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	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	60
Program Memory Size	48KB (48K 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4823ft020eg

Address Space

The eZ8 CPU can access three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral and general-purpose I/O port control registers
- The program memory contains addresses for all memory locations having executable code and/or data
- The Data Memory consists of the addresses for all memory locations that hold only data

These three address spaces are covered briefly in the following sections. For more information about the eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com.

Register File

The Register File address space in the Z8 Encore! XP F64xx Series is 4KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers are read from when defined as sources and written to when defined as destinations. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256-byte control register section are reserved (unavailable). Reading from an reserved Register File addresses returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8 Encore! XP F64xx Series provide 2KB to 4KB of on-chip RAM depending upon the device. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect. To determine the amount of RAM available for the specific Z8 Encore! XP F64xx Series device, see the Part Selection Guide section on page 2.

Program Memory

The eZ8 CPU supports 64KB of program memory address space. The Z8 Encore! XP F64xx Series contains 16KB to 64KB of on-chip Flash in the program memory address space, depending upon the device. Reading from program memory addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented program memory addresses produces no effect. Table 5 describes the program memory maps for the Z8 Encore! XP F64xx Series products.

Table 5. Z8 Encore! XP F64xx Series Program Memory Maps

Program Memory Address (Hex)	Function
Z8F162x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFF	Program Memory
Z8F242x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFF	Program Memory
Z8F322x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFF	Program Memory
Z8F482x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector

Note: *See Table 23 on page 48 for a list of the interrupt vectors.

Port A–H Alternate Function Subregisters

The Port A–H Alternate Function Subregister, shown in Table 17, is accessed through the Port A–H Control Register by writing 02H to the Port A–H Address Register. The Port A–H Alternate Function subregisters select the alternate functions for the selected pins. To determine the alternate function associated with each port pin, see the GPIO Alternate Functions section on page 37.

! **Caution:** Do not enable alternate function for GPIO port pins which do not have an associated alternate function. Failure to follow this guideline may result in unpredictable operation.

Table 17. Port A–H Alternate Function Subregisters

Bit	7	6	5	4	3	2	1	0
Field	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	0							
R/W	R/W							
Address	See note.							

Note: If a 02H exists in the Port A–H Address Register, it is accessible through the Port A–H Control Register.

Bit	Description
[7:0] AFx	**Port Alternate Function Enabled** 0 = The port pin is in NORMAL Mode and the DDx bit in the Port A–H Data Direction Subregister determines the direction of the pin. 1 = The alternate function is selected. Port pin operation is controlled by the alternate function.

Note: x indicates register bits in the range [7:0].

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) Register, shown in Table 26, stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 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 can read the Interrupt Request 1 Register to determine if any interrupt requests are pending.

Table 26. Interrupt Request 2 Register (IRQ2)

Bit	7	6	5	4	3	2	1	0
Field	T3I	U1RXI	U1TXI	DMAI	PC3I	PC2I	PC1I	PC0I
RESET	0							
R/W	R/W							
Address	FC6H							

Bit	Description
[7] T3I	**Timer 3 Interrupt Request** 0 = No interrupt request is pending for Timer 3. 1 = An interrupt request from Timer 3 is awaiting service.
[6] U1RXI	**UART 1 Receive Interrupt Request** 0 = No interrupt request is pending for the UART1 receiver. 1 = An interrupt request from UART1 receiver is awaiting service.
[5] U1TXI	**UART 1 Transmit Interrupt Request** 0 = No interrupt request is pending for the UART 1 transmitter. 1 = An interrupt request from the UART 1 transmitter is awaiting service.
[4] DMAI	**DMA Interrupt Request** 0 = No interrupt request is pending for the DMA. 1 = An interrupt request from the DMA is awaiting service.
[3:0] PCxI	**Port C Pin *x* Interrupt Request** 0 = No interrupt request is pending for GPIO Port C pin *x*. 1 = An interrupt request from GPIO Port C pin *x* is awaiting service.

Note: x indicates the specific GPIO Port C pin in the range [3:0].

it is appropriate to have the timer output make a permanent state change upon a One-Shot time-out, first set the TPOL bit in the Timer Control 1 Register to the start value before beginning ONE-SHOT Mode. Then, after starting the timer, set TPOL to the opposite bit value.

Observe the following procedure for configuring a timer for ONE-SHOT Mode and initiating the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for ONE-SHOT Mode
   - Set the prescale value
   - If using the timer output alternate function, set the initial output level (High or Low)
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, 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 1 Register to enable the timer and initiate counting.

