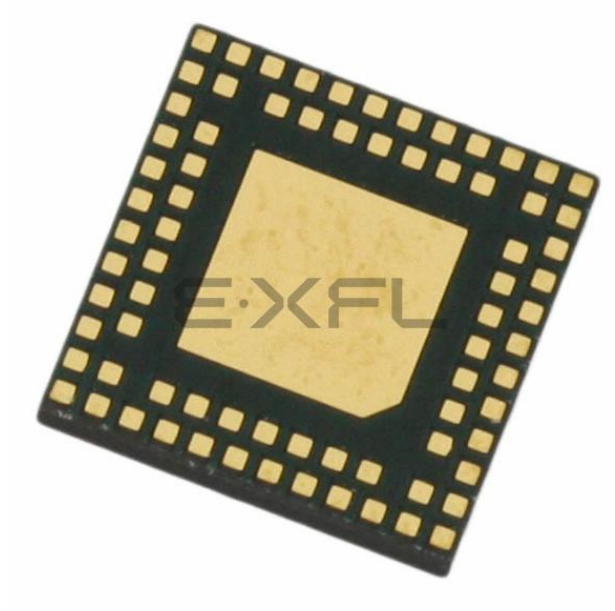

Welcome to E-XFL.COM

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	Active
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	57
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.8V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	76-VFQFN Dual Rows, Exposed Pad
Supplier Device Package	76-DQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f966-b-gm

