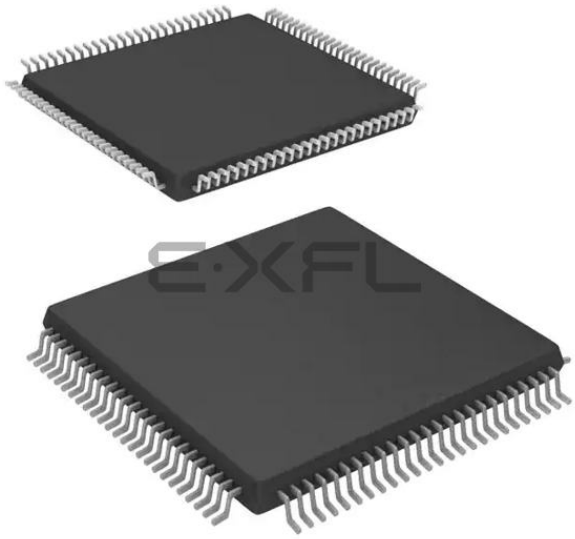


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

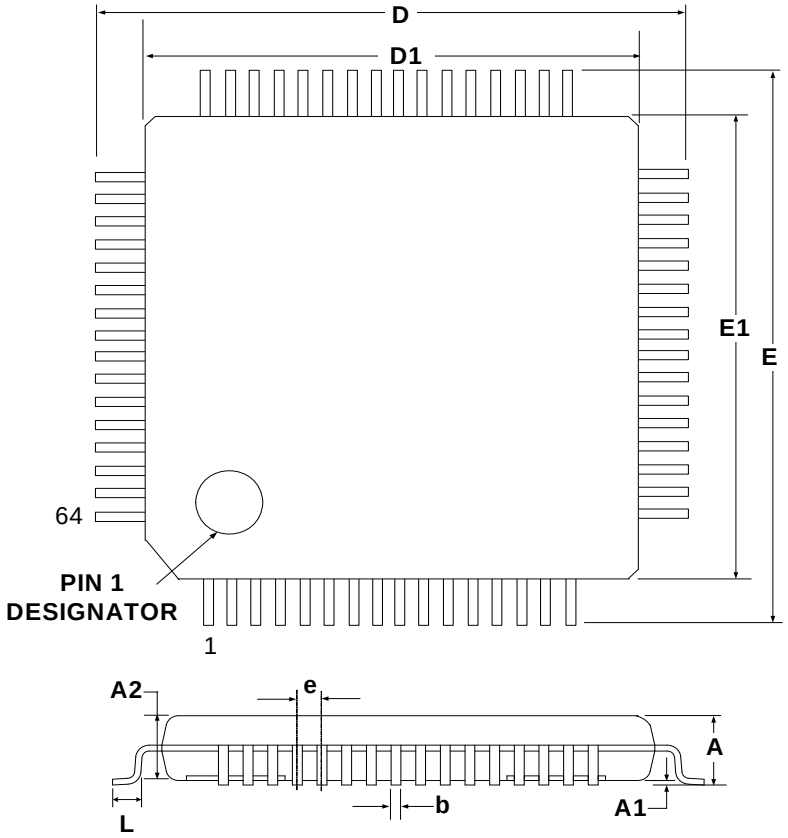
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	100MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	64
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
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/c8051f132-gq

Table 4.1. Pin Definitions (Continued)

