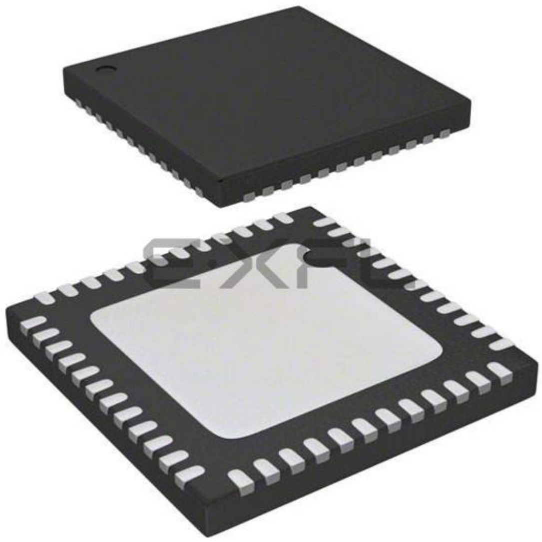

What is "Embedded - Microcontrollers"?

"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	Cap Sense, POR, PWM, WDT
Number of I/O	39
Program Memory Size	15KB (15K x 8)
Program Memory Type	FLASH
EEPROM Size	32 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f705-gm

List of Registers

3. Pin Definitions

Table 3.1. Pin Definitions for the C8051F70x/71x

Name	TQFP64	TQFP48 QFN48	QFN32	QFN24	Type	Description
V_{DD}	8, 24, 41, 57	8, 20, 44	27	21		Power Supply Voltage.
GND	9, 25, 40, 56	9, 21, 30, 43	Center	20		Ground.
$\overline{RST}$ / C2CK	58	45	28	22	D I/O D I/O	Device Reset. Open-drain output of internal POR or V_{DD} monitor. Clock signal for the C2 Debug Interface.
C2D	59	46	29	23	D I/O	Bi-directional data signal for the C2 Debug Interface.
P0.0 / VREF	55	42	—	—	D I/O or A In A In	Port 0.0. ADC0 Input. External VREF input.
P0.1/ AGND	54	41	—	—	D I/O or A In	Port 0.1. ADC0 Input. External AGND input.
P0.2 / XTAL1	53	40	—	—	D I/O or A In A In	Port 0.2. ADC0 Input. External Clock Pin. This pin can be used for crystal clock mode.
P0.3 / XTAL2	52	39	26	—	D I/O or A In A I/O or D In	Port 0.3. ADC0 Input. External Clock Pin. This pin can be used for RC, crystal, and CMOS clock modes.
P0.4	51	38	25	19	D I/O or A In	Port 0.4. ADC0 Input.
P0.5	50	37	24	18	D I/O or A In	Port 0.5. ADC0 Input.
P0.6	49	36	—	—	D I/O or A In	Port 0.6. ADC0 Input.

