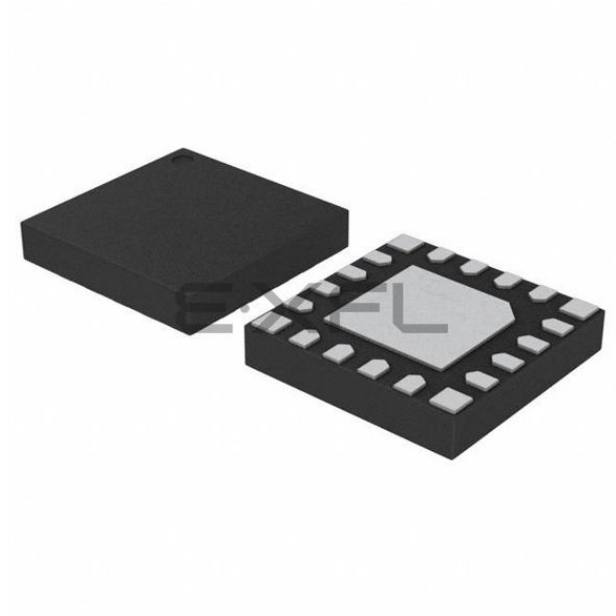


Welcome to E-XFL.COM

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	CIP-51 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	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 9x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f980-gmr

