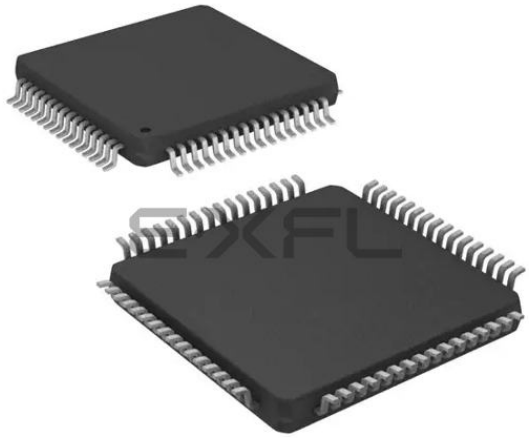


Welcome to E-XFL.COM

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	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f015-gqr

1.8. Comparators and DACs

The C8051F000 MCU Family has two 12-bit DACs and two comparators on chip (the second comparator, CP1, is not bonded out on the F002, F007, F012, and F017). 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. The comparators' output state can also be polled in software. These interrupts are capable of waking up the MCU from idle mode. The comparator outputs can be programmed to appear on the Port I/O pins via the Crossbar.

The DACs are voltage output mode and use the same voltage reference as the ADC. They are especially useful as references for the comparators or offsets for the differential inputs of the ADC.