In ONE-SHOT Mode, the system clock always provides the timer input. The timer period is calculated using the following equation:

$$\text{ONE-SHOT Mode Time-Out Period (s)} = \frac{(\text{Reload Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

CONTINUOUS Mode

In CONTINUOUS Mode, 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. Upon 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. Also, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) upon timer reload.

Observe the following procedure for configuring a timer for CONTINUOUS Mode and initiating the count:

1. Write to the Timer Control 1 Register to:

Timer 0–3 Control 0 Registers

The Timer 0–3 Control 0 (TxCTL0) registers, shown in Tables 45 and 46, allow cascading of the timers.

Table 45. Timer 0–3 Control 0 Register (TxCTL0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			CSC	Reserved			
RESET	0							
R/W	R/W							
Address	F06H, F0EH, F16H, F1EH							

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 000.
[4] CSC	**Cascade Timers** 0 = Timer input signal comes from the pin. 1 = For Timer 0, the input signal is connected to Timer 3 output. For Timer 1, the input signal is connected to the Timer 0 output. For Timer 2, the input signal is connected to the Timer 1 output. For Timer 3, the input signal is connected to the Timer 2 output.
[3:0]	**Reserved** These bits are reserved and must be programmed to 0000.

Bit	Description (Continued)
[5:3] PRES	**Prescale Value** The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the timer is disabled to ensure 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.
[2:0] 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.

unlock the Watchdog Timer Reload Byte registers (WDTU, WDTH, and WDTL) for write access.

1. Write 55H to the Watchdog Timer Control Register (WDTCTL).
2. Write AAH to the Watchdog Timer Control Register (WDTCTL).
3. Write the Watchdog Timer Reload Upper Byte Register (WDTU).
4. Write the Watchdog Timer Reload High Byte Register (WDTH).
5. Write the Watchdog Timer Reload Low Byte Register (WDTL).

All steps of the Watchdog Timer reload unlock sequence must be written in the sequence described above; 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 can occur, unless the sequence is restarted. The value in the Watchdog Timer Reload registers is loaded into the counter when the Watchdog Timer is first enabled and every time a WDT instruction is executed.

Watchdog Timer Control Register Definitions

This section defines the features of the following Watchdog Timer Control registers.

Watchdog Timer Control Register: see page 83

Watchdog Timer Reload Upper, High and Low Byte Registers: see page 85

Watchdog Timer Control Register

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

Writing the 55H, AAH unlock sequence to the Watchdog Timer Control (WDTCTL) Register 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 Register address produce no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the Reload registers.

