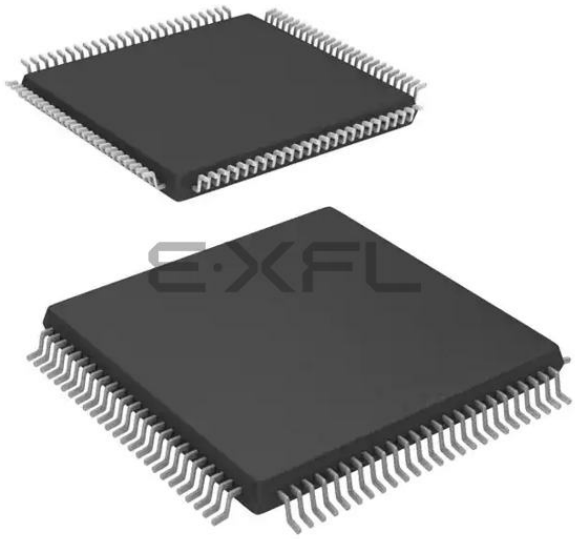


Welcome to E-XFL.COM

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	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	59
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 2x16b, 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/c8051f062-gqr

1.1.3. Additional Features

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

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

There are up to seven reset sources for the MCU: an on-board VDD monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator0, a forced software reset, the CNVSTR2 input pin, and the /RST pin. The /RST pin is bi-directional, accommodating an external reset, or allowing the internally generated POR to be output on the /RST pin. Each reset source except for the VDD monitor and Reset Input pin may be disabled by the user in software; the VDD monitor is enabled/disabled via the MONEN pin. The Watchdog Timer may be permanently enabled in software after a power-on reset during MCU initialization.

The MCU has an internal, stand alone clock generator which is used by default as the system clock after any reset. If desired, the clock source may be switched on the fly to the external oscillator, which can use a crystal, ceramic resonator, capacitor, RC, or external clock source to generate the system clock. This can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the fast (up to 25 MHz) internal oscillator as needed.

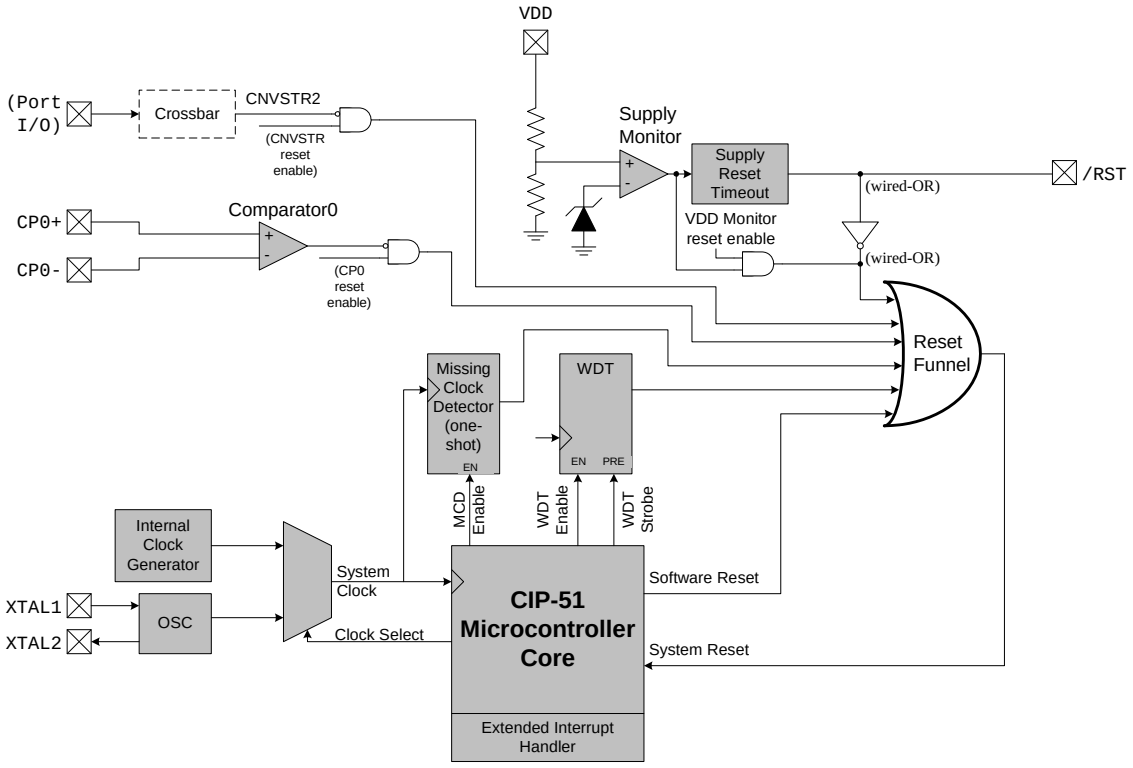


Figure 1.6. On-Board Clock and Reset

Table 4.1. Pin Definitions (Continued)

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

6.2. DMA0 Instruction Format

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

Table 6.1. DMA0 Instruction Set

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

6.3. XRAM Addressing and Setup

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

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

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

.

