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, POR, PWM, WDT
Number of I/O	19
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0412sj020ec

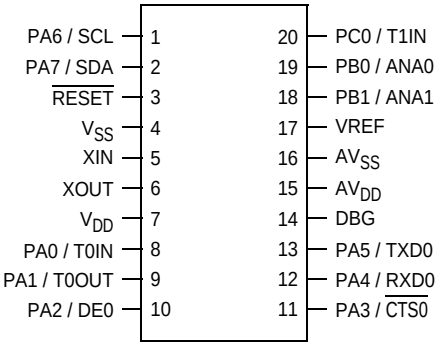

Figure 2. Z8F0821 and Z8F0421 in 20-Pin SSOP and PDIP Packages

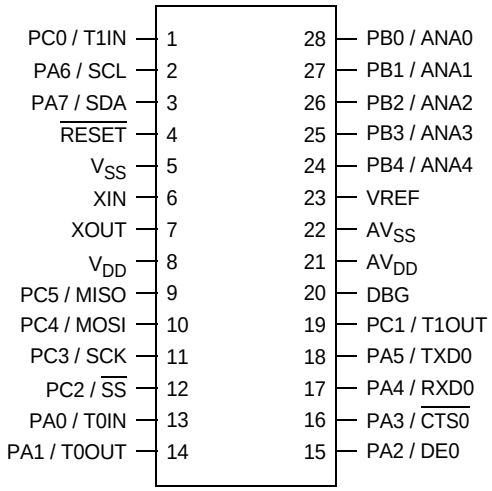

Figure 3. Z8F0822 and Z8F0422 in 28-Pin SOIC and PDIP Packages

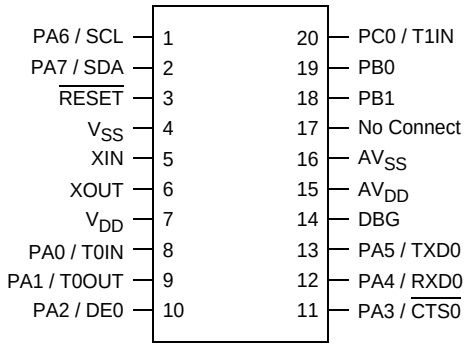

Figure 4. Z8F0811 and Z8F0411 in 20-Pin SSOP and PDIP Packages

Power-On Reset

Each device in the Z8 Encore! XP® F0822 Series contains an internal POR circuit. The POR circuit monitors the supply voltage and holds the device in the Reset state until the supply voltage reaches a safe operating level. After the supply voltage exceeds the POR voltage threshold (V_{POR}), the POR Counter is enabled and counts 66 cycles of the WDT oscillator. After the POR counter times out, the XTAL Counter is enabled to count a total of 16 system clock pulses. The device is held in the Reset state until both the POR Counter and XTAL counter have timed out. After the Z8 Encore! XP F0822 Series device exits the POR state, the eZ8 CPU fetches the Reset vector. Following POR, the POR status bit in the Watchdog Timer Control Register (WDTCTL) is set to 1.

Figure 6 displays POR operation. See Electrical Characteristics for POR threshold voltage (V_{POR}).