Name	Pin Numbers				Type	Description
	‘F120 ‘F122 ‘F124 ‘F126	‘F121 ‘F123 ‘F125 ‘F127	‘F130 ‘F132	‘F131 ‘F133		
VREF	12	7	12	7	A I/O	Bandgap Voltage Reference Output (all devices). DAC Voltage Reference Input (C8051F121/3/5/7 only).
VREFA		8			A In	ADC0 and ADC2 Voltage Reference Input.
VREF0	16		16	8	A In	ADC0 Voltage Reference Input.
VREF2	17		17		A In	ADC2 Voltage Reference Input.
VREFD	15		15		A In	DAC Voltage Reference Input.
AIN0.0	18	9	18	9	A In	ADC0 Input Channel 0 (See ADC0 Specification for complete description).
AIN0.1	19	10	19	10	A In	ADC0 Input Channel 1 (See ADC0 Specification for complete description).
AIN0.2	20	11	20	11	A In	ADC0 Input Channel 2 (See ADC0 Specification for complete description).
AIN0.3	21	12	21	12	A In	ADC0 Input Channel 3 (See ADC0 Specification for complete description).
AIN0.4	22	13	22	13	A In	ADC0 Input Channel 4 (See ADC0 Specification for complete description).
AIN0.5	23	14	23	14	A In	ADC0 Input Channel 5 (See ADC0 Specification for complete description).
AIN0.6	24	15	24	15	A In	ADC0 Input Channel 6 (See ADC0 Specification for complete description).
AIN0.7	25	16	25	16	A In	ADC0 Input Channel 7 (See ADC0 Specification for complete description).
CP0+	9	4	9	4	A In	Comparator 0 Non-Inverting Input.
CP0-	8	3	8	3	A In	Comparator 0 Inverting Input.
CP1+	7	2	7	2	A In	Comparator 1 Non-Inverting Input.
CP1–	6	1	6	1	A In	Comparator 1 Inverting Input.
DAC0	100	64			A Out	Digital to Analog Converter 0 Voltage Output. (See DAC Specification for complete description).



	MIN (mm)	NOM (mm)	MAX (mm)
A	-	-	1.20
A1	0.05	-	0.15
A2	0.95	1.00	1.05
b	0.17	0.22	0.27
D	-	12.00	-
D1	-	10.00	-
e	-	0.50	-
E	-	12.00	-
E1	-	10.00	-
L	0.45	0.60	0.75

Figure 4.6. TQFP-64 Package Drawing

SFR Definition 5.3. ADC0CF: ADC0 Configuration

SFR Page: 0
SFR Address: 0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AMP0GN2	AMP0GN1	AMP0GN0	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
The SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in AD0SC4-0, and CLK_{SAR0} refers to the desired ADC0 SAR clock (Note: the ADC0 SAR Conversion Clock should be less than or equal to 2.5 MHz).

$$AD0SC = \frac{SYSCLK}{2 \times CLK_{SAR0}} - 1 \qquad (AD0SC > 00000\text{b})$$

When the AD0SC bits are equal to 00000b, the SAR Conversion clock is equal to SYSCLK to facilitate faster ADC conversions at slower SYSCLK speeds.

Bits2–0: AMP0GN2–0: ADC0 Internal Amplifier Gain (PGA).
000: Gain = 1
001: Gain = 2
010: Gain = 4
011: Gain = 8
10x: Gain = 16
11x: Gain = 0.5

10-bit ADC0 Data Word appears in the ADC0 Data Word Registers as follows:
ADC0H[1:0]:ADC0L[7:0], if AD0LJST = 0
(ADC0H[7:2] will be sign-extension of ADC0H.1 for a differential reading, otherwise =
000000b).

ADC0H[7:0]:ADC0L[7:6], if AD0LJST = 1
(ADC0L[5:0] = 00b).

Example: ADC0 Data Word Conversion Map, AIN0.0 Input in Single-Ended Mode
(AMX0CF = 0x00, AMX0SL = 0x00)

AIN0.0–AGND (Volts)	**ADC0H:ADC0L** (AD0LJST = 0)	**ADC0H:ADC0L** (AD0LJST = 1)
VREF x (1023/1024)	0x03FF	0xFFC0
VREF / 2	0x0200	0x8000
VREF x (511/1024)	0x01FF	0x7FC0
0	0x0000	0x0000

Example: ADC0 Data Word Conversion Map, AIN0.0-AIN0.1 Differential Input Pair
(AMX0CF = 0x01, AMX0SL = 0x00)

AIN0.0–AIN0.1 (Volts)	**ADC0H:ADC0L** (AD0LJST = 0)	**ADC0H:ADC0L** (AD0LJST = 1)
VREF x (511/512)	0x01FF	0x7FC0
VREF / 2	0x0100	0x4000
VREF x (1/512)	0x0001	0x0040
0	0x0000	0x0000
–VREF x (1/512)	0xFFFF (–1d)	0xFFC0
–VREF / 2	0xFF00 (–256d)	0xC000
–VREF	0xFE00 (–512d)	0x8000

For AD0LJST = 0:

$$Code = Vin \times \frac{Gain}{VREF} \times 2^n$$; 'n' = 10 for Single-Ended; 'n'= 9 for Differential.

