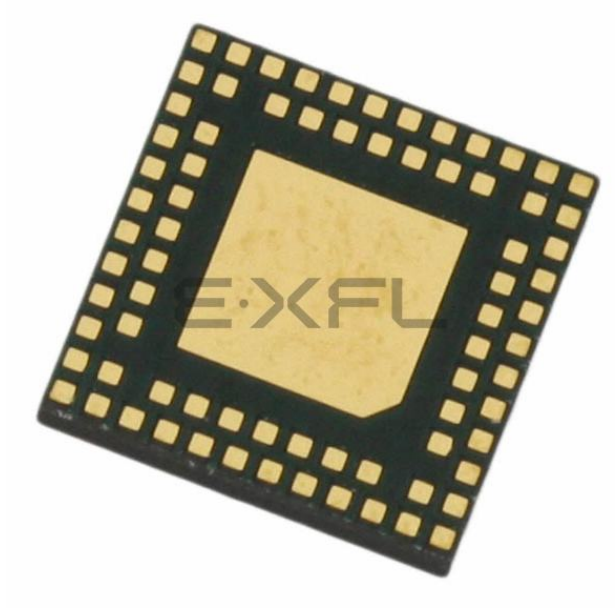


Welcome to E-XFL.COM

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, POR, PWM, WDT
Number of I/O	57
Program Memory Size	128KB (128K 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/c8051f962-b-gm

Table 4.16. Comparator Electrical Characteristics

V_{BAT} = 1.8 to 3.8 V, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	120	—	ns
	CP0+ – CP0– = –100 mV	—	110	—	ns
Response Time: Mode 1, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	180	—	ns
	CP0+ – CP0– = –100 mV	—	220	—	ns
Response Time: Mode 2, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	350	—	ns
	CP0+ – CP0– = –100 mV	—	600	—	ns
Response Time: Mode 3, V_{BAT} = 2.4 V, V_{CM}* = 1.2 V	CP0+ – CP0– = 100 mV	—	1240	—	ns
	CP0+ – CP0– = –100 mV	—	3200	—	ns
Common-Mode Rejection Ratio		—	1.5	—	mV/V
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{BAT} + 0.25	V
Input Capacitance		—	12	—	pF
Input Bias Current		—	1	—	nA
Input Offset Voltage		–10	—	+10	mV
Power Supply					
Power Supply Rejection		—	0.1	—	mV/V
Power-up Time	V_{BAT} = 3.8 V	—	0.6	—	µs
	V_{BAT} = 3.0 V	—	1.0	—	µs
	V_{BAT} = 2.4 V	—	1.8	—	µs
	V_{BAT} = 1.8 V	—	10	—	µs
Supply Current at DC	Mode 0	—	23	—	µA
	Mode 1	—	8.8	—	µA
	Mode 2	—	2.6	—	µA
	Mode 3	—	0.4	—	µA

***Note:** Vcm is the common-mode voltage on CP0+ and CP0–.

7. Comparators

C8051F96x devices include two on-chip programmable voltage comparators: Comparator 0 (CPT0) is shown in Figure 7.1; Comparator 1 (CPT1) is shown in Figure 7.2. The two comparators operate identically, but may differ in their ability to be used as reset or wake-up sources. See the Reset Sources chapter and the Power Management chapter for details on reset sources and low power mode wake-up sources, respectively.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output when the device is in some low power modes.

7.1. Comparator Inputs

Each Comparator performs an analog comparison of the voltage levels at its positive (CP0+ or CP1+) and negative (CP0- or CP1-) input. Both comparators support multiple port pin inputs multiplexed to their positive and negative comparator inputs using analog input multiplexers. The analog input multiplexers are completely under software control and configured using SFR registers. See Section “7.6. Comparator0 and Comparator1 Analog Multiplexers” on page 112 for details on how to select and configure Comparator inputs.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs and skipped by the Crossbar. See the Port I/O chapter for more details on how to configure Port I/O pins as Analog Inputs. The Comparator may also be used to compare the logic level of digital signals, however, Port I/O pins configured as digital inputs must be driven to a valid logic state (HIGH or LOW) to avoid increased power consumption.