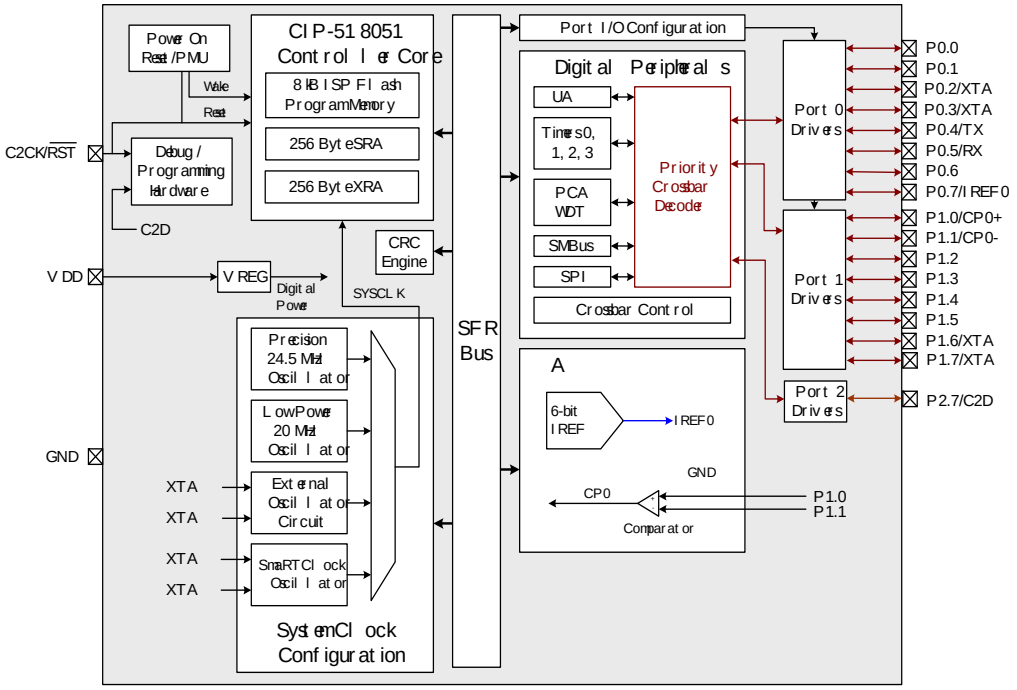


Figure 1.7. C8051F987 Block Diagram

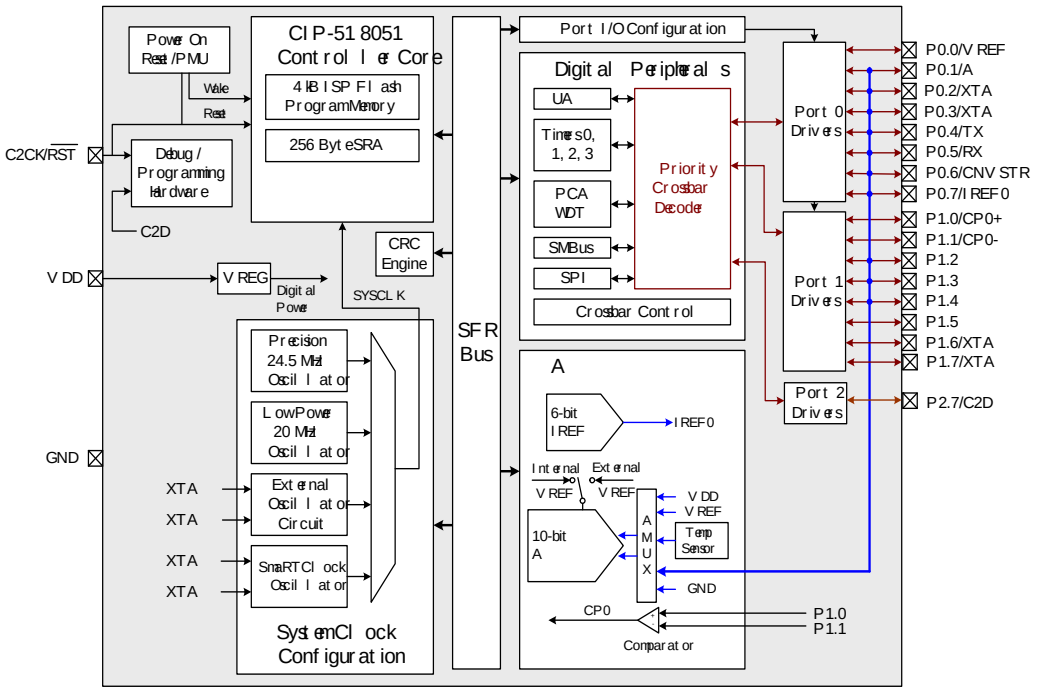


Figure 1.8. C8051F988 Block Diagram

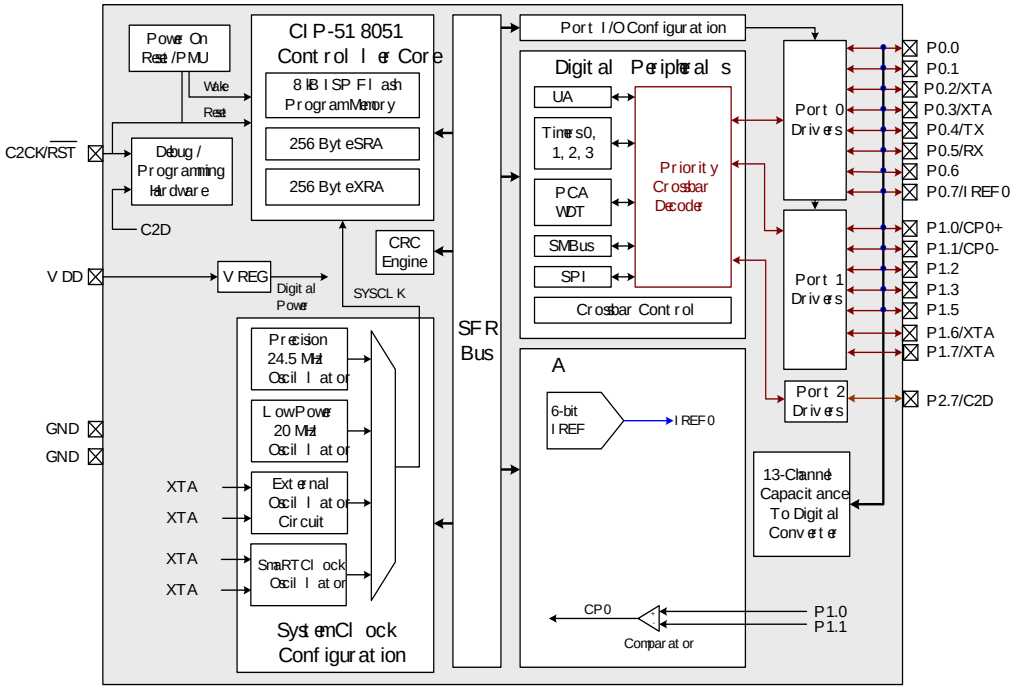


Figure 1.11. C8051F991 Block Diagram

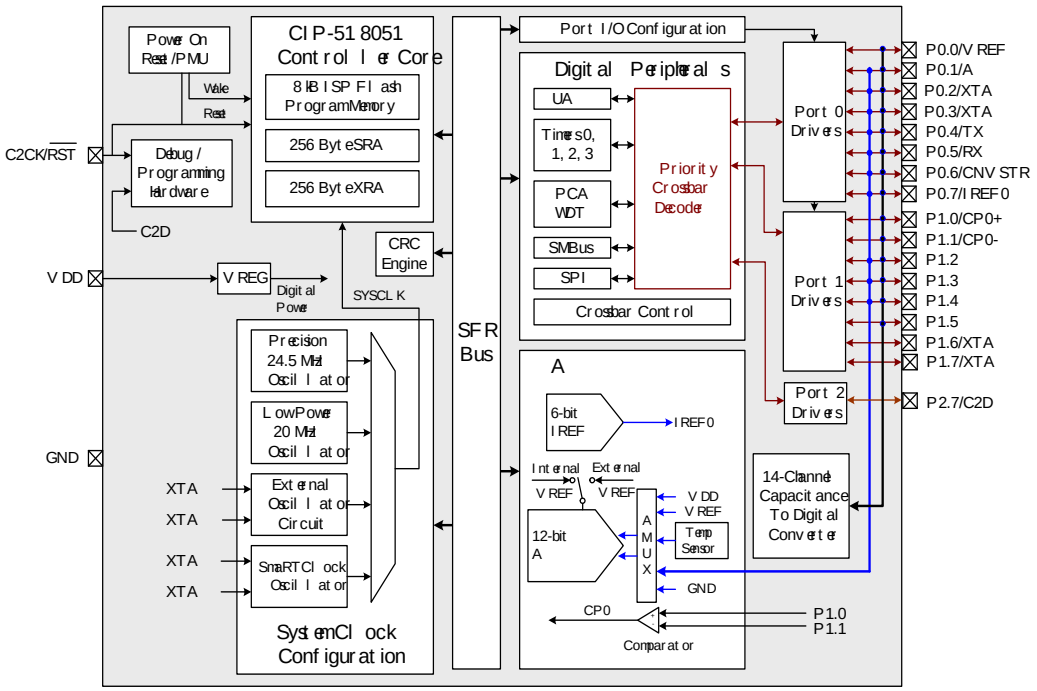


Figure 1.12. C8051F996 Block Diagram

1.5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode

C8051F99x-C8051F98x devices have a 300 ksps, 10-bit or 75 ksps 12-bit successive-approximation-register (SA d programmable window detector. A autonomous low power Burst Mode which can automatically enable A samples, then place A mode without CPU intervention. It also has a 16-bit accumulator that can automatically average the A sults, providing an effective 11, 12, or 13 bit A result without any additional CPU intervention.