6. QFN-48 Package Specifications

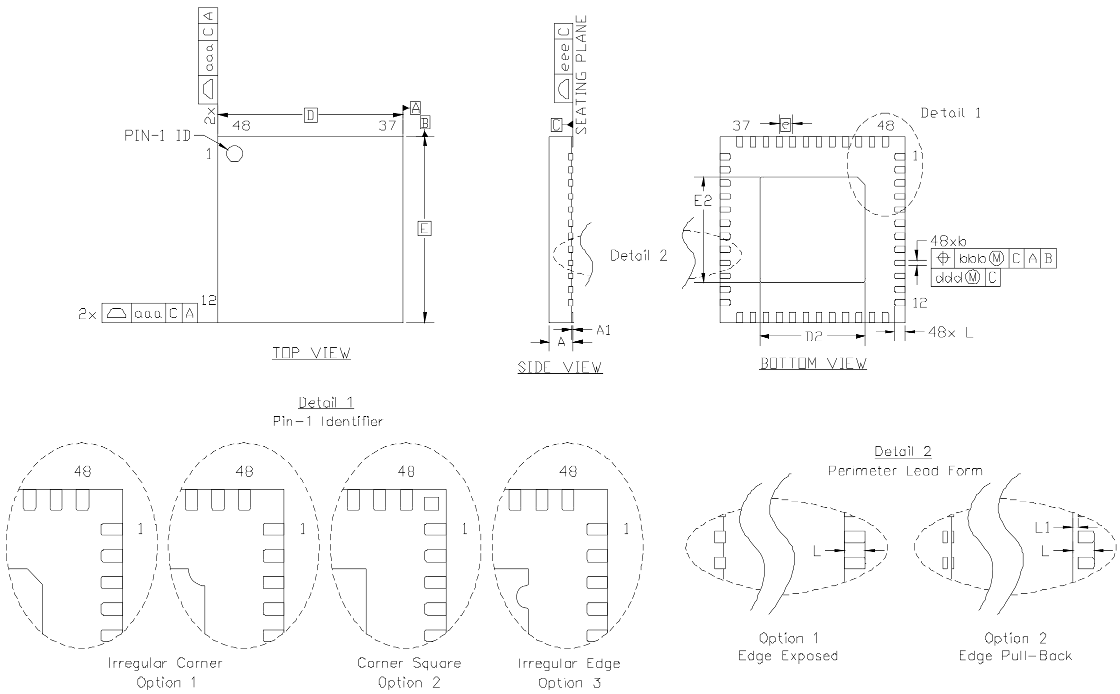

Figure 6.1. QFN-48 Package Drawing

Table 6.1. QFN-48 Package Dimensions

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	0.80	0.90	1.00	E2	3.90	4.00	4.10
A1	0.00	—	0.05	L	0.30	0.40	0.50
b	0.18	0.23	0.30	L1	0.00	—	0.10
D		7.00 BSC.		aaa	—	—	0.10
D2	3.90	4.00	4.10	bbb	—	—	0.10
e		0.50 BSC.		ccc	—	—	0.05
E		7.00 BSC.		ddd	—	—	0.08

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. 3.This drawing conforms to JEDEC outline MO-220, variation VKKD-4 except for features D2 and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

15.1. Configuring Port Pins as Capacitive Sense Inputs

In order for a port pin to be measured by CS0, that port pin must be configured as an analog input (see "28. Port Input/Output"). Configuring the input multiplexer to a port pin not configured as an analog input will cause the capacitance-to-digital converter to output incorrect measurements.

Note: When CS0 begins a conversion to measure capacitance on a port pin, CS0 grounds all other port pins that meet the following requirements:
- The port pin is accessible by the CS0 input multiplexer.
- The port pin is configured as an analog input.
- The port latch contains a 0.

15.2. CS0 Gain Adjustment

The gain of the CS0 circuit can be adjusted in integer increments from 1x to 8x (8x is the default). High gain gives the best sensitivity and resolution for small capacitors, such as those typically implemented as touch-sensitive PCB features. To measure larger capacitance values, the gain can be lowered. However, lower gain values will affect the overall conversion time. SeeTable 15.1 for more details on the gain adjustment. The bits CS0CG[2:0] in register CS0MD1 set the gain value.

Table 15.1. Gain Setting vs. Maximum Capacitance and Conversion Time

CS0CG[2:0] (Gain)	Maximum Total Capacitance (pF)[1]	Conversion Time (µs)[2]
000b (1x)	520	178
001b (2x)	260	93
010b (3x)	175	66
011b (4x)	130	52
100b (5x)	105	43
101b (6x)	85	38
110b (7x)	75	34
111b (8x)	65	31

Notes:
1. The maximum total capacitance values listed in this table are for guidance only, and are not a specification. The total measured capacitance will include internal capacitance as well as external parasitics, and the actual external capacitance being measured. Please refer to the Electrical Specifications for details on the maximum external capacitance.
2. Conversion times are nominal, and listed for 13-bit conversions with all other CS0 settings at their default values.