Figure 1.11. Comparator and DAC Diagram

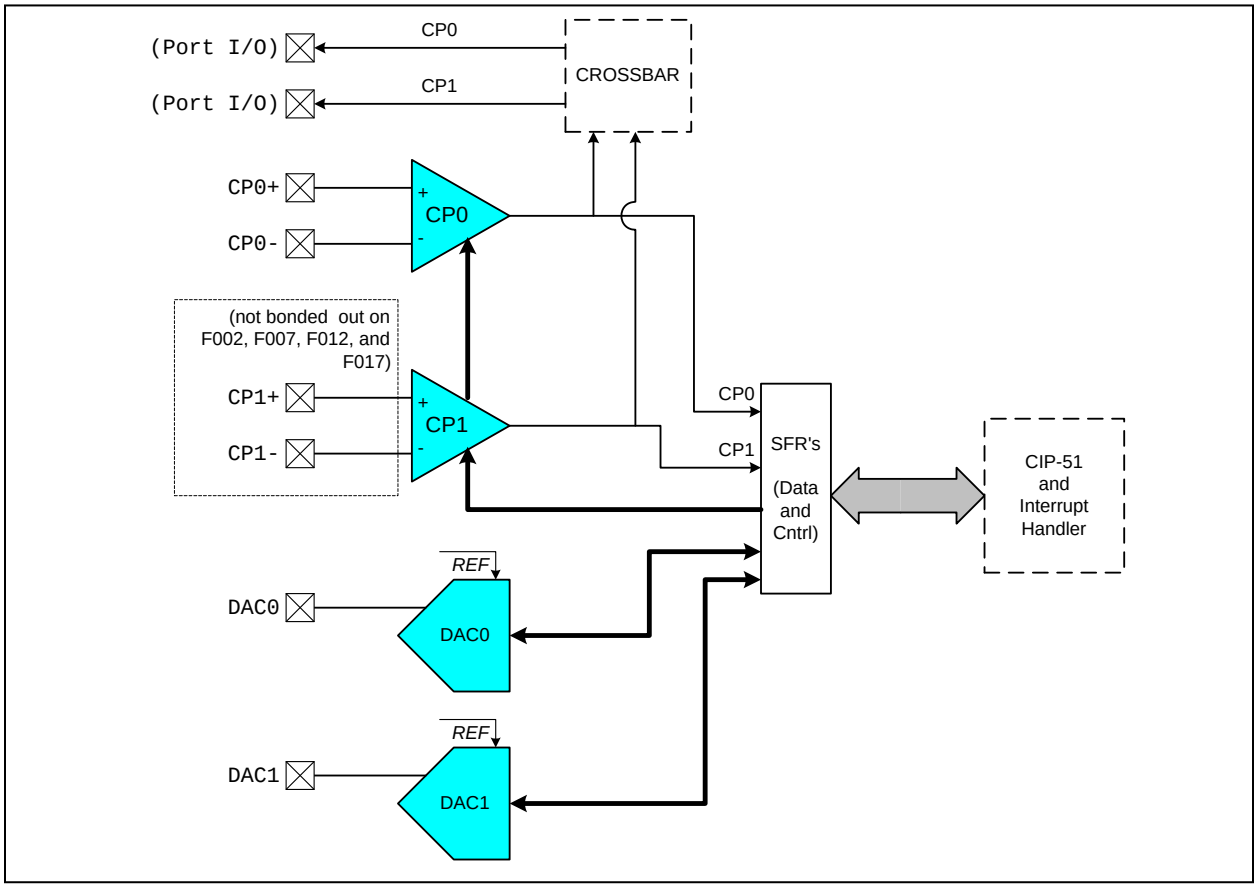


Figure 5.3. Temperature Sensor Transfer Function

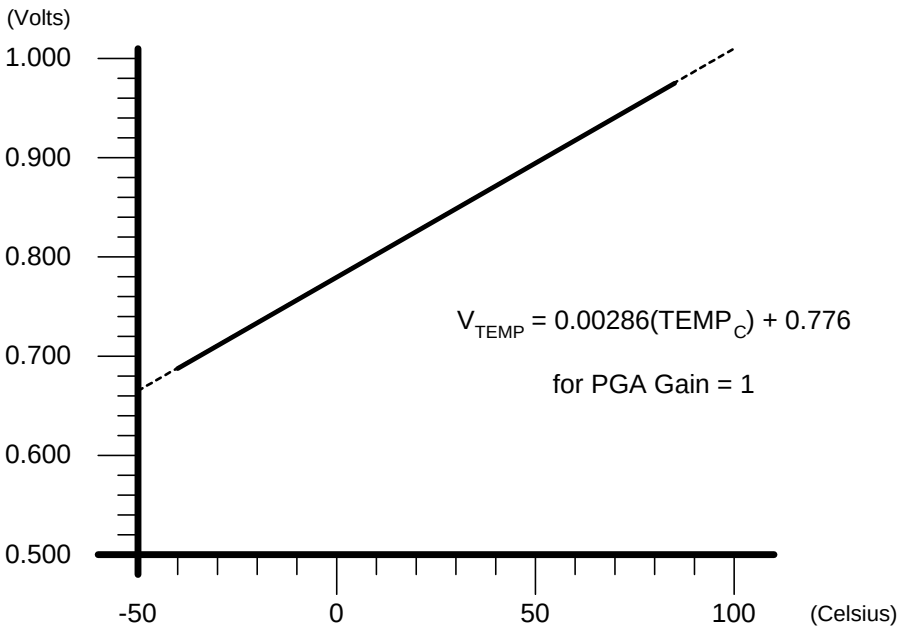


Figure 5.4. AMX0CF: AMUX Configuration Register (C8051F00x)

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 singled-ended inputs
1: AIN6, AIN7 are (respectively) +, - differential input pair
Bit2: AIN45IC: AIN4, AIN5 Input Pair Configuration Bit
0: AIN4 and AIN5 are independent singled-ended inputs
1: AIN4, AIN5 are (respectively) +, - differential input pair
Bit1: AIN23IC: AIN2, AIN3 Input Pair Configuration Bit
0: AIN2 and AIN3 are independent singled-ended inputs
1: AIN2, AIN3 are (respectively) +, - differential input pair
Bit0: AIN01IC: AIN0, AIN1 Input Pair Configuration Bit
0: AIN0 and AIN1 are independent singled-ended inputs
1: AIN0, AIN1 are (respectively) +, - differential input pair

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

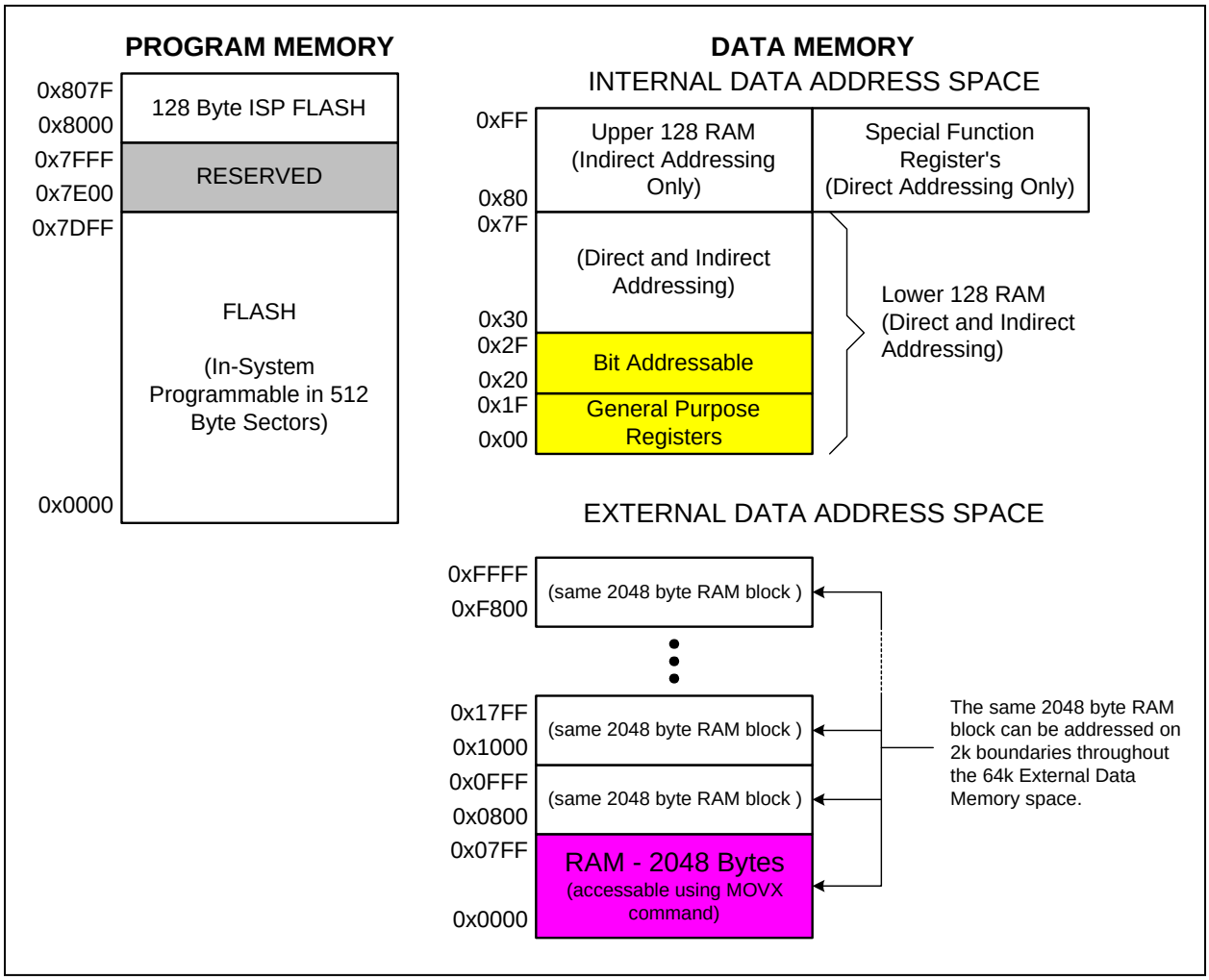
Table 7.1. DAC Electrical Characteristics