The A Figure 1.17) and has an on-chip attenuator that allows it to measure voltages up to twice the voltage reference A onal A temperature sensor, the V DD supply voltage, and the internal digital supply voltage

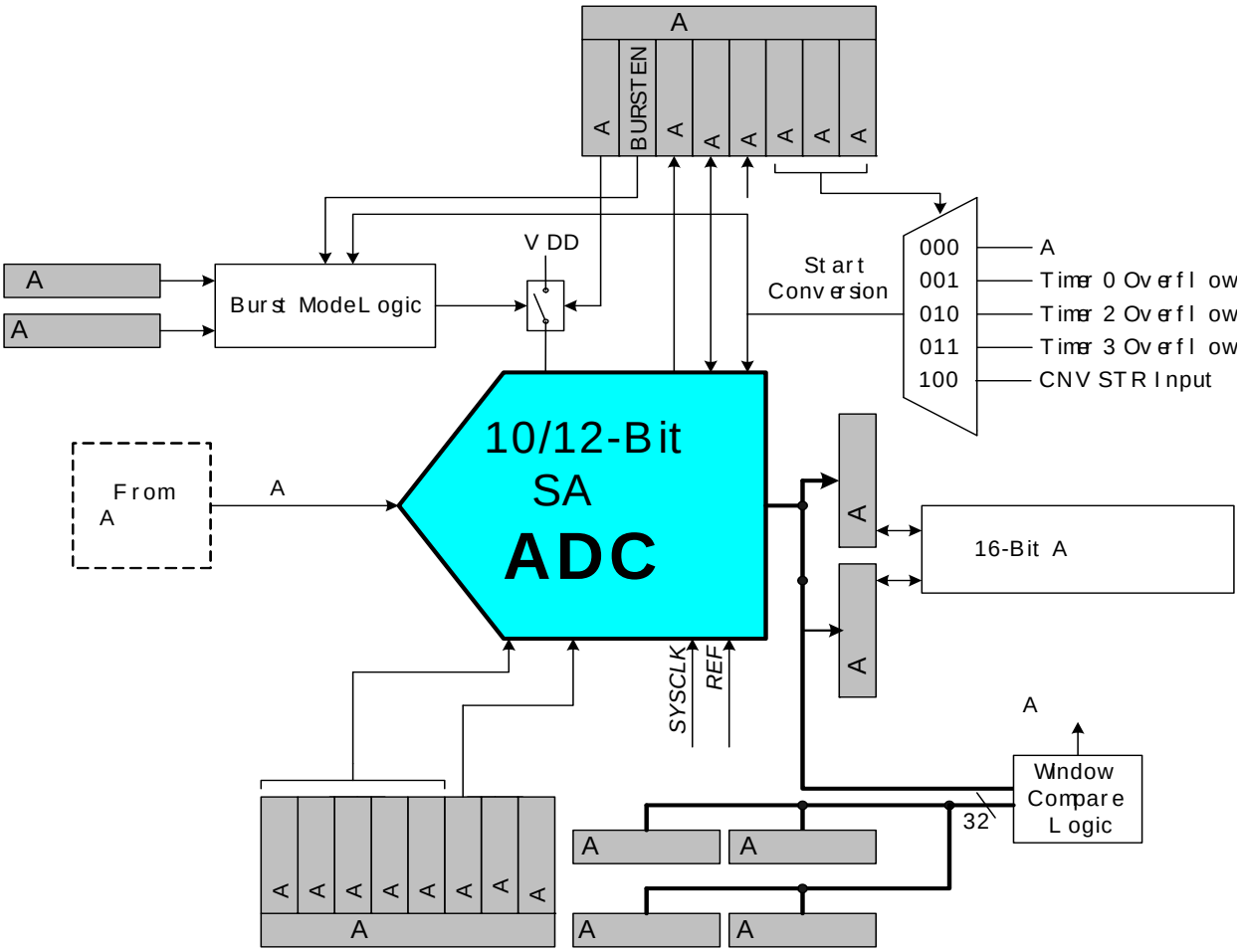
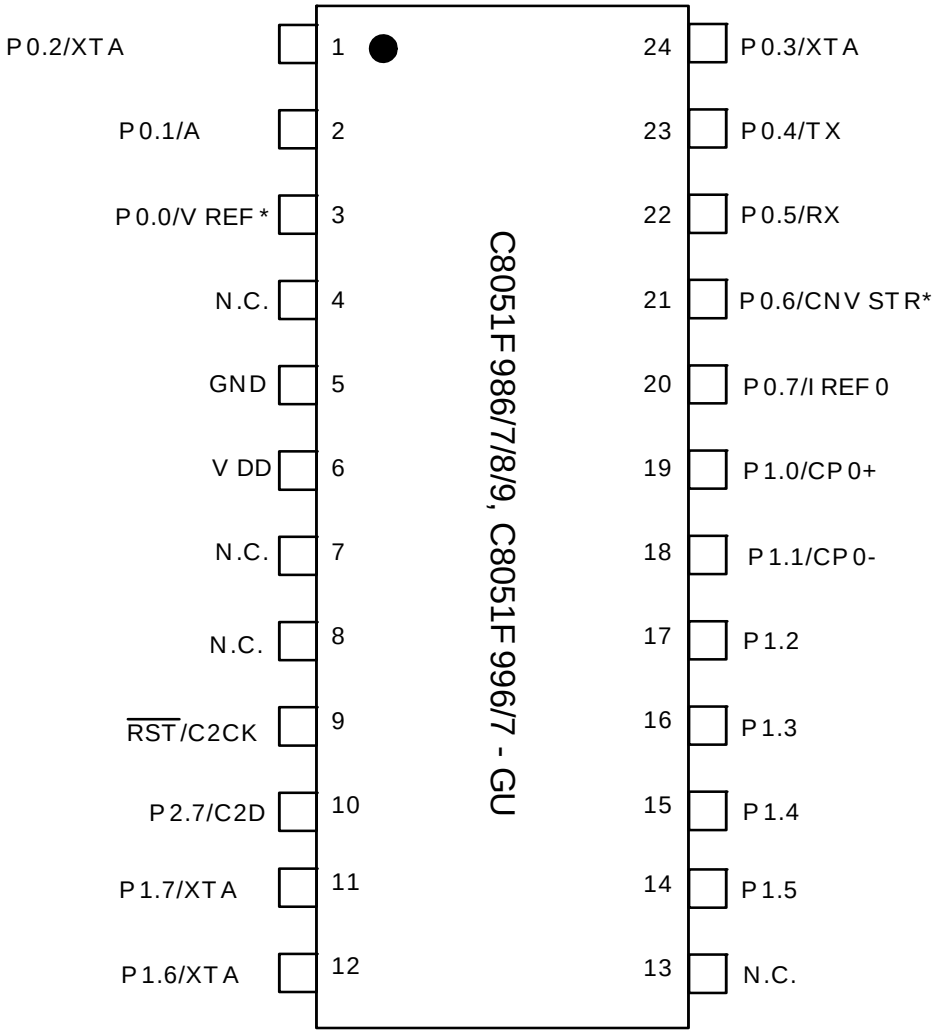