Table 8.1. DAC Electrical Characteristics

VDD = 3.0 V, AV+ = 3.0 V, VREF = 2.40 V (REFBE = 0), No Output Load unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Static Performance					
Resolution			12		bits
Integral Nonlinearity			±1.5		LSB
Differential Nonlinearity				±1	LSB
Output Noise	No Output Filter 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
Full-Scale Error			±20	±60	mV
Full-Scale Error Tempco			10		ppm/°C
VDD Power Supply Rejection Ratio			-60		dB
Output Impedance in Shutdown Mode	DACnEN = 0		100		kΩ
Output Sink Current			300		µA
Output Short-Circuit Current	Data Word = 0xFFF		15		mA
Dynamic Performance					
Voltage Output Slew Rate	Load = 40pF		0.44		V/µs
Output Settling Time to 1/2 LSB	Load = 40pF, Output swing from code 0xFFF to 0x014		10		µs
Output Voltage Swing		0		VREF-1LSB	V
Startup Time			10		µs
Analog Outputs					
Load Regulation	I_L = 0.01mA to 0.3mA at code 0xFFF		60		ppm
Power Consumption (each DAC)					
Power Supply Current (AV+ supplied to DAC)	Data Word = 0x7FF		300	500	µA

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

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

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

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

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

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

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

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

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

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

Table 13.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
TMR4L	0xCC	2	Timer/Counter 4 Low	page 301
WDTCN	0xFF	All Pages	Watchdog Timer Control	page 167
XBR0	0xE1	F	Port I/O Crossbar Control 0	page 210
XBR1	0xE2	F	Port I/O Crossbar Control 1	page 211
XBR2	0xE3	F	Port I/O Crossbar Control 2	page 212
XBR3	0xE4	F	Port I/O Crossbar Control 3	page 213

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

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

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

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

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

13.2.7. Register Descriptions

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

Figure 13.13. SP: Stack Pointer

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

SFR Address: 0x81
SFR Page: All Pages

Bits7-0: SP: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

Figure 13.14. DPL: Data Pointer Low Byte

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

SFR Address: 0x82
SFR Page: All Pages

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

Figure 13.15. DPH: Data Pointer High Byte

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

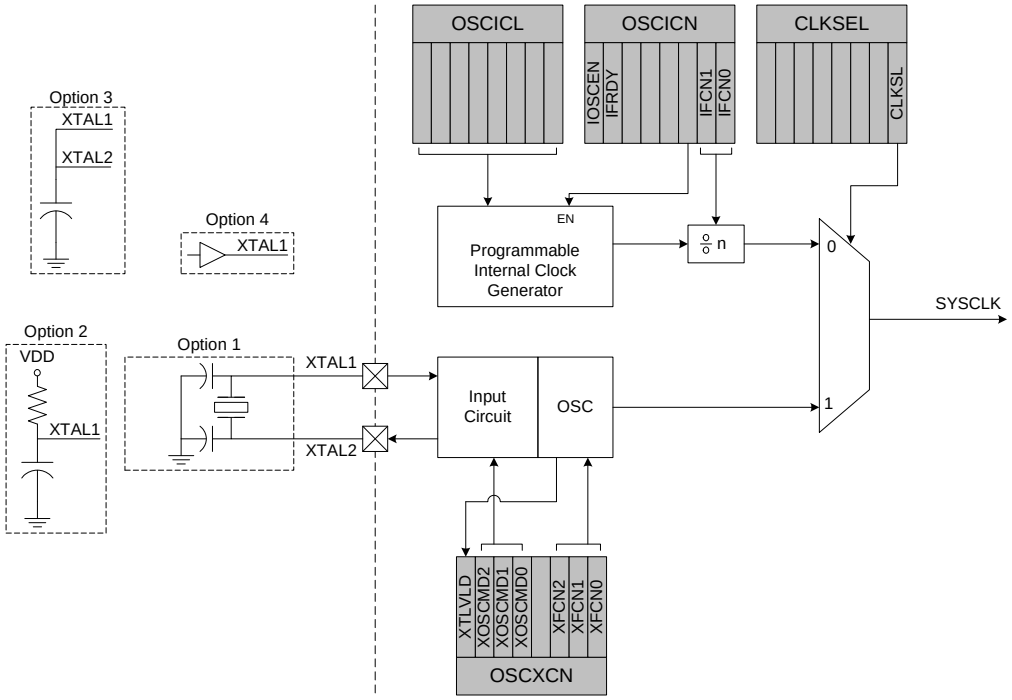
SFR Address: 0x83
SFR Page: All Pages

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

15. Oscillators

C8051F060/1/2/3/4/5/6/7 devices include a programmable internal oscillator and an external oscillator drive circuit. The internal oscillator can be enabled, disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 15.1. The system clock can be sourced by the external oscillator circuit, the internal oscillator, or a scaled version of the internal oscillator. The internal oscillator's electrical specifications are given in Table 15.1.

