**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 | 64-LQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f6422ar020sc |

# Manual Objectives

This Product Specification provides detailed operating information for the Flash devices within Zilog's Z8 Encore! XP® 64K Series Flash Microcontrollers Microcontroller (MCU) products. Within this document, the Z8F642x, Z8F482x, Z8F322x, Z8F242x, and Z8F162x devices are referred to collectively as the Z8 Encore! XP® 64K Series Flash Microcontrollers unless specifically stated otherwise.

## About This Manual

Zilog® recommends that you read and understand everything in this manual before setting up and using the product. However, we recognize that there are different styles of learning. Therefore, we have designed this Product Specification to be used either as a *how to* procedural manual or a reference guide to important data.

## Intended Audience

This document is written for Zilog customers who are experienced at working with microcontrollers, integrated circuits, or printed circuit assemblies.

## Manual Conventions

The following assumptions and conventions are adopted to provide clarity and ease of use:

### Courier Typeface

Commands, code lines and fragments, bits, equations, hexadecimal addresses, and various executable items are distinguished from general text by the use of the `Courier` typeface. Where the use of the font is not indicated, as in the Index, the name of the entity is presented in upper case.

- Example: FLAGS[1] is `smrf`.

### Hexadecimal Values

Hexadecimal values are designated by uppercase *H* suffix and appear in the `Courier` typeface.

- Example: R1 is set to `F8H.`

### Brackets

The square brackets, [ ], indicate a register or bus.

- Example: For the register R1[7:0], R1 is an 8-bit register, R1[7] is the most significant bit, and R1[0] is the least significant bit.