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

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
STATIC PERFORMANCE					
Resolution			12		bits
Integral Nonlinearity	For Data Word Range 0x014 to 0xFEB		±2		LSB
Differential Nonlinearity	Guaranteed Monotonic (codes 0x014 to 0xFEB)			±1	LSB
Output Noise	No Output Filter 100kHz Output Filter 10kHz 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 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 ½ LSB	Load = 40pF, Output swing from code 0xFFF to 0x014		10		µs
Output Voltage Swing		0		REF-1LSB	V
Startup Time	DAC Enable asserted		10		µs
ANALOG OUTPUTS					
Load Regulation	I_L = 0.01mA to 0.3mA at code 0xFFF		60		ppm
CURRENT CONSUMPTION (each DAC)					
Power Supply Current (AV+ supplied to DAC)	Data Word = 0x7FF		110	400	µA

register configured for accessing the external data memory space. Refer to Section 11 (Flash Memory) for further details.

Figure 10.2. Memory Map



10.2.5. 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, 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 or interrupt pushes two record bits onto the register. (A Pop or decrement SP pops one record bit, and a Return pops two record bits, also.) The stack record circuitry can also detect an overflow or underflow on the Stack, and can notify the debug software even with the MCU running full-speed debug.

Table 10.4. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Interrupt-Pending Flag	Enable
Reset	0x0000	Top	None	Always enabled
External Interrupt 0 (/INT0)	0x0003	0	IE0 (TCON.1)	EX0 (IE.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	ET0 (IE.1)
External Interrupt 1 (/INT1)	0x0013	2	IE1 (TCON.3)	EX1 (IE.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	ET1 (IE.3)
Serial Port (UART)	0x0023	4	RI (SCON.0) TI (SCON.1)	ES (IE.4)
Timer 2 Overflow (or EXF2)	0x002B	5	TF2 (T2CON.7)	ET2 (IE.5)
Serial Peripheral Interface	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	ESPI0 (EIE1.0)
SMBus Interface	0x003B	7	SI (SMB0CN.3)	ESMB0 (EIE1.1)
ADC0 Window Comparison	0x0043	8	ADWINT (ADC0CN.2)	EWADC0 (EIE1.2)
Programmable Counter Array 0	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	EPCA0 (EIE1.3)
Comparator 0 Falling Edge	0x0053	10	CP0FIF (CPT0CN.4)	ECP0F (EIE1.4)
Comparator 0 Rising Edge	0x005B	11	CP0RIF (CPT0CN.5)	ECP0R (EIE1.5)
Comparator 1 Falling Edge	0x0063	12	CP1FIF (CPT1CN.4)	ECP1F (EIE1.6)
Comparator 1 Rising Edge	0x006B	13	CP1RIF (CPT1CN.5)	ECP1R (EIE1.7)
Timer 3 Overflow	0x0073	14	TF3 (TMR3CN.7)	ET3 (EIE2.0)
ADC0 End of Conversion	0x007B	15	ADCINT (ADC0CN.5)	EADC0 (EIE2.1)
External Interrupt 4	0x0083	16	IE4 (PRT1IF.4)	EX4 (EIE2.2)
External Interrupt 5	0x008B	17	IE5 (PRT1IF.5)	EX5 (EIE2.3)
External Interrupt 6	0x0093	18	IE6 (PRT1IF.6)	EX6 (EIE2.4)
External Interrupt 7	0x009B	19	IE7 (PRT1IF.7)	EX7 (EIE2.5)
Unused Interrupt Location	0x00A3	20	None	Reserved (EIE2.6)
External Crystal OSC Ready	0x00AB	21	XTLVLD (OSCXCN.7)	EXVLD (EIE2.7)