15.3. Capacitive Sense Start-Of-Conversion Sources

A capacitive sense conversion can be initiated in one of seven ways, depending on the programmed state of the CS0 start of conversion bits (CS0CF6:4). Conversions may be initiated by one of the following:

1. Writing a 1 to the CS0BUSY bit of register CS0CN
2. Timer 0 overflow
3. Timer 2 overflow
4. Timer 1 overflow
5. Timer 3 overflow
6. Convert continuously
7. Convert continuously with auto-scan enabled

17. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051F70x/71x device family is shown in Figure 17.1

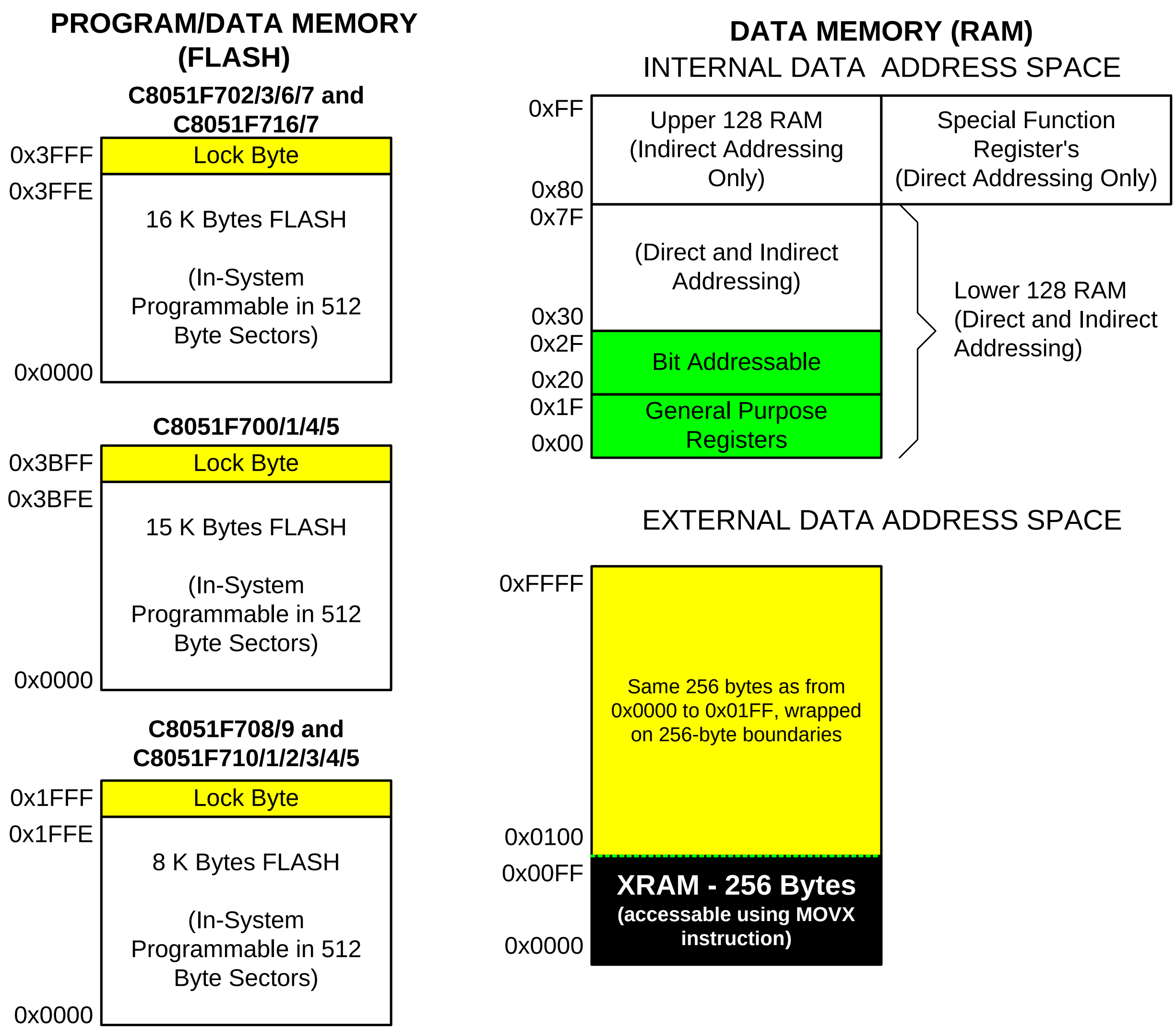

Figure 17.1. C8051F70x/71x Memory Map

18.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Configure the Output Modes of the associated port pins as either push-pull or open-drain (push-pull is most common).
2. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic 1).
3. Select Multiplexed mode or Non-multiplexed mode.
4. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
5. Set up timing to interface with off-chip memory or peripherals.

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

18.3. Port Configuration

The EMIF pinout is shown in Figure 18.2 on Page 127

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches for those pins. See Section “28. Port Input/Output” on page 180 for more information about Port operation and configuration. **The Port latches should be explicitly configured to “park” the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode.

18.5.3. Split Mode with Bank Select

When EMI0CF[3:2] are set to 10, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. The upper 8-bits of the Address Bus A[15:8] are determined by EMI0CN, and the lower 8-bits of the Address Bus A[7:0] are determined by R0 or R1. All 16-bits of the Address Bus A[15:0] are driven in “Bank Select” mode.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

18.5.4. External Only

When EMI0CF[3:2] are set to 11, all MOVX operations are directed to off-chip space. On-chip XRAM is not visible to the CPU. This mode is useful for accessing off-chip memory located between 0x0000 and the internal XRAM size boundary.

- 8-bit MOVX operations ignore the contents of EMI0CN. The upper Address bits A[15:8] are not driven (identical behavior to an off-chip access in “Split Mode without Bank Select” described above). This allows the user to manipulate the upper address bits at will by setting the Port state directly. The lower 8-bits of the effective address A[7:0] are determined by the contents of R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine the effective address A[15:0]. The full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

