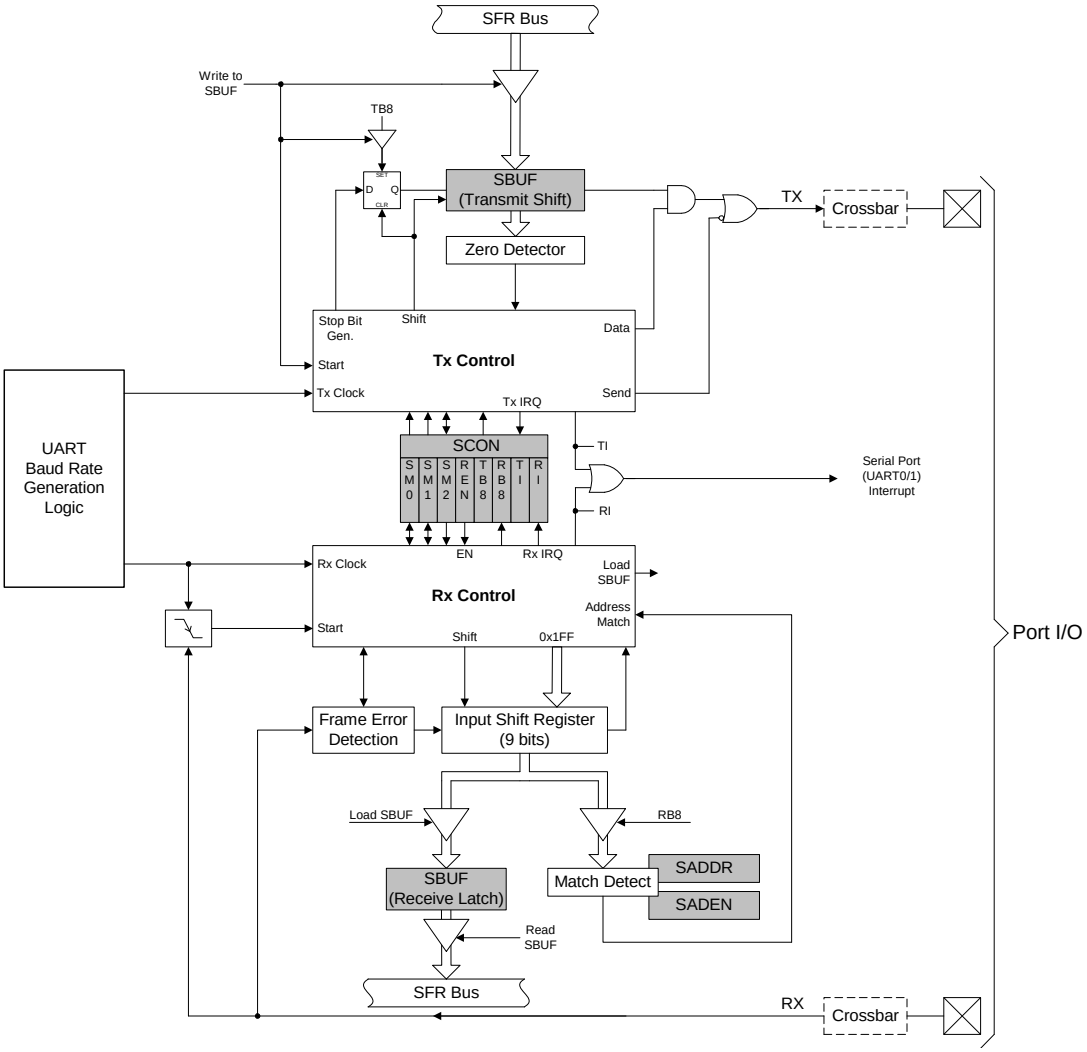
21. UART1

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



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

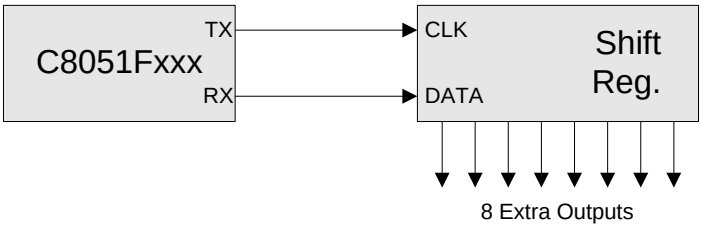
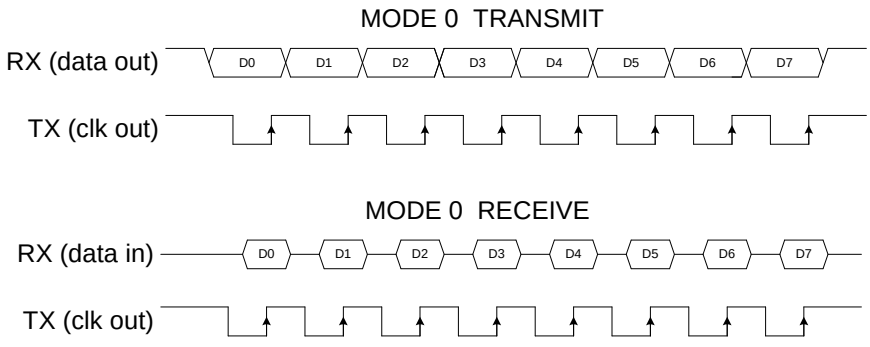


Figure 21.3. UART1 Mode 0 Timing Diagram



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)*

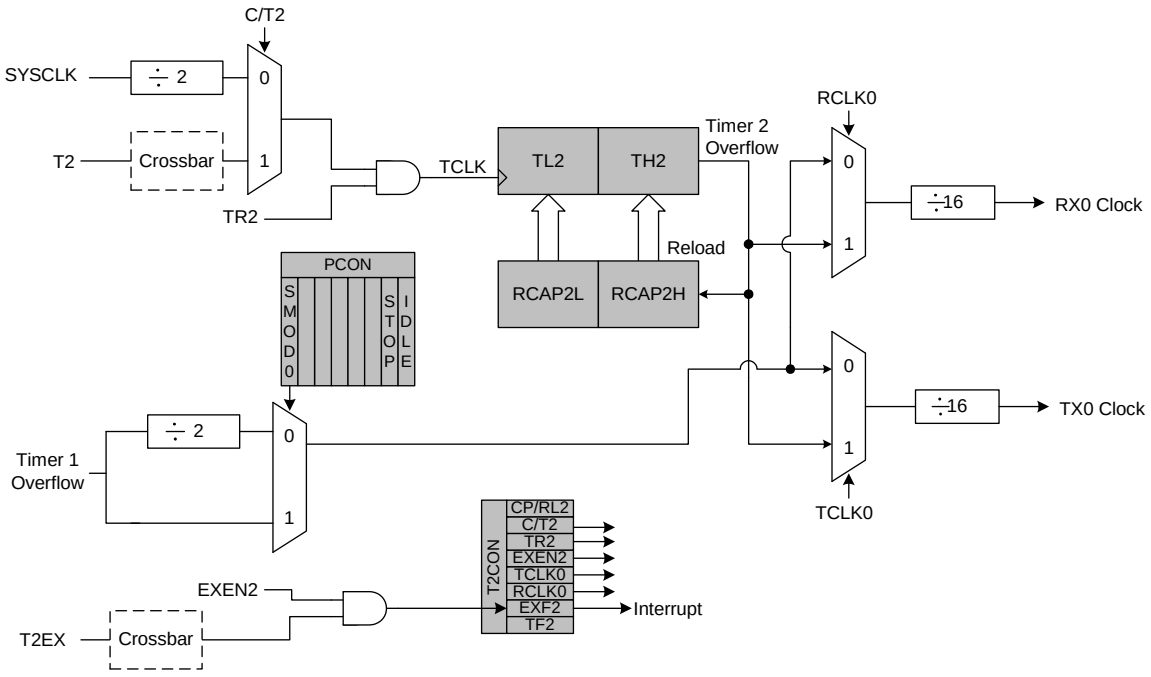
If a different time base is required, setting the C/T4 bit to logic 1 will allow the timebase to be derived from the external input pin T4. In this case, the baud rate for the UART is calculated as:

*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



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

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).