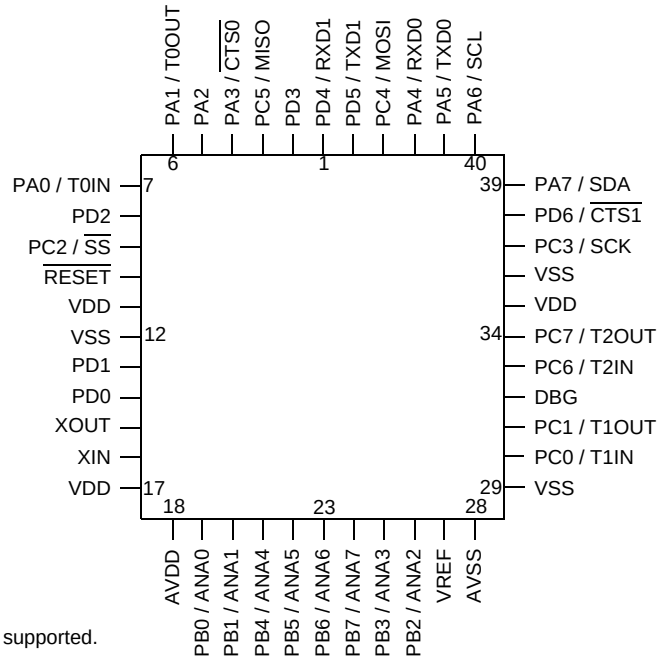
Welcome to E-XFL.COM

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	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6402vs020sc



Note: Timer 3 is not supported.

Figure 57. Z8Fxx01 in 44-Pin Plastic Leaded Chip Carrier (PLCC)

Program Memory

The eZ8 CPU supports 64KB of Program Memory address space. The Z8F640x family devices contain 16KB to 64KB of on-chip Flash memory in the Program Memory address space. Reading from Program Memory addresses outside the available Flash memory addresses returns FFH. Writing to these unemployments Program Memory addresses produces no effect. Table 4 describes the Program Memory Maps for the Z8F640x family products.

Table 4. Z8F640x Family Program Memory Maps

Program Memory Address (Hex)	Function
Z8F160x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFFH	Program Memory
Z8F240x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFFH	Program Memory
Z8F320x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFFH	Program Memory

*** See Table 22 on page 45 for a list of the interrupt vectors.**

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
Timer 3 (not available in 40- and 44- Pin Packages)				
F18	Timer 3 High Byte	T3H	00	66
F19	Timer 3 Low Byte	T3L	01	66
F1A	Timer 3 Reload High Byte	T3RH	FF	67
F1B	Timer 3 Reload Low Byte	T3RL	FF	67
F1C	Timer 3 PWM High Byte	T3PWMH	00	69
F1D	Timer 3 PWM Low Byte	T3PWML	00	69
F1E	Reserved	—	XX	
F1F	Timer 3 Control	T3CTL	00	70
F20-F3F	Reserved	—	XX	
UART 0				
F40	UART0 Transmit Data	U0TXD	XX	86
	UART0 Receive Data	U0RXD	XX	87
F41	UART0 Status 0	U0STAT0	0000011Xb	87
F42	UART0 Control 0	U0CTL0	00	89
F43	UART0 Control 1	U0CTL1	00	89
F44	UART0 Status 1	U0STAT1	00	87
F45	Reserved	—	XX	
F46	UART0 Baud Rate High Byte	U0BRH	FF	91
F47	UART0 Baud Rate Low Byte	U0BRL	FF	91
UART 1				
F48	UART1 Transmit Data	U1TXD	XX	86
	UART1 Receive Data	U1RXD	XX	87
F49	UART1 Status 0	U1STAT0	0000011Xb	87
F4A	UART1 Control 0	U1CTL0	00	89
F4B	UART1 Control 1	U1CTL1	00	89
F4C	UART1 Status 1	U1STAT1	00	87
F4D	Reserved	—	XX	
F4E	UART1 Baud Rate High Byte	U1BRH	FF	91
F4F	UART1 Baud Rate Low Byte	U1BRL	FF	91
I^2C				
F50	I^2C Data	I2CDATA	00	118
F51	I^2C Status	I2CSTAT	80	118
F52	I^2C Control	I2CCTL	00	119
F53	I^2C Baud Rate High Byte	I2CBRH	FF	121
F54	I^2C Baud Rate Low Byte	I2CBRL	FF	121
F55-F5F	Reserved	—	XX	
Serial Peripheral Interface (SPI)				
F60	SPI Data	SPIDATA	XX	106
F61	SPI Control	SPICTL	00	107