Figure 15.1. Oscillator Diagram



15.1. Programmable Internal Oscillator

All C8051F060/1/2/3/4/5/6/7 devices include a programmable internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register as defined by Figure 15.2.

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

Electrical specifications for the precision internal oscillator are given in Table 15.1. The programmed internal oscillator frequency must not exceed 25 MHz. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN.

Figure 16.5. PSCTL: Program Store Read/Write Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	SFLE	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:

SFR Address: 0x8F
SFR Page: 0

Bits 7-3: UNUSED. Read = 00000b, Write = don't care.

Bit 2: SFLE: Scratchpad Flash Memory Access Enable
When this bit is set, Flash MOVC reads and writes from user software are directed to the 128-byte Scratchpad Flash sector. When SFLE is set to logic 1, Flash accesses out of the address range 0x00-0x7F should not be attempted. Reads/Writes out of this range will yield undefined results.
0: Flash access from user software directed to the Program/Data Flash sector.
1: Flash access from user software directed to the Scratchpad sector.

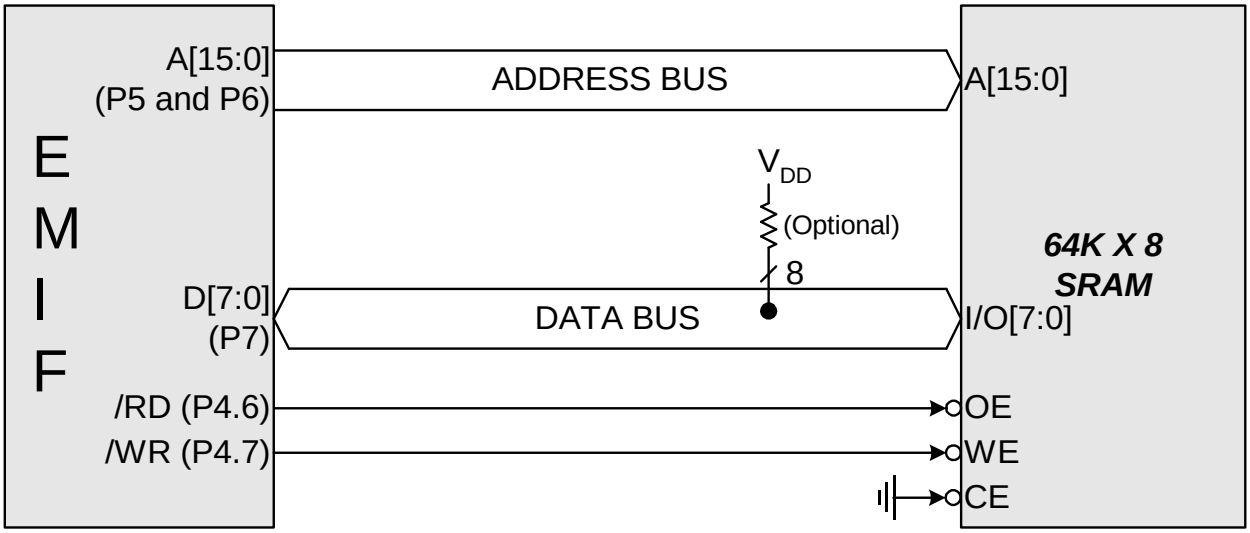
Bit 1: PSEE: Program Store Erase Enable.
Setting this bit allows an entire page of the Flash program memory to be erased provided the PSWE bit is also set. After setting this bit, a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter. **Note: The Flash page containing the Read Lock Byte and Write/Erase Lock Byte cannot be erased by software.**
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit 0: PSWE: Program Store Write Enable.
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The location must be erased prior to writing data.
0: Write to Flash program memory disabled. MOVX write operations target External RAM.
1: Write to Flash program memory enabled. MOVX write operations target Flash memory.

17.4.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 17.4. See Section “17.6.1. Non-multiplexed Mode” on page 196 for more information about Non-multiplexed operation.