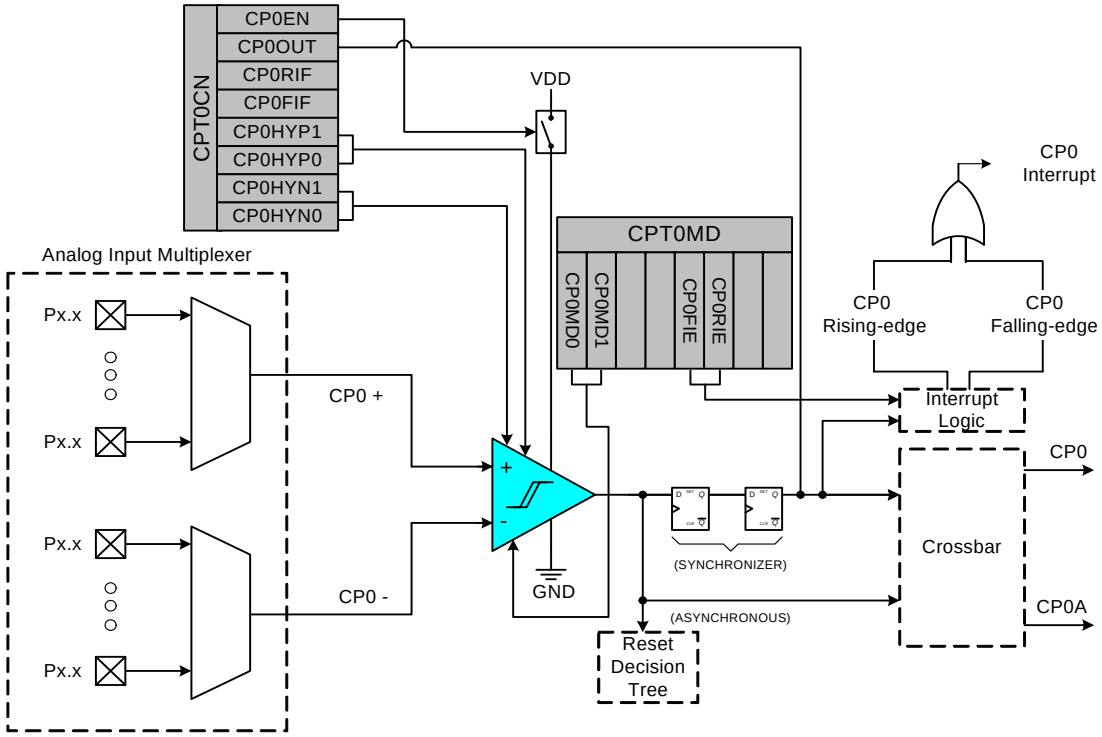


Figure 7.1. Comparator 0 Functional Block Diagram

SFR Definition 7.5. CPT0MX: Comparator0 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX0N[3:0]				CMX0P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0x9F

Bit	Name	Function
7:4	CMX0N	**Comparator0 Negative Input Selection.** Selects the negative input channel for Comparator0. 0000: P0.1 0001: P0.3 0010: P0.5 0011: Reserved 0100: Reserved 0101: Reserved 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 1010: Reserved 1011: Reserved 1100: Compare 1101: VBAT divided by 2 1110: Digital Supply Voltage 1111: Ground
3:0	CMX0P	**Comparator0 Positive Input Selection.** Selects the positive input channel for Comparator0. 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: Reserved 0101: Reserved 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 1010: Reserved 1011: Reserved 1100: Compare 1101: VBAT divided by 2 1110: VBAT Supply Voltage 1111: VBAT Supply Voltage

SFR Definition 8.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	PARITY
Type	R/W	R/W	R/W	R/W		R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xD0; Bit-Addressable

Bit	Name	Function
7	CY	**Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	**Auxiliary Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under software control.
4:3	RS[1:0]	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: ■ An ADD, ADDC, or SUBB instruction causes a sign-change overflow. ■ A MUL instruction results in an overflow (result is greater than 255). ■ A DIV instruction causes a divide-by-zero condition. The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under software control.
0	PARITY	**Parity Flag.** This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

10.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 8 kB boundaries. As an example, the addresses 0x2000 and 0x4000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

10.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, 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. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.5.3. 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.