XX=Undefined

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
FCE	Interrupt Port Select	IRQPS	00	55
FCF	Interrupt Control	IRQCTL	00	56
GPIO Port A				
FD0	Port A Address	PAADDR	00	37
FD1	Port A Control	PACTL	00	38
FD2	Port A Input Data	PAIN	XX	42
FD3	Port A Output Data	PAOUT	00	43
GPIO Port B				
FD4	Port B Address	PBADDR	00	37
FD5	Port B Control	PBCTL	00	38
FD6	Port B Input Data	PBIN	XX	42
FD7	Port B Output Data	PBOUT	00	43
GPIO Port C				
FD8	Port C Address	PCADDR	00	37
FD9	Port C Control	PCCTL	00	38
FDA	Port C Input Data	PCIN	XX	42
FDB	Port C Output Data	PCOUT	00	43
GPIO Port D				
FDC	Port D Address	PDADDR	00	37
FDD	Port D Control	PDCTL	00	38
FDE	Port D Input Data	PDIN	XX	42
FDF	Port D Output Data	PDOUT	00	43
GPIO Port E				
FE0	Port E Address	PEADDR	00	37
FE1	Port E Control	PECTL	00	38
FE2	Port E Input Data	PEIN	XX	42
FE3	Port E Output Data	PEOUT	00	43
GPIO Port F				
FE4	Port F Address	PFADDR	00	37
FE5	Port F Control	PFCTL	00	38
FE6	Port F Input Data	PFIN	XX	42
FE7	Port F Output Data	PFOUT	00	43
GPIO Port G				
FE8	Port G Address	PGADDR	00	37
FE9	Port G Control	PGCTL	00	38
FEA	Port G Input Data	PGIN	XX	42
FEB	Port G Output Data	PGOUT	00	43
GPIO Port H				
FEC	Port H Address	PHADDR	00	37

XX=Undefined

information on approximate time-out delays for the minimum and maximum WDT reload values.

Table 45. Watch-Dog Timer Approximate Time-Out Delays

WDT Reload Value	WDT Reload Value	Approximate Time-Out Delay (with 50kHz typical WDT oscillator frequency)	
(Hex)	(Decimal)	Typical	Description
000004	4	80µs	Minimum time-out delay
FFFFFF	16,777,215	335.5s	Maximum time-out delay

Watch-Dog Timer Refresh

When first enabled, the Watch-Dog Timer is loaded with the value in the Watch-Dog Timer Reload registers. The Watch-Dog Timer then counts down to 000000H unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the downcounter to be reloaded with the WDT Reload value stored in the Watch-Dog Timer Reload registers. Counting resumes following the reload operation.\

When the Z8F640x family device is operating in Debug Mode (via the On-Chip Debugger), the Watch-Dog Timer is continuously refreshed to prevent spurious Watch-Dog Timer time-outs.

Watch-Dog Timer Time-Out Response

The Watch-Dog Timer times out when the counter reaches 000000H. A time-out of the Watch-Dog Timer generates either an interrupt or a Short Reset. The WDT_RES Option Bit determines the time-out response of the Watch-Dog Timer. Refer to the **Option Bits** chapter for information regarding programming of the WDT_RES Option Bit.

WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the Watch-Dog Timer issues an interrupt request to the interrupt controller and sets the WDT status bit in the Watch-Dog Timer Control register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the Watch-Dog Timer interrupt vector and executing code from the vector address. After time-out and interrupt generation, the Watch-Dog Timer counter rolls over to its maximum value of FFFFFH and continues counting. The Watch-Dog Timer counter is not automatically returned to its Reload Value.