Figure 1.16. ADC0 Functional Block Diagram



*__Note:__ Signal only available on ‘F986, ‘F988, and ‘F996 devices.

Figure 3.3. QSOP-24 Pinout Diagram (Top View)

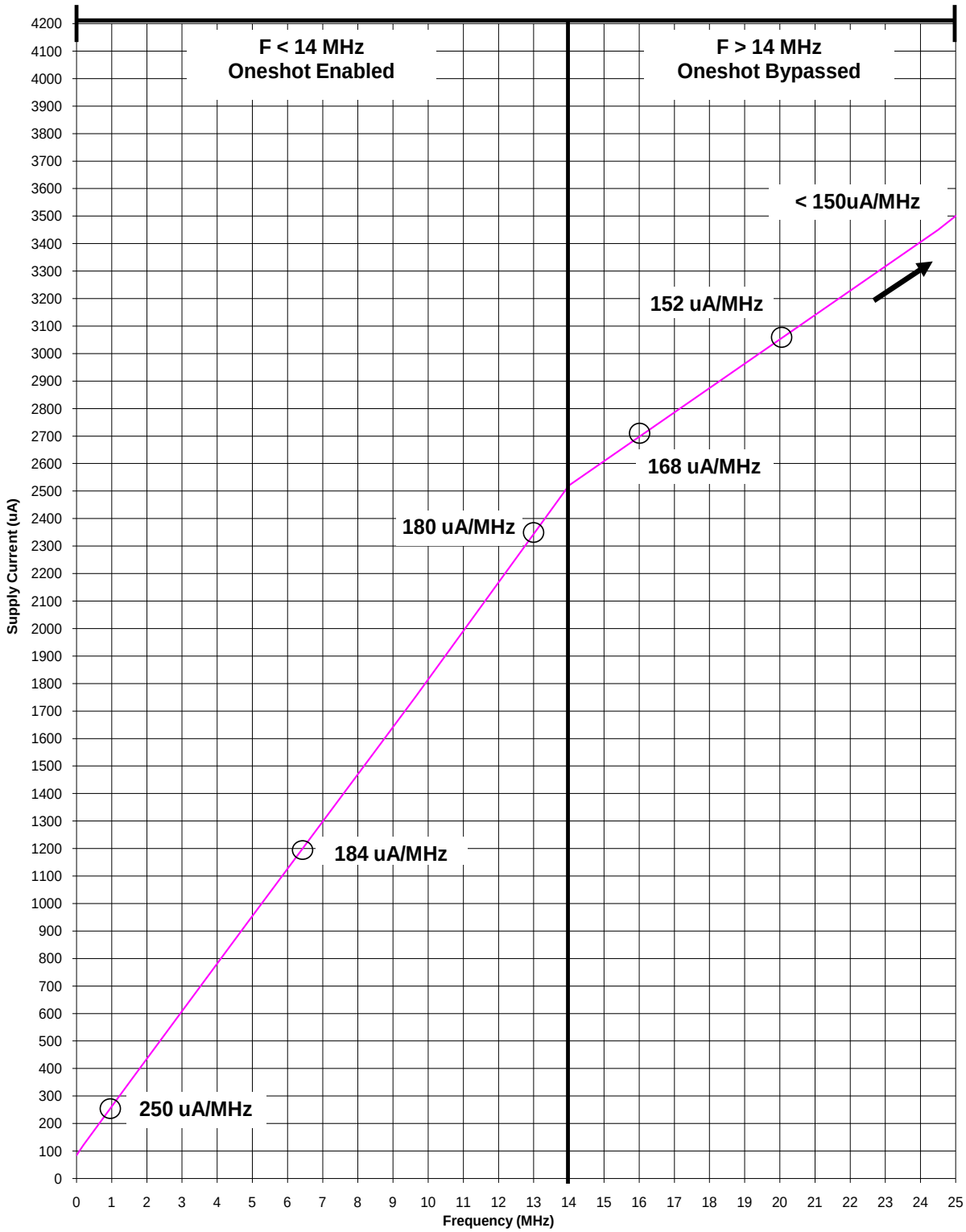


Figure 4.1. Active Mode Current (External CMOS Clock)

5.2.3. Burst Mode

Burst Mode is a power saving feature that allows A low power state between conversions. When Burst Mode is enabled, A from a low power state, accumulates 1, 4, 8, 16, 32, or 64 using an internal Burst Mode clock (approximately 2MHz), then reenters a low power state. Since the Burst Mode clock is independent of the system clock, A rm multiple conversions then enter a low power state within a single system clock cycle, even if the system clock is slow (eg. 32.768 kHz), or suspended.

Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, A idle power state (i.e. the state A to logic 0, A 1, A se nabled after each burst. On each convert start signal , A ened from its Idle Power State. If A down, it will automatically power up and wait pt ogrammable Power-Up Time controlled by the A tracking and converting immediately. Figure 5.3 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the A Interrupt Flag (A will be set after "repeat count" conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until "repeat count" conversions have been accumulated.