Figure 6.5. ADC0 Data Word Example

SFR Definition 6.9. ADC0LTH: ADC0 Less-Than Data High Byte

SFR Page: 0
SFR Address: 0xC7

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	

Bits7–0: High byte of ADC0 Less-Than Data Word.

SFR Definition 6.10. ADC0LTL: ADC0 Less-Than Data Low Byte

SFR Page: 0
SFR Address: 0xC6

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	

Bits7–0: Low byte of ADC0 Less-Than Data Word.

SFR Definition 7.6. ADC2GT: ADC2 Greater-Than Data Byte

SFR Page: 2
SFR Address: 0xC4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–0: ADC2 Greater-Than Data Word.

SFR Definition 7.7. ADC2LT: ADC2 Less-Than Data Byte

SFR Page: 2
SFR Address: 0xC6

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	

Bits7–0: ADC2 Less-Than Data Word.

11.2. 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. There are 256 bytes of internal data memory and 128k bytes (C8051F12x and C8051F130/1) or 64k bytes (C8051F132/3) of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 11.2.

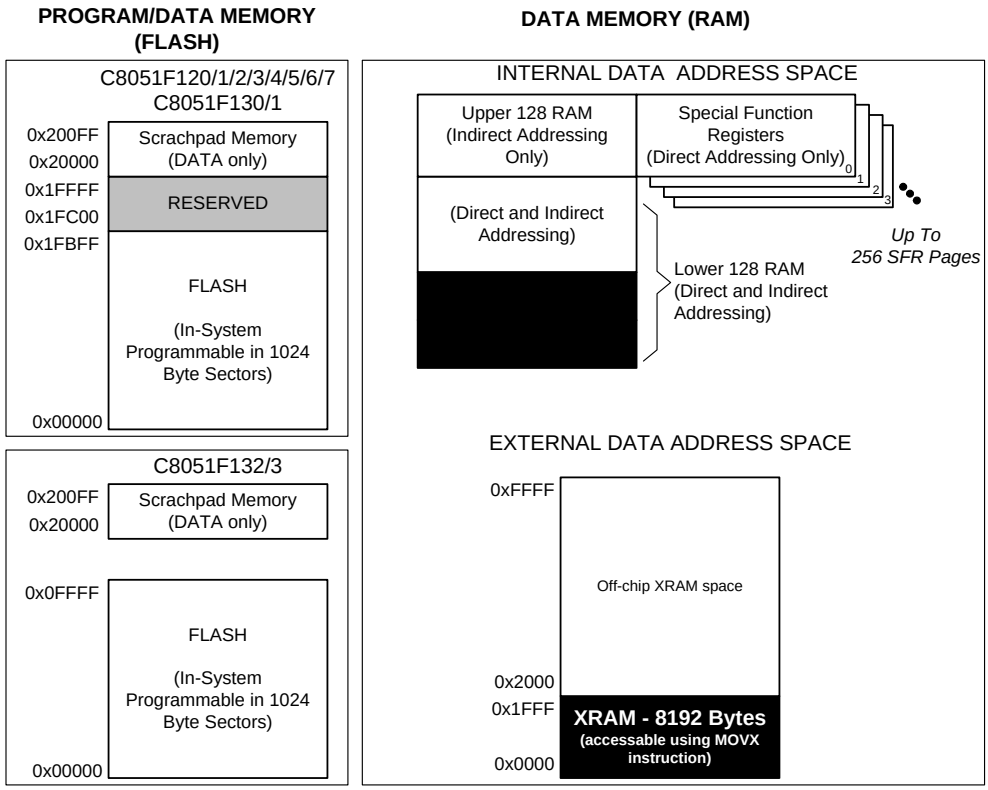


Figure 11.2. Memory Map

11.2.1. Program Memory

The C8051F12x and C8051F130/1 have a 128 kB program memory space. The MCU implements this program memory space as in-system re-programmable Flash memory in four 32 kB code banks. A common code bank (Bank 0) of 32 kB is always accessible from addresses 0x0000 to 0x7FFF. The three upper code banks (Bank 1, Bank 2, and Bank 3) are each mapped to addresses 0x8000 to 0xFFFF, depending on the selection of bits in the PSBANK register, as described in SFR Definition 11.1. The IFBANK bits select which of the upper banks are used for code execution, while the COBANK bits select the bank to be used for direct writes and reads of the Flash memory. Note: 1024 bytes of the memory in Bank 3 (0x1FC00 to 0x1FFFF) are reserved and are not available for user program or data storage. The C8051F132/3 have a 64k byte program memory space implemented as in-system re-programmable Flash memory, and organized in a contiguous block from address 0x00000 to 0x0FFFF.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section “15. Flash Memory” on page 199** for further details.