WDT Interrupt in Stop Mode

If configured to generate an interrupt when a time-out occurs and the Z8F640x family device is in STOP mode, the Watch-Dog Timer automatically initiates a STOP Mode Recovery and generates an interrupt request. Both the WDT status bit and the STOP bit in the Watch-Dog Timer Control register are set to 1 following WDT time-out in STOP

Serial Peripheral Interface

Overview

The Serial Peripheral Interface™ (SPI) is a synchronous interface allowing several SPI-type devices to be interconnected. SPI-compatible devices include EEPROMs, Analog-to-Digital Converters, and ISDN devices. Features of the SPI include:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface
- Data transfers rates up to a maximum of one-fourth the system clock frequency
- Error detection
- Write and mode collision detection
- Dedicated Baud Rate Generator

Architecture

The SPI may be configured as either a Master (in single or multi-master systems) or a Slave as illustrated in Figures 74 through 76.

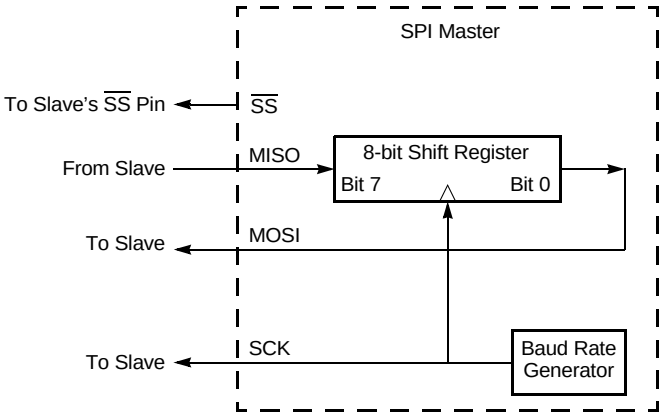


Figure 74. SPI Configured as a Master in a Single Master, Single Slave System

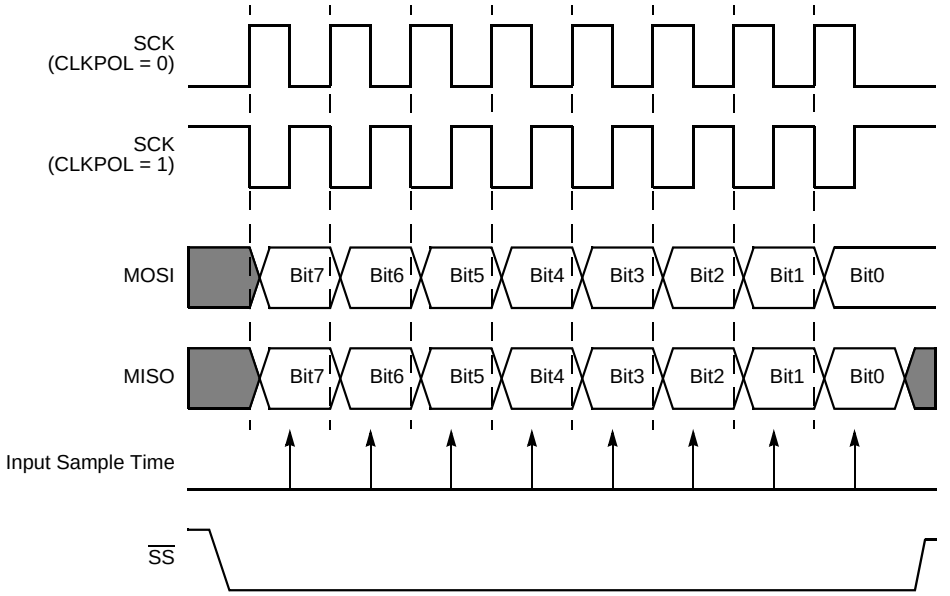


Figure 78. SPI Timing When PHASE is 1

Multi-Master Operation

In a multi-master SPI system, all SCK pins are tied together, all MOSI pins are tied together and all MISO pins are tied together. All SPI pins must then be configured in open-drain mode to prevent bus contention. At any one time, only one SPI device is configured as the Master and all other SPI devices on the bus are configured as Slaves. The Master enables a single Slave by asserting the $\overline{SS}$ pin on that Slave only. Then, the single Master drives data out its SCK and MOSI pins to the SCK and MOSI pins on the Slaves (including those which are not enabled). The enabled Slave drives data out its MISO pin to the MISO Master pin.