In Burst Mode, tracking is determined by the settings in A these registers will work in most applications without modification; however, settling time requirements may need adjustment in some applications. Refer to "5.2.4. Settling Time Requirements" on page 71 for more details.

Notes:

- Setting A king before each conversion, regardless of the settings of A
- When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.

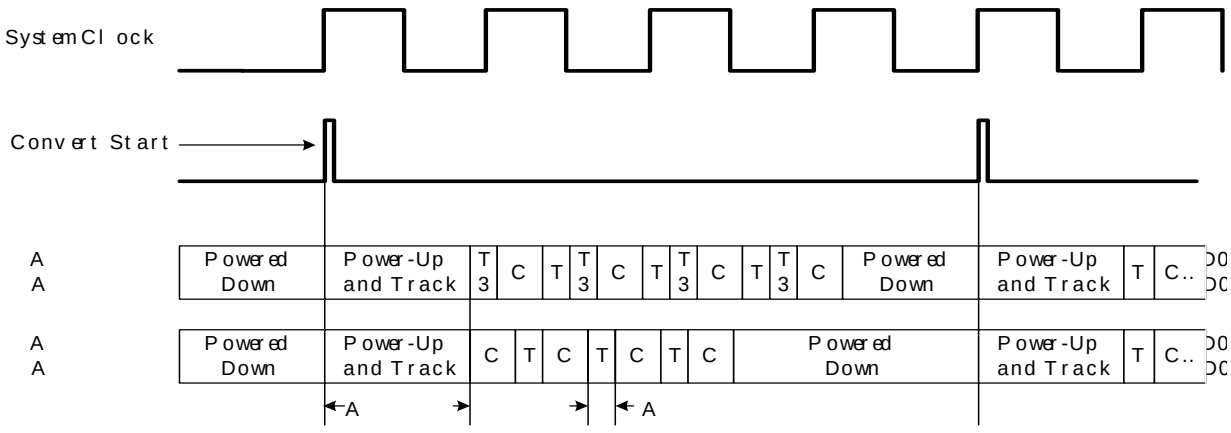


Figure 5.3. Burst Mode Tracking Example with Repeat Count Set to 4

8.12. CS0 Pin Monitor

The CS0 module provides accurate conversions in all operating modes of the CPU, peripherals and I/O ports. Pin monitoring circuits are provided to improve interference immunity from high-current output pin switching. The CS0 Pin Monitor register (CS0PM, SFR Definition 8.14) controls the operation of these pin monitors.

Conversions in the CS0 module are immune to any change on digital inputs and immune to most output switching. Even high-speed serial data transmission will not affect CS0 operation as long as the output load is limited. Output changes that switch large loads such as LEDs and heavily-loaded communications lines can affect conversion accuracy. For this reason, the CS0 module includes pin monitoring circuits that will, if enabled, automatically adjust conversion timing if necessary to eliminate any effect from high-current output pin switching.

The pin monitor enable bit should be set for any output signal that is expected to drive a large load.

Example: The SMBus in a system is heavily loaded with multiple slaves and a long PCB route. Set the SMBus pin monitor enable, SMBPM = 1.

Example: Timer 2 controls an LED on Port 1, pin 3 to provide variable dimming. Set the Port SFR write monitor enable, PIOPM = 1.

Example: The SPI bus is used to communicate to a nearby host. The pin monitor is not needed because the output is not heavily loaded, SPIPM remains = 0, the default reset state.

Pin monitors should not be enabled unless they are required. The pin monitor works by repeating any portion of a conversion that may have been corrupted by a change on an output pin. Setting pin monitor enable bits will slow CS0 conversions.

The frequency of CS0 retry operations can be limited by setting the CSPMMD bits. In the default (reset) state, all converter retry requests will be performed. This is the recommended setting for all applications. The number of retries per conversion can be limited to either two or four retries by changing CSPMMD. Limiting the number of retries per conversion ensures that even in circumstances where extremely frequent high-power output switching occurs, conversions will be completed, though there may be some loss of accuracy due to switching noise.

A an be detected by reading the Pin Monitor Event bit, CS0PME, in register CS0CN. This bit will be set if any CS0 converter retries have occurred. It remains set until cleared by software or a device reset.

8.13. Adjusting CS0 For Special Situations

There are several configuration options in the CS0 module designed to modify the operation of the circuit and address special situations. In particular, any circuit with more than 500 ohms of series impedance between the sensor and the device pin may require adjustments for optimal performance. Typical applications which may require adjustment include:

- Touchpanel sensors fabricated using a resistive conductor such as indium-tin-oxide (ITO).
- Circuits using a high-value series resistor to isolate the sensor element for high ESD protection.

Most systems will require no fine tuning, and the default settings for CS0DT, CS0DR, CS0IA, CS0RP and CS0LP should be used.

12.1. SFR Paging

To accommodate more than 128 SFRs in the 0x80 to 0xFF address space, SFR paging has been implemented. By default, all SFR accesses target SFR Page 0x0 to allow access to the registers listed in Table 12.1. During device initialization, some SFRs located on SFR Page 0xF may need to be accessed. Table 12.2 lists the SFRs accessible from SFR Page 0x0F. Some SFRs are accessible from both pages, including the SFRPA register. SFRs only accessible from Page 0xF are in **bold**.

The following procedure should be used when accessing SFRs on Page 0xF:

1. Save the current interrupt state (EA)
2. Disable Interrupts (EA)
3. Set SFRPA
4. A
5. Set SFRPA
6. Restore interrupt state (EA)

Table 12.2. Special Function Register (SFR) Memory Map (Page 0xF)

F8								
F0	B			**CS0MD3**			EIP1	EIP2
E8								
E0	A		**REVID**	**DEVICEID**		FLWR	EIE1	EIE2
D8							**CS0PM**	
D0	PSW							
C8								
C0								
B8		**IREF0CF**		A	A			
B0						**PMU0MD**		
A	IE	CLKSEL						
A	P2							SFRPA
98		**P0DRV**	CRC0CNT	**P1DRV**	CRC0FLIP	**P2DRV**	CRC0A	
90	P1							
88						**TOFFL**	**TOFFH**	
80	P0	SP	DPL	DPH	CRC0CN	CRC0IN	CRC0DA	PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

14.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. A
constants from being read or altered across the C2 interface.

A Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. See **Section "10. Memory Organization" on page 128** for the location of the security byte. The Flash security mechanism allows the user to lock *n* 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where *n* is the 1s complement number represented by the Security Lock Byte. **The page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are 1) and locked when any other Flash pages are locked (any bit of the Lock Byte is 0).**

Security Lock Byte	1111 1011b
ones Complement:	0000 0100b
Flash pages locked:	5 (First four Flash pages + Lock Byte Page)

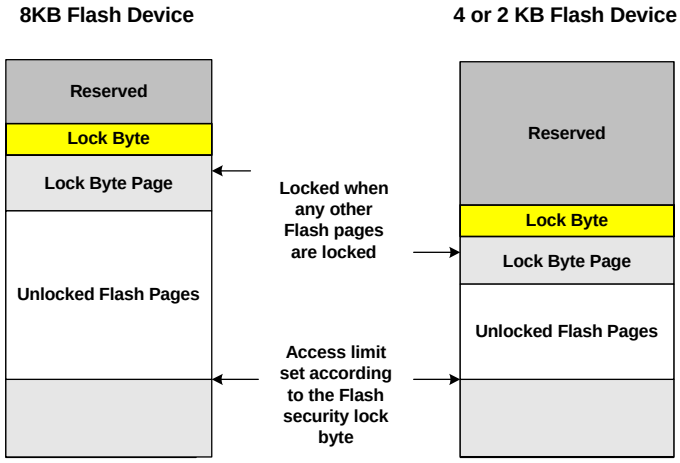


Figure 14.1. Flash Program Memory Map (8 kB and smaller devices)

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 14.1 summarizes the Flash security features of the C8051F99x-C8051F98x devices.

15.1. Normal Mode

The MCU is fully functional in normal mode. Figure 15.1 shows the on-chip power distribution to various peripherals. There are two supply voltages powering various sections of the chip: V_{DD} and the 1.8 V internal core supply. A V_{DD} pin. A CIP-51 core are powered from the 1.8 V internal core supply. RA al ways powered directly from the V_{DD} pin in sleep mode and powered fr om the core supply in all other power modes.

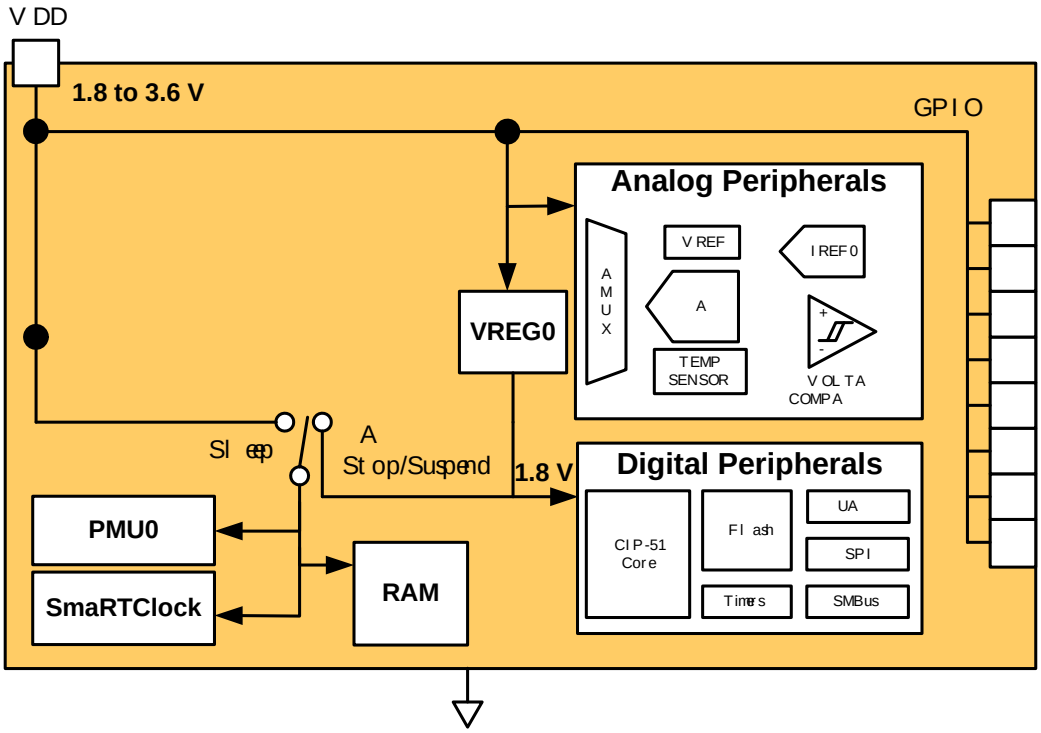


Figure 15.1. C8051F99x-C8051F98x Power Distribution

18.8. SmaRTClock (Real Time Clock) Reset

The SmaRTClock can generate a system reset on two events: SmaRTClock Oscillator Fail or SmaRTClock A
il event occurs when the SmaRTClock Missing Clock Detector is enabled and the SmaRTClock clock is below approximately 20kHz. A
event occurs when the SmaRTClock A
led and the SmaRTClock timer value matches the A
flag (RSTSRC.7). The SmaRTClock reset remains functional even when the device is in the low power Suspend or Sleep mode. The state of the RST pin is unaffected by this reset.

18.9. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit (RSTSRC.4). The SWRSF bit will read 1 following a software forced reset. The state of the RST pin is unaffected by this reset.