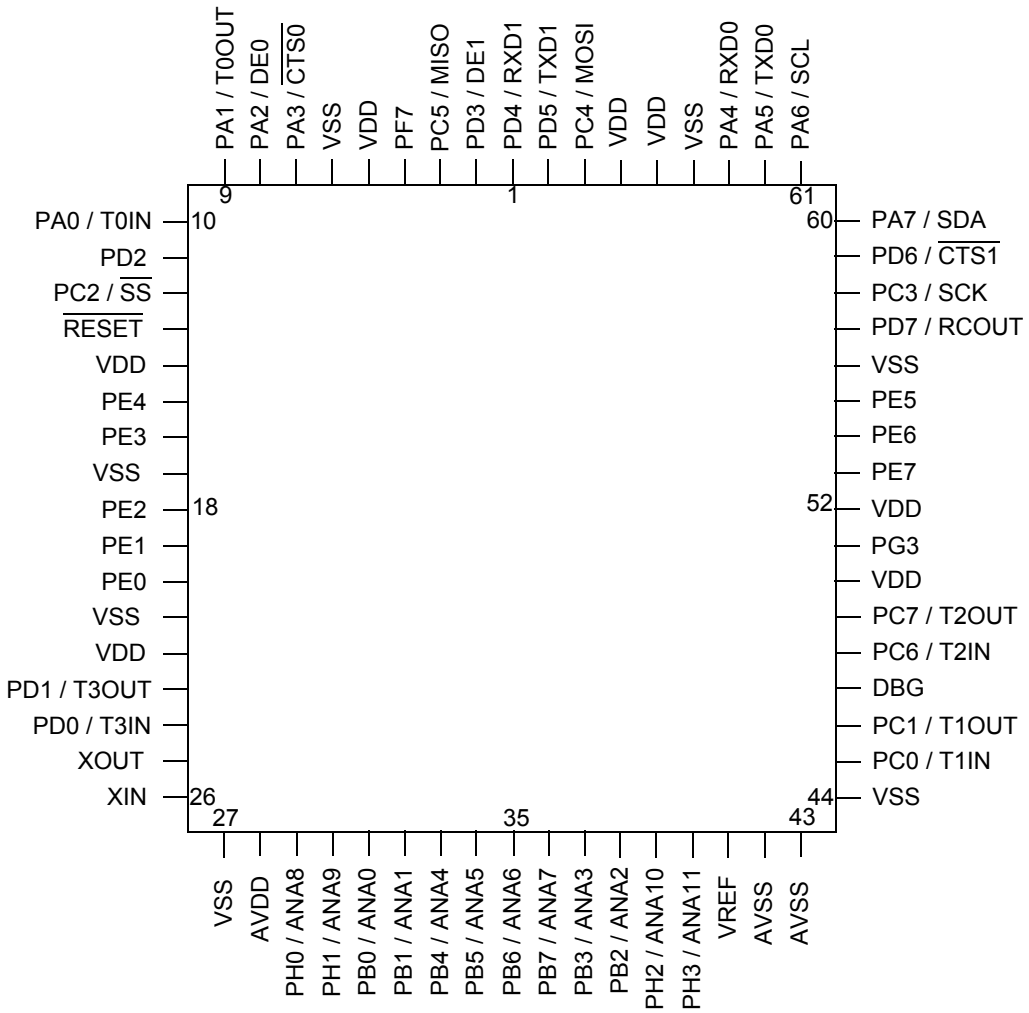



**Figure 6. Z8 Encore! XP 64K Series Flash Microcontrollers in 68-Pin Plastic Leaded Chip Carrier (PLCC)**

### Port A–H Data Direction Sub-Registers

The Port A–H Data Direction sub-register is accessed through the Port A–H Control register by writing 01H to the Port A–H Address register (Table 16).

**Table 16. Port A–H Data Direction Sub-Registers**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | DD7 | DD6 | DD5 | DD4 | DD3 | DD2 | DD1 | DD0 |
| RESET | 1 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | If 01H in Port A–H Address Register, accessible through Port A–H Control Register | | | | | | | |

DD[7:0]—Data Direction
These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.
0 = Output. Data in the Port A–H Output Data register is driven onto the port pin.
1 = Input. The port pin is sampled and the value written into the Port A–H Input Data Register. The output driver is tri-stated.

### Port A–H Alternate Function Sub-Registers

The Port A–H Alternate Function sub-register (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 sub-registers select the alternate functions for the selected pins. To determine the alternate function associated with each port pin, see GPIO Alternate Functions on page 59.

**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 Sub-Registers**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | AF7 | AF6 | AF5 | AF4 | AF3 | AF2 | AF1 | AF0 |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | If 02H in Port A–H Address Register, accessible through Port A–H Control Register | | | | | | | |

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.

U0RXI—UART 0 Receiver Interrupt Request
0 = No interrupt request is pending for the UART 0 receiver.
1 = An interrupt request from the UART 0 receiver is awaiting service.

U0TXI—UART 0 Transmitter Interrupt Request
0 = No interrupt request is pending for the UART 0 transmitter.
1 = An interrupt request from the UART 0 transmitter is awaiting service.

$I^2CI$— $I^2C$ Interrupt Request
0 = No interrupt request is pending for the $I^2C$.
1 = An interrupt request from the $I^2C$ is awaiting service.

SPII—SPI Interrupt Request
0 = No interrupt request is pending for the SPI.
1 = An interrupt request from the SPI is awaiting service.

ADCI—ADC Interrupt Request
0 = No interrupt request is pending for the Analog-to-Digital Converter.
1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

## Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 25) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 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 25. Interrupt Request 1 Register (IRQ1)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | PAD7I | PAD6I | PAD5I | PAD4I | PAD3I | PAD2I | PAD1I | PAD0I |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | FC3H | | | | | | | |

PAD*x*I—Port A or Port D Pin *x* Interrupt Request
0 = No interrupt request is pending for GPIO Port A or Port D pin *x*.
1 = An interrupt request from GPIO Port A or Port D pin *x* is awaiting service.

where *x* indicates the specific GPIO Port pin number (0 through 7). For each pin, only 1 of either Port A or Port D can be enabled for interrupts at any one time. Port selection (A or D) is determined by the values in the Interrupt Port Select Register.

## Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register (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)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | T3I | U1RXI | U1TXI | DMAI | PC3I | PC2I | PC1I | PC0I |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | FC6H | | | | | | | |

T3I—Timer 3 Interrupt Request
0 = No interrupt request is pending for Timer 3.
1 = An interrupt request from Timer 3 is awaiting service.

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.

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.

DMAI—DMA Interrupt Request
0 = No interrupt request is pending for the DMA.
1 = An interrupt request from the DMA is awaiting service.

PC*x*I—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.

**Table 33. IRQ2 Enable and Priority Encoding (Continued)**

| IRQ2ENH[*x*] | IRQ2ENL[*x*] | Priority | Description |
|---|---|---|---|
| 1 | 1 | Level 3 | High |

**Note:** where *x* indicates the register bits from 0 through 7.

**Table 34. IRQ2 Enable High Bit Register (IRQ2ENH)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | T3ENH | U1RENH | U1TENH | DMAENH | C3ENH | C2ENH | C1ENH | C0ENH |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | FC7H | | | | | | | |

T3ENH—Timer 3 Interrupt Request Enable High Bit
U1RENH—UART 1 Receive Interrupt Request Enable High Bit
U1TENH—UART 1 Transmit Interrupt Request Enable High Bit
DMAENH—DMA Interrupt Request Enable High Bit
C3ENH—Port C3 Interrupt Request Enable High Bit
C2ENH—Port C2 Interrupt Request Enable High Bit
C1ENH—Port C1 Interrupt Request Enable High Bit
C0ENH—Port C0 Interrupt Request Enable High Bit

**Table 35. IRQ2 Enable Low Bit Register (IRQ2ENL)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | T3ENL | U1RENL | U1TENL | DMAENL | C3ENL | C2ENL | C1ENL | C0ENL |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | FC8H | | | | | | | |

T3ENL—Timer 3 Interrupt Request Enable Low Bit
U1RENL—UART 1 Receive Interrupt Request Enable Low Bit
U1TENL—UART 1 Transmit Interrupt Request Enable Low Bit
DMAENL—DMA Interrupt Request Enable Low Bit
C3ENL—Port C3 Interrupt Request Enable Low Bit
C2ENL—Port C2 Interrupt Request Enable Low Bit

**Table 59. UART Baud Rate High Byte Register (U*x*BRH)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | BRH | | | | | | | |
| RESET | 1 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | F46H and F4EH | | | | | | | |

**Table 60. UART Baud Rate Low Byte Register (U*x*BRL)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | BRL | | | | | | | |
| RESET | 1 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | F47H and F4FH | | | | | | | |

For a given UART data rate, the integer baud rate divisor value is calculated using the following equation:

$$\text{UART Baud Rate Divisor Value (BRG)} = Round\left(\frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Data Rate (bits/s)}}\right)$$

The baud rate error relative to the desired baud rate is calculated using the following equation:

$$\text{UART Baud Rate Error (\%)} = 100 \times \left(\frac{\text{Actual Data Rate} - \text{Desired Data Rate}}{\text{Desired Data Rate}}\right)$$

For reliable communication, the UART baud rate error must never exceed 5 percent. Table 61 provides information on data rate errors for popular baud rates and commonly used crystal oscillator frequencies.

repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal. This action allows the Endec to tolerate jitter and baud rate errors in the incoming data stream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

## Infrared Encoder/Decoder Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined in UART Control Register Definitions on page 114.

**Caution:** *To prevent spurious signals during IrDA data transmission, set the* `IREN` *bit in the UARTx Control 1 register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.*

- Master receives from a 7-bit slave
- Master receives from a 10-bit slave

## SDA and SCL Signals

I²C sends all addresses, data and acknowledge signals over the SDA line, most-significant bit first. SCL is the common clock for the I²C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I²C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a high level. When the slave releases the clock, the I²C Controller continues the transaction. All data is transferred in bytes and there is no 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 four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The Transmit interrupt is enabled by the IEN and TXI bits of the Control register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I²C Controller and neither the START or STOP bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status register and can only be cleared by setting the START or STOP bit in the I²C Control register. When this interrupt occurs, the I²C Controller waits until either the STOP or START bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I²C Controller (master reading data from slave). This procedure sets the RDRF bit of the I²C Status register. The RDRF bit is cleared by reading the I²C Data register. The RDRF bit is set during the acknowledge phase. The I²C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