10.4.3. 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.

10.4.4. 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.

Figure 10.10. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PT2	PS	PT1	PX1	PT0	PX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 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 interrupts.
0: Timer 2 interrupts set to low priority level.
1: Timer 2 interrupts set to high priority level.

Bit4: PS: Serial Port (UART) Interrupt Priority Control.
This bit sets the priority of the Serial Port (UART) interrupts.
0: UART interrupts set to low priority level.
1: UART interrupts set to high priority level.

Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupts.
0: Timer 1 interrupts set to low priority level.
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 interrupts.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.

Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupts.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.

Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupts.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

Figure 10.13. 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: Comparator 1 (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: Comparator 1 (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: Comparator 0 (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: Comparator 0 (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: SMBus 0 Interrupt Priority Control.
This bit sets the priority of the SMBus interrupt.
0: SMBus interrupt set to low priority level.
1: SMBus interrupt set to high priority level.

Bit0: PSPI0: Serial Peripheral Interface 0 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 15.11. P2: Port2 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	Bit	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xA0

Bits7-0: P2.[7:0]
(Write – Output appears on I/O pins per XBR0, XBR1, and XBR2 registers)
0: Logic Low Output.
1: Logic High Output (high-impedance if corresponding PRT2CF.n bit = 0)
(Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P2.n is logic low.
1: P2.n is logic high.

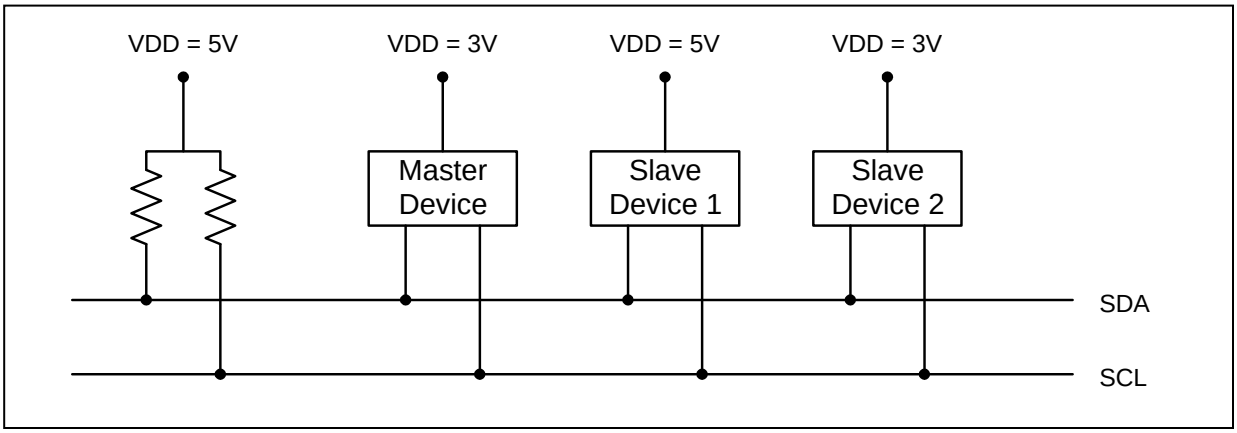
Figure 15.12. PRT2CF: Port2 Configuration Register

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: PRT2CF.[7:0]: Output Configuration Bits for P2.7-P2.0 (respectively)
0: Corresponding P2.n Output mode is Open-Drain.
1: Corresponding P2.n Output mode is Push-Pull.

Figure 16.2 shows a typical SMBus configuration. The SMBus interface will work at any voltage between 3.0V and 5.0V and different devices on the bus may operate at different voltage levels. The SCL (serial clock) and SDA (serial data) lines are bi-directional. They must be connected to a positive power supply voltage through a pull-up resistor or similar circuit. When the bus is free, both lines are pulled high. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines. 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 300ns and 1000ns, respectively.

Figure 16.2. Typical SMBus Configuration



16.1. 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.

Figure 17.7. SPI0CKR: SPI 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: SPI Clock Rate
These bits determine the frequency of the SCK output when the SPI 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 equations:

$f_{SCK} = 0.5 * f_{SYSCLK} / (SPI0CKR + 1)$, for $0 \leq SPI0CKR \leq 255$,

Figure 17.8. SPI0DAT: SPI 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 SPI 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.