10.5.4. 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.

13.3. CRC Seed Value

Normally, the initial value or the CRC results is cleared to 0x0000. However, a CRC might be specified with an initial value preset to all ones (0xFFFF).

The steps to preset the CRC with all ones is as follows:

1. Set the SEED bit to 1.
2. Reset the CRC1 module by setting the CLR bit to 1 in CRC1CN.
3. Clear the SEED bit to 0.

The CRC1 module is now ready to calculate a CRC using a CRC seed value of 0xFFFF.

13.4. Inverting the Final Value

Sometimes it is necessary to invert the final value. This will take the ones complement of the final result.

The steps to flip the final CRC results are as follows:

1. Clear the CRC module by setting the CLR bit in CRC1CN SFR.
2. Write the polynomial to CRC1POLH:L.
3. Write all data bytes to CRC1IN.
4. Set the INV bit in the CRC1CN SFR to invert the final results.
5. Read the final CRC results from CRC1OUTH:L.

Clear the FLIP bit in the CRC1CN SFR.

13.5. Flipping the Final Value

The steps to flip the final CRC results are as follows:

1. Clear the CRC module by setting the CLR bit in CRC1CN SFR.
2. Write the polynomial to CRC1POLH:L.
3. Write all data bytes to CRC1IN.
4. Set the FLIP bit in the CRC1CN SFR to flip the final results.
5. Read the final CRC results from CRC1OUTH:L.
6. Clear the FLIP bit in the CRC1CN SFR.

The flip operation will exchange bit 15 with bit 0, bit 14 with bit 1, bit 13 with bit 2, and so on.

14.6.5. Counter Mode

The Counter (CTR) Mode uses a sequential counter which is incremented after each block. This turns the block cipher into a stream cipher. This algorithm is shown inFigure 14.4. Note that the decryption operation actually uses the encryption key and encryption block cipher. The XOR operation is always on the output of the Cipher. The counter is a 16-byte block. Often the several bytes of the counter are initialized to a nonce (number used once). The last byte of the counter is incremented and propagated. Thus, the counter is treated as a 16-byte big endian integer.