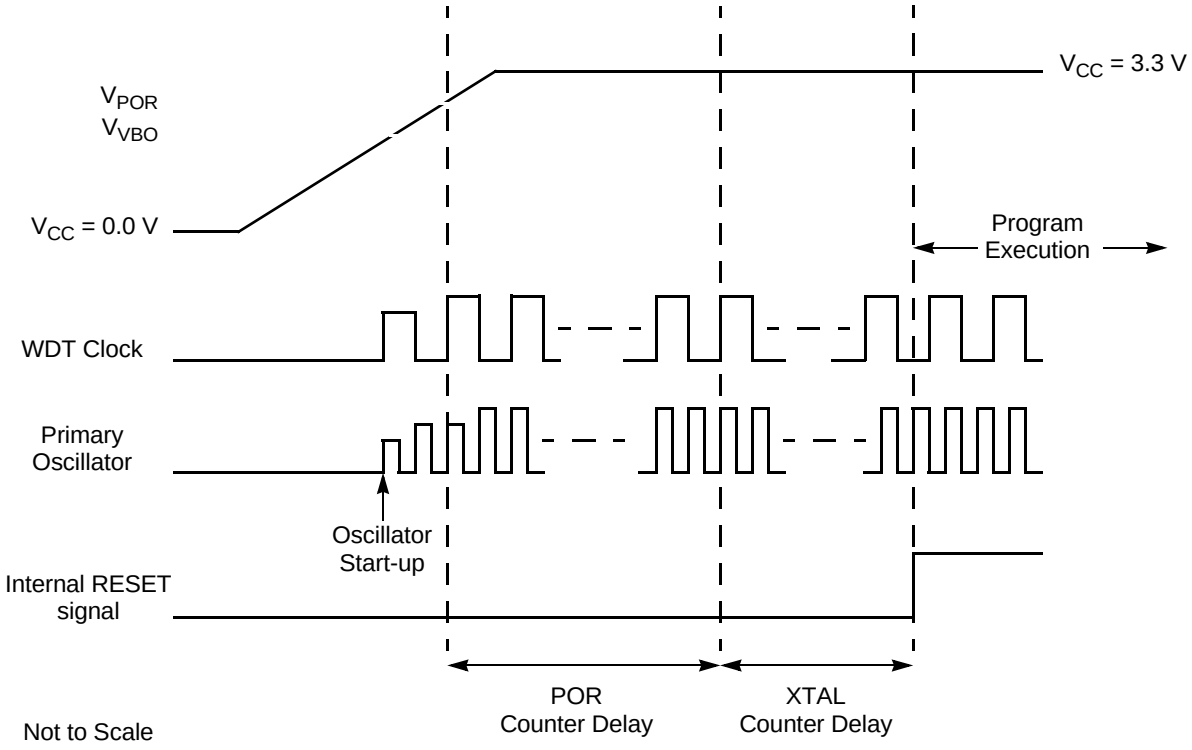

Figure 6. Power-On Reset Operation

Voltage Brownout Reset

The devices in Z8 Encore! XP F0822 Series provide low Voltage Brownout protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage

External Pin Reset

The $\overline{\text{RESET}}$ pin contains a Schmitt-triggered input, an internal pull-up, an analog filter, and a digital filter to reject noise. After the $\overline{\text{RESET}}$ pin is asserted for at least 4 system clock cycles, the device progresses through the System Reset sequence. While the $\overline{\text{RESET}}$ input pin is asserted Low, Z8 Encore! XP F0822 Series device continues to be held in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state immediately following $\overline{\text{RESET}}$ pin deassertion. Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the Watchdog Timer Control Register (WDTCTL) is set to 1.

On-Chip Debugger Initiated Reset

A POR is initiated using the OCD by setting the RST bit in the OCD Control Register. The OCD block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the system reset. Following the system reset, the POR bit in the WDT Control Register is set.

Stop Mode Recovery

STOP mode is entered by execution of a STOP instruction by the eZ8 CPU. For detailed information on STOP mode, see Low-Power Modes on page 45. During Stop Mode Recovery, the device is held in reset for 66 cycles of the WDT oscillator followed by 16 cycles of the system clock. Stop Mode Recovery only affects the contents of the WDT Control Register and does not affect any other values in the Register File, including the Stack Pointer, Register Pointer, Flags, Peripheral Control Registers, and General-Purpose RAM.

The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the STOP bit in the WDT Control Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions. The text following provides more detailed information on each of the Stop Mode Recovery sources.

Table 10. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
STOP mode	WDT time-out when configured for Reset	Stop Mode Recovery
	WDT time-out when configured for interrupt	Stop Mode Recovery followed by interrupt (if interrupts are enabled)
	Data transition on any GPIO Port pin enabled as a Stop Mode Recovery source	Stop Mode Recovery

Poor coding style that resulting in lost interrupt requests:

LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0

Note: *To avoid missing interrupts, the following style of coding to set bits in the Interrupt Request Registers is recommended*

Good coding style that avoids lost interrupt requests:

ORX IRQ0, MASK

Interrupt Control Register Definitions

For all interrupts other than the WDT interrupt, the Interrupt Control Registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register (Table 25) stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU reads the IRQ0 Register to determine if any interrupt requests are pending.