For a Master device operating in a multi-master system, if the $\overline{SS}$ pin is configured as an input and is driven Low by another Master, the COL bit is set to 1 in the SPI Status Register. The COL bit indicates the occurrence of a multi-master collision (mode fault error condition).

limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the high period of SCL.

I²C Interrupts

the I²C Controller contains three sources of interrupts—Transmit, Receive and Not Acknowledge (NAK) interrupts. NAK interrupts occur when a Not Acknowledge is received from the slave or sent by the I²C Controller and the Start or Stop bit is set. This source sets bit 0 and can only be cleared by setting the Start or Stop bit. When this interrupt occurs, the I²C Controller waits until it is cleared before performing any action. In an interrupt service routine, this interrupt must be the first thing polled. Receive interrupts occur when a byte of data has been received by the I²C master. This interrupt is cleared by reading from the I²C Data register. If no action is taken, the I²C Controller waits until this interrupt is cleared before performing any other action.

For Transmit interrupts to occur, the TXI bit must be 1 in the I²C Control register. Transmit interrupts occur under the following conditions when the transmit data register is empty:

- The I²C Controller is idle (not performing an operation).
- The START bit is set and there is no valid data in the I²C Shift or I²C Data register to shift out.
- The first bit of the byte of an address is shifting out and the RD bit of the I²C Status register is deasserted.
- The first bit of a 10-bit address shifts out.
- The first bit of write data shifted out.

Note: Writing to the I²C Data register always clears a Transmit interrupt.

Start and Stop Conditions

The master (I²C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I²C Controller generates a START condition by pulling the SDA signal low while SCL is high. Then a high-to-low transition occurs on the SDA signal while the clock is High. To complete a transaction, the I²C Controller generates a Stop condition by creating a low-to-high transition of the SDA signal in the middle of the high period of the SCL signal.When the SCL signal is High, the master generates a Start bit by pulling a High SDA signal Low and generates a Stop bit by releasing the SDA signal. The Start and Stop signals are found in the I²C Control register and must be written by software when the Z8F640x family device must begin or end a transaction.

Writing a Transaction with a 7-Bit Address

1. The I²C Controller shifts the I²C Shift register out onto SDA signal.

2. The I2C Controller waits for the slave to send an Acknowledge (by pulling the SDA signal Low). If the slave pulls the SDA signal High (Not-Acknowledge), the I2C Controller sends a Stop signal.
3. If the slave needs to service an interrupt, it pulls the SCL signal Low, which halts I2C operation.
4. If there is no other data in the I2C Data register or the STOP bit in the I2C Control register is set by software, then the Stop signal is sent.

Figure 79 illustrates the data transfer format for a 7-bit addressed slave. Shaded regions indicate data transferred from the I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

Figure 79. 7-Bit Addressed Slave Data Transfer Format

The data transfer format for a transmit operation on a 7-bit addressed slave is as follows:

1. Software asserts the IEN bit in the I2C Control register.
2. Software asserts the TXI bit of the I2C Control register to enable Transmit interrupts.
3. The I2C interrupt asserts, because the I2C Data register is empty
4. Software responds to the TDRE bit by writing a 7-bit slave address followed by a 0 (write) to the I2C Data register.
5. Software asserts the START bit of the I2C Control register.
6. The I2C Controller sends the START condition to the I2C slave.
7. The I2C Controller loads the I2C Shift register with the contents of the I2C Data register.
8. After one bit of address has been shifted out by the SDA signal, the Transmit interrupt is asserted.
9. Software responds by writing the contents of the data into the I2C Data register.
10. The I2C Controller shifts the rest of the address and write bit out by the SDA signal.
11. The I2C slave sends an acknowledge (by pulling the SDA signal low) during the next high period of SCL. The I2C Controller sets the ACK bit in the I2C Status register.
12. The I2C Controller loads the contents of the I2C Shift register with the contents of the I2C Data register.
13. The I2C Controller shifts the data out of via the SDA signal. After the first bit is sent, the Transmit interrupt is asserted.