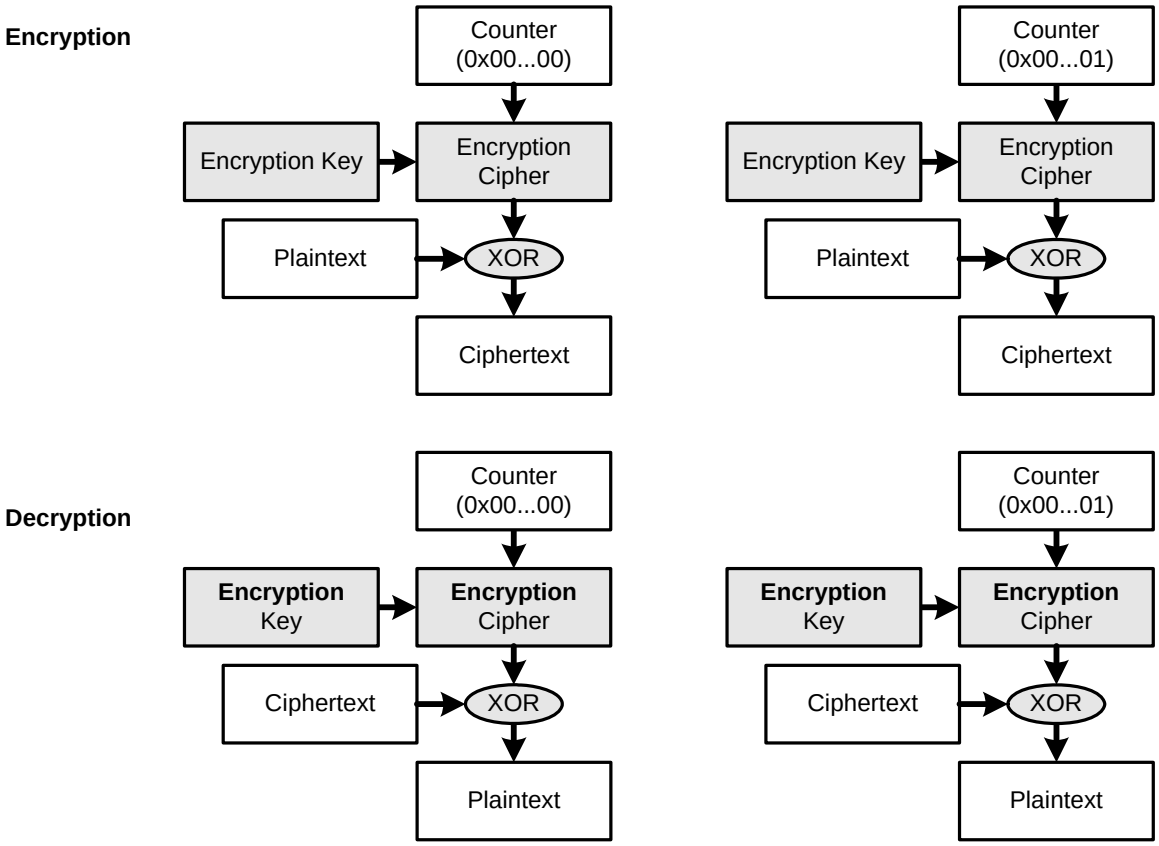


Figure 14.7. Counter Mode

15. Encoder/Decoder

The Encoder/Decoder consists of three 8-bit data registers, a control register and an encoder/decoder logic block.

The size of the input data depends on the mode. The input data for Manchester encoding is one byte. For Manchester decoding it is two bytes. Three-out-of-Six encoding is two bytes. Three-out-of six decoding is three bytes.

The output size also depends on the mode selected. The input and output data size are shown below:

Table 15.1. Encoder Input and Output Data Sizes

	Input Data Size	Output Data Size
Operation	**Bytes**	**Bytes**
Manchester Encode	1	2
Manchester Decode	2	1
Three out of Six Encode	2	3
Three out of Six Decode	3	2

The input and output data is always right justified. So for Manchester mode the input uses only ENC0L and the output data is only in ENC0M and ENC0L. ENC0H is not used for Manchester mode

SFR Definition 15.1. ENC0CN: Encoder Decoder 0 Control

Bit	7	6	5	4	3	2	1	0
Name	READY	ERROR	ENC	DEC		DMA	ENDIAN	MODE
Type	R	R	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC5; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
7	READY	**Ready Flag.**
6	ERROR	**Error Flag.**
5	ENC	**Encode.** Setting this bit will initiate an Encode operation.
4	DEC	**Decode.** Setting this bit will initiate a Decode operation.
2	DMA	**DMA Mode Enable.** This bit should be set when using the encoder/decoder with the DMA.
1	ENDIAN	**Big-Endian DMA Mode Select.** This bit should be set when using the DMA with big-endian multiple byte DMA transfers. The DMA must also be configured for the same endian mode.
0	MODE	**Mode.** 0: Select Manchester encoding or decoding. 1:Select Three-out-of-Six encoding or decoding.