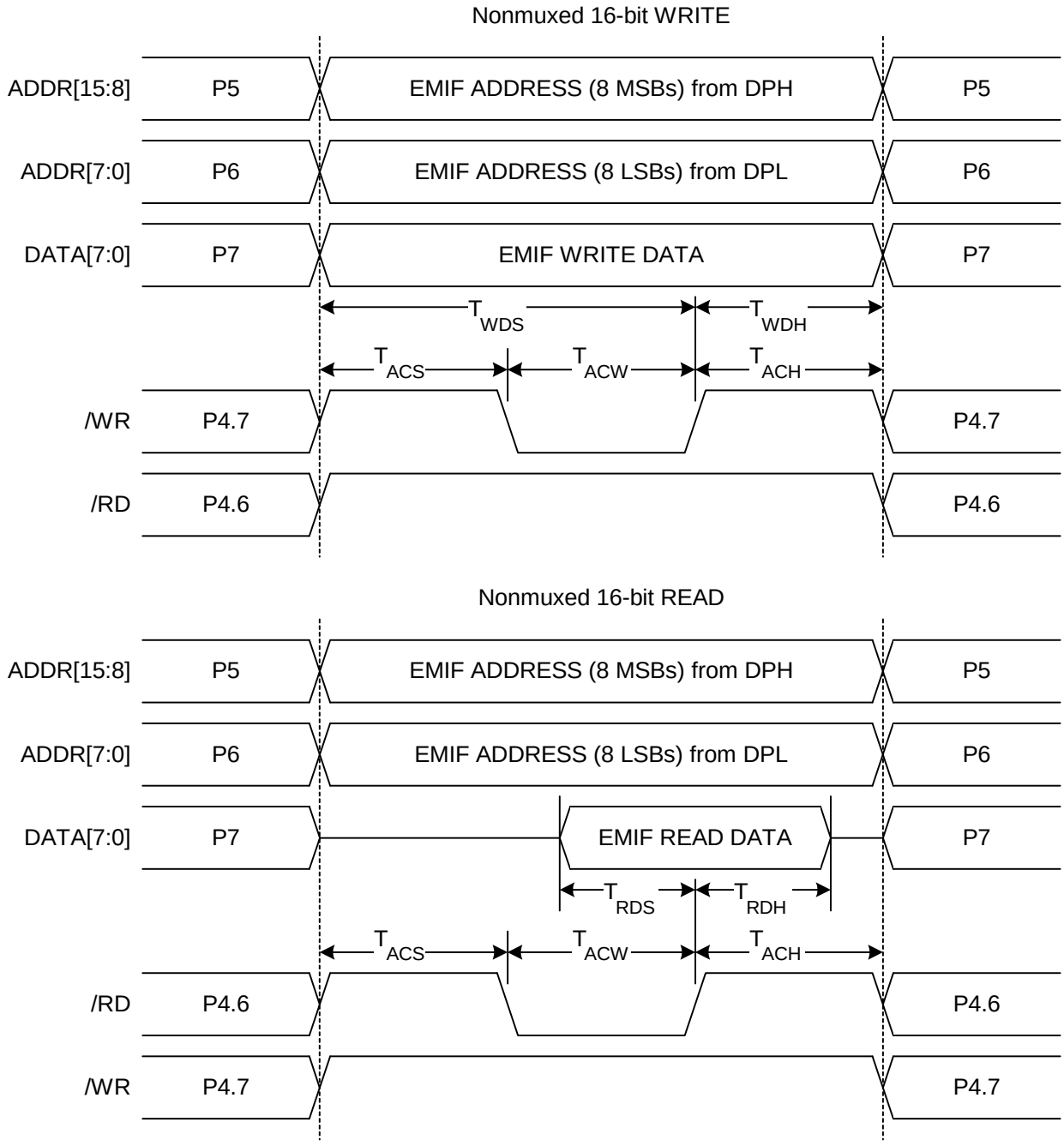
Figure 17.4. Non-multiplexed Configuration Example



17.6.1. Non-multiplexed Mode

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

Figure 17.7. Non-multiplexed 16-bit MOVX Timing



18.1.6. Crossbar Pin Assignment Example

In this example (Figure 18.4), we configure the Crossbar to allocate Port pins for UART0, the SMBus, all 6 PCA modules, /INT0, and /INT1 (12 pins total). Additionally, we configure P1.2, P1.3, and P1.4 for Analog Input mode so that the voltages at these pins can be measured by ADC2. The configuration steps are as follows:

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

1. We configure the desired Port 1 pins to Analog Input mode by setting P1MDIN to 0xE3 (P1.4, P1.3, and P1.2 are Analog Inputs, so their associated P1MDIN bits are set to logic 0).
2. We enable the Crossbar by setting XBARE = 1: XBR2 = 0x40.
   - UART0 has the highest priority, so P0.0 is assigned to TX0, and P0.1 is assigned to RX0.
   - The SMBus is next in priority order, so P0.2 is assigned to SDA, and P0.3 is assigned to SCL.
   - PCA0 is next in priority order, so P0.4 through P1.1 are assigned to CEX0 through CEX5
   - P1MDIN is set to 0xE3, which configures P1.2, P1.3, and P1.4 as Analog Inputs, causing the Crossbar to skip these pins.
   - /INT0 is next in priority order, so it is assigned to the next non-skipped pin, which is P1.5.
   - /INT1 is next in priority order, so it is assigned to P1.6.
3. We set the UART0 TX pin (TX0, P0.0) output and the CEX0-3 outputs to Push-Pull by setting P0MDOUT = 0xF1.
4. We explicitly disable the output drivers on the 3 Analog Input pins by setting the corresponding bits in the P1MDOUT register to '0', and in P1 to '1'. Additionally, the CEX5-4 output pins are set to Push-Pull mode. Therefore, P1MDOUT = 0x03 (configure unused pins to Open-Drain) and P1 = 0xFF (a logic 1 selects the high-impedance state).

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

18.2.5. External Memory Interface

If the External Memory Interface is enabled on the High ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus during the MOVX execution. See Section “17. External Data Memory Interface and On-Chip XRAM” on page 187 for more information about the External Memory Interface.