11.2.2. Data Memory

The CIP-51 implements 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFR's. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 11.2 illustrates the data memory organization of the CIP-51.

11.2.3. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 11.9). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

11.2.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination). The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte.

For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

11.2.5. 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, address 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 which is accessed by the debug logic. 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 pushes two record bits onto the register. (A POP or decrement SP pops one record bit,

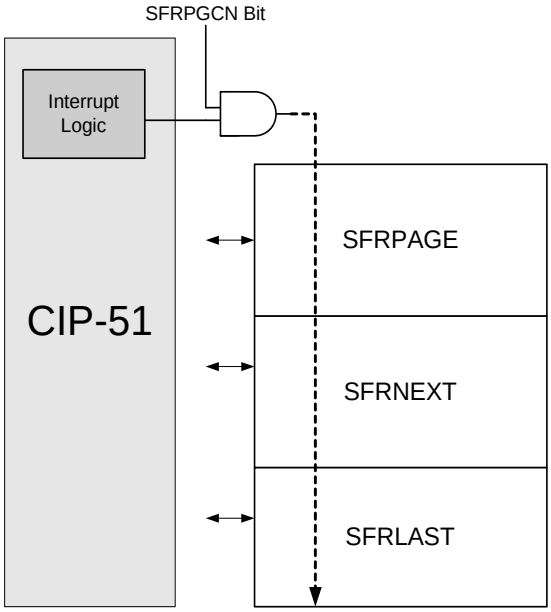


Figure 11.4. SFR Page Stack

Automatic hardware switching of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFRPGCN). This function defaults to ‘enabled’ upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) is provided in Table 11.2. in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Note that certain SFR’s are accessible from ALL SFR pages, and are denoted by the “**(ALL PAGES)**” designation. For example, the Port I/O registers P0, P1, P2, and P3 all have the “**(ALL PAGES)**” designation, indicating these SFR’s are accessible from all SFR pages regardless of the SFRPAGE register value.

11.3.2. External Interrupts

Two of the external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of bits IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flag for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag follows the state of the external interrupt's input pin. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

Table 11.4. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flags	Bit addressable?	Cleared by HW?	SFRPAGE (SFRPGEN = 1)	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	0	Always Enabled	Always Highest
External Interrupt 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	0	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	0	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	0	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	0	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y		0	ES0 (IE.4)	PS0 (IP.4)
Timer 2	0x002B	5	TF2 (TMR2CN.7) EXF2 (TMR2CN.6)	Y		0	ET2 (IE.5)	PT2 (IP.5)
Serial Peripheral Interface	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y		0	ESPI0 (EIE1.0)	PSPI0 (EIP1.0)
SMBus Interface	0x003B	7	SI (SMB0CN.3)	Y		0	ESMB0 (EIE1.1)	PSMB0 (EIP1.1)
ADC0 Window Comparator	0x0043	8	AD0WINT (ADC0CN.1)	Y		0	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
Programmable Counter Array	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y		0	EPCA0 (EIE1.3)	PPCA0 (EIP1.3)
Comparator 0 Falling Edge	0x0053	10	CP0FIF (CPT0CN.4)	Y		1	ECP0F (EIE1.4)	PCP0F (EIP1.4)
Comparator 0 Rising Edge	0x005B	11	CP0RIF (CPT0CN.5)	Y		1	ECP0R (EIE1.5)	PCP0R (EIP1.5)
Comparator 1 Falling Edge	0x0063	12	CP1FIF (CPT1CN.4)	Y		2	ECP1F (EIE1.6)	PCP1F (EIP1.6)

11.4. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the external peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the system clock is stopped. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. SFR Definition 11.18 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

Although the CIP-51 has Idle and Stop modes built in (as with any standard 8051 architecture), power management of the entire MCU is better accomplished by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and put into low power mode. Digital peripherals, such as timers or serial buses, draw little power whenever they are not in use. Turning off the Flash memory saves power, similar to entering Idle mode. Turning off the oscillator saves even more power, but requires a reset to restart the MCU.