Table 25. Interrupt Request 0 Register (IRQ0)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved	T1I	T0I	U0RXI	U0TXI	I2CI	SPII	ADCI
RESET	0							
R/W	R/W							
ADDR	FC0H							

Reserved—Must be 0

T1I—Timer 1 Interrupt Request
0 = No interrupt request is pending for Timer 1.
1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request
0 = No interrupt request is pending for Timer 0.
1 = An interrupt request from Timer 0 is awaiting service.

4. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function
6. Write to the Timer Control Register to enable the timer and initiate counting

In COMPARE mode, the system clock always provides the timer input. The Compare time is calculated by the following equation:

$$\text{Compare Mode Time (s)} = \frac{(\text{Compare Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

GATED Mode

In GATED mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control Register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte Registers. The timer input is the system clock. When reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte Registers is reset to 0001H and counting resumes (assuming the Timer Input signal is still asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

Follow the steps below for configuring a timer for GATED mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for GATED mode
   - Set the prescale value
2. Write to the Timer High and Low Byte Registers to set the starting count value. This only affects the first pass in GATED mode. After the first timer reset in GATED mode, counting always begins at the reset value of 0001H
3. Write to the Timer Reload High and Low Byte Registers to set the Reload value
4. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers
5. Configure the associated GPIO port pin for the Timer Input alternate function
6. Write to the Timer Control Register to enable the timer
7. Assert the Timer Input signal to initiate the counting

TH and TL—Timer High and Low Bytes
These 2 bytes, {TMRH[7:0], TMRL[7:0]}, contain the current 16-bit timer count value.

Timer Reload High and Low Byte Registers

The Timer 0–1 Reload High and Low Byte (TxRH and TxRL) Registers (Table 41) store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte register are stored in a temporary holding register. When a write to the Timer Reload Low Byte Register occurs, the temporary holding register value is written to the Timer High Byte Register. This operation allows simultaneous updates of the 16-bit Timer Reload value.

In COMPARE mode, the Timer Reload High and Low Byte Registers store the 16-bit Compare value.

Table 41. Timer 0–1 Reload High Byte Register (TxRH)

BITS	7	6	5	4	3	2	1	0
FIELD	TRH							
RESET	1							
R/W	R/W							
ADDR	F02H, F0AH							

Table 42. Timer 0–1 Reload Low Byte Register (TxRL)

BITS	7	6	5	4	3	2	1	0
FIELD	TRL							
RESET	1							
R/W	R/W							
ADDR	F03H, F0BH							

TRH and TRL—Timer Reload Register High and Low
These two bytes form the 16-bit Reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to 0001H. In COMPARE mode, these two bytes form the 16-bit Compare value.

Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers (Table 43 and Table 44) are used for Pulse-Width Modulator (PWM) operations. These registers also store the Capture values for the CAPTURE and CAPTURE/COMPARE modes.

COMPARE Mode
When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

GATED Mode
0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input.
1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.

CAPTURE/COMPARE Mode
0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal.
1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.

PRES—Prescale value
The timer input clock is divided by 2^{PRES}, where PRES is set from 0 to 7. The prescaler is reset each time the Timer is disabled. This insures proper clock division each time the Timer is restarted.
000 = Divide by 1
001 = Divide by 2
010 = Divide by 4
011 = Divide by 8
100 = Divide by 16
101 = Divide by 32
110 = Divide by 64
111 = Divide by 128

TMODE—Timer Mode
000 = ONE-SHOT mode
001 = CONTINUOUS mode
010 = COUNTER mode
011 = PWM mode
100 = CAPTURE mode
101 = COMPARE mode
110 = GATED mode
111 = CAPTURE/COMPARE mode

All three Watchdog Timer Reload Registers must be written in this order. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes occur unless the sequence is restarted. The value in the Watchdog Timer Reload Registers is loaded into the counter when the WDT is first enabled and every time a WDT instruction is executed.

Watchdog Timer Control Register Definitions

Watchdog Timer Control Register

The Watchdog Timer Control Register (WDTCTL), detailed in Table 48, is a Read-Only Register that indicates the source of the most recent Reset event, a Stop Mode Recovery event, and a WDT time-out. Reading this register resets the upper four bits to 0.