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
PC0INT0	0xFB	0x2	PC0 Interrupt 0	332
PC0INT1	0xFC	0x2	PC0 Interrupt 1	333
PC0MD	0xD9	0x2	PC0 Mode	321
PC0PCF	0xD7	0x2	PC0 Pull-up Configuration	322
PC0STAT	0xC1	0x2	PC0 Status	324
PC0TH	0xE4	0x2	PC0 Threshold	323
PCA0CN	0xD8	All Pages	PCA0 Control	480
PCA0CPH0	0xFC	0x0	PCA0 Capture 0 High	485
PCA0CPH1	0xEA	0x0	PCA0 Capture 1 High	485
PCA0CPH2	0xEC	0x0	PCA0 Capture 2 High	485
PCA0CPH3	0xEE	0x0	PCA0 Capture 3 High	485
PCA0CPH4	0xFE	0x0	PCA0 Capture 4 High	485
PCA0CPH5	0xD3	0x0	PCA0 Capture 5 High	485
PCA0CPL0	0xFB	0x0	PCA0 Capture 0 Low	485
PCA0CPL1	0xE9	0x0	PCA0 Capture 1 Low	485
PCA0CPL2	0xEB	0x0	PCA0 Capture 2 Low	485
PCA0CPL3	0xED	0x0	PCA0 Capture 3 Low	485
PCA0CPL4	0xFD	0x0	PCA0 Capture 4 Low	485
PCA0CPL5	0xD2	0x0	PCA0 Capture 5 Low	485
PCA0CPM0	0xDA	0x0	PCA0 Module 0 Mode Register	483
PCA0CPM1	0xDB	0x0	PCA0 Module 1 Mode Register	483
PCA0CPM2	0xDC	0x0	PCA0 Module 2 Mode Register	483
PCA0CPM3	0xDD	0x0	PCA0 Module 3 Mode Register	483
PCA0CPM4	0xDE	0x0	PCA0 Module 4 Mode Register	483
PCA0CPM5	0xCE	0x0	PCA0 Module 5 Mode Register	483
PCA0H		0x0	PCA0 Counter High	484
PCA0L	0xF9	0x0	PCA0 Counter Low	484
PCA0MD	0xD9	0x0	PCA0 Mode	481
PCA0PWM	0xDF	0x0	PCA0 PWM Configuration	482
PCLKACT	0xF5	0xF	Peripheral Clock Enable Active Mode	260
PCLKEN	0xFE	0xF	Peripheral Clock Enables (LP Idle)	261
PCON	0x87	All Pages	Power Control	268
PMU0CF	0xB5	0x0	PMU0 Configuration 0	265
PMU0FL	0xB6	0x0	PMU0 flag	266

18.4. Determining the Device Part Number at Run Time

In many applications, user software may need to determine the MCU part number at run time in order to determine the hardware capabilities. The part number can be determined by reading the value of the DEVICEID Special Function Register.

The value of the DEVICEID register can be decoded as follows:

0xD0—C8051F960
0xD1—C8051F961
0xD2—C8051F962
0xD3—C8051F963
0xD4—C8051F964
0xD5—C8051F965
0xD6—C8051F966
0xD7—C8051F967
0xD8—C8051F968

SFR Definition 18.1. DEVICEID: Device Identification

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

SFR Page = 0xF; SFR Address = 0xE9

Bit	Name	Function
7:0	DEVICEID[7:0]	**Device Identification.** These bits contain a value that can be decoded to determine the device part number.

SFR Definition 18.2. REVID: Revision Identification

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

SFR Page = 0xF; SFR Address = 0xEA

Bit	Name	Function
7:0	REVID[7:0]	**Revision Identification.** These bits contain a value that can be decoded to determine the silicon revision. 0x01 = Revision A. 0x02 = Revision B.

20.1. Startup Behavior

The dc-dc converter is enabled by setting bit DC0EN (DC0MD.0) to logic 1. When first enabled, the M1 switch turns on and continues to supply current into the output capacitor through the inductor until the VDC output voltage reaches the programmed level set by by the VSEL bits (DC0CF.[6:3]).

The peak transient current in the inductor is limited for safe operation. The peak inductor current is programmable using the ILIMIT bits (DC0MD.[6:4]). The peak inductor current, size of the output capacitor and the amount of dc load current present during startup will determine the length of time it takes to charge the output capacitor. The RDYH and RDYL bits (DC0RDY.7 and DC0DRY.6) may be used to determine when the output voltage is within approximately 100 mV of the programmed voltage.

In order to ensure reliable startup of the dc-dc converter, the following restrictions have been imposed:

- The maximum dc load current allowed during startup is given in Table 4.20 on page 77. If the dc-dc converter is powering external sensors or devices through the VDC pin, then the current supplied to these sensors or devices is counted towards this limit. The in-rush current into capacitors does not count towards this limit.

- The maximum total output capacitance is given in Table 4.20 on page 77. This value includes the required 2.2 µF ceramic output capacitor and any additional capacitance connected to the VDC pin.