11.4.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt or $\overline{\text{RST}}$ is asserted. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x00000.

If enabled, the WDT will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to **Section 13** for more information on the use and configuration of the WDT.

Note: Any instruction which sets the IDLE bit should be immediately followed by an instruction which has two or more opcode bytes. For example:

```
// in ‘C’:
PCON |= 0x01;    // Set IDLE bit
PCON = PCON;     // ... Followed by a 3-cycle Dummy Instruction

; in assembly:
ORL PCON, #01h   ; Set IDLE bit
MOV PCON, PCON   ; ... Followed by a 3-cycle Dummy Instruction
```

If the instruction following the write to the IDLE bit is a single-byte instruction and an interrupt occurs during the execution of the instruction of the instruction which sets the IDLE bit, the CPU may not wake from IDLE mode when a future interrupt occurs.

12. Multiply And Accumulate (MAC0)

The C8051F120/1/2/3 and C8051F130/1/2/3 devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. MAC0 contains a 16-by-16 bit multiplier and a 40-bit adder, which can perform integer or fractional multiply-accumulate and multiply operations on signed input values in two SYSCLK cycles. A rounding engine provides a rounded 16-bit fractional result after an additional (third) SYSCLK cycle. MAC0 also contains a 1-bit arithmetic shifter that will left or right-shift the contents of the 40-bit accumulator in a single SYSCLK cycle. Figure 12.1 shows a block diagram of the MAC0 unit and its associated Special Function Registers.

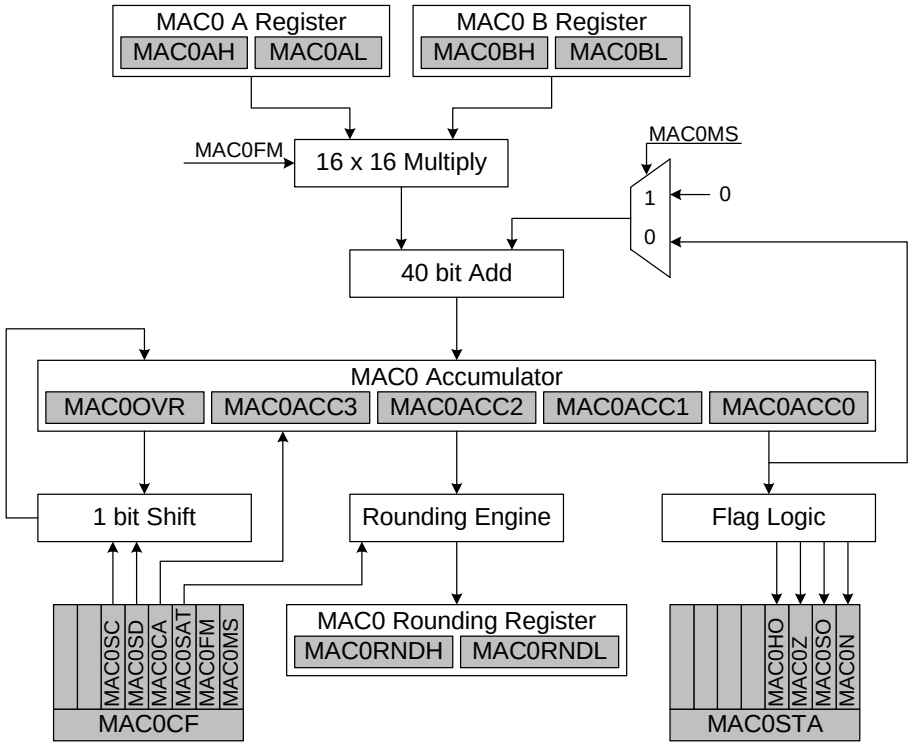


Figure 12.1. MAC0 Block Diagram

12.1. Special Function Registers

There are thirteen Special Function Register (SFR) locations associated with MAC0. Two of these registers are related to configuration and operation, while the other eleven are used to store multi-byte input and output data for MAC0. The Configuration register MAC0CF (SFR Definition 12.1) is used to configure and control MAC0. The Status register MAC0STA (SFR Definition 12.2) contains flags to indicate overflow conditions, as well as zero and negative results. The 16-bit MAC0A (MAC0AH:MAC0AL) and MAC0B (MAC0BH:MAC0BL) registers are used as inputs to the multiplier. The MAC0 Accumulator register is 40 bits long, and consists of five SFRs: MAC0OVR, MAC0ACC3, MAC0ACC2, MAC0ACC1, and MAC0ACC0. The primary results of a MAC0 operation are stored in the Accumulator registers. If they are needed, the rounded results are stored in the 16-bit Rounding Register MAC0RND (MAC0RNDH:MAC0RNDL).

