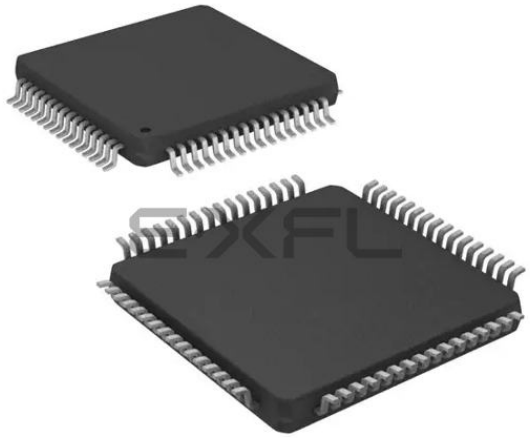


Welcome to E-XFL.COM

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	Obsolete
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	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	2.8V ~ 3.6V
Data Converters	A/D 8x10b
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/c8051f018

TABLE OF CONTENTS

1. SYSTEM OVERVIEW 7
Table 1.1. Product Selection Guide 7
Figure 1.1. C8051F018 Block Diagram 8
Figure 1.2. C8051F019 Block Diagram 9
1.1. CIP-51™ CPU 10
Figure 1.3. Comparison of Peak MCU Execution Speeds 10
Figure 1.4. On-Board Clock and Reset 11
1.2. On-Board Memory 12
Figure 1.5. On-Board Memory Map 12
1.3. JTAG Debug and Boundary Scan 13
Figure 1.6. Debug Environment Diagram 13
1.4. Programmable Digital I/O and Crossbar 14
Figure 1.7. Digital Crossbar Diagram 14
1.5. Programmable Counter Array 15
Figure 1.8. PCA Block Diagram 15
1.6. Serial Ports 15
1.7. Analog to Digital Converter 16
Figure 1.9. ADC Diagram 16
1.8. Comparators 17
Figure 1.10. Comparator Diagram 17
2. ABSOLUTE MAXIMUM RATINGS* 18
3. GLOBAL DC ELECTRICAL CHARACTERISTICS 18
4. PINOUT AND PACKAGE DEFINITIONS 19
Table 4.1. Pin Definitions 19
Figure 4.1. TQFP-64 Pinout Diagram 21
Figure 4.2. TQFP-64 Package Drawing 22
Figure 4.3. TQFP-48 Pinout Diagram 23
Figure 4.4. TQFP-48 Package Drawing 24
5. ADC 25
Figure 5.1. 10-Bit ADC Functional Block Diagram 25
5.1. Analog Multiplexer 25
5.2. ADC Modes of Operation 26
Figure 5.2. 10-Bit ADC Track and Conversion Example Timing 27
Figure 5.3. Temperature Sensor Transfer Function 27
Figure 5.4. AMX0CF: AMUX Configuration Register 28
Figure 5.5. AMX0SL: AMUX Channel Select Register (C8051F01x) 29
Figure 5.6. ADC0CF: ADC Configuration Register (C8051F01x) 30
Figure 5.7. ADC0CN: ADC Control Register 31
Figure 5.8. ADC0H: ADC Data Word MSB Register 32
Figure 5.9. ADC0L: ADC Data Word LSB Register 32
5.3. ADC Programmable Window Detector 33
Figure 5.10. ADC0GTH: ADC Greater-Than Data High Byte Register 33
Figure 5.11. ADC0GTL: ADC Greater-Than Data Low Byte Register 33
Figure 5.12. ADC0LTH: ADC Less-Than Data High Byte Register 33
Figure 5.13. ADC0LTL: ADC Less-Than Data Low Byte Register 33
Figure 5.14. 10-Bit ADC Window Interrupt Examples, Right Justified Data 34
Figure 5.15. 10-Bit ADC Window Interrupt Examples, Left Justified Data 35
Table 5.1. 10-Bit ADC Electrical Characteristics 36
6. COMPARATORS 37
Figure 6.1. Comparator Functional Block Diagram 37
Figure 6.2. Comparator Hysteresis Plot 38

1.3. JTAG Debug and Boundary Scan

The C8051F018/9 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. The JTAG port is fully compliant to IEEE 1149.1, providing full boundary scan for test and manufacturing purposes.