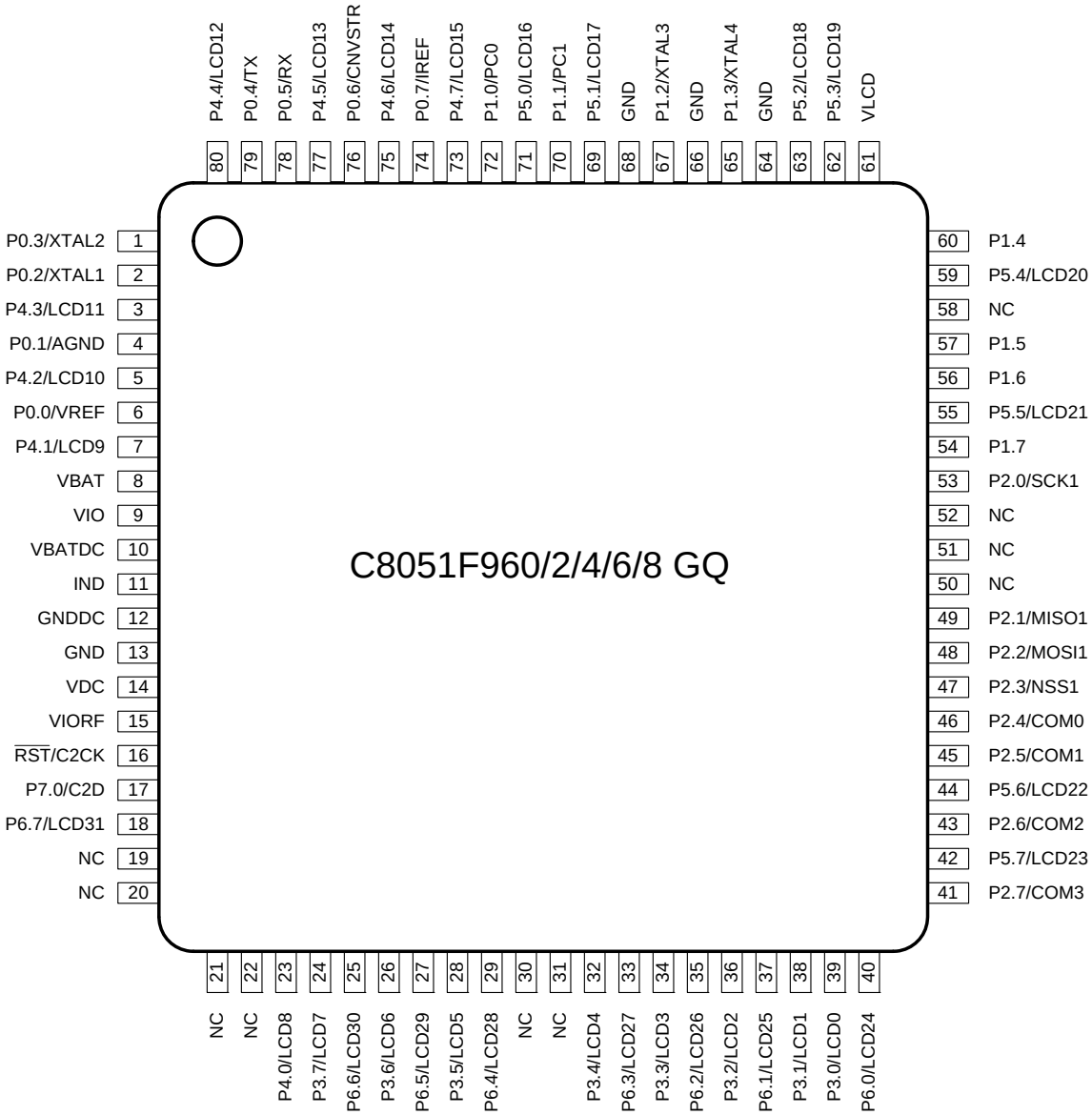

Figure 3.3. TQFP-80 Pinout Diagram (Top View)

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Condition	Min	Typ	Max	Unit
Digital Supply Current— Low Power Idle Mode, All peripheral clocks enabled (PCLKEN = 0x0F) (CPU Inactive, not fetching instructions from flash)					
I_{BAT}[2, 6]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	1.5	1.9	mA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	1.07	—	mA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	270 280	— —	µA µA
	V_{BAT} = 1.8–3.8 V, F = 32.768 kHz (includes SmaRTClock oscillator current)	—	232[5]	—	µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	47[5]	—	µA/MHz
Digital Supply Current— Low Power Idle Mode, All Peripheral Clocks Disabled (PCLKEN = 0x00) (CPU Inactive, not fetching instructions from flash)					
I_{BAT}[2, 7]	V_{BAT} = 1.8–3.8 V, F = 24.5 MHz (includes precision oscillator current)	—	487	—	µA
	V_{BAT} = 1.8–3.8 V, F = 20 MHz (includes low power oscillator current)	—	340	—	µA
	V_{BAT} = 1.8 V, F = 1 MHz V_{BAT} = 3.8 V, F = 1 MHz (includes external oscillator/GPIO current)	— —	90 94	— —	µA µA
I_{BAT} Frequency Sensitivity[3]	V_{BAT} = 1.8–3.8 V, T = 25 °C	—	11[5]	—	µA/MHz
Digital Supply Current—Suspend Mode					
Digital Supply Current (Suspend Mode)	V_{BAT} = 1.8 V V_{BAT} = 3.8 V	— —	77 84	— —	µA