Figure 19.1. CAN Controller Diagram

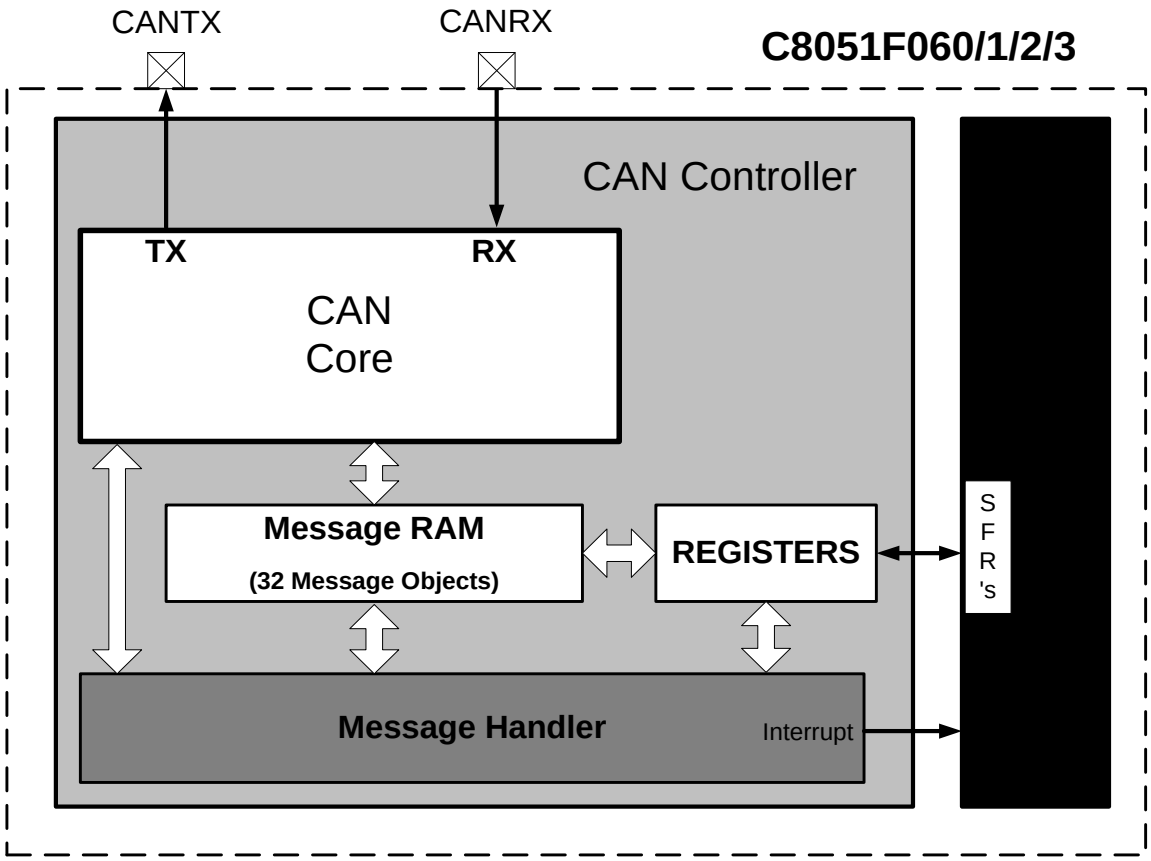


Figure 19.2. Typical CAN Bus Configuration

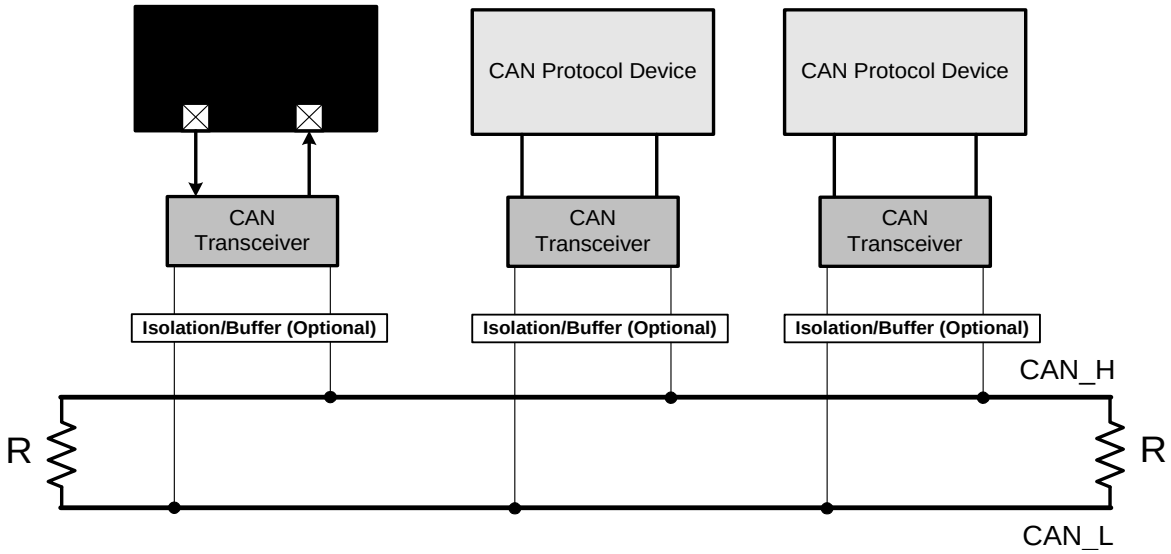
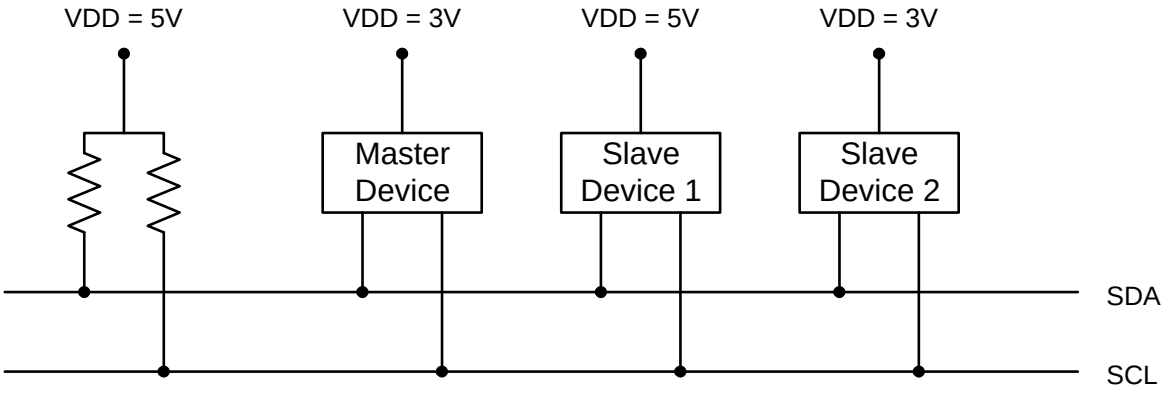


Figure 20.2 shows a typical SMBus configuration. The SMBus0 interface will work at any voltage between 3.0 V and 5.0 V and different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pull-up resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus will not exceed 300 ns and 1000 ns, respectively.

Figure 20.2. Typical SMBus Configuration



20.1. Supporting Documents

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

1. The I2C-bus and how to use it (including specifications), Philips Semiconductor.
2. The I2C-Bus Specification -- Version 2.0, Philips Semiconductor.
3. System Management Bus Specification -- Version 1.1, SBS Implementers Forum.

20.2. SMBus Protocol

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. Note: multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. Note that it is not necessary to specify one device as the master in a system; any device who transmits a START and a slave address becomes the master for that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7-1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Each byte that is received (by a master or slave) must be acknowledged (ACK) with a low SDA during a high SCL (see Figure 20.3). If the receiving device does not ACK, the transmitting device will read a “not acknowledge” (NACK), which is a high SDA during a high SCL.