Writing the 55H, AAH unlock sequence to the Watchdog Timer Control Register (WDTCTL) address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL address produce no effect on the bits in the WDTCTL. The locking mechanism prevents spurious writes to the Reload registers.

Table 48. Watchdog Timer Control Register (WDTCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	POR	STOP	WDT	EXT	Reserved			
RESET	See descriptions below			0				
R/W	R							
ADDR	FF0H							

Reset or Stop Mode Recovery Event	POR	STOP	WDT	EXT
Power-On Reset	1	0	0	0
Reset through $\overline{\text{RESET}}$ pin assertion	0	0	0	1
Reset through WDT time-out	0	0	1	0
Reset through the OCD (OCTCTL[1] set to 1)	1	0	0	0
Reset from STOP Mode through the DBG Pin driven Low	1	0	0	0
Stop Mode Recovery through GPIO pin transition	0	1	0	0
Stop Mode Recovery through WDT time-out	0	1	1	0

POR—Power-On Reset Indicator
If this bit is set to 1, a POR event occurred. This bit is reset to 0, if a WDT time-out or Stop Mode Recovery occurs. This bit is also reset to 0, when the register is read.

STOP—Stop Mode Recovery Indicator
If this bit is set to 1, a Stop Mode Recovery occurred. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurred due to a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the Stop Mode Recovery was not caused by a WDT time-out. This bit is reset by a POR or a WDT time-out that occurred while not in STOP mode. Reading this register also resets this bit.

WDT—Watchdog Timer Time-Out Indicator
If this bit is set to 1, a WDT time-out occurred. A POR resets this pin. A Stop Mode Recovery due a change in an input pin also resets this bit. Reading this register resets this bit.

EXT—External Reset Indicator
If this bit is set to 1, a Reset initiated by the external $\overline{\text{RESET}}$ pin occurred. A POR or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.

Reserved
These bits are reserved and must be 0.

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) Registers (Table 49 through Table 51) form the 24-bit reload value that is loaded into the WDT, when a WDT instruction executes. The 24-bit reload value is {WDTU[7:0], WDTH[7:0], WDTL[7:0]}. Writing to these registers sets the required Reload Value. Reading from these registers returns the current WDT count value.

! Caution: *The 24-bit WDT Reload Value must not be set to a value less than* 000004H.

Table 49. Watchdog Timer Reload Upper Byte Register (WDTU)

BITS	7	6	5	4	3	2	1	0
FIELD	WDTU							
RESET	1							
R/W	R/W*							
ADDR	FF1H							
R/W*—Read returns the current WDT count value. Write sets the desired Reload Value.								

WDTU—WDT Reload Upper Byte
Most significant byte (MSB), Bits[23:16], of the 24-bit WDT reload value.

SPI Control Register Definitions

SPI Data Register

The SPI Data Register stores both the outgoing (transmit) data and the incoming (receive) data. Reads from the SPI Data Register always return the current contents of the 8-bit Shift Register. Data is shifted out starting with bit 7. The last bit received resides in bit position 0.

With the SPI configured as a Master, writing a data byte to this register initiates the data transmission. With the SPI configured as a Slave, writing a data byte to this register loads the shift register in preparation for the next data transfer with the external Master. In either the Master or Slave modes, if a transmission is already in progress, writes to this register are ignored and the Overrun error Flag, OVR, is set in the SPI Status Register.

When the character length is less than 8 bits (as set by the NUMBITS field in the SPI Mode Register), the transmit character must be left justified in the SPI Data Register. A received character of less than 8 bits is right justified (last bit received is in bit position 0). For example, if the SPI is configured for 4-bit characters, the transmit characters must be written to SPIDATA[7:4] and the received characters are read from SPIDATA[3:0].

Table 63. SPI Data Register (SPIDATA)

BITS	7	6	5	4	3	2	1	0
FIELD	DATA							
RESET	X							
R/W	R/W							
ADDR	F60H							

DATA—Data
Transmit and/or receive data.

I²C Controller

The I²C Controller makes the F0822 Series products bus-compatible with the I²C protocol. The I²C Controller consists of two bidirectional bus lines—a serial data signal (SDA) and a serial clock signal (SCL). Features of the I²C Controller include:

- Transmit and Receive Operation in MASTER mode.
- Maximum data rate of 400 kbit/s.
- 7-bit and 10-bit addressing modes for Slaves.
- Unrestricted number of data bytes transmitted per transfer.