1. Software writes the I2C Data register with a 7-bit slave address followed by a 1 (read).
2. Software asserts the START bit of the I2C Control register.
3. Software asserts the NAK bit of the I2C Control register so that after the first byte of data has been read by the I2C Controller, a Not Acknowledge is sent to the I2C slave.
4. The I2C Controller sends the START condition.
5. The I2C Controller sends the address and read bit by the SDA signal.
6. The I2C slave sends an Acknowledge by pulling the SDA signal Low during the next high period of SCL.
7. The I2C Controller reads the first byte of data from the I2C slave.
8. The I2C Controller asserts the Receive interrupt.
9. Software responds by reading the I2C Data register.
10. The I2C Controller sends a NAK to the I2C slave.
11. A NAK interrupt is generated by the I2C Controller.
12. Software responds by setting the STOP bit of the I2C Control register.
13. A STOP condition is sent to the I2C slave.

Reading a Transaction with a 10-Bit Address

Figure 82 illustrates the receive format for a 10-bit addressed slave. The shaded regions indicate data transferred from the I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

S	Slave Address 1st 7 bits	W=0	A	Slave address 2nd Byte	A	S	Slave Address 1st 7 bits	R=1	A	Data	A	Data	$\overline{A}$	P

Figure 82. Receive Data Format for a 10-Bit Addressed Slave

The first seven bits transmitted in the first byte are 11110XX. The two bits XX are the two most-significant bits of the 10-bit address. The lowest bit of the first byte transferred is the write signal.

The data transfer format for a receive operation on a 10-bit addressed slave is as follows:

1. Software writes an address 11110B followed by the two address bits and a 0 (write).
2. Software asserts the START bit of the I2C Control register.
3. The I2C Controller sends the Start condition.

DMA_ADC Control Register

The DMA_ADC Control register enables and sets options (DMA enable and interrupt enable) for ADC operation.

Table 78. DMA_ADC Control Register (DMAACTL)

BITS	7	6	5	4	3	2	1	0
FIELD	DAEN	IRQEN	Reserved		ADC_IN			
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FBEH							

DAEN—DMA_ADC Enable
0 = DMA_ADC is disabled and the ADC Analog Input Number (ADC_IN) is reset to 0.
1 = DMA_ADC is enabled.

IRQEN—Interrupt Enable
0 = DMA_ADC does not generate any interrupts.
1 = DMA_ADC generates an interrupt after transferring data from the last ADC Analog Input specified by the ADC_IN field.

Reserved
These bits are reserved and must be 0.

ADC_IN—ADC Analog Input Number
These bits set the number of ADC Analog Inputs to be used in the continuous update (data conversion followed by DMA data transfer). The conversion always begins with ADC Analog Input 0 and then progresses sequentially through the other selected ADC Analog Inputs.
0000 = ADC Analog Input 0 updated.
0001 = ADC Analog Inputs 0-1 updated.
0010 = ADC Analog Inputs 0-2 updated.
0011 = ADC Analog Inputs 0-3 updated.
0100 = ADC Analog Inputs 0-4 updated.
0101 = ADC Analog Inputs 0-5 updated.
0110 = ADC Analog Inputs 0-6 updated.
0111 = ADC Analog Inputs 0-7 updated.
1000 = ADC Analog Inputs 0-8 updated.
1001 = ADC Analog Inputs 0-9 updated.
1010 = ADC Analog Inputs 0-10 updated.
1011 = ADC Analog Inputs 0-11 updated.
1100-1111 = Reserved.

this bit to 0 when a conversion has been completed.
1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.