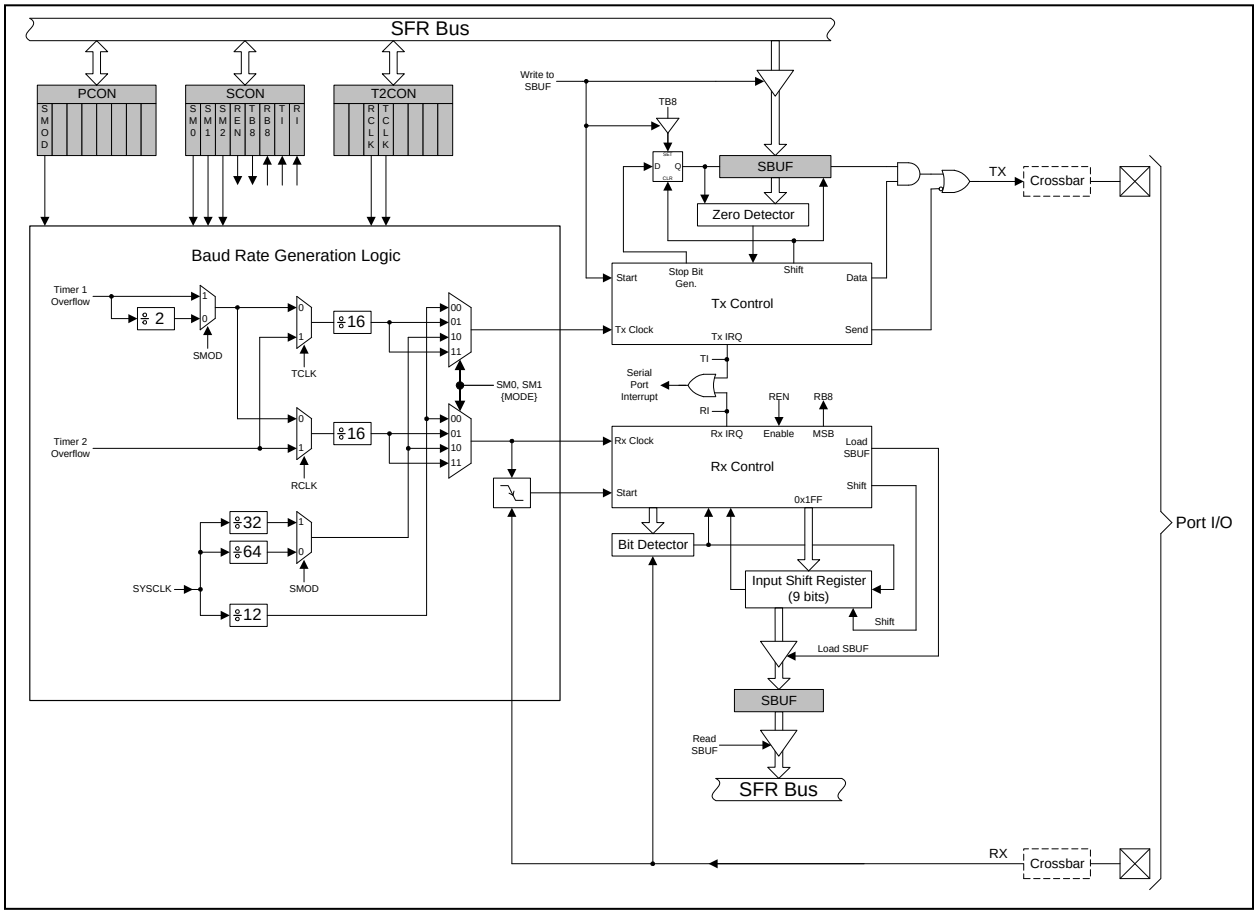
18. UART

The UART is a serial port capable of asynchronous transmission. The UART can function in full duplex mode. In all modes, receive data is buffered in a holding register. This allows the UART to start reception of a second incoming data byte before software has finished reading the previous data byte.

The UART has an associated Serial Control Register (SCON) and a Serial Data Buffer (SBUF) in the SFRs. The single SBUF location provides access to both transmit and receive registers. Reads access the Receive register and writes access the Transmit register automatically.

The UART is capable of generating interrupts if enabled. The UART has two sources of interrupts: a Transmit Interrupt flag, TI (SCON.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI (SCON.0) set when reception of a data byte is complete. The UART 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 UART interrupt (transmit complete or receive complete).