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

10.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 010

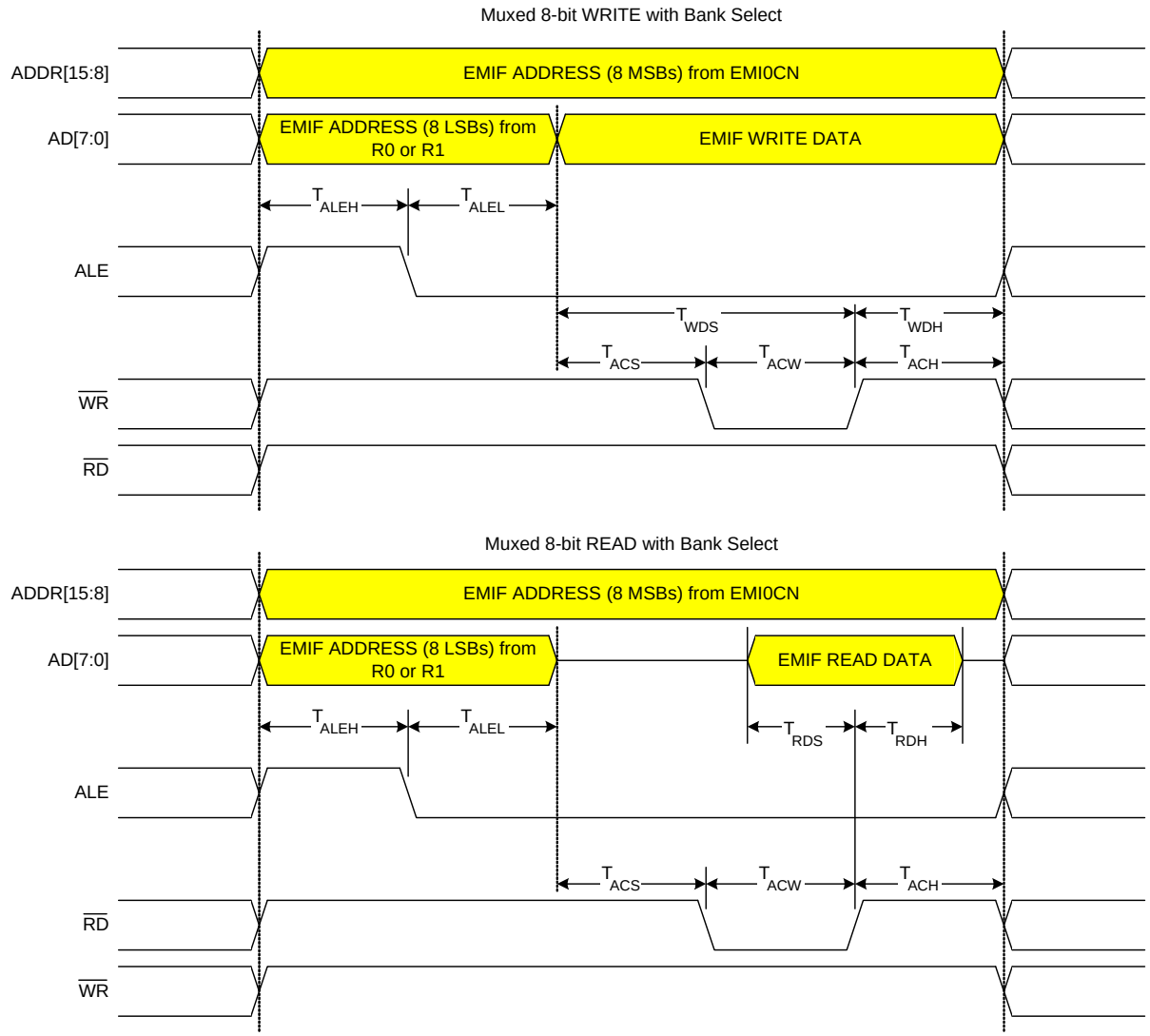

Figure 10.9. Multiplexed 8-bit MOVX with Bank Select Timing

SFR Definition 11.8. DMA0NBAH: Memory Base Address High Byte

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

SFR Page = 0x2; SFR Address = 0xCB

Bit	Name	Function
7:4	Unused	Read = 0b, Write = Don't Care
3:0	NBAH[3:0]	**Memory Base Address High Byte.** Sets high byte of the memory base address which is the DMA0 XRAM starting address of the selected channel if the channel's address offset DMA0NAO is reset to 0.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

SFR Definition 11.9. DMA0NBAL: Memory Base Address Low Byte

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

SFR Page = 0x2; SFR Address = 0xCA

Bit	Name	Function
7:0	NBAL[7:0]	**Memory Base Address Low Byte.** Sets low byte of the memory base address which is the DMA0 XRAM starting address of the selected channel if the channel's address offset DMA0NAO is reset to 0.
Note: This sfr is a DMA channel indirect register. Select the desired channel first using the DMA0SEL sfr.		

14.2.1. Key Inversion using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use the code examples. The steps are listed here for completeness.