Reserved
This bit is reserved and must be 0.

$\overline{\text{VREF}}$
0 = Internal voltage reference generator enabled. The VREF pin should be left unconnected (or capacitively coupled to analog ground).
1 = Internal voltage reference generator disabled. An external voltage reference must be provided through the VREF pin.

CONT
0 = Single-shot conversion. ADC data is output once at completion of the 5129 system clock cycles.
1 = Continuous conversion. ADC data updated every 256 system clock cycles.

ANAIN—Analog Input Select
These bits select the analog input for conversion. Not all Port pins in this list are available in all packages for the Z8F640x family of products. Refer to the **Signal and Pin Descriptions** chapter for information regarding the Port pins available with each package style. Do not enable unavailable analog inputs.
0000 = ANA0
0001 = ANA1
0010 = ANA2
0011 = ANA3
0100 = ANA4
0101 = ANA5
0110 = ANA6
0111 = ANA7
1000 = ANA8
1001 = ANA9
1010 = ANA10
1011 = ANA11
11XX = Reserved.

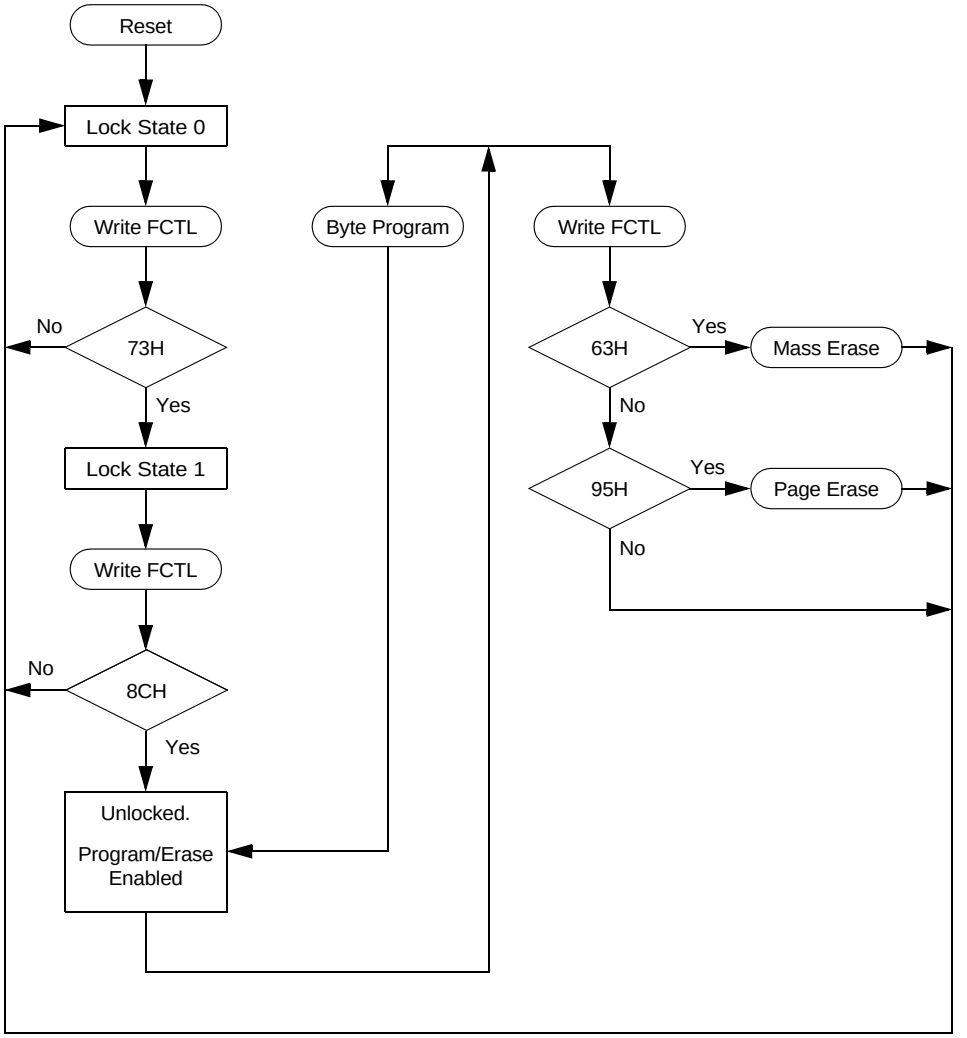


Figure 85. Flash Controller Operation Flow Chart

Table 100. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units	Notes
68-Pin PLCC Maximum Ratings at 70^0C to 105^0C				
Total power dissipation		500	mW	
Maximum current into V_{DD} or out of V_{SS}		140	mA	
64-Pin LQFP Maximum Ratings at -40°C to 70°C				
Total power dissipation		1000	mW	
Maximum current into V_{DD} or out of V_{SS}		275	mA	
64-Pin LQFP Maximum Ratings at 70^0C to 105^0C				
Total power dissipation		540	mW	