Figure 18.1. UART Block Diagram



19.1.3. 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 count in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter 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 19.2. T0 Mode 2 Block Diagram

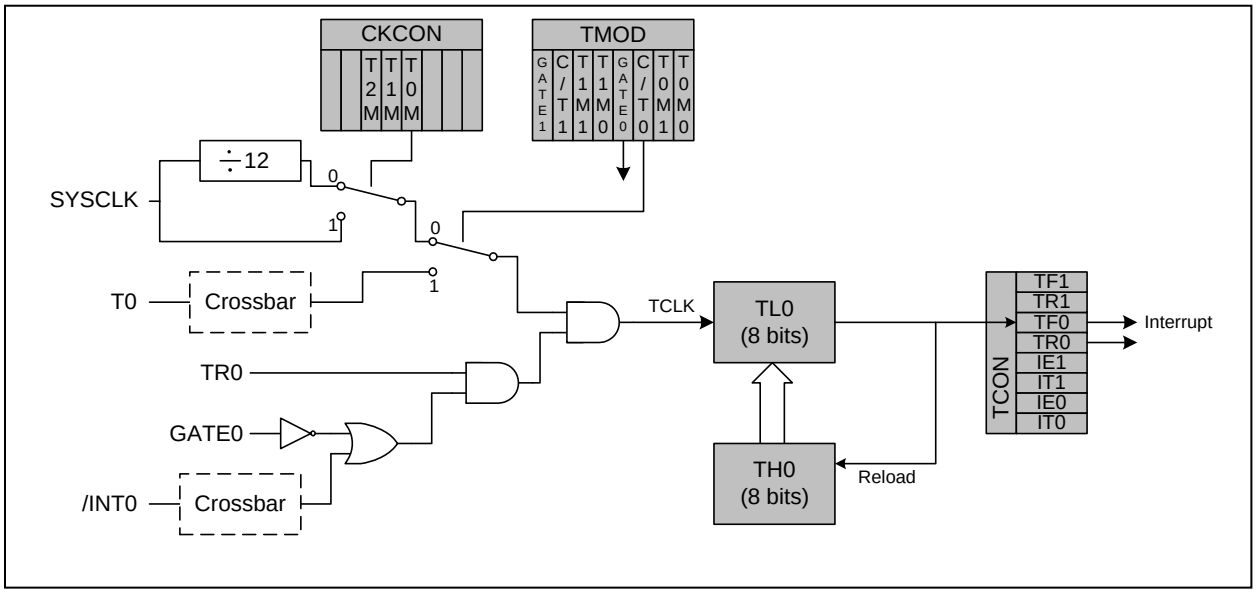


Figure 19.5. TMOD: Timer Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x89

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 level one.

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/stopped

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 level one.

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

19.2. 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 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 14 for information on selecting and configuring external I/O pins.) 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 (T2CON) register. 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.

RCLK	TCLK	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 TX
1	0	X	1	Baud Rate Generator for RX
1	1	X	1	Baud Rate Generator for TX and RX
X	X	X	0	Off

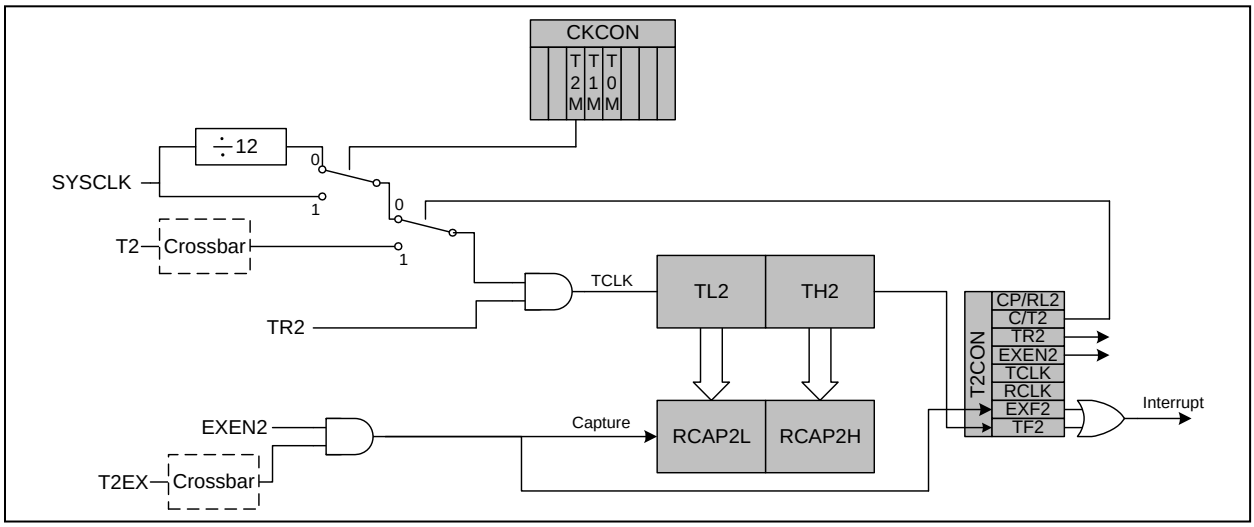
19.2.1. Mode 0: 16-bit Counter/Timer with Capture

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 16-bit value in Timer 2 (TH2, TL2) to be loaded into the capture registers (RCAP2H, RCAP2L).