15. The I2C Controller sends the repeated START condition.
16. The I2C Controller loads the I2C Shift register with the contents of the I2C Data register (third address transfer).
17. The I2C Controller sends 11110B followed by the two most significant bits of the slave read address and a 1 (read).
18. The I2C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL

    If the slave were to Not Acknowledge at this point (this should not happen because the slave did acknowledge the first two address bytes), software would respond 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).
19. The I2C Controller shifts in a byte of data from the I2C slave on the SDA signal. The I2C Controller sends a Not Acknowledge to the I2C slave if the NAK bit is set (last byte), else it sends an Acknowledge.
20. The I2C Controller asserts the Receive interrupt (RDRF bit set in the Status register).
21. Software responds by reading the I2C Data register which clears the RDRF bit. If there is only one more byte to receive, set the NAK bit of the I2C Control register.
22. If there are one or more bytes to transfer, return to step 19.
23. After the last byte is shifted in, a Not Acknowledge interrupt is generated by the I2C Controller.
24. Software responds by setting the STOP bit of the I2C Control register.
25. A STOP condition is sent to the I2C slave and the STOP and NCKI bits are cleared.

## I2C Control Register Definitions

### I2C Data Register

The I2C Data register (see Table 70 on page 157) holds the data that is to be loaded into the I2C Shift register during a write to a slave. This register also holds data that is loaded from the I2C Shift register during a read from a slave. The I2C Shift Register is not accessible in the Register File address space, but is used only to buffer incoming and outgoing data.

**Table 105. Absolute Maximum Ratings (Continued)**

| Parameter | Minimum | Maximum | Units | Notes |
|---|---|---|---|---|
| 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 °C to 125 °C** | | | | |
| 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 °C to 125 °C** | | | | |
| 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 °C to 125 °C** | | | | |
| Total power dissipation | | 360 | mW | |
| Maximum current into $V_{DD}$ or out of $V_{SS}$ | | 100 | 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 125 °C** | | | | |
| Total power dissipation | | 540 | mW | |
| Maximum current into $V_{DD}$ or out of $V_{SS}$ | | 150 | mA | |

**Note:** This voltage applies to all pins except the following: VDD, AVDD, pins supporting analog input (Ports B and H), $\overline{\text{RESET}}$, and where noted otherwise.

## On-Chip Peripheral AC and DC Electrical Characteristics

### Table 107. Power-On Reset and Voltage Brownout Electrical Characteristics and Timing

| Symbol | Parameter | $T_A$ = –40 °C to 125 °C | | | Units | Conditions |
|---|---|---|---|---|---|---|
| | | Minimum | Typical[1] | Maximum | | |
| $V_{POR}$ | Power-On Reset Voltage Threshold | 2.40 | 2.70 | 2.90 | V | $V_{DD} = V_{POR}$ |
| $V_{VBO}$ | Voltage Brownout 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 (10 kHz) + 16 System Clock cycles (20 MHz) |
| $T_{VBO}$ | Voltage Brownout Pulse Rejection Period | – | 10 | – | µs | $V_{DD} < V_{VBO}$ to generate a Reset. |
| $T_{RAMP}$ | Time for VDD to transition from $V_{SS}$ to $V_{POR}$ to ensure valid Reset | 0.10 | – | 100 | ms | |

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

## On-Chip Debugger Timing

Figure 52 and Table 116 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 µs maximum rise and fall time.