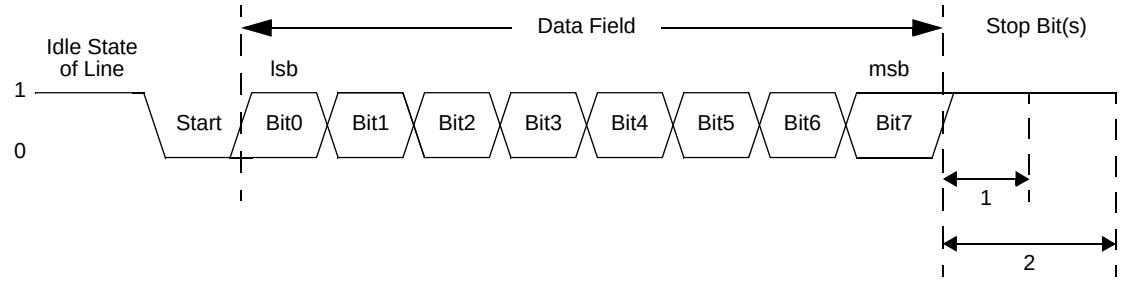

Figure 14. UART Asynchronous Data Format without Parity

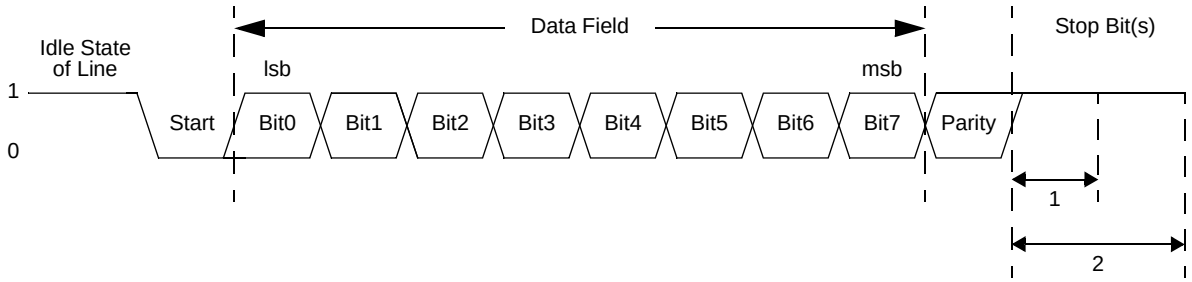

Figure 15. UART Asynchronous Data Format with Parity

Transmitting Data using the Polled Method

Observe the following procedure to transmit data using the polled method of operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. If MULTIPROCESSOR Mode is appropriate, write to the UART Control 1 Register to enable MULTIPROCESSOR (9-Bit) Mode functions.
   - Set the MULTIPROCESSOR Mode Select (MPEN) to Enable MULTIPROCESSOR Mode
4. Write to the UART Control 0 Register to:
   - Set the transmit enable bit (TEN) to enable the UART for data transmission
   - If parity is appropriate and MULTIPROCESSOR Mode is not enabled, set the parity enable bit (PEN) and select either Even or Odd parity (PSEL)

- 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 shifts out

Note: Writing to the I²C Data Register always clears the TRDE bit to 0. When TDRE is asserted, the I²C Controller pauses at the beginning of the Acknowledge cycle of the byte currently shifting out. It does not resume until the Data Register is written with the next value to send or until the stop or start bits are set, indicating that the current byte is the last one to send.

The fourth interrupt source is the baud rate generator. If the I²C Controller is disabled (IEN bit in the I2CCTL Register = 0) and the BIRQ bit in the I2CCTL Register = 1, an interrupt is generated when the baud rate generator counts down to 1. This allows the I²C baud rate generator to be used by software as a general purpose timer when IEN = 0.

Software Control of I²C Transactions

Software can control I²C transactions by using the I²C Controller interrupt, by polling the I²C Status Register or by DMA. Note that not all products include a DMA Controller.

To use interrupts, the I²C interrupt must be enabled in the Interrupt Controller. The TXI bit in the I²C Control Register must be set to enable transmit interrupts.

To control transactions by polling, the interrupt bits (TDRE, RDRF and NCKI) in the I²C Status Register should be polled. The TDRE bit asserts regardless of the state of the TXI bit.

Either or both transmit and receive data movement can be controlled by the DMA Controller. The DMA Controller channel(s) must be initialized to select the I²C transmit and receive requests. Transmit DMA requests require that the TXI bit in the I²C Control Register be set.

Caution: A transmit (write) DMA operation hangs if the slave responds with a Not Acknowledge before the last byte has been sent. After receiving the Not Acknowledge, the I²C Controller sets the NCKI bit in the Status Register and pauses until either the stop or start bits in the Control Register are set.

For a receive (read) DMA transaction to send a Not Acknowledge on the last byte, the receive DMA must be set up to receive n-1 bytes, then software must set the NAK bit and receive the last (nth) byte directly.

Table 71. I²C Data Register (I2CDATA)

Bit	7	6	5	4	3	2	1	0
Field	DATA							
RESET	0							
R/W	R/W							
Address	F50H							

I²C Status Register

The read-only I²C Status Register, shown in Table 72, indicates the status of the I²C Controller.

Table 72. I²C Status Register (I2CSTAT)

Bit	7	6	5	4	3	2	1	0
Field	TDRE	RDRF	ACK	10B	RD	TAS	DSS	NCKI
RESET	1	0						
R/W	R							
Address	F51H							

Bit	Description
[7] TDRE	**Transmit Data Register Empty** When the I²C Controller is enabled, this bit is 1 when the I²C Data Register is empty. When this bit is set, an interrupt is generated if the TXI bit is set, except when the I²C Controller is shifting in data during the reception of a byte or when shifting an address and the RD bit is set. This bit is cleared by writing to the I2CDATA Register.
[6] RDRF	**Receive Data Register Full** This bit is set = 1 when the I²C Controller is enabled and the I²C Controller has received a byte of data. When asserted, this bit causes the I²C Controller to generate an interrupt. This bit is cleared by reading the I²C Data Register (unless the read is performed using execution of the On-Chip Debugger's Read Register command).

Bit	Description (Continued)
[4] CONT	**Conversion** 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.
[3:0] ANAIN[3:0]	**Analog Input Select** These bits select the analog input for conversion. For information about the Port pins available with each package style, see the Signal and Pin Descriptions chapter on page 7. 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.

interrupts are typically disabled during critical sections of code where interrupts should not occur (such as adjusting the stack pointer or modifying shared data).

Software can poll the IDLE bit of the OCDSTAT Register to determine if the OCD is looping on a BRK instruction. When software stops the CPU on the BRK instruction that it is looping on, it should not set the DBGMODE bit of the OCDCTL Register. The CPU may have vectored to and be in the middle of an interrupt service routine when this bit gets set. Instead, software must clear the BRKLP bit. This action allows the CPU to finish the interrupt service routine it may be in and return the BRK instruction. When the CPU returns to the BRK instruction it was previously looping on, it automatically sets the DBGMODE bit and enters DEBUG Mode.