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

The C8051F015DK is a development kit with all the hardware and software necessary to develop application code and perform in-circuit debug with the C8051F018/9 MCUs. The kit includes software with a developer's studio and debugger, an integrated 8051 assembler, and an RS-232 to JTAG protocol translator module referred to as the EC. It also has a target application board with a C8051F015 MCU installed and a large prototyping area, plus the RS-232 and JTAG cables, and wall-mount power supply. The Development Kit requires a Windows 95/98/NT/2000/XP computer with one available RS-232 serial port. As shown in Figure 1.6, the PC is connected via RS-232 to the EC. A six-inch ribbon cable connects the EC to the user's application board, picking up the four JTAG pins and VDD and GND. The EC takes its power from the application board. It requires roughly 20mA at 2.8-3.6V. For applications where there is not sufficient power available from the target board, the provided power supply can be connected directly to the EC.

This is a vastly superior configuration for developing and debugging embedded applications compared to standard MCU Emulators, which use on-board "ICE Chips" and target cables and require the MCU in the application board to be socketed. Silicon Labs' debug environment both increases ease of use and preserves the performance of the precision analog peripherals.

Figure 1.6. Debug Environment Diagram

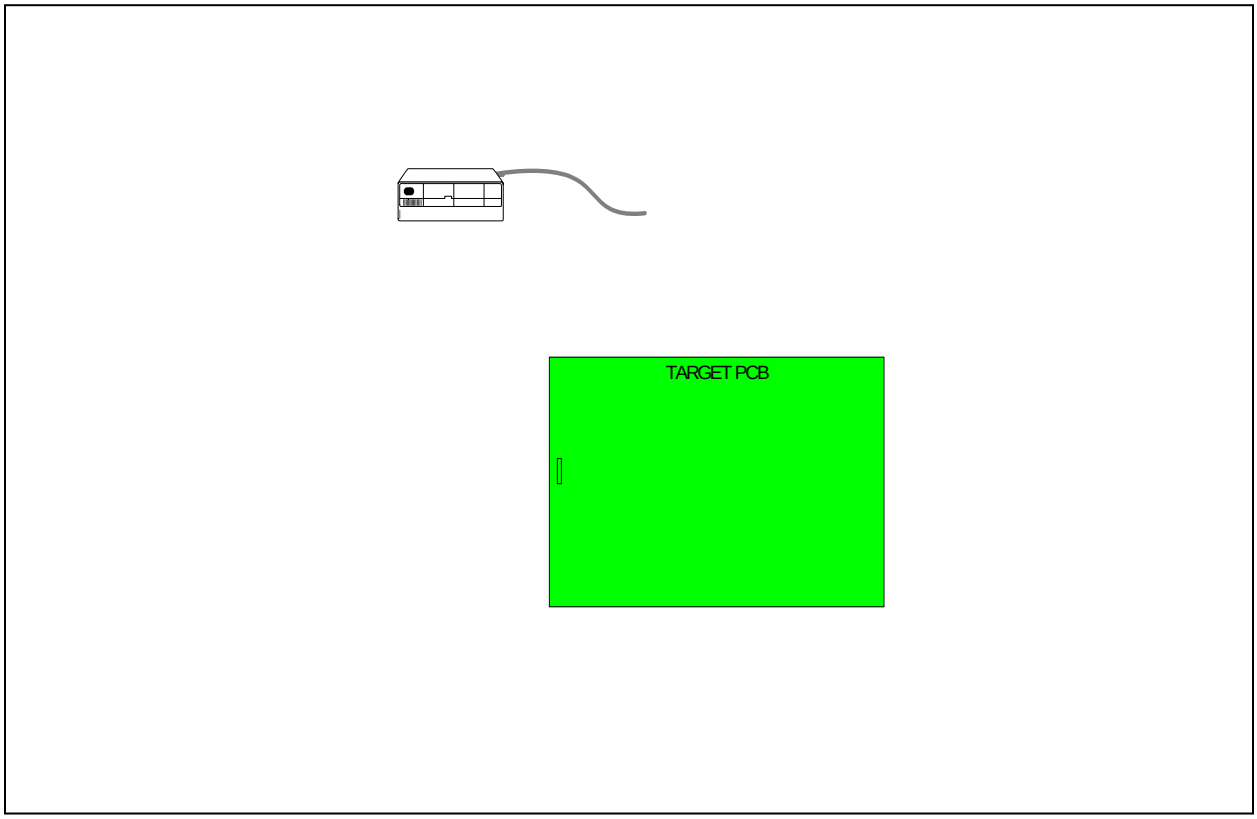


Figure 14.4. SMB0CN: SMBus Control Register

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
BUSY	ENSMB	STA	STO	SI	AA	FTE	TOE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xC0

Bit7: BUSY: Busy Status Flag.
0: SMBus is free
1: SMBus is busy

Bit6: ENSMB: SMBus Enable.
This bit enables/disables the SMBus serial interface.
0: SMBus disabled.
1: SMBus enabled.