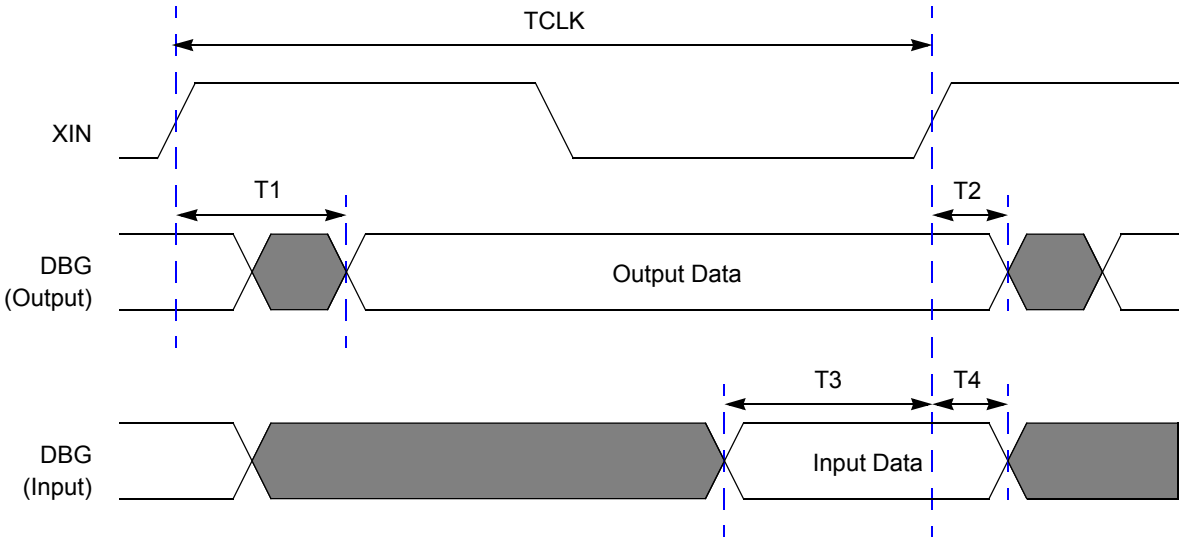



**Figure 52. On-Chip Debugger Timing**

**Table 116. On-Chip Debugger Timing**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| **DBG** | | | |
| $T_1$ | XIN Rise to DBG Valid Delay | – | 30 |
| $T_2$ | 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 |

## I²C Timing

Figure 55 and Table 119 provide timing information for I²C pins.