SFR Definition 13.1. WDTCN: Watchdog Timer Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								xxxxx111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xFF
SFR Page: All Pages

Bits7–0: WDT Control
Writing 0xA5 both enables and reloads the WDT.
Writing 0xDE followed within 4 system clocks by 0xAD disables the WDT.
Writing 0xFF locks out the disable feature.

Bit4: Watchdog Status Bit (when Read)
Reading the WDTCN.[4] bit indicates the Watchdog Timer Status.
0: WDT is inactive
1: WDT is active

Bits2–0: Watchdog Timeout Interval Bits
The WDTCN.[2:0] bits set the Watchdog Timeout Interval. When writing these bits, WDTCN.7 must be set to 0.

SFR Definition 14.3. CLKSEL: System Clock Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CLKDIV1	CLKDIV0	-	-	CLKSL1	CLKSL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x97
SFR Page: F

Bits 7–6: Reserved.

Bits 5–4: CLKDIV1–0: Output SYSCLK Divide Factor.
These bits can be used to pre-divide SYSCLK before it is output to a port pin through the crossbar.
00: Output will be SYSCLK.
01: Output will be SYSCLK/2.
10: Output will be SYSCLK/4.
11: Output will be SYSCLK/8.
See **Section “18. Port Input/Output” on page 235** for more details about routing this output to a port pin.

Bits 3–2: Reserved.

Bits 1–0: CLKSL1–0: System Clock Source Select Bits.
00: SYSCLK derived from the Internal Oscillator, and scaled as per the IFCN bits in OSCICN.
01: SYSCLK derived from the External Oscillator circuit.
10: SYSCLK derived from the PLL.
11: Reserved.

SFR Definition 14.8. PLL0FLT: PLL Filter

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PLLICO1	PLLICO0	PLLLP3	PLLLP2	PLLLP1	PLLLP0	00110001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8F
SFR Page: F

Bits 7–6: UNUSED: Read = 00b; Write = don't care.
Bits 5–4: PLLICO1-0: PLL Current-Controlled Oscillator Control Bits.
Selection is based on the desired output frequency, according to the following table:

PLL Output Clock	PLLICO1-0
65–100 MHz	00
45–80 MHz	01
30–60 MHz	10
25–50 MHz	11

Bits 3–0: PLLLP3-0: PLL Loop Filter Control Bits.
Selection is based on the divided PLL reference clock, according to the following table:

Divided PLL Reference Clock	PLLLP3-0
19–30 MHz	0001
12.2–19.5 MHz	0011
7.8–12.5 MHz	0111
5–8 MHz	1111

Table 14.2. PLL Frequency Characteristics

–40 to +85 °C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Input Frequency (Divided Reference Frequency)		5		30	MHz
PLL Output Frequency		25		100*	MHz
***Note:** The maximum operating frequency of the C8051F124/5/6/7 is 50 MHz					

Table 15.1. Flash Electrical Characteristics

V_{DD} = 2.7 to 3.6 V; –40 to +85 °C

Parameter	Conditions	Min	Typ	Max	Units
Flash Size[1]	C8051F12x and C8051F130/1		131328[2]		Bytes
Flash Size[1]	C8051F132/3		65792		Bytes
Endurance		20k	100k		Erase/Write
Erase Cycle Time		10	12	14	ms
Write Cycle Time		40	50	60	µs

Notes:
1. Includes 256-byte Scratch Pad Area
2. 1024 Bytes at location 0x1FC00 to 0x1FFFF are reserved.

15.1.1. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written and erased using the MOVX write instruction (as described in **Section 15.1.2** and **Section 15.1.3**) and read using the MOVC instruction. The COBANK bits in register PSBANK (SFR Definition 11.1) control which portion of the Flash memory is targeted by writes and erases of addresses above 0x07FFF. For devices with 64 kB of Flash. the COBANK bits should always remain set to '01' to ensure that Flash write, erase, and read operations are valid.