The peak inductor current limit is programmable by software as shown in Table 20.1. Limiting the peak inductor current can allow the dc-dc converter to start up using a high impedance power source (such as when a battery is near its end of life) or allow inductors with a low current rating to be utilized. By default, the peak inductor current is set to 500 mA.

.

Table 20.1. IPeak Inductor Current Limit Settings

ILIMIT	Peak Current (mA)
001	200
010	300
011	400
100	500
101	600

The peak inductor current is dependent on several factors including the dc load current and can be estimated using following equation:

$$I_{PK} = \frac{2 \times I_{LOAD} \times (VDC - VBATDC)}{efficiency \times inductance \times frequency}$$

efficiency = 0.80

inductance = 0.68 µH

frequency = 2.4 MHz

Important Notes:

- The Power-on Reset (POR) delay is not incurred after a supply monitor reset. See Section “4. Electrical Characteristics” on page 56 for complete electrical characteristics of the active mode supply monitors.
- Software should take care not to inadvertently disable the supply monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to 1 to keep the supply monitor enabled as a reset source.
- The supply monitor must be enabled before selecting it as a reset source**.** Selecting the supply monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the supply monitor and selecting it as a reset source. See Section “4. Electrical Characteristics” on page 56 for minimum supply monitor turn-on time. **No delay should be introduced in systems where software contains routines that erase or write Flash memory.** The procedure for enabling the V_{DD} supply monitor and selecting it as a reset source is shown below:
  1. Enable the Supply Monitor (VDMEN bit in VDM0CN = 1).
  2. Wait for the Supply Monitor to stabilize (optional).
  3. Select the Supply Monitor as a reset source (PORSF bit in RSTSRC = 1).

ness. As shown in Figure 24.2, duty cycles less than 65% indicate a robust oscillation. As the duty cycle approaches 68%, oscillation becomes less reliable and the risk of clock failure increases. Increasing the bias current (by disabling AGC) will always improve oscillation robustness and will reduce the output clock's duty cycle. This test should be performed at the worst case system conditions, as results at very low temperatures or high supply voltage will vary from results taken at room temperature or low supply voltage.

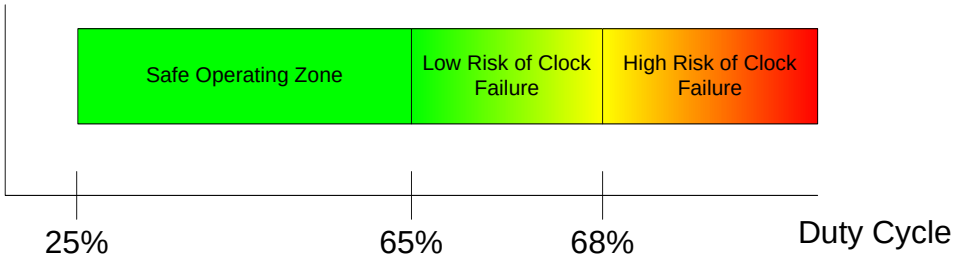


Figure 24.2. Interpreting Oscillation Robustness (Duty Cycle) Test Results

As an alternative to performing the oscillation robustness test, Automatic Gain Control may be disabled at the cost of increased power consumption (approximately 200 nA). Disabling Automatic Gain Control will provide the crystal oscillator with higher immunity against external factors which may lead to clock failure. Automatic Gain Control must be disabled if using the SmaRTClock oscillator in self-oscillate mode.

Table 24.3 shows a summary of the oscillator bias settings. The SmaRTClock Bias Doubling feature allows the self-oscillation frequency to be increased (almost doubled) and allows a higher crystal drive strength in crystal mode. High crystal drive strength is recommended when the crystal is exposed to poor environmental conditions such as excessive moisture. SmaRTClock Bias Doubling is enabled by setting BIASX2 (RTC0XCN.5) to 1.

.