Bit5: STA: SMBus Start Flag.
0: No START condition is transmitted.
1: When operating as a master, a START condition is transmitted if the bus is free. (If the bus is not free, the START is transmitted after a STOP is received.) If STA is set after one or more bytes have been transmitted or received and before a STOP is received, a repeated START condition is transmitted. STO should be explicitly cleared before setting STA to logic 1.

Bit4: STO: SMBus Stop Flag.
0: No STOP condition is transmitted.
1: Setting STO to logic 1 causes a STOP condition to be transmitted. When a STOP condition is received, hardware clears STO to logic 0. If both STA and STO are set, a STOP condition is transmitted followed by a START condition. In slave mode, setting the STO flag causes SMBus to behave as if a STOP condition was received.

Bit3: SI: SMBus Serial Interrupt Flag.
This bit is set by hardware when one of 27 possible SMBus states is entered. (Status code 0xF8 does not cause SI to be set.) When the SI interrupt is enabled, setting this bit causes the CPU to vector to the SMBus interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: AA: SMBus Assert Acknowledge Flag.
This bit defines the type of acknowledge returned during the acknowledge cycle on the SCL line.
0: A “not acknowledge” (high level on SDA) is returned during the acknowledge cycle.
1: An “acknowledge” (low level on SDA) is returned during the acknowledge cycle.

Bit1: FTE: SMBus Free Timer Enable Bit
0: No timeout when SCL is high
1: Timeout when SCL high time exceeds limit specified by the SMB0CR value.

Bit0: TOE: SMBus Timeout Enable Bit
0: No timeout when SCL is low.
1: Timeout when SCL low time exceeds limit specified by Timer 3, if enabled.

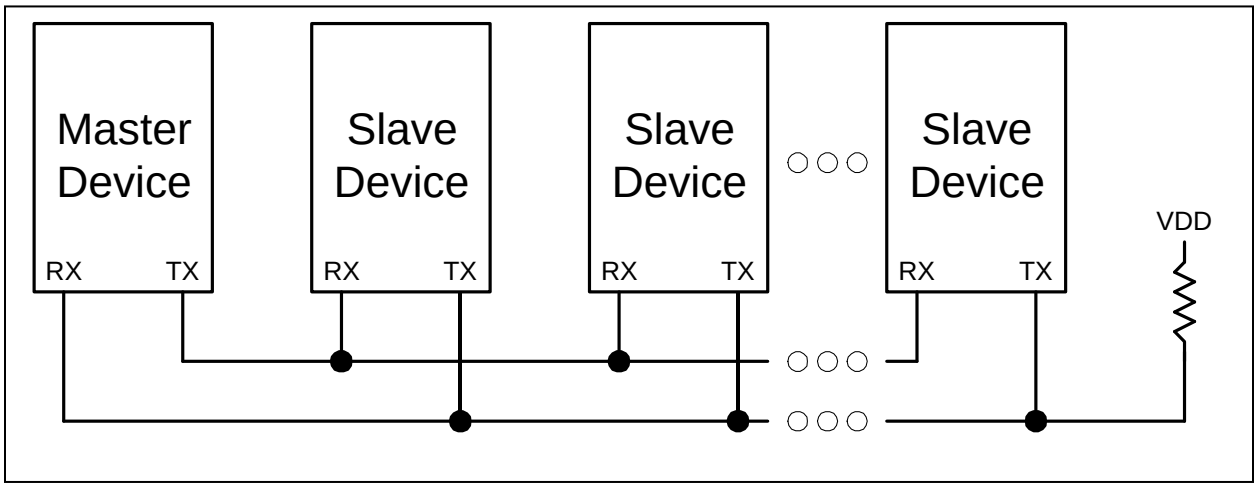
16.2. Multiprocessor Communications

Modes 2 and 3 support multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the SM2 bit (SCON.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic one (RB8 = 1) signifying an address byte has been received. In the UART's interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its SM2 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their SM2 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its SM2 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