The I²C Controller in the F0822 Series products does not operate in Slave mode.

Architecture

Figure 25 displays the architecture of the I²C Controller.

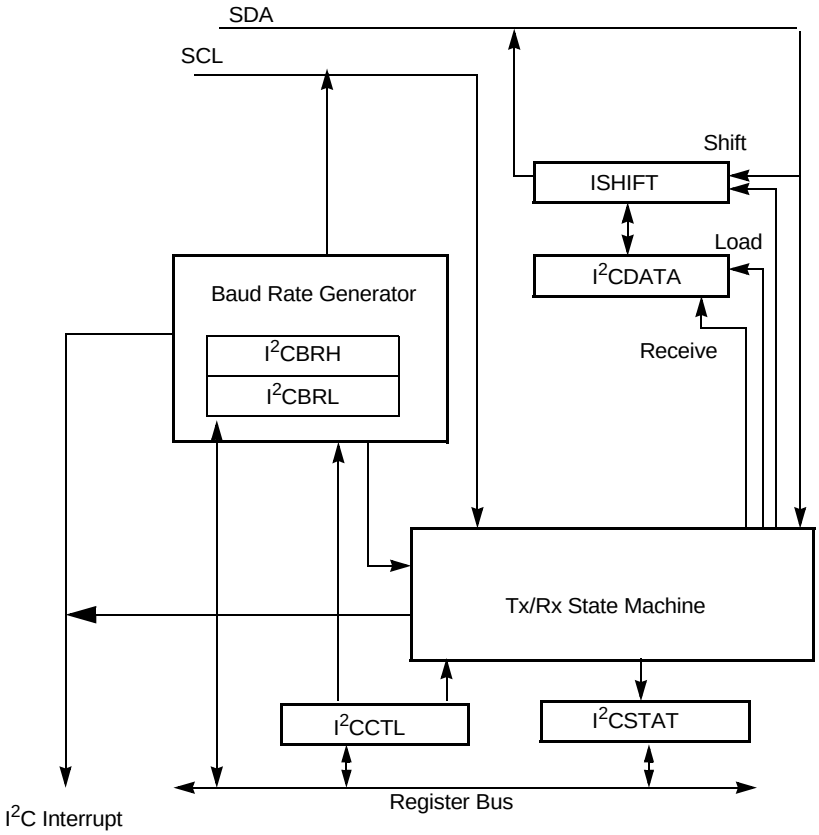

Figure 25. I²C Controller Block Diagram

Follow the steps below for a transmit operation on a 10-bit addressed slave:

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 interrupt by writing the first slave address byte to the I2C Data Register. The least-significant bit must be 0 for the write operation.
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 is shifted out by the SDA signal, the Transmit Interrupt is asserted.
9. Software responds by writing the second byte of address into the contents of the I2C Data Register.
10. The I2C Controller shifts the rest of the first byte of address and write bit out the SDA signal.
11. If the I2C Slave acknowledges the first address byte 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. Continue with step 12.

    If the slave does not acknowledge the first address byte, the I2C Controller sets the NCKI bit and clears the ACK bit in the I2C Status register. Software responds to the Not Acknowledge interrupt by setting the STOP and FLUSH bits and clearing the TXI bit. The I2C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore the following steps).
12. The I2C Controller loads the I2C Shift register with the contents of the I2C Data Register.
13. The I2C Controller shifts the second address byte out the SDA signal. After the first bit has been sent, the Transmit Interrupt is asserted.
14. Software responds by writing a data byte to the I2C Data Register.
15. The I2C Controller completes shifting the contents of the shift register on the SDA signal.
16. If 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. Continue with step 17.

    If the slave does not acknowledge the second address byte or one of the data bytes, the

Flash Controller Behavior in Debug Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the OCD:

- The Flash Write Protect option bit is ignored
- The Flash Sector Protect Register is ignored for programming and erase operations
- Programming operations are not limited to the page selected in the Page Select Register
- Bits in the Flash Sector Protect Register can be written to 1 or 0
- The second write of the Page Select Register to unlock the Flash Controller is not necessary
- The Page Select Register is written when the Flash Controller is unlocked
- The Mass Erase command is enabled

Flash Control Register Definitions

Flash Control Register

The Flash Control Register (Table 83) is used to unlock the Flash Controller for programming and erase operations, or to select the Flash Sector Protect Register. The Write-only Flash Control Register shares its Register File address with the Read-only Flash Status Register.