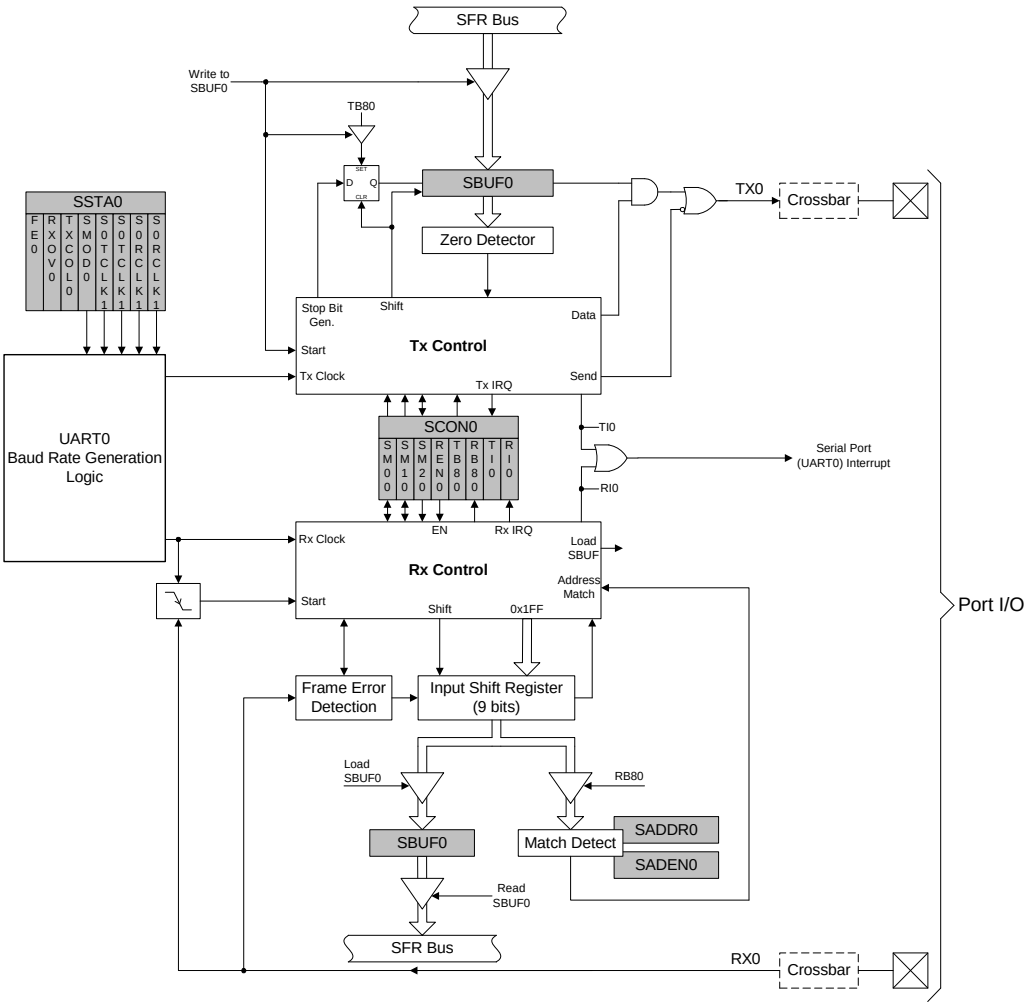
22. UART0

UART0 is an enhanced serial port with frame error detection and address recognition hardware. UART0 may operate in full-duplex asynchronous or half-duplex synchronous modes, and mutiproccessor communication is fully supported. Receive data is buffered in a holding register, allowing UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte. A Receive Overrun bit indicates when new received data is latched into the receive buffer before the previously received byte has been read.

UART0 is accessed via its associated SFRs, Serial Control (SCON0) and Serial Data Buffer (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. Reading SCON0 accesses the Receive register and writing SCON0 accesses the Transmit register.

UART0 may be operated in polled or interrupt mode. UART0 has two sources of interrupts: a Transmit Interrupt flag, TI0 (SCON0.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI0 (SCON0.0) set when reception of a data byte is complete. UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine; they must be cleared manually by software. This allows software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

Figure 22.1. UART0 Block Diagram



The baud rate generated in Mode 1 is a function of timer overflow. UART0 can use Timer 1 operating in *8-Bit Auto-Reload Mode*, or Timer 2, 3, or 4 operating in *Auto-reload Mode* to generate the baud rate (note that the TX and RX clocks are selected separately). On each timer overflow event (a rollover from all ones - (0xFF for Timer 1, 0xFFFF for Timer 2, 3, or 4) - to zero) a clock is sent to the baud rate logic.

Timers 1, 2, 3, or 4 are selected as the baud rate source with bits in the SSTA0 register (see Figure 22.9). The transmit baud rate clock is selected using the S0TCLK1 and S0TCLK0 bits, and the receive baud rate clock is selected using the S0RCLK1 and S0RCLK0 bits.