Figure 16.7. UART Multi-Processor Mode Interconnect Diagram

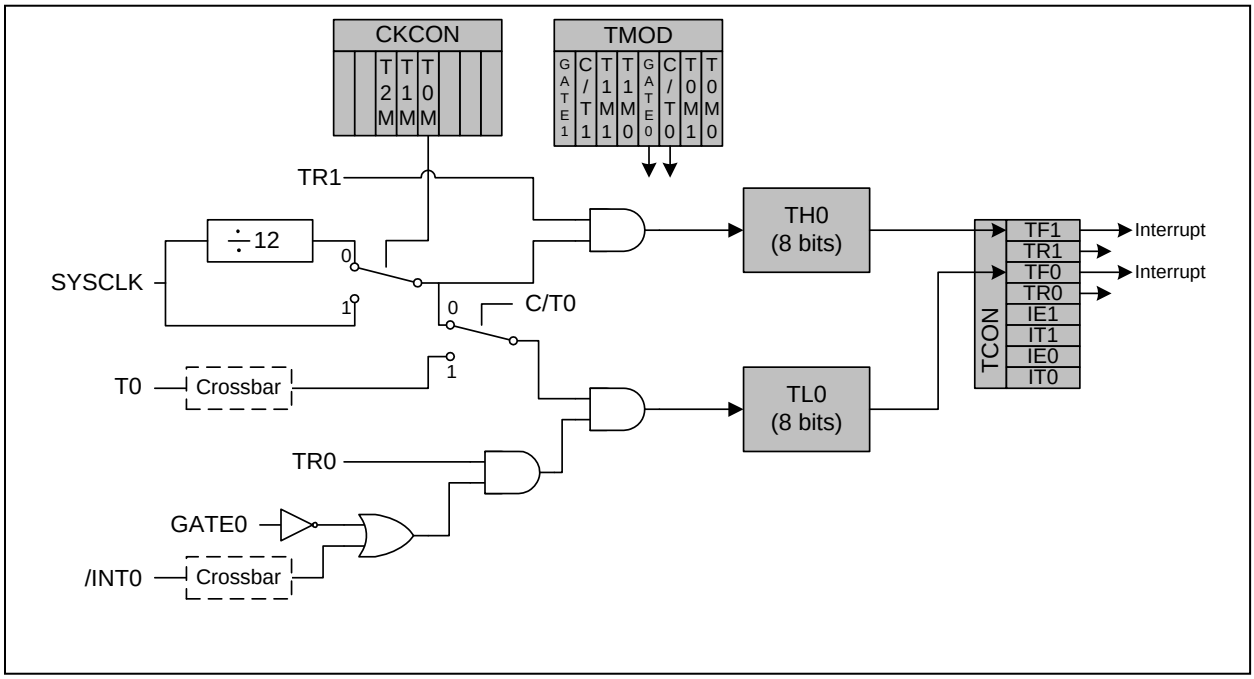


17.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

Timer 0 and Timer 1 behave differently in Mode 3. Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. It can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3, so with Timer 0 in Mode 3, Timer 1 can be turned off and on by switching it into and out of its Mode 3. When Timer 0 is in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used for baud rate generation. Refer to Section 16 (UART) for information on configuring Timer 1 for baud rate generation.