18.6. Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, $\overline{RD}$ and $\overline{WR}$ strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in SFR Definition 18.3, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for $\overline{RD}$ or $\overline{WR}$ pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for /ALE + 1 for $\overline{RD}$ or $\overline{WR}$ + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 18.1 lists the ac parameters for the External Memory Interface, and Figure 18.4 through Figure 18.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

Table 20.2. Special Function Registers (Continued)

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

Register	Address	Page	Description	Page
P0MDIN	0xF1	F	Port 0 Input Mode Configuration	195
P0MDOUT	0xA4	F	Port 0 Output Mode Configuration	196
P0SKIP	0xD4	F	Port 0 Skip	196
P1	0x90	All Pages	Port 1 Latch	197
P1DRV	0xFA	F	Port 1 Drive Strength	199
P1MASK	0xE2	0	P0 Mask	193
P1MAT	0xE1	0	P1 Match	194
P1MDIN	0xF2	F	Port 1 Input Mode Configuration	198
P1MDOUT	0xA5	F	Port 1 Output Mode Configuration	198
P1SKIP	0xD5	F	Port 1 Skip	199
P2	0xA0	All Pages	Port 2 Latch	200
P2DRV	0xFB	F	Port 2 Drive Strength	202
P2MDIN	0xF3	F	Port 2 Input Mode Configuration	200
P2MDOUT	0xA6	F	Port 2 Output Mode Configuration	201
P2SKIP	0xD6	F	Port 2 Skip	201
P3	0xB0	All Pages	Port 3 Latch	202
P3DRV	0xFC	F	Port 3 Drive Strength	204
P3MDIN	0xF4	F	Port 3 Input Mode Configuration	203
P3MDOUT	0xAF	F	Port 3 Output Mode Configuration	203
P4	0xAC	All Pages	Port 4 Latch	204
P4DRV	0xFD	F	Port 4 Drive Strength	206
P4MDIN	0xF5	F	Port 4 Input Mode Configuration	205
P4MDOUT	0x9A	F	Port 4 Output Mode Configuration	205
P5	0xB3	All Pages	Port 5 Latch	206
P5DRV	0xFE	F	Port 5 Drive Strength	208
P5MDIN	0xF6	F	Port 5 Input Mode Configuration	207
P5MDOUT	0x9B	F	Port 5 Output Mode Configuration	207
P6	0xB2	All Pages	Port 6 Latch	208
P6DRV	0xC1	F	Port 6 Drive Strength	210
P6MDIN	0xF7	F	Port 6 Input Mode Configuration	209
P6MDOUT	0x9C	F	Port 6 Output Mode Configuration	209
PCA0CN	0xD8	All Pages	PCA Control	295
PCA0CPH0	0xFC	0	PCA Capture 0 High	300
PCA0CPH1	0xEA	0	PCA Capture 1 High	300
PCA0CPH2	0xEC	0	PCA Capture 2 High	300
PCA0CPL0	0xFB	0	PCA Capture 0 Low	300
PCA0CPL1	0xE9	0	PCA Capture 1 Low	300
PCA0CPL2	0xEB	0	PCA Capture 2 Low	300