Steps to generate the Decryption Key from Encryption Key

- Prepare encryption key and dummy data in xram.
- Reset AES module by clearing bit 3 of AES0BCFG.
- Disable the first three DMA channels by clearing bits 0 to 2 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr.
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr.
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the encryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr.
  - Clear DMA0NSZH
  - Clear DMA0NAOH and DMA0NAOL.
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of dummy data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write 0x10 (16) to the DMA0NSZL sfr.
  - Clear DMA0NSZH
  - Clear DMA0NAOH and DMA0NAOL
- Configure the third DMA channel for the AES0YOUT sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr.
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address for the decryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr.
  - Clear DMA0NSZH.
  - Clear DMA0NAOH and DMA0NAOL.
- Clear first three DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first three DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for inverse key generation by writing 0x04 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG.
- Configure the AES core for encryption by setting the bit 2 of AES0BCFG.
- Initiate the encryption operation by setting bit 3 of AES0BCFG.
- Wait on the DMA interrupt from DMA channel 2.
- Disable the AES Module by clearing bit 2 of AES0BCFG.
- Disable the DMA by writing 0x00 to DMA0EN.

Table 16.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page
ENC0H	0xC4	0x2	ENC0 High	215
ENC0L	0xC2	0x2	ENC0 Low	215
ENC0M	0xC3	0x2	ENC0 Middle	215
FLKEY	0xB7	All Pages	Flash Lock And Key	254
FLSCL	0xB6	0xF	Flash Scale Register	255
FLWR	0xE5	0x0	Flash Write Only	255
FRBCN	0xB5	0xF	Flash Read Buffer Control	256
IE	0xA8	All Pages	Interrupt Enable	236
IP	0xB8	All Pages	Interrupt Priority	237
IREF0CF	0xB9	0xF	Current Reference IREF0 Configuration	104
IREF0CN	0xB9	0x0	Current Reference IREF0 Configuration	103
IT01CF	0xE4	0x0 and 0xF	INT0/INT1 Configuration	243
LCD0BLINK	0x9E	0x2	LCD0 Blink Mask	346
LCD0BUFCF	0xAC	0xF	LCD0 Buffer Configuration	350
LCD0BUFCN	0x9C	0xF	LCD0 Buffer Control	349
LCD0BUFMD	0xB6	0x2	LCD0 Buffer Mode	350
LCD0CF	0xA5	0x2	LCD0 Configuration	348
LCD0CHPCF	0xAD	0x2	LCD0 Charge Pump Configuration	349
LCD0CHPCN	0xB5	0x2	LCD0 Charge Pump Control	348
LCD0CHPMD	0xAE	0x2	LCD0 Charge Pump Mode	349
LCD0CLKDIVH	0xAA	0x2	LCD0 Clock Divider High	345
LCD0CLKDIVL	0xA9	0x2	LCD0 Clock Divider Low	345
LCD0CN	0x9D	0x2	LCD0 Control	337
LCD0CNTRST	0x9C	0x2	LCD0 Contrast	341
LCD0D0	0x89	0x2	LCD0 Data 0	335
LCD0D1	0x8A	0x2	LCD0 Data 1	335
LCD0D2	0x8B	0x2	LCD0 Data 2	335
LCD0D3	0x8C	0x2	LCD0 Data 3	335
LCD0D4	0x8D	0x2	LCD0 Data 4	335
LCD0D5	0x8E	0x2	LCD0 Data 5	335
LCD0D6	0x91	0x2	LCD0 Data 6	335
LCD0D7	0x92	0x2	LCD0 Data 7	335
LCD0D8	0x93	0x2	LCD0 Data 8	335