Maximum current into V_{DD} or out of V_{SS}		150	mA	
44-Pin PLCC Maximum Ratings at -40°C to 70°C				
Total power dissipation		750	mW	
Maximum current into V_{DD} or out of V_{SS}		200	mA	
44-Pin PLCC Maximum Ratings at 70^0C to 105^0C				
Total power dissipation		295	mW	
Maximum current into V_{DD} or out of V_{SS}		83	mA	
44-pin LQFP Maximum Ratings at -40°C to 70°C				
Total power dissipation		750	mW	
Maximum current into V_{DD} or out of V_{SS}		200	mA	
44-pin LQFP Maximum Ratings at 70^0C to 105^0C				
Total power dissipation		410	mW	
Maximum current into V_{DD} or out of V_{SS}		114	mA	
40-Pin PDIP Maximum Ratings at -40°C to 70°C				
Total power dissipation		1000	mW	
Maximum current into V_{DD} or out of V_{SS}		275	mA	
40-Pin PDIP Maximum Ratings at 70°C to 105°C				
Total power dissipation		540	mW	
Maximum current into V_{DD} or out of V_{SS}		150	mA	

Notes:
1. This voltage applies to all pins except the following: V_{DD}, AV_{DD}, pins supporting analog input (Port B and Port H), $\overline{\text{RESET}}$, and where noted otherwise.

Table 101. DC Characteristics

Symbol	Parameter	T_A = -40°C to 105°C			Units	Conditions
		Minimum	Typical	Maximum		
I_{PU}	Weak Pull-up Current	30	100	350	µA	V_{DD} = 3.0 - 3.6V
I_{CCS}	Supply Current in Stop Mode		600		µA	V_{DD} = 3.3V

[1] This condition excludes all pins that have on-chip pull-ups, when driven Low.

[2] These values are provided for design guidance only and are not tested in production.

Figure 91 illustrates the typical current consumption while operating at 25ºC, 3.3V, versus the system clock frequency.