Timer 2 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the external T2 pin as its clock source when operating in Counter/Timer with 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 19.11. T2 Mode 0 Block Diagram



20. PROGRAMMABLE COUNTER ARRAY

The Programmable Counter Array (PCA) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA 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 15.1 for details on configuring the Crossbar). The counter/timer is driven by a configurable timebase that can select between four inputs as its source: system clock divided by twelve, system clock divided by four, Timer 0 overflow, or an external clock signal on the ECI line. The PCA is configured and controlled through the system controller's Special Function Registers. The basic PCA block diagram is shown in Figure 20.1.

Figure 20.1. PCA Block Diagram

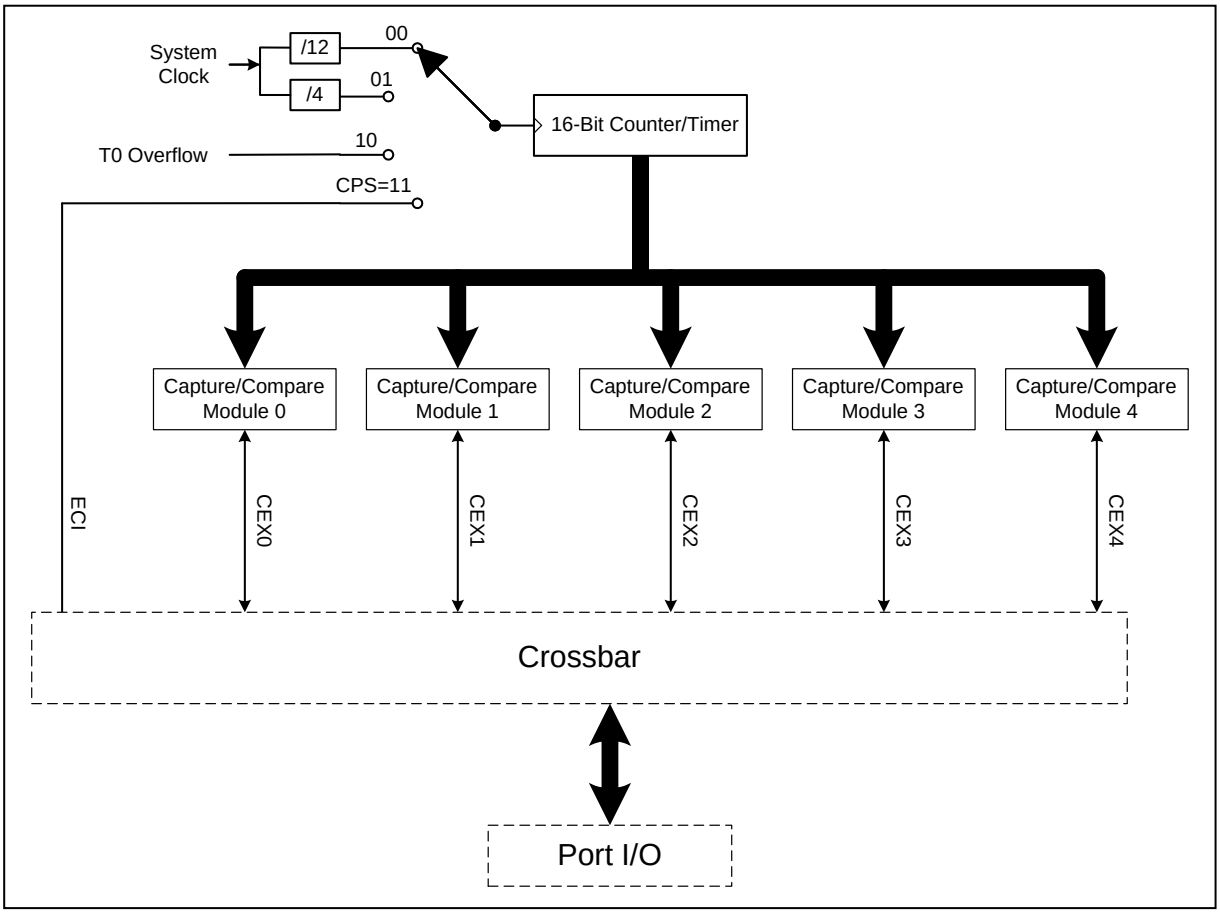


Figure 20.10. PCA0CPMn: PCA Capture/Compare Registers

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	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: UNUSED. Read = 0, Write = don't care.

Bit6: ECOMn: Comparator Function Enable.
This bit enables/disables the comparator function for PCA module *n*.
0: Disabled.
1: Enabled.

Bit5: CAPPn: Capture Positive Function Enable.
This bit enables/disables the positive edge capture for PCA module *n*.
0: Disabled.
1: Enabled.