Table 16.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page
PMU0MD	0xB3	0x0	Power Management Unit Mode	267
PSBANK	0x84	All Pages	Flash Page Switch Bank SFR	127
PSCTL	0x8F	All Pages	Program Store R/W Control	253
PSW	0xD0	All Pages	Program Status Word	123
REF0CN	0xD1	0x0	Voltage Reference Control	102
REG0CN	0xC9	0x0	Voltage Regulator (REG0) Control	277
REVID	0xEA	0xF	Revision ID	249
RSTSRC	0xEF	0x0	Reset Source Configuration/Status	285
RTC0ADR	0xAC	0x0	RTC0 Address	298
RTC0DAT	0xAD	0x0	RTC0 Data	299
RTC0KEY	0xAE	0x0	RTC0 Key	298
SBUF0	0x99	0x0	UART0 Data Buffer	408
SCON0	0x98	All Pages	UART0 Control	407
SFRLAST	0x86	All Pages	SFR Page Stack Last	221
SFRNEXT	0x85	All Pages	SFR Page Stack Next	220
SFRPAGE	0xA7	All Pages	SFR Page	219
SFRPGCN	0x8E	0xF	SFR Page Control	218
SMB0ADM	0xF5	0x0	SMBus Slave Address Mask	392
SMB0ADR	0xF4	0x0	SMBus Slave Address	391
SMB0CF	0xC1	0x0	SMBus0 Configuration	387
SMB0CN	0xC0	All Pages	SMBus0 Control	389
SMB0DAT	0xC2	0x0	SMBus0 Data	393
SPI0CFG	0xA1	0x0	SPI0 Configuration	418
SPI0CKR	0xA2	0x0	SPI0 Clock Rate Control	420
SPI0CN	0xF8	0x0	SPI0 Control	419
SPI0DAT	0xA3	0x0	SPI0 Data	420
SPI1CFG	0xA1	0x2	SPI1 Configuration	438
SPI1CKR	0xA2	0x2	SPI1 Clock Rate Control	440
SPI1CN	0xF8	0x2	SPI1 Control	439
SPI1DAT	0xA3	0x2	SPI1 Data	440
SP	0x81	All Pages	Stack Pointer	122
TCON	0x88	All Pages	Timer/Counter Control	450
TH0	0x8C	0x0	Timer/Counter 0 High	453
TH1	0x8D	0x0	Timer/Counter 1 High	453

18. 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, a single byte at a time, 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 4.8 for complete flash memory electrical characteristics.

18.1. 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 “34. C2 Interface” on page 486.

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. For detailed guidelines on programming flash from firmware, please see Section “18.5. Flash Write and Erase Guidelines” on page 250.

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.

18.1.1. 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 18.4.

18.1.2. Flash Erase Procedure

The flash memory is organized in 1024-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 1024-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. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank.
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a data byte to any location within the 1024-byte page to be erased.
8. Clear the PSWE and PSEE bits.

SFR Definition 18.5. FLSCL: Flash Scale

Bit	7	6	5	4	3	2	1	0
Name		BYPASS						
Type	R	R/W	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xB6

Bit	Name	Function
7	Reserved	Always Write to 0.
6	BYPASS	**Flash Read Timing One-Shot Bypass.** 0: The one-shot determines the flash read time. 1: The system clock determines the flash read time. Leaving the one-shot enabled will provide the lowest power consumption up to 25 MHz.
5:0	Reserved	Always Write to 000000.
Note: Operations which clear the BYPASS bit do not need to be immediately followed by a benign 3-byte instruction. For code compatibility with C8051F930/31/20/21 devices, a benign 3-byte instruction whose third byte is a don't care should follow the clear operation. See the C8051F93x-C8051F92x data sheet for more details.		

SFR Definition 18.6. FLWR: Flash Write Only

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

SFR Page = 0x0; SFR Address = 0xE5

Bit	Name	Function
7:0	FLWR[7:0]	**Flash Write Only.** All writes to this register have no effect on system operation.