Figure 17.3. T0 Mode 3 Block Diagram



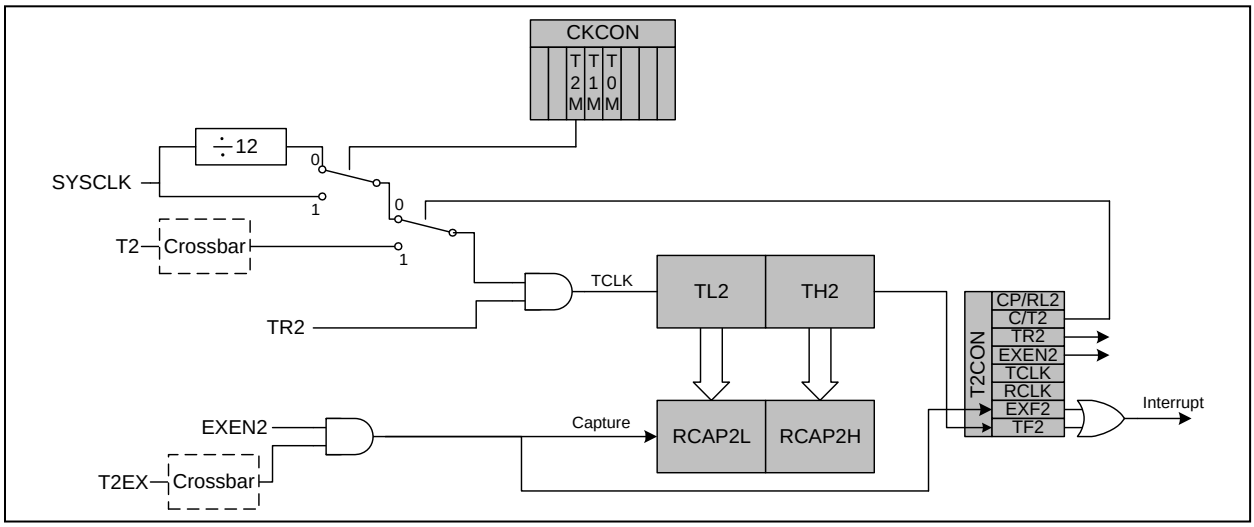
17.2.1. Mode 0: 16-bit Counter/Timer with Capture

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 17.11. T2 Mode 0 Block Diagram

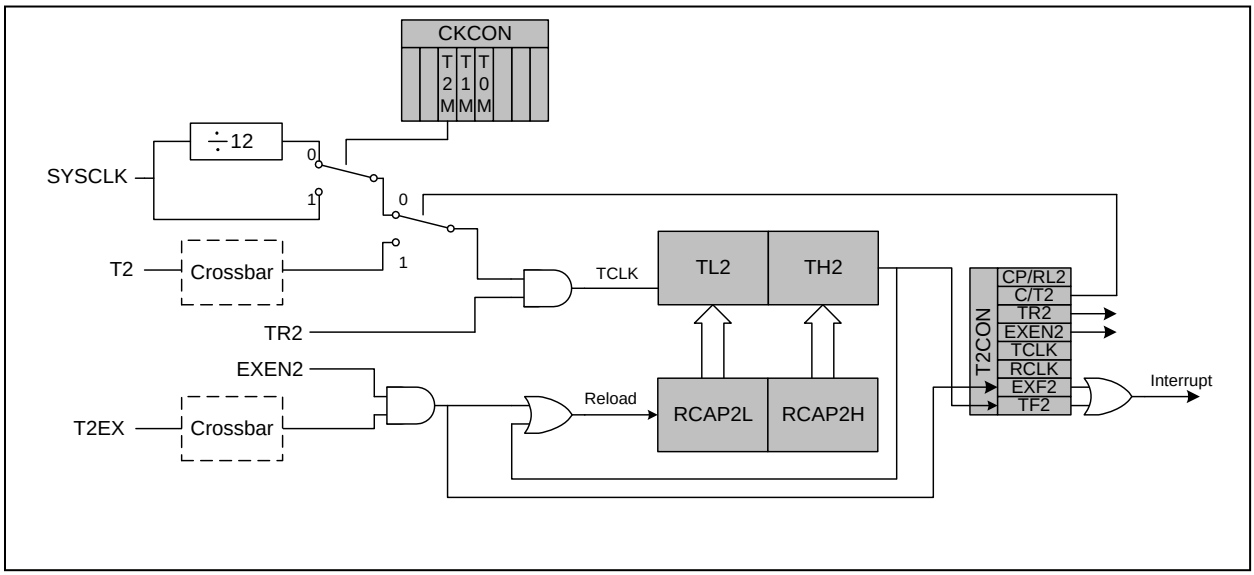


17.2.2. Mode 1: 16-bit Counter/Timer with Auto-Reload

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

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

Figure 17.12. T2 Mode 1 Block Diagram



18.2. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H at the same time. By reading the PCA0L Register first, this allows the PCA0H value to be held (at the time PCA0L was read) until the user reads the PCA0H Register. Reading PCA0H or PCA0L does not disturb the counter operation. The CPS1 and CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 18.2.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic 1.) Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the microcontroller core is in Idle mode.