Table 20.2. Special Function Registers (Continued)

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

Register	Address	Page	Description	Page
TMR3CN	0x91	0	Timer/Counter 3 Control	281
TMR3H	0x95	0	Timer/Counter 3 High	283
TMR3L	0x94	0	Timer/Counter 3 Low	283
TMR3RLH	0x93	0	Timer/Counter 3 Reload High	282
TMR3RLL	0x92	0	Timer/Counter 3 Reload Low	282
VDM0CN	0xFF	All Pages	VDD Monitor Control	166
WDTCN	0xE3	All Pages	Watchdog Timer Control	170
XBR0	0xE1	F	Port I/O Crossbar Control 0	190
XBR1	0xE2	F	Port I/O Crossbar Control 1	191
All other SFR Locations			Reserved	

22. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during Flash write/erase operations. Refer to Table 9.6 for complete Flash memory electrical characteristics.

22.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “35. C2 Interface” on page 301.

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before programming Flash memory using MOVX, Flash programming operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

Note: A minimum SYSCLK frequency is required for writing or erasing Flash memory, as detailed in Section “Table 9.6. Flash Electrical Characteristics” on page 50.

For detailed guidelines on programming Flash from firmware, please see Section “22.4. Flash Write and Erase Guidelines” on page 150.

To ensure the integrity of the Flash contents, the on-chip VDD Monitor must be enabled and enabled as a reset source in any system that includes code that writes and/or erases Flash memory from software. Furthermore, there should be no delay between enabling the V_{DD} Monitor and enabling the V_{DD} Monitor as a reset source. Any attempt to write or erase Flash memory while the V_{DD} Monitor is disabled, or not enabled as a reset source, will cause a Flash Error device reset.

22.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 22.2.

22.1.2. Flash Erase Procedure

The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

1. Save current interrupt state and disable interrupts.
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. Write the first key code to FLKEY: 0xA5.
5. Write the second key code to FLKEY: 0xF1.

23.4. EEPROM Security

RAM can only be downloaded to EEPROM after firmware writes a sequence of two bytes to EEKEY. In order to enable EEPROM writes:

1. Write the first EEPROM key code byte to EEKEY: 0x55
2. Write the second EEPROM key code byte to EEKEY: 0xAA

After a EEPROM writes have been enabled and a single write has executed, the control logic locks EEPROM writes until the two-byte unlock sequence has been entered into EEKEY again.

The protection state of the EEPROM can be observed by reading EEPSTATE (EEKEY2:0). This state can be read at any time without affecting the EEPROM's protection state.