20.1.2. Using RTC0ADR and RTC0DAT to Access SmaRTClock Internal Registers

The SmaRTClock internal registers can be read and written using RTC0A
RTC0A ckinternal register that will be targeted by subsequent reads or writes. Recommended instruction timing is provided in this section. If the recommended instruction timing is not followed, then BUSY (RTC0A
sure the SmaRTClock Interface is not busy performing the previous read or write operation. A -
Clock Write operation is initiated by writing to the RTC0DA wis an example of writing to a SmaRTClock internal register.

1. Poll BUSY (RTC0A it returns 0 or follow recommended instruction timing.
2. Write 0x05 to RTC0A al RTC0CN register at SmaRTClock A
3. Write 0x00 to RTC0DA

A the SmaRTClock Interface Busy bit. This transfers the contents of the internal register selected by RTC0A ansferred data will remain in RTC0DA
register.

1. Poll BUSY (RTC0A it returns 0 or follow recommended instruction timing.
2. Write 0x05 to RTC0A al RTC0CN register at SmaRTClock A
3. Write 1 to BUSY. This initiates the transfer of data from RTC0CN to RTC0DA
4. Poll BUSY (RTC0A it returns 0 or follow mmend instruction timing.
5. Read data from RTC0DA is a copy of the RTC0CN register.

Note: The RTC0A

20.1.3. RTC0ADR Short Strobe Feature

Reads and writes to indirect SmaRTClock registers normally take 7 system clock cycles. To minimize the indirect register access time, the Short Strobe feature decreases the read and write access time to 6 system clocks. The Short Strobe feature is automatically enabled on reset and can be manually enabled/disabled using the SHORT (RTC0A

Recommended Instruction Timing for a single register read with short strobe enabled:

```
mov RTC0ADR, #095h
nop
nop
nop
mov A, RTC0DAT
```

Recommended Instruction Timing for a single register write with short strobe enabled:

```
mov RTC0ADR, #095h
mov RTC0DAT, #000h
nop
```

20.1.4. SmaRTClock Interface Autoread Feature

When A
SmaRTClock internal register selected by RTC0A -
ning of each series of consecutive reads. Software should follow recommended instruction timing or check if the SmaRTClock Interface is busy prior to reading RTC0DA
(RTC0A

22.3.2. Arbitration

A ebusisfreeafter a STOP condition or after theSCL and SDA Section “22.3.5. SCL High(SMBusFree) Timeout”on page 238). In theevent that two or more devicesattempt to begin a transfer at thesametime, an arbitra - tion schemeisemployed to forceonemaster to give up thebus. Themaster devicescontinuetransmitting until oneattemptsa HIGH,whil ethe other transmits a LOW. Sincethebusisopen-drain, thebuswil l be pul led LOW. Themaster attempting theHIGHwil l detect a LOWSDA set hearbitration. Thewinning master continuesitstransmission without interruption; thel osing master becomesa sl aveandreceivesthe rest of thetransfer if addressed. Thisarbitration schemeisnon-destru ctive: onedevicealwayswins, and no data isl ost.

22.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I 2C, whichal lowsdeviceswithdifferent speed capabilitiesto coexist on the bus. A ring a transfer in order to allow slower slavedevicesto communicatewithfaster masters. Theslavemay temporarily hold theSCL line LOWto extend theclock low period, effectively decreasing theserial clock frequency.

22.3.4. SCL Low Timeout

If theSCL lineisheld low by a slavedeviceon thebus, no further communication ispossible. Furthermore, themaster cannot forcetheSCL linehigh to correct theerror condition. To solvethisproblem, theSMBus protocol specifiesthat devicesparticipating in a transfer must detect any clock cycleheld low longer than 25 ms as a “timeout”condition. Devicesthat haveddetected thetimeout condition must resetthecommuni - cation no later than 10 after detecting thetimeout condition.

When theSMBTOE bit in SMB0CF isset, Timer 3 isused to detect SCL low timeouts. Timer 3 isforced to reload when SCL ishigh, and allowed to count when SCL islow. WithTimer 3 enabled and configured to overflow after 25 ms(and SMBTOE set), theTimer 3 interrupt serviceroutinecan beused to reset (disable and reenable) theSMBusin theevent of an SCL low timeout.

22.3.5. SCL High (SMBus Free) Timeout

TheSMBusspecification stipulatesthat if theSCL and SDA ps, thebus isdesignated asfree. When theSM BFTE bit in SMB0CF isset, thebuswil l beconsidered free if SCL and SDA SMBusclocksourceperiods(asdefined by thetimer configured for the SMBus clocksource). If theSMBusiswaiting to generatea Master STA following thistimeout. Notethat a clocksourceisrequired for freetimeout detection, even in a slaveonly implementation.

22.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the STA containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic (REA received from the slave on SDA the serial clock. The slave transmits one or more bytes of serial data.

If hardware A received byte Software must write the A

With hardware A led, the SMBus hardware will automatically generate the A and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

Writing a 1 to the A an A the A NA the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0 DA Figure 22.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur at different places in the sequence, depending on whether hardware A eration is enabled. The interrupt occurs **before** the A **after** the A