Table 24.3. SmaRTClock Bias Settings

Mode	Setting	Power Consumption
Crystal	Bias Double Off, AGC On	Lowest
	Bias Double Off, AGC Off	Low
	Bias Double On, AGC On	High
	Bias Double On, AGC Off	Highest
Self-Oscillate	Bias Double Off	Low
	Bias Double On	High

SFR Definition 26.3. LCD0CNTRST: LCD0 Contrast Adjustment

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

SFR Page = 0x2; SFR Address = 0x9C

Bit	Name	Function
7:5	Reserved	Read = 000. Write = Must write 000.
4:0	CNTRST	**Contrast Setpoint.** Determines the setpoint for the VLCD voltage necessary to achieve the desired contrast. 00000: 1.90 00001: 1.96 00010: 2.02 00011: 2.08 00100: 2.13 00101: 2.19 00110: 2.25 00111: 2.31 01000: 2.37 01001: 2.43 01010: 2.49 01011: 2.55 01100: 2.60 01101: 2.66 01110: 2.72 01111: 2.78 10000: 2.84 10001: 2.90 10010: 2.96 10011: 3.02 10100: 3.07 10101: 3.13 10110: 3.19 10111: 3.25 11000: 3.31 11001: 3.37 11010: 3.43 11011: 3.49 11100: 3.54 11101: 3.60 11110: 3.66 11111: 3.72

SFR Definition 30.3. SPI0CKR: SPI0 Clock Rate

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

SFR Page = 0x0; SFR Address = 0xA2

Bit	Name	Function
7:0	SCR[7:0]	**SPI0 Clock Rate.** These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR[7:0] + 1)}$ for 0 <= SPI0CKR <= 255 Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 30.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xA3

Bit	Name	Function
7:0	SPI0DAT[7:0]	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

31.6. Using SPI1 with the DMA

SPI1 is a DMA-enabled peripheral that can provide autonomous data transfers when used with the DMA. The DMA-enabled SPI1 supports both master and slave mode. The SPI requires two DMA channels for a bidirectional data transfer and also supports unidirectional data transfers using a single DMA channel.

There are no additional control bits in the SPI1 control and configuration SFRs. The configuration is the same in DMA and non-DMA mode. While the SPIF flag and/or SPI interrupts are normally used for non-DMA SPI transfers, a DMA transfer is managed using the DMA enable and DMA full transfer complete flags.

More information on using the SPI1 peripheral can be found in the detailed example code for SPI1 Master and Slave modes.

31.7. Master Mode SPI1 DMA Transfers

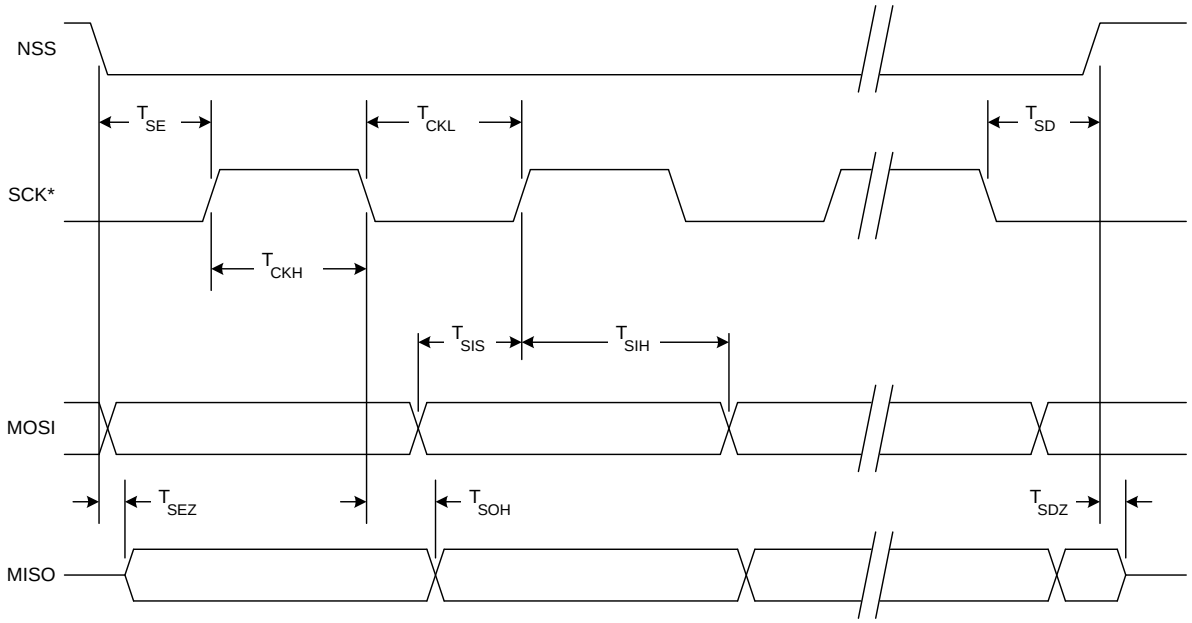
The SPI interface does not normally have any handshaking or flow control. Therefore, the Master will transmit all of the output data without waiting on the slave peripheral. The system designer must ensure that the slave peripheral can accept all of the data at the transfer rate.