Table 83. Flash Control Register (FCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	FCMD							
RESET	0							
R/W	W							
ADDR	FF8H							

FCMD—Flash Command
73H = First unlock command.
8CH = Second unlock command.
95H = Page erase command.
63H = Mass erase command
5EH = Flash Sector Protect Register select.

* All other commands, or any command out of sequence, lock the Flash Controller.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

- **Read OCD Control Register (05H)**—The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

- **Write Program Counter (06H)**—The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter. If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the Program Counter values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

- **Read Program Counter (07H)**—The Read Program Counter command reads the value in the eZ8 CPU's Program Counter. If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

- **Write Register (08H)**—The Write Register command writes data to the Register File. Data can be written 1-256 bytes at a time (256 bytes can be written by setting size to zero). If the device is not in DEBUG mode, the address and data values are discarded. If the Read Protect Option Bit is enabled, then only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0,Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

- **Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1-256 bytes at a time (256 bytes can be read by setting size to zero). Reading peripheral control registers through the OCD does not effect peripheral operation. For example, register bits that are normally cleared upon a read operation will not be effected (WDTSTAT register is affected by OCD read register operation). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

Figure 44 displays the maximum HALT mode current consumption across the full operating temperature range of the device and versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

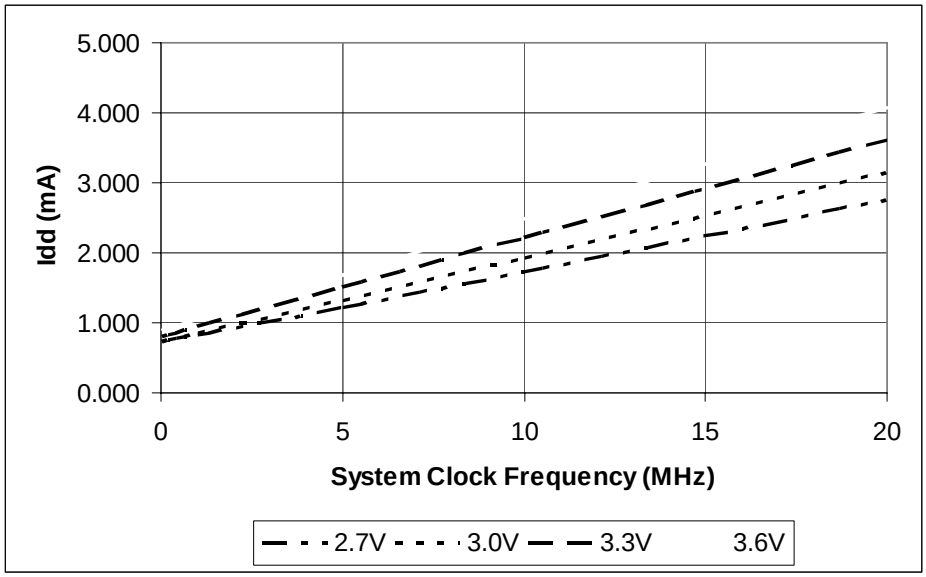

Figure 44. Maximum HALT Mode I_{CC} Versus System Clock Frequency

Table 104. Analog-to-Digital Converter Electrical Characteristics and Timing

Symbol	Parameter	V_{DD} = 3.0–3.6 V T_A = -40 °C to 105 °C			Units	Conditions
		Minimum	Typical	Maximum		
	Resolution	10	–	–	bits	External V_{REF} = 3.0 V;
	Differential Nonlinearity (DNL)	-0.25	–	0.25	lsb	Guaranteed by design
	Integral Nonlinearity (INL)	-2.0	–	2.0	lsb	External V_{REF} = 3.0 V
	DC Offset Error	-35	–	25	mV	
V_{REF}	Internal Reference Voltage	1.9	2.0	2.4	V	V_{DD} = 3.0 - 3.6 V T_A = -40 °C to 105 °C