SFR Definition 25.2. PC0PCF: PC0 Mode Pull-Up Configuration

Bit	7	6	5	4	3	2	1	0
Name	PUCAL	CALRES	CALPORT	RES[2:0]			DUTY[1:0]	
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	1	0	0

SFR Address = 0xD7; SFR Page = 0x2

Bit	Name	Function
7	PUCAL	**Pull-Up Driver Calibration** 0: Calibration complete or not running. 1: Start calibration of pull up (Self clearing). Calibration determines the lowest usable pull-up strength.
6	CALRES	**Calibration Result** 0: Fail (switch may be closed preventing detection of pull ups). Writes value of 0x11111 to PC0PCF[4:0] 1: Pass (writes calibrated value into PC0PCF[4:0]).
5	CALPORT	**Calibration Port** 0: Calibration on PC0 only. 1: Calibration on PC1 only.
4:2	RES[2:0]	**Pull-Up Resistor Select** Current with force pull-up on bit set (PC0TH.2=1) and VBAT=3.6V. 000: Pull-up disabled. 001: 1 µA.* 010: 4 µA.* 011: 16 µA.* 100: 64 µA.* 101: 256 µA.* 110: 1 mA.* 111: 4 mA.* *The effective average pull-up current depends on selected resistor, pull-up resistor duty-cycle multiplier, and sample rate duty-cycle multiplier.
1:0	DUTY[1:0]	**Pull-Up Resistor Duty Cycle Multiplier** 000: 1/4 (25%)* 001: 3/8 (37.5%)* 010: 1/2 (50%)* 011: 3/4 (75%)* *The final pull-up resistor duty cycle is the sample rate duty-cycle multiplier times the pull-up duty-cycle multiplier.

SFR Definition 25.19. PC0HIST: PC0 History

Bit	7	6	5	4	3	2	1	0
Name	PC0HIST[7:0]							
Type	R							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4; SFR Page = 0x2

Bit	Name	Function
7:0	PC0HIST[7:0]	**PC0 History.** Contains the last 8 recorded directions (1: clock-wise, 0: counter clock-wise) on the previous 8 counts. Values of 0x55 or 0xAA may indicate flutter during quadrature mode.

26.3. LCD Contrast Adjustment

The LCD Bias voltages which determine the LCD contrast are generated using the VBAT supply voltage or the on-chip charge pump. There are four contrast control modes to accomodate a wide variety of applications and supply voltages. The target contrast voltage is programmable in 60 mV steps from 1.9 to 3.72 V. The LCD contrast voltage is controlled by the LCD0CNTRST register and the contrast control mode is selected by setting the appropriate bits in the LCD0MSCN, LCD0MSCF, LCD0PWR, and LCD0VBMCN registers.

Note: An external 10 µF decoupling capacitor is required on the VLCD pin to create a charge reservoir at the output of the charge pump.

Table 26.1. Bit Configurations to select Contrast Control Modes

Mode	LCD0MSCN.2	LCD0MSCF.0	LCD0PWR.3	LCD0VBMCN.7
1	0	1	0	0
2	0	1	1	1
3	1*	0	1	1
4	1*	0	0	1
*** May be set to 0 to support increased load currents.**				

26.3.1. Contrast Control Mode 1 (Bypass Mode)

In Contrast Control Mode 1, the contrast control circuitry is disabled and the VLCD voltage follows the VBAT supply voltage, as shown in Figure 26.3. This mode is useful in systems where the VBAT voltage always remains constant and will provide the lowest LCD power consumption. Bypass Mode is selected using the following procedure:

1. Clear Bit 2 of the LCD0MSCN register to 0b (LCD0MSCN &= ~0x04)
2. Set Bit 0 of the LCD0MSCF register to 1b (LCD0MSCF |= 0x01)
3. Clear Bit 3 of the LCD0PWR register to 0b (LCD0PWR &= ~0x08)
4. Clear Bit 7 of the LCD0VBMCN register to 0b (LCD0VBMCN &= ~0x80)

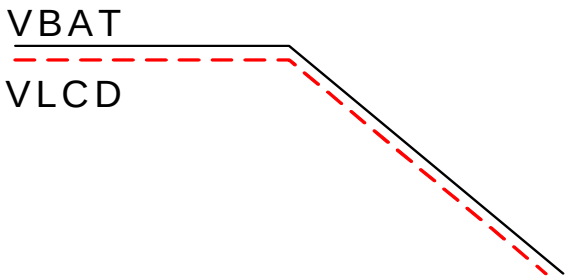

Figure 26.3. Contrast Control Mode 1

26.7. Advanced LCD Optimizations

The special function registers described in this section should be left at their reset value for most systems. Some systems with specific low power or large load requirments will benefit from tweaking the values in these registers to achieve minimum power consumption or maximum drive level.