31.8. Master Mode Bidirectional Data Transfer

A bidirectional SPI Master Mode DMA transfer will transmit a specified number of bytes out on the MOSI pin and receive the same number of bytes on the MISO pin. The MOSI data must be stored in XRAM before initiating the DMA transfers. The DMA will also transfer all the MISO data to XRAM, overwriting any data at the target location.

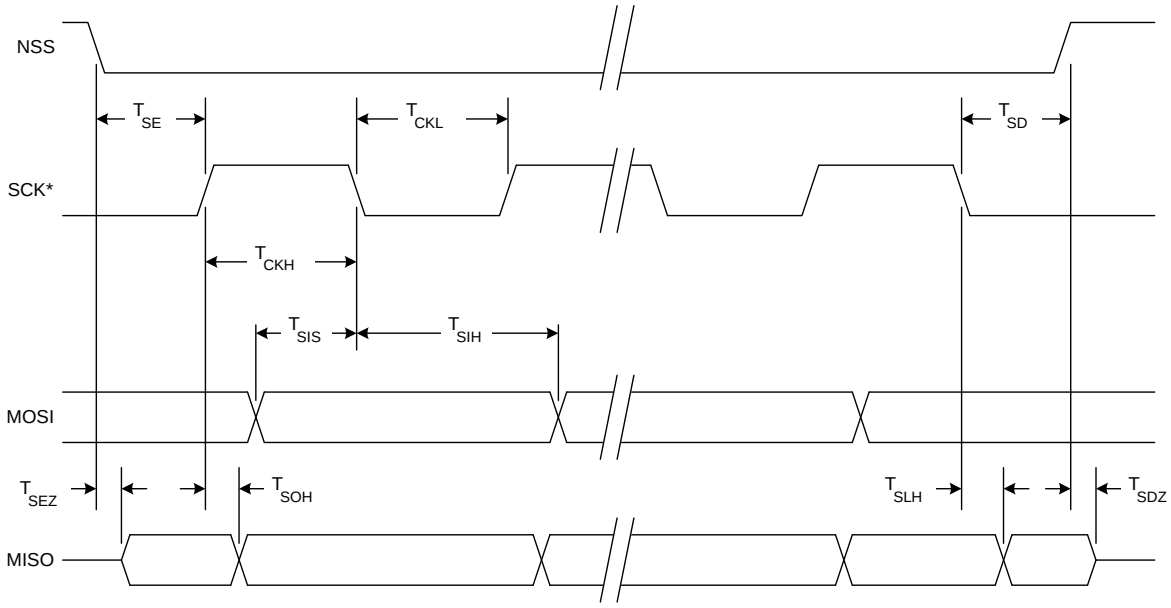
A bidirectional transfer requires two DMA channels. The first DMA channel transfers data from XRAM to the SPI1DAT SFR and the second DMA channel transfers data from the SPI1DAT SFR to XRAM. The second channel DMA interrupt indicates SPI transfer completion.

In master mode, the NSS pin is an output and the hardware does not manage the NSS pin automatically. Normally, firmware should assert the NSS pin before the SPI transfer and deassert it upon completion of the transfer. When using 4-wire Master mode, bit 2 of SPI1CN controls the state of the NSS pin. When using 3-wire master mode, firmware may use any GPIO pin as NSS.



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 31.11. SPI Slave Timing (CKPHA = 1)