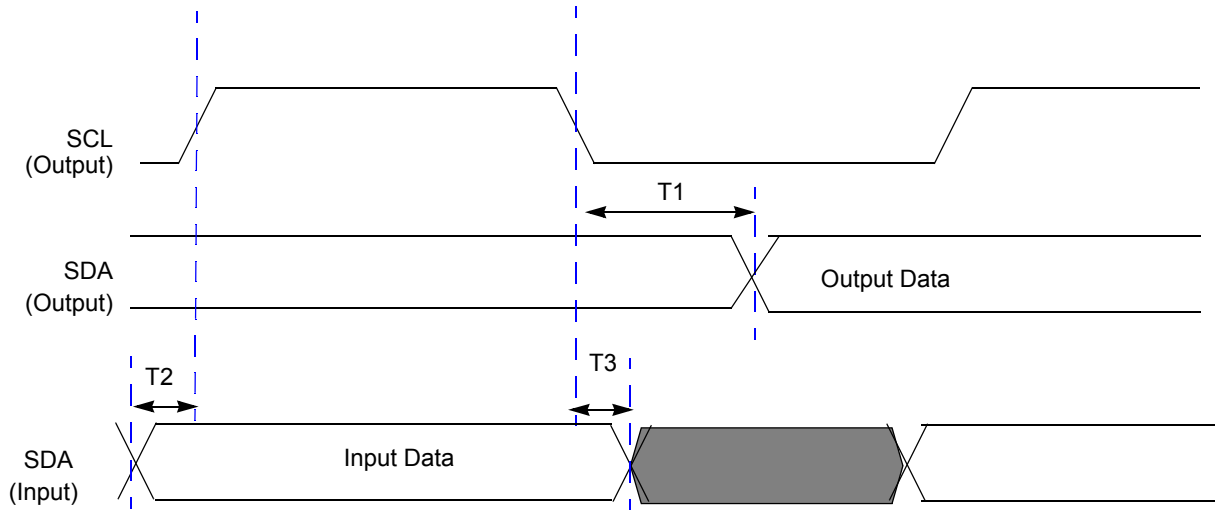



**Figure 55. I²C Timing**

**Table 119. I²C Timing**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| **I²C** | | | |
| $T_1$ | SCL Fall to SDA output delay | | SCL period/4 |
| $T_2$ | SDA Input to SCL rising edge Setup Time | 0 | |
| $T_3$ | SDA Input to SCL falling edge Hold Time | 0 | |

**Table 133. eZ8 CPU Instruction Summary (Continued)**

| Assembly Mnemonic | Symbolic Operation | Address Mode dst | Address Mode src | Opcode(s) (Hex) | Flags C | Z | S | V | D | H | Fetch Cycles | Instr. Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AND dst, src | dst ← dst AND src | r | r | 52 | - | * | * | 0 | - | - | 2 | 3 |
| | | r | Ir | 53 | | | | | | | 2 | 4 |
| | | R | R | 54 | | | | | | | 3 | 3 |
| | | R | IR | 55 | | | | | | | 3 | 4 |
| | | R | IM | 56 | | | | | | | 3 | 3 |
| | | IR | IM | 57 | | | | | | | 3 | 4 |
| ANDX dst, src | dst ← dst AND src | ER | ER | 58 | - | * | * | 0 | - | - | 4 | 3 |
| | | ER | IM | 59 | | | | | | | 4 | 3 |
| ATM | Block all interrupt and DMA requests during execution of the next 3 instructions | | | 2F | - | - | - | - | - | - | 1 | 2 |
| BCLR bit, dst | dst[bit] ← 0 | r | | E2 | - | * | * | 0 | - | - | 2 | 2 |
| BIT p, bit, dst | dst[bit] ← p | r | | E2 | - | * | * | 0 | - | - | 2 | 2 |
| BRK | Debugger Break | | | 00 | - | - | - | - | - | - | 1 | 1 |
| BSET bit, dst | dst[bit] ← 1 | r | | E2 | - | * | * | 0 | - | - | 2 | 2 |
| BSWAP dst | dst[7:0] ← dst[0:7] | R | | D5 | X | * | * | 0 | - | - | 2 | 2 |
| BTJ p, bit, src, dst | if src[bit] = p PC ← PC + X | | r | F6 | - | - | - | - | - | - | 3 | 3 |
| | | | Ir | F7 | | | | | | | 3 | 4 |
| BTJNZ bit, src, dst | if src[bit] = 1 PC ← PC + X | | r | F6 | - | - | - | - | - | - | 3 | 3 |
| | | | Ir | F7 | | | | | | | 3 | 4 |
| BTJZ bit, src, dst | if src[bit] = 0 PC ← PC + X | | r | F6 | - | - | - | - | - | - | 3 | 3 |
| | | | Ir | F7 | | | | | | | 3 | 4 |
| CALL dst | SP ← SP -2 @SP ← PC PC ← dst | IRR | | D4 | - | - | - | - | - | - | 2 | 6 |
| | | DA | | D6 | | | | | | | 3 | 3 |
| CCF | C ← ~C | | | EF | * | - | - | - | - | - | 1 | 2 |
| CLR dst | dst ← 00H | R | | B0 | - | - | - | - | - | - | 2 | 2 |
| | | IR | | B1 | | | | | | | 2 | 3 |

# Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 59 and Table 134 on page 262. Figure 60 on page 263 and Figure 61 on page 264 provide information on each of the eZ8™ CPU instructions.