If the two-byte unlock sequence is entered incorrectly, or if a write is attempted without first entering the two-byte sequence, EEPROM writes will be locked until the next power-on reset.

SFR Definition 23.1. EEADDR: EEPROM Byte Address

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name				EEADDR[4:0]				
Type	R	R	R	R/W				
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xB6; SFR Page = All Pages

Bit	Name	Description
7:5	Unused	Read = 000b; Write = Don't Care
4:0	EEADDR[4:0]	**EEPROM Byte Address** Selects one of 32 EEPROM bytes to read/write.

29.2. 32-bit CRC Algorithm

The C8051F70x/71x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

1. XOR the least-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x00000000 or 0xFFFFFFFF).
2. Right-shift the CRC result.
3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit C8051F70x/71x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input){
   unsigned char i; // loop counter
   #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ CRC_input;
   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x00000001) == 0x00000001)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc >> 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc >> 1;
      }
   }
   return CRC_acc; // Return the final remainder (CRC value)
}
```

Table 29.2 lists example input values and the associated outputs using the 32-bit C8051F70x/71x CRC algorithm (an initial value of 0xFFFFFFFF is used):

Table 29.2. Example 32-bit CRC Outputs

Input	Output
0x63	0xF9462090
0xAA, 0xBB, 0xCC	0x41B207B3
0x00, 0x00, 0xAA, 0xBB, 0xCC	0x78D129BC

30.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 30.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 30.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

30.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

30.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 30.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 30.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 30.5 for SMBus status decoding using the SMB0CN register.

32. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “32.1. Enhanced Baud Rate Generation” on page 255). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

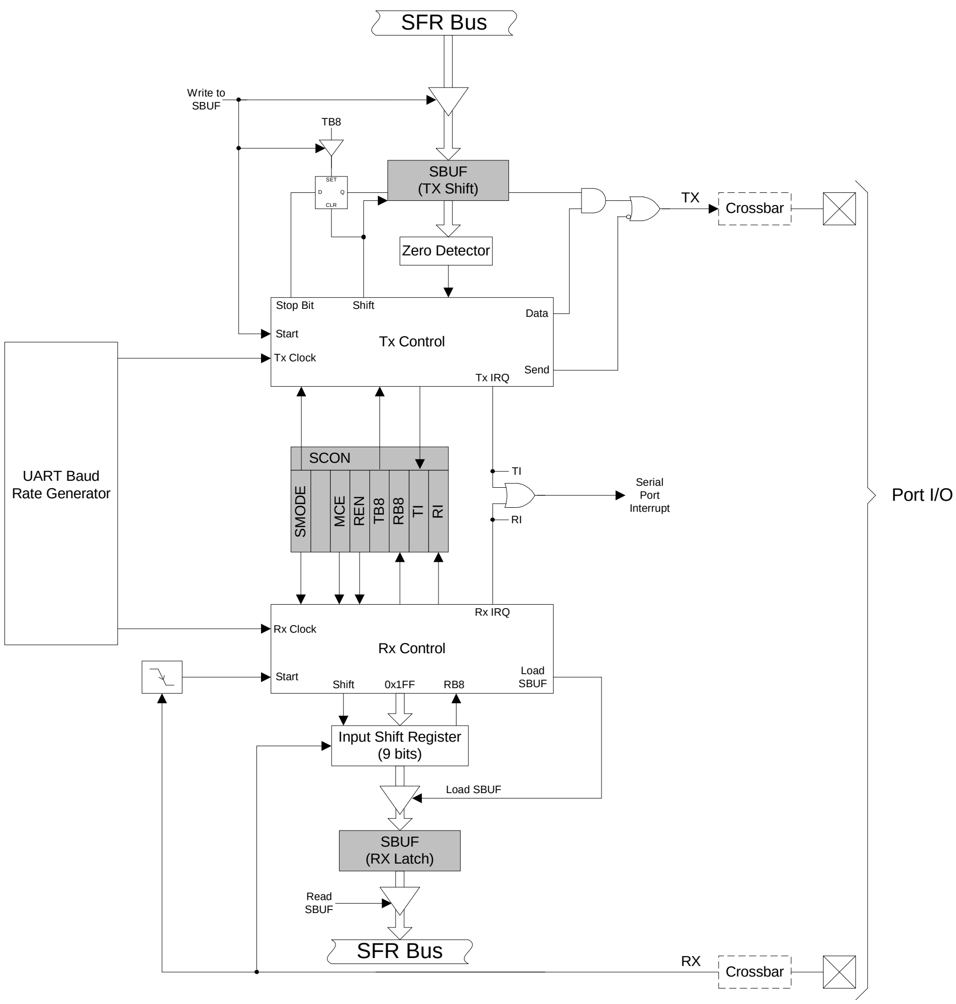

Figure 32.1. UART0 Block Diagram

33. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload. Additionally, Timer 3 offers the ability to be clocked from the external oscillator while the device is in Suspend mode, and can be used as a wake-up source. This allows for implementation of a very low-power system, including RTC capability.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 33.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

SFR Definition 34.4. PCA0CPMn: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPM0 = 0xDA, PCA0CPM1 = 0xDB, PCA0CPM2 = 0xDC