Software detects that the majority of the OCD commands are still disabled when the eZ8 CPU is looping on a BRK instruction. The eZ8 CPU must be stopped and the part must be in DEBUG Mode before these commands can be issued.

Breakpoints in Flash Memory

The BRK instruction is op code 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a breakpoint, write 00H to the appropriate address, overwriting the current instruction. To remove a breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG Mode, all OCD commands become available unless the user code and control registers are protected by programming the Read Protect option bit (RP). The Read Protect option bit prevents the code in memory from being read out of the Z8 Encore! XP F64xx Series products. When this option is enabled, several of the OCD commands are disabled.

Table 102 contains a summary of the On-Chip Debugger commands. Table 102 lists those commands that operate when the device is not in DEBUG Mode (normal operation) and those commands that are disabled by programming the Read Protect option bit.

Each OCD command is further described in the list that follows the table.

```
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). 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
```

Write Program Memory (0AH). The Write Program Memory command writes data to program memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Program Memory (0BH). The Read Program Memory command reads data from program memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG 1-65536 data bytes
```

Write Data Memory (0CH). The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). If the device is not in DEBUG Mode or if the Read Protect option bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Oscillator Operation with an External RC Network

The External RC Oscillator mode is applicable to timing-insensitive applications. Figure 41 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

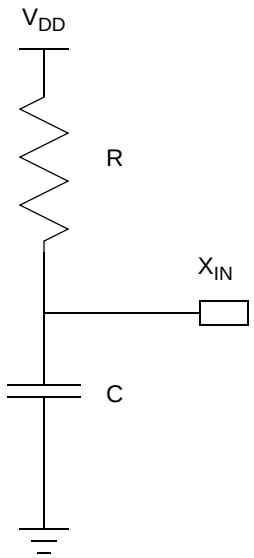

Figure 41. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45kΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40kΩ. The typical oscillator frequency can be estimated from the values of the resistor (*R* in kΩ) and capacitor (*C* in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1\times10^{6}}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 42 displays the typical (3.3V and 25°C) oscillator frequency as a function of the capacitor (*C* in pF) employed in the RC network assuming a 45kΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board should be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20pF are recommended.

On-Chip Peripheral AC and DC Electrical Characteristics

Table 108. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Minimum**	**Typical***	**Maximum**	**Units**	**Conditions**
V_{POR}	Power-On Reset Voltage Threshold	2.40	2.70	2.90	V	$V_{DD} = V_{POR}$
V_{VBO}	Voltage Brown-Out Reset Voltage Threshold	2.30	2.60	2.85	V	$V_{DD} = V_{VBO}$
	V_{POR} to V_{VBO} hysteresis	50	100	–	mV	
	Starting V_{DD} voltage to ensure valid Power-On Reset.	–	V_{SS}	–	V	
T_{ANA}	Power-On Reset Analog Delay	–	50	–	µs	$V_{DD} > V_{POR}$; T_{POR} Digital Reset delay follows T_{ANA}
T_{POR}	Power-On Reset Digital Delay	–	6.6	–	ms	66 WDT Oscillator cycles (10kHz) + 16 System Clock cycles (20MHz)
T_{VBO}	Voltage Brown-Out Pulse Rejection Period	–	10	–	µs	$V_{DD} < V_{VBO}$ to generate a Reset.
T_{RAMP}	Time for V_{DD} to transition from V_{SS} to V_{POR} to ensure valid Reset	0.10	–	100	ms	

Note: *Data in the typical column is from characterization at 3.3V and 0°C. These values are provided for design guidance only and are not tested in production.

Table 135. 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 136 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 136. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
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

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Hex Address: F40

Table 170. UART Transmit Data Register (U*x*TXD)

Bit	7	6	5	4	3	2	1	0
Field	TXD							
RESET	X							
R/W	W							
Address	F40H and F48H							

Table 171. UART Receive Data Register (U*x*RXD)

Bit	7	6	5	4	3	2	1	0
Field	RXD							
RESET	X							
R/W	R							
Address	F40H and F48H							

Hex Address: F41

Table 172. UART Status 0 Register (U*x*STAT0)

Bit	7	6	5	4	3	2	1	0
Field	RDA	PE	OE	FE	BRKD	TDRE	TXE	CTS
RESET	0						1	X
R/W	R							
Address	F41H and F49H							

Hex Address: F42

Table 173. UART Control 0 Register (U*x*CTL0)

Bit	7	6	5	4	3	2	1	0
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
RESET	0							
R/W	R/W							
Address	F42H and F4AH							

Index

Numerics

A

B

C