When Timer 1 is selected as a baud rate source, the SMOD0 bit (SSTA0.4) selects whether or not to divide the Timer 1 overflow rate by two. On reset, the SMOD0 bit is logic 0, thus selecting the lower speed baud rate by default. The SMOD0 bit affects the baud rate generated by Timer 1 as shown in Equation 22.1.

Equation 22.1. Mode 1 Baud Rate using Timer 1

When SMOD0 = 0:

$$\text{Mode1\_BaudRate} = 1/32 \cdot \text{Timer1\_OverflowRate}$$

When SMOD0 = 1:

$$\text{Mode1\_BaudRate} = 1/16 \cdot \text{Timer1\_OverflowRate}$$

The Timer 1 overflow rate is determined by the Timer 1 clock source (T1CLK) and reload value (TH1). The frequency of T1CLK is selected as described in Section “24.1. Timer 0 and Timer 1” on page 287. The Timer 1 overflow rate is calculated as shown in Equation 22.2.

Equation 22.2. Timer 1 Overflow Rate

$$\text{Timer1\_OverflowRate} = \text{T1CLK}/(256 - \text{TH1})$$

When Timers 2, 3, or 4 are selected as a baud rate source, the baud rate is generated as shown in Equation 22.3.

Equation 22.3. Mode 1 Baud Rate using Timer 2, 3, or 4

$$\text{Mode1\_BaudRate} = 1/16 \cdot \text{Timer234\_OverflowRate}$$

The overflow rate for Timer 2, 3, or 4 is determined by the clock source for the timer (TnCLK) and the 16-bit reload value stored in the RCAPn register (n = 2, 3, or 4), as shown in Equation 22.4.

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

$$\text{Timer234\_OverflowRate} = \text{TnCLK}/(65536 - \text{RCAPn})$$

26.1. Boundary Scan

The DR in the Boundary Scan path is a 126-bit shift register for the C8051F060/2/4/6 and a 118-bit shift register for the C8051F061/3/5/7. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature via the EXTEST and SAMPLE commands.

Table 26.1. Boundary Data Register Bit Definitions (C8051F060/2/4/6)

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

Bit	Action	Target
0	Capture	Reset Enable from MCU
	Update	Reset Enable to /RST pin
1	Capture	Reset Input from /RST pin
	Update	Not used
2	Capture	CAN RX Output Enable to pin
	Update	CAN RX Output Enable to pin
3	Capture	CAN RX Input from pin
	Update	CAN RX Output to pin
4	Capture	CAN TX Output Enable to pin
	Update	CAN TX Output Enable to pin
5	Capture	CAN TX Input from pin
	Update	CAN TX Output to pin
6	Capture	External Clock from XTAL1 pin
	Update	Not used
7	Capture	Weak Pullup Enable from MCU
	Update	Weak Pullup Enable to Port Pins
8, 10, 12, 14, 16, 18, 20, 22	Capture	P0.n output enable from MCU (e.g. Bit 8 = P0.0, Bit 10 = P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit 8 = P0.0oe, Bit 10 = P0.1oe, etc.)
9, 11, 13, 15, 17, 19, 21, 23	Capture	P0.n input from pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.)
24, 26, 28, 30, 32, 34, 36, 38	Capture	P1.n output enable from MCU (follows P0.n numbering scheme)
	Update	P1.n output enable to pin (follows P0.n numbering scheme)
25, 27, 29, 31, 33, 35, 37, 39	Capture	P1.n input from pin (follows P0.n numbering scheme)
	Update	P1.n output to pin (follows P0.n numbering scheme)
40, 42, 44, 46, 48, 50, 52, 54	Capture	P2.n output enable from MCU (follows P0.n numbering scheme)
	Update	P2.n output enable to pin (follows P0.n numbering scheme)
41, 43, 45, 47, 49, 51, 53, 55	Capture	P2.n input from pin (follows P0.n numbering scheme)
	Update	P2.n output to pin (follows P0.n numbering scheme)
56, 58, 60, 62, 64, 66, 68, 70	Capture	P3.n output enable from MCU (follows P0.n numbering scheme)
	Update	P3.n output enable to pin (follows P0.n numbering scheme)
57, 59, 61, 63, 65, 67, 69, 71	Capture	P3.n input from pin (follows P0.n numbering scheme)
	Update	P3.n output to pin (follows P0.n numbering scheme)
72, 74, 76	Capture	P4.5, P4.6, P4.7 (respectively) output enable from MCU
	Update	P4.5, P4.6, P4.7 (respectively) output enable to pin
73, 75, 77	Capture	P4.5, P4.6, P4.7 (respectively) input from pin
	Update	P4.5, P4.6, P4.7 (respectively) output to pin
78, 80, 82, 84, 86, 88, 90, 92	Capture	P5.n output enable from MCU (follows P0.n numbering scheme)
	Update	P5.n output enable to pin (follows P0.n numbering scheme)