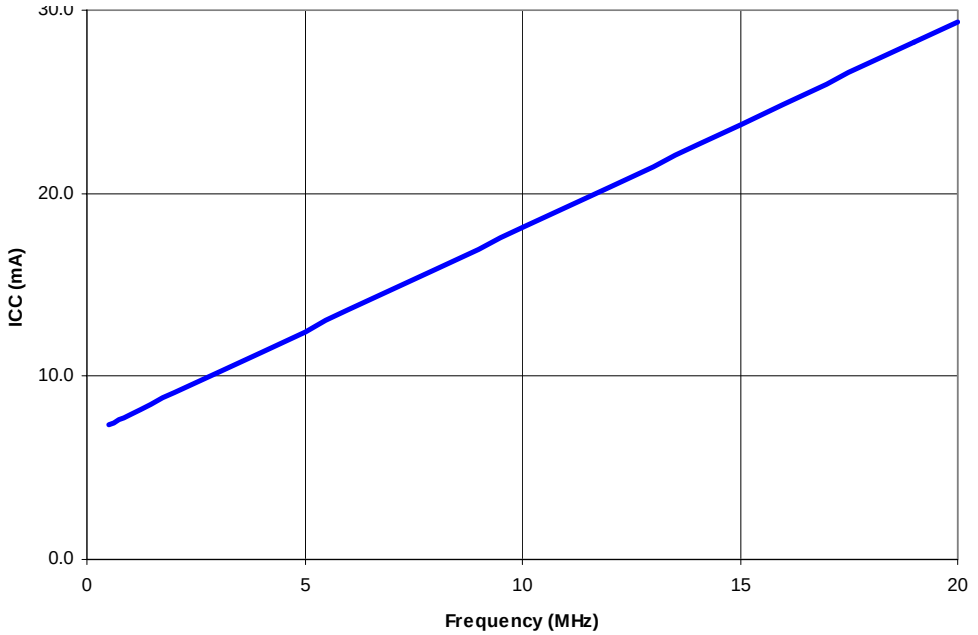


Figure 91. Nominal ICC Versus System Clock Frequency

AC Characteristics

The section provides information on the AC characteristics and timing of the Z8 Encore!™. All AC timing information assumes a standard load of 50pF on all outputs.

Table 102. AC Characteristics

Symbol	Parameter	V_{DD} = 3.0 - 3.6V T_A =-40^{0}C to 105^{0}C		Units	Conditions
		Minimum	Maximum		
F_{sysclk}	System Clock Frequency	–	20.0	MHz	Read-only from Flash memory.
		0.032768	20.0	MHz	Program or erasure of the Flash memory.
F_{XTAL}	Crystal Oscillator Frequency	1.0	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
T_{XIN}	System Clock Period	50	–	ns	$T_{CLK} = 1/F_{sysclk}$
T_{XINH}	System Clock High Time	20	30	ns	T_{CLK} = 50ns
T_{XINL}	System Clock Low Time	20	30	ns	T_{CLK} = 50ns
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50ns
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50ns

On-Chip Debugger Timing

Figure 95 and Table 109 provide timing information for DBG pins. The timing specifications presume a rise and fall time on DBG of less than 4µs.

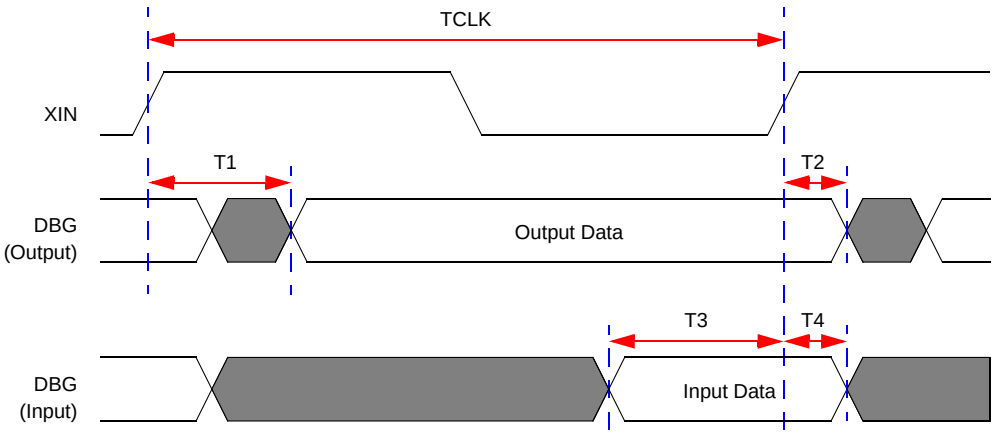


Figure 95. On-Chip Debugger Timing

Table 109. On-Chip Debugger Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
DBG			
T_1	XIN Rise to DBG Valid Delay	–	15
T_2	XIN Rise to DBG Output Hold Time	2	–
T_3	DBG to XIN Rise Input Setup Time	10	–
T_4	DBG to XIN Rise Input Hold Time	5	–
	DBG frequency		System Clock / 4

S