Bit4: CAPNn: Capture Negative Function Enable.
This bit enables/disables the negative edge capture for PCA module *n*.
0: Disabled.
1: Enabled.

Bit3: MATn: Match Function Enable.
This bit enables/disables the match function for PCA module *n*. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set.
0: Disabled.
1: Enabled.

Bit2: TOGn: Toggle Function Enable.
This bit enables/disables the toggle function for PCA module *n*. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle.
0: Disabled.
1: Enabled.

Bit1: PWMn: Pulse Width Modulation Mode Enable.
This bit enables/disables the comparator function for PCA module *n*. When enabled, a pulse width modulated signal is output on the CEXn pin.
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.

21.1. Boundary Scan

The Data Register in the Boundary Scan path is an 87-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 21.1. 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
	Update	Reset Enable to /RST pin
1	Capture	Reset input from /RST pin
	Update	Reset output to /RST pin
2	Capture	External Clock from XTAL1 pin
	Update	Not used
3	Capture	Weak pullup enable from MCU
	Update	Weak pullup enable to Port Pins
4-11	Capture	SFR Address Bus bit from CIP-51 (e.g. Bit4=SFRA0, Bit5=SFRA1…)
	Update	SFR Address Bus bit to SFR Address Bus (e.g. Bit4=XSFRA0, Bit5=XSFRA1)
12-19	Capture	SFR Data Bus bit read from SFR (e.g. Bit12=SFRD0, Bit13=SFRD1…)
	Update	SFR Data Bus bit written to SFR (e.g. Bit12=SFRD0, Bit13=SFRD1…)
20	Capture	SFR Write Strobe from CIP-51
	Update	SFR Write Strobe to SFR Bus
21	Capture	SFR Read Strobe from CIP-51
	Update	SFR Read Strobe to SFR Bus
22	Capture	SFR Read/Modify/Write Strobe from CIP-51
	Update	SFR Read/Modify/Write Strobe to SFR Bus
23,25,27,29, 31,33,35,37	Capture	P0.n output enable from MCU (e.g. Bit23=P0.0, Bit25=P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit23=P0.0oe, Bit25=P0.1oe, etc.)
24,26,28,30, 32,34,36,38	Capture	P0.n input from pin (e.g. Bit24=P0.0, Bit26=P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit24=P0.0, Bit26=P0.1, etc.)
39,41,43,45, 47,49,51,53	Capture	P1.n output enable from MCU (e.g. Bit39=P1.0, Bit41=P1.1, etc.)
	Update	P1.n output enable to pin (e.g. Bit39=P1.0oe, Bit41=P1.1oe, etc.)
40,42,44,46, 48,50,52,54	Capture	P1.n input from pin (e.g. Bit40=P1.0, Bit42=P1.1, etc.)
	Update	P1.n output to pin (e.g. Bit40=P1.0, Bit42=P1.1, etc.)
55,57,59,61, 63,65,67,69	Capture	P2.n output enable from MCU (e.g. Bit55=P2.0, Bit57=P2.1, etc.)
	Update	P2.n output enable to pin (e.g. Bit55=P2.0oe, Bit57=P2.1oe, etc.)
56,58,60,62, 64,66,68,70	Capture	P2.n input from pin (e.g. Bit56=P2.0, Bit58=P2.1, etc.)
	Update	P2.n output to pin (e.g. Bit56=P2.0, Bit58=P2.1, etc.)
71,73,75,77, 79,81,83,85	Capture	P3.n output enable from MCU (e.g. Bit71=P3.0, Bit73=P3.1, etc.)
	Update	P3.n output enable to pin (e.g. Bit71=P3.0oe, Bit73=P3.1oe, etc.)
72,74,76,78, 80,82,84,86	Capture	P3.n input from pin (e.g. Bit72=P3.0, Bit74=P3.1, etc.)
	Update	P3.n output to pin (e.g. Bit72=P3.0, Bit74=P3.1, etc.)

21.3. Debug Support

Each MCU has on-chip JTAG and debug circuitry that provide *non-intrusive, full speed, in-circuit debug using the production part installed in the end application* using the four pin JTAG I/F. Silicon Labs' debug system supports inspection and modification of memory and registers, setting breakpoints, watchpoints, single stepping, and run and halt commands. No additional target RAM, program memory, or communications channels are required. All the digital and analog peripherals are functional and work correctly (remain in sync) while debugging. The WDT is disabled when the MCU is halted during single stepping or at a breakpoint.

The C8051F000DK, C8051F005DK, C8051F010DK, and C8051F015DK are development kits with all the hardware and software necessary to develop application code and perform in-circuit debugging with each MCU in the C8051F000 family. Each kit includes an Integrated Development Environment (IDE) which has a debugger and integrated 8051 assembler. It has an RS-232 to JTAG protocol translator module referred to as the EC. There is also a target application board with a C8051F000, F005, F010, or F015 installed and with a large prototyping area. The kit also includes RS-232 and JTAG cables, and wall-mount power supply.