SFR Definition 26.12. LCD0CF: LCD0 Configuration

Bit	7	6	5	4	3	2	1	0
Name			CMPBYP					
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 Page = 0x2; SFR Address = 0xA5

Bit	Name	Function
7:6	Reserved	Read = 00b. Must write 00b.
5	CMPBYP	**VLCD/VIO Supply Comparator Disable.** Setting this bit to '1' disables the supply voltage comparator which determines if the VIO supply is lower than VLCD. This comparator should only be disabled, as a power saving measure, if VIO is internally or externally shorted to VBAT.
4:0	Reserved	Read = 00b. Must write 00000b.

SFR Definition 26.13. LCD0CHPCN: LCD0 Charge Pump Control

Bit	7	6	5	4	3	2	1	0
Name								
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	0	1	1

SFR Page = 0x2; SFR Address = 0xB5

Bit	Name	Function
7:0	Reserved	Must write 0x4B.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

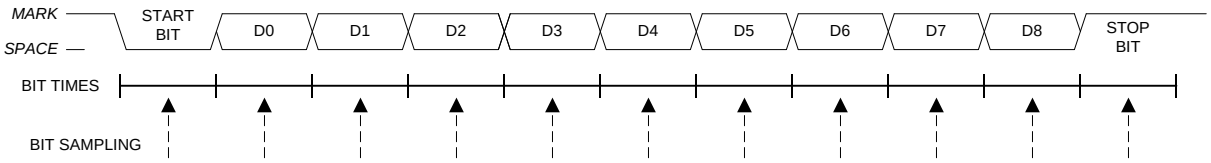

Figure 29.5. 9-Bit UART Timing Diagram

29.3. Multiprocessor Communications

9-Bit UART mode supports 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 MCE0 bit (SCON0.5) 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 1 (RB80 = 1) signifying an address byte has been received. In the UART 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 MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 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 MCE0 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).

SFR Definition 31.1. SPI1CFG: SPI1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Page = 0x0; SFR Address = 0x84

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI1 Clock Phase.** 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	**SPI1 Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** This bit is set to logic 1 whenever the NSS pin is low indicating SPI1 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	**NSS Instantaneous Pin Input.** This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	**Shift Register Empty (valid in slave mode only).** This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
0	RXBMT	**Receive Buffer Empty (valid in slave mode only).** This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.

Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 31.1 for timing parameters.