SFR Pages: PCA0CPM0 = F, PCA0CPM1 = F, PCA0CPM2 = F

Bit	Name	Function
7	PWM16n	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOMn	**Comparator Function Enable.** This bit enables the comparator function for PCA module n when set to 1.
5	CAPPn	**Capture Positive Function Enable.** This bit enables the positive edge capture for PCA module n when set to 1.
4	CAPNn	**Capture Negative Function Enable.** This bit enables the negative edge capture for PCA module n when set to 1.
3	MATn	**Match Function Enable.** This bit enables the match function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	**Toggle Function Enable.** This bit enables the toggle function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function for PCA module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	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.

C2 Register Definition 35.4. FPCTL: C2 Flash Programming Control

Bit	7	6	5	4	3	2	1	0
Name	FPCTL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**C2 Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 35.5. FPDAT: C2 Flash Programming Data

Bit	7	6	5	4	3	2	1	0
Name	FPDAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xBF

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase

Document Change List

Revision 0.5 to Revision 1.0

- Updated “Electrical Characteristics” on page 47.
- Updated “Port Input/Output” on page 180.

Revision 0.4 to Revision 0.5

- Removed Incorrect Pin Connections in Figure 1.4 on page 21 and Figure 1.6 on page 23.
- Updated Specifications in Section “9. Electrical Characteristics” on page 47.
- Updated Section “15. Capacitive Sense (CS0)” on page 80 for clarity.
- Corrected “CJNE A, direct, rel” instruction timing in Table 16.1.
- Noted that a minimum SYSCLK speed is required for Flash writes or erases in Section “22.1. Programming The Flash Memory” on page 148, and for EEPROM writes in Section “23.3. Interfacing with the EEPROM” on page 155.
- Corrected P0.3 overvoltage capabilities throughout document.

Revision 0.3 to Revision 0.4

- Updated Section “15. Capacitive Sense (CS0)” on page 80 to reflect Revision B enhancements.
- Added C8051F716 and C8051F717 devices, package information, and features.
- Updated Register 19.1, “HWID: Hardware Identification Byte,” on page 128.
- Corrected minor typographical and formatting errors throughout document.

Revision 0.2 to Revision 0.3

- Corrected Dimension D in the QFN-48 Package Specifications.
- Updated Table 9.1 on page 47.
- Updated Register 10.1, “ADC0CF: ADC0 Configuration,” on page 59.
- Updated Register 14.3, “CPT0MX: Comparator0 MUX Selection,” on page 79.
- Updated Section “28.1.1. Port Pins Configured for Analog I/O” on page 181.
- Updated Register 35.2, “DEVICEID: C2 Device ID,” on page 302.