Table 18.2. PCA Timebase Input Options

CPS1	CPS0	Timebase
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	Timer 0 overflow
1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)

Figure 18.7. PCA Counter/Timer Block Diagram

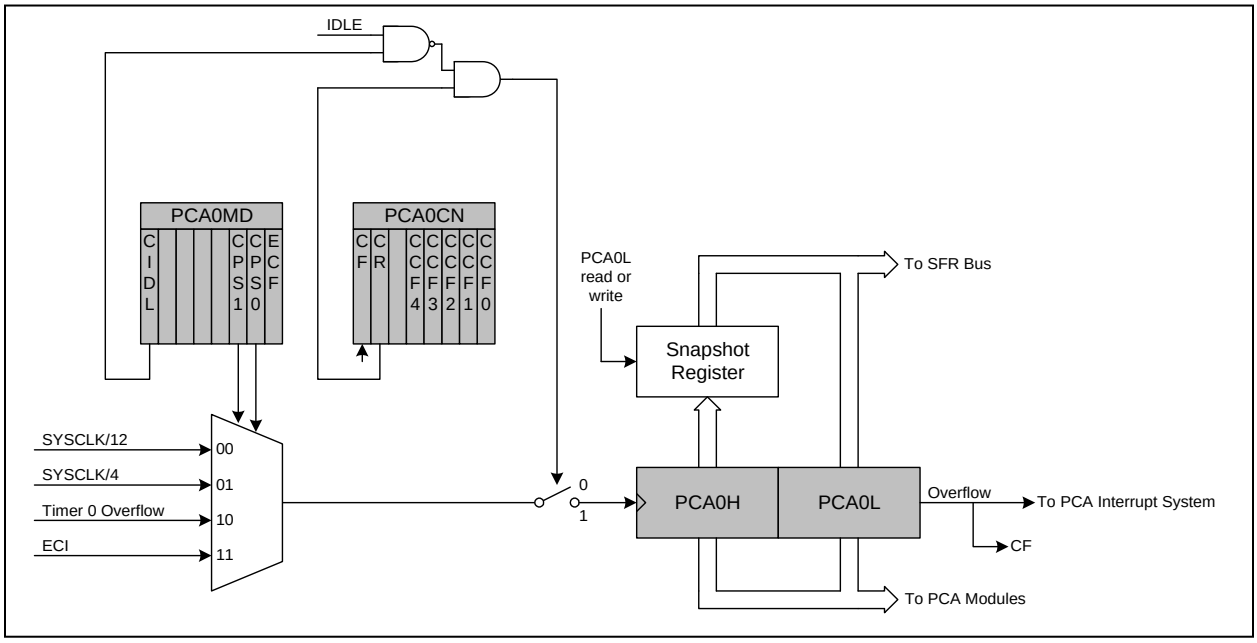


Figure 18.11. PCA0L: PCA Counter/Timer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE9

Bits 7-0: PCA0L: PCA Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

Figure 18.12. PCA0H: PCA Counter/Timer High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF9

Bits 7-0: PCA0H: PCA Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Note the value read is actually from the snapshot register in order to synchronize it with PCA0L.

Figure 18.13. PCA0CPLn: PCA Capture Module Low Byte

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

PCA0CPLn Address: PCA0CPL0 = 0xEA (n = 0)
PCA0CPL1 = 0xEB (n = 1)
PCA0CPL2 = 0xEC (n = 2)
PCA0CPL3 = 0xED (n = 3)
PCA0CPL4 = 0xEE (n = 4)

Bits7-6: PCA0CPLn: PCA Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

Figure 18.14. PCA0CPHn: PCA Capture Module High Byte

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

PCA0CPHn Address: PCA0CPH0 = 0xFA (n = 0)
PCA0CPH1 = 0xFB (n = 1)
PCA0CPH2 = 0xFC (n = 2)
PCA0CPH3 = 0xFD (n = 3)
PCA0CPH4 = 0xFE (n = 4)

Bits7-0: PCA0CPHn: PCA Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

19.3. 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 Laboratories' 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 C8051F015DK is a development kit with all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F018/9. 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 C8051F015 installed and with a large prototyping area. The kit also includes RS-232 and JTAG cables, and wall-mount power supply.