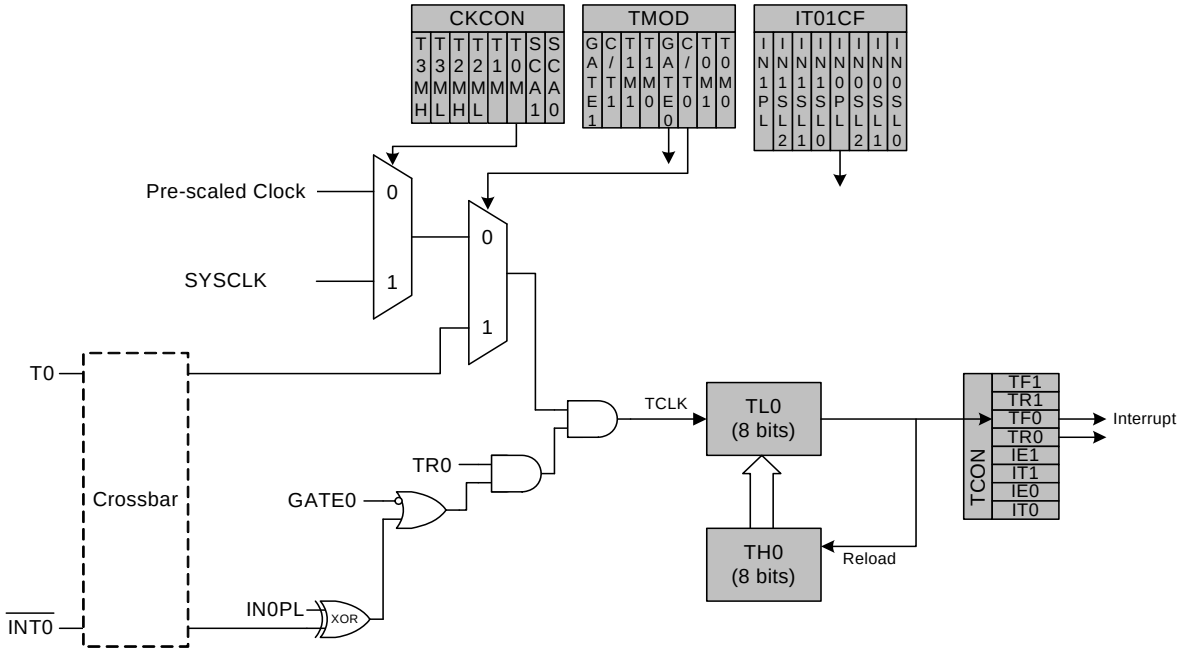

Figure 32.2. T0 Mode 2 Block Diagram

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

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. TL0 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 or prescaled 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. When Timer 0 is operating 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 to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

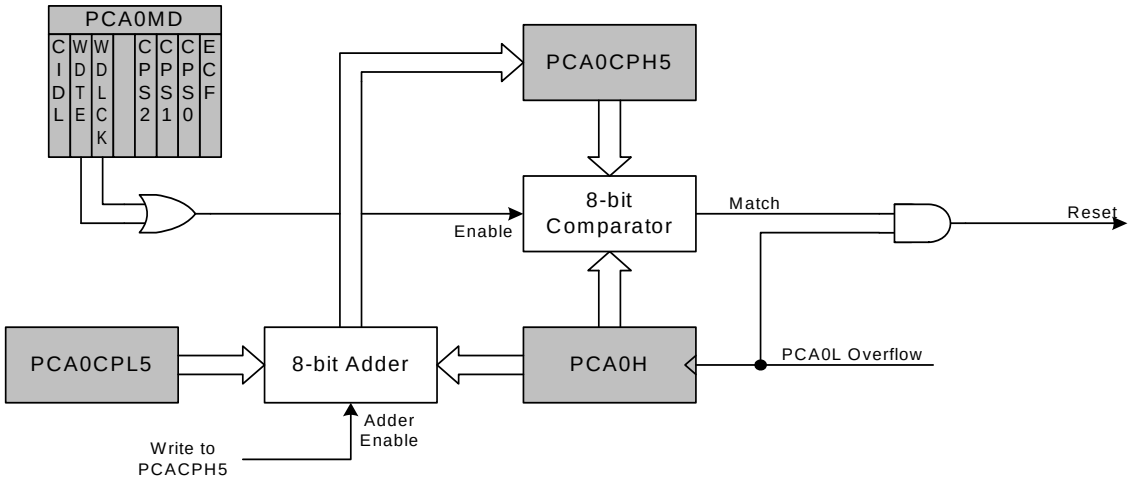

Figure 33.11. PCA Module 5 with Watchdog Timer Enabled

Note that the 8-bit offset held in PCA0CPH5 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 33.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL5) + (256 - PCA0L)$$

Equation 33.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH5 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF5 flag (PCA0CN.5) while the WDT is enabled.

33.4.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a 0 to the WDTE bit.
- Select the desired PCA clock source (with the CPS2–CPS0 bits).
- Load PCA0CPL5 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to 1.
- Reset the WDT timer by writing to PCA0CPH5.

The PCA clock source and idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL5 defaults to 0x00. Using Equation 33.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 33.3 lists some example timeout intervals for typical system clocks.