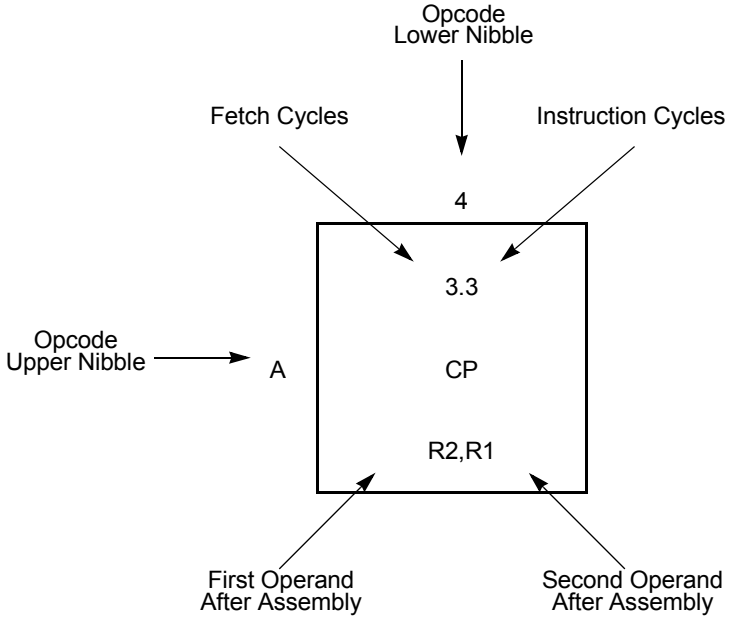
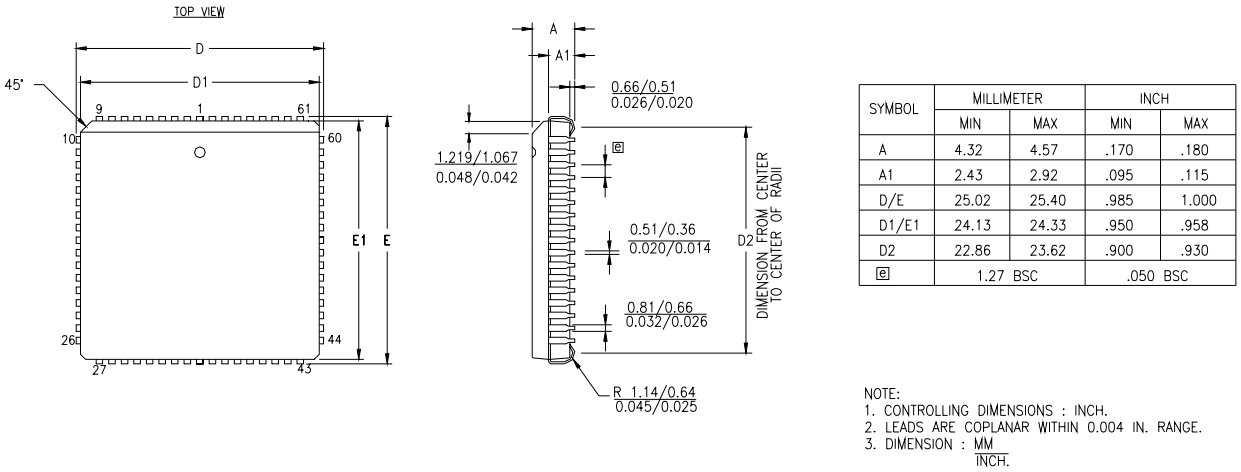



**Figure 59. Opcode Map Cell Description**

Figure 66 displays the 68-pin Plastic Lead Chip Carrier (PLCC) package available for the Z8X1622, Z8X2422, Z8X3222, Z8X4822, and Z8X6422 devices.




| SYMBOL | MILLIMETER | | INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | 4.32 | 4.57 | .170 | .180 |
| A1 | 2.43 | 2.92 | .095 | .115 |
| D/E | 25.02 | 25.40 | .985 | 1.000 |
| D1/E1 | 24.13 | 24.33 | .950 | .958 |
| D2 | 22.86 | 23.62 | .900 | .930 |
| e | 1.27 BSC | | .050 BSC | |

**Figure 66. 68-Lead Plastic Lead Chip Carrier Package (PLCC)**

## V



## W



## X



## Z