VC_{REF}	Voltage Coefficient of Internal Reference Voltage	–	78	–	mV/V	V_{REF} variation as a function of AVDD.
TC_{REF}	Temperature Coefficient of Internal Reference Voltage	–	1	–	mV/^{0}C	
	Single-Shot Conversion Period	5129			cycles	System clock cycles
	Continuous Conversion Period	256			cycles	System clock cycles
R_S	Analog Source Impedance	–	–	150	W	Recommended
Zin	Input Impedance		150		KΩ	
V_{REF}	External Reference Voltage			AVDD	V	AVDD <= VDD. When using an external reference voltage, decoupling capacitance should be placed from VREF to AVSS.
I_{REF}	Current draw into VREF pin when driving with external source.		25.0	40.0	µA	

Condition Codes

The C, Z, S, and V flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the flag settings are encoded in a 4-bit field called the condition code (cc), which forms Bits 7:4 of the conditional jump instructions. The condition codes are summarized in Table 117. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the flag test operation decides if the conditional jump is executed.

Table 117. Condition Codes

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	–
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1
0011	3	ULE	Unsigned Less Than or Equal	(C OR Z) = 1
0100	4	OV	Overflow	V = 1
0101	5	MI	Minus	S = 1
0110	6	Z	Zero	Z = 1
0110	6	EQ	Equal	Z = 1
0111	7	C	Carry	C = 1
0111	7	ULT	Unsigned Less Than	C = 1
1000	8	T (or blank)	Always True	–
1001	9	GE	Greater Than or Equal	(S XOR V) = 0
1010	A	GT	Greater Than	(Z OR (S XOR V)) = 0
1011	B	UGT	Unsigned Greater Than	(C = 0 AND Z = 0) = 1
1100	C	NOV	No Overflow	V = 0
1101	D	PL	Plus	S = 0
1110	E	NZ	Non-Zero	Z = 0
1110	E	NE	Not Equal	Z = 0
1111	F	NC	No Carry	C = 0
1111	F	UGE	Unsigned Greater Than or Equal	C = 0

Table 125. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

eZ8 CPU Instruction Summary

Table 126 summarizes the eZ8 CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch, and the number of CPU clock cycles required for the instruction execution.

Table 126. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	13							2	4
		R	R	14							3	3
		R	IR	15							3	4
		R	IM	16							3	3
		IR	IM	17							3	4
ADCX dst, src	dst ← dst + src + C	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3

Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 57 and Table 127 on page 230. Figure 58 on page 231 and Figure 59 on page 232 provide information on each of the eZ8 CPU instructions.

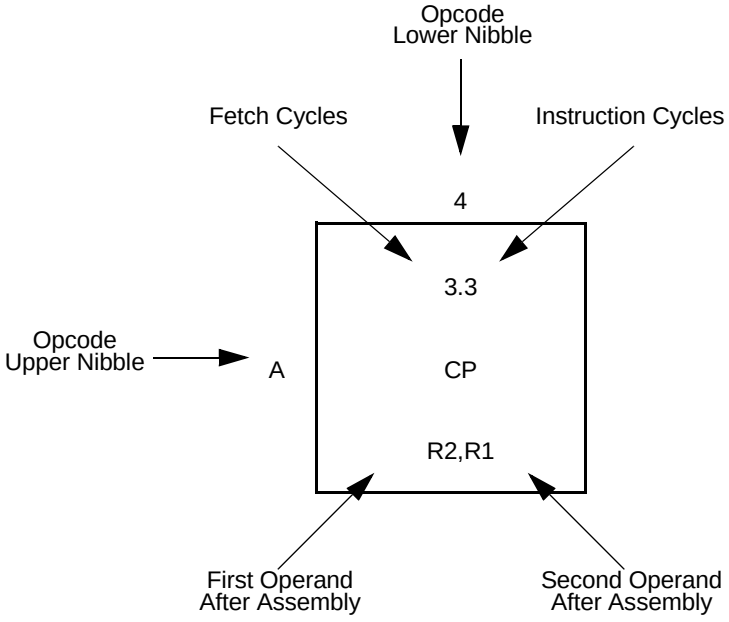

Figure 57. Opcode Map Cell Description

Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at http://www.zilog.com/kb.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at http://support.zilog.com.