Two additional 128-byte sectors (256 bytes total) of Flash memory are included for non-volatile data storage. The smaller sector size makes them particularly well suited as general purpose, non-volatile scratch-pad memory. Even though Flash memory can be written a single byte at a time, an entire sector must be erased first. In order to change a single byte of a multi-byte data set, the data must be moved to temporary storage. The 128-byte sector-size facilitates updating data without wasting program memory or RAM space. The 128-byte sectors are double-mapped over the normal Flash memory for MOVC reads and MOVX writes only; their addresses range from 0x00 to 0x7F and from 0x80 to 0xFF (see Figure 15.2). To access the 128-byte sectors, the SFLE bit in PSCTL must be set to logic 1. Code execution from the 128-byte Scratchpad areas is not permitted. The 128-byte sectors can be erased individually, or both at the same time. To erase both sectors simultaneously, the address 0x0400 should be targeted during the erase operation with SFLE set to '1'. See Figure 15.1 for the memory map under different COBANK and SFLE settings.

Table 17.1. AC Parameters for External Memory Interface

Parameter	Description	Min	Max	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20	—	ns
T_{RDH}	Read Data Hold Time	0	—	ns

Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

ple, to assign TX0 to a Port pin without assigning RX0 as well. Each combination of enabled peripherals results in a unique device pinout.

All Port pins on Ports 0 through 3 that are not allocated by the Crossbar can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See SFR Definition 18.4, SFR Definition 18.6, SFR Definition 18.9, and SFR Definition 18.11), a set of SFR's which are both byte- and bit-addressable. The output states of Port pins that are allocated by the Crossbar are controlled by the digital peripheral that is mapped to those pins. Writes to the Port Data registers (or associated Port bits) will have no effect on the states of these pins.

A Read of a Port Data register (or Port bit) will always return the logic state present at the pin itself, regardless of whether the Crossbar has allocated the pin for peripheral use or not. An exception to this occurs during the execution of a *read-modify-write* instruction (ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR, SETB, and the bitwise MOV write operation). During the *read* cycle of the *read-modify-write* instruction, it is the contents of the Port Data register, not the state of the Port pins themselves, which is read. Note that at clock rates above 50 MHz, when a pin is written and then immediately read (i.e. a write instruction followed immediately by a read instruction), the propagation delay of the port drivers may cause the read instruction to return the previous logic level of the pin.

Because the Crossbar registers affect the pinout of the peripherals of the device, they are typically configured in the initialization code of the system before the peripherals themselves are configured. Once configured, the Crossbar registers are typically left alone.

Once the Crossbar registers have been properly configured, the Crossbar is enabled by setting XBARE (XBR2.4) to a logic 1. **Until XBARE is set to a logic 1, the output drivers on Ports 0 through 3 are explicitly disabled in order to prevent possible contention on the Port pins while the Crossbar registers and other registers which can affect the device pinout are being written**.

The output drivers on Crossbar-assigned input signals (like RX0, for example) are explicitly disabled; thus the values of the Port Data registers and the PnMDOUT registers have no effect on the states of these pins.

18.1.2. Configuring the Output Modes of the Port Pins

The output drivers on Ports 0 through 3 remain disabled until the Crossbar is enabled by setting XBARE (XBR2.4) to a logic 1.

The output mode of each port pin can be configured to be either Open-Drain or Push-Pull. In the Push-Pull configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and writing a logic 1 will cause the Port pin to be driven to V_{DD}. In the Open-Drain configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire (like the SDA signal on an SMBus connection).

The output modes of the Port pins on Ports 0 through 3 are determined by the bits in the associated PnMDOUT registers (See SFR Definition 18.5, SFR Definition 18.8, SFR Definition 18.10, and SFR Definition 18.12). For example, a logic 1 in P3MDOUT.7 will configure the output mode of P3.7 to Push-Pull; a logic 0 in P3MDOUT.7 will configure the output mode of P3.7 to Open-Drain. All Port pins default to Open-Drain output.

Table 20.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing* (See Figure 20.8 and Figure 20.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}		ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}		ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20		ns
T_{MIH}	SCK Shift Edge to MISO Change	0		ns
Slave Mode Timing* (See Figure 20.10 and Figure 20.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}		ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}		ns
T_{SEZ}	NSS Falling to MISO Valid		4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z		4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}		ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}		ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}		ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}		ns
T_{SOH}	SCK Shift Edge to MISO Change		4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).