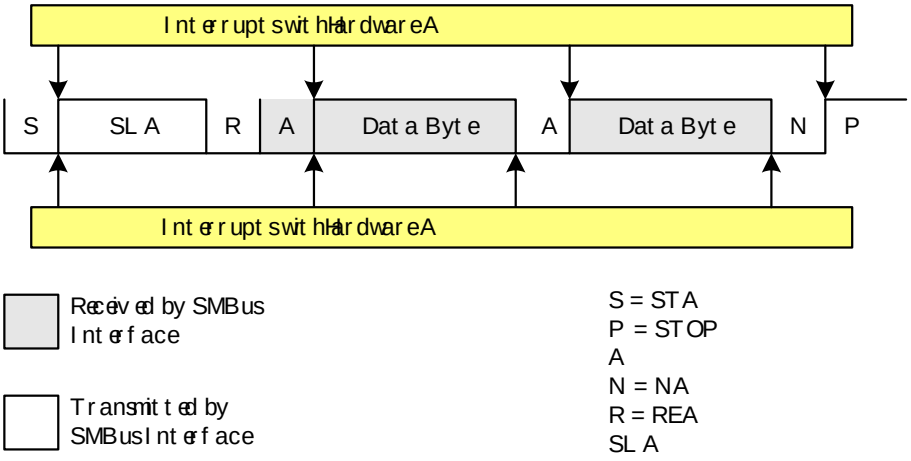


Figure 22.6. Typical Master Read Sequence

Table 22.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A - at ed.	L oad sl av eaddr ess+ R/Wint o SMB 0DA	0	0	X	1100
	1100	0	0	0	A was t ransmit t ed; NA r ecei v ed.	Set STA	1	0	X	1110
						A	0	1	X	—
		0	0	1	A was t ransmit t ed; A r ecei v ed.	L oad next dat a byt e int o SMB 0DA	0	0	X	1100
						End t r ansf er wit h STOP.	0	1	X	—
						End t r ansf er wit h STOP and st ar t anot her t r ansf er.	1	1	X	—
						Send r epeat ed STA	1	0	X	1110
						Swit ch t o Mast er Recei v er Mode (cl ear SI wit hout wr it ing new dat a t o SMB 0DA dat a byt e	0	0	1	1000
Master Receiver	1000	0	0	1	A r ecei v ed; A sent .	Set A Read SMB 0DA	0	0	1	1000
						Set NA byt e as t hel ast dat a byt e; Read SMB 0DA	0	0	0	1000
						I nit iat e r epeat ed STA	1	0	0	1110
						Swit ch t o Mast er Tr ansmit t er Mode (wr it e t o SMB 0DA cl ear ing SI).	0	0	X	1100
		0	0	0	A r ecei v ed; NA byt e.	Read SMB 0DA	0	1	0	—
						Read SMB 0DA f ol l owed by STA	1	1	0	1110
						I nit iat e r epeat ed STA	1	0	0	1110
						Swit ch t o Mast er Tr ansmit t er Mode (wr it e t o SMB 0DA cl ear ing SI).	0	0	X	1100

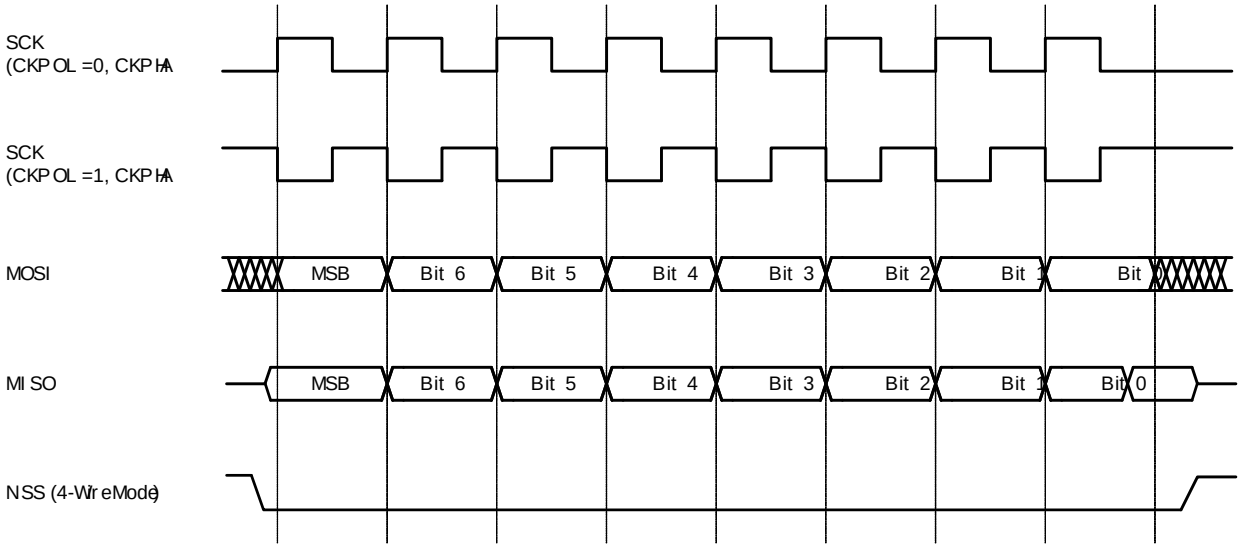


Figure 24.7. Slave Mode Data/Clock Timing (CKPHA = 1)

24.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 24.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD[1:0]		TXBMT	SPIEN
Type	R/W	R/W	R/W	R/W	R/W		R	R/W
Reset	0	0	0	0	0	1	1	0

SFR Page = 0x0; SFR Address = 0xF8

Bit	Name	Function
7	SPIF	**SPI0 Interrupt Flag.** This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
6	WCOL	**Write Collision Flag.** This bit is set to logic 1 if a write to SPI0DAT occurs while SPI0DAT is not ready for a new byte to be transmitted. When this occurs, the write to SPI0DAT is ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
5	MODF	**Mode Fault Flag.** This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
4	RXOVRN	**Receive Overrun Flag (valid in slave mode only).** This bit is set to logic 1 by hardware when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by software.
3:2	NSSMD[1:0]	**Slave Select Mode.** Selects between the following NSS operation modes: (See Section 24.2 and Section 24.3). 00: 3-Wire Slave or 3-Wire Master Mode. NSS signal is not routed to a port pin. 01: 4-Wire Slave or Multi-Master Mode (Default). NSS is an input to the device. 1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.
1	TXBMT	**Transmit Buffer Empty.** This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.
0	SPIEN	**SPI0 Enable.** 0: SPI disabled. 1: SPI enabled.

SFR Definition 25.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR A

Bit	Name	Function
7:0	TMR2L[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value

SFR Definition 25.12. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR A

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value