



Welcome to [E-XFL.COM](https://www.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	I <sup>2</sup> C, 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f6402vs020ec">https://www.e-xfl.com/product-detail/zilog/z8f6402vs020ec</a>



DMA_ADC Operation .....	123
Configuring DMA_ADC for Data Transfer .....	124
DMA Control Register Definitions .....	124
DMAx Control Register .....	124
DMAx I/O Address Register .....	125
DMAx Address High Nibble Register .....	126
DMAx Start/Current Address Low Byte Register .....	127
DMAx End Address Low Byte Register .....	128
DMA_ADC Address Register .....	128
DMA_ADC Control Register .....	130
DMA Status Register .....	131
<b>Analog-to-Digital Converter .....</b>	<b>132</b>
Overview .....	132
Architecture .....	132
Operation .....	133
Automatic Power-Down .....	133
Single-Shot Conversion .....	133
Continuous Conversion .....	134
DMA Control of the ADC .....	135
ADC Control Register Definitions .....	135
ADC Control Register .....	135
ADC Data High Byte Register .....	137
ADC Data Low Bits Register .....	137
<b>Flash Memory .....</b>	<b>138</b>
Overview .....	138
Operation .....	139
Flash Operation Timing Using the Flash Frequency Registers ..	141
Flash Code Protection Against External Access .....	141
Flash Code Protection Against Accidental Program and Erasure	141
Byte Programming .....	142
Page Erase .....	143
Mass Erase .....	143
Flash Controller Bypass .....	143
Flash Control Register Definitions .....	144
Flash Control Register .....	144
Flash Status Register .....	145
Flash Page Select Register .....	146
Flash Frequency High and Low Byte Registers .....	147



## *List of Tables*

Table 1.	Z8F640x Family Part Selection Guide	2
Table 2.	Z8F640x Family Package Options	6
Table 3.	Signal Descriptions	13
Table 4.	Pin Characteristics of the Z8F640x family	15
Table 5.	Z8F640x Family Program Memory Maps	18
Table 6.	Z8F640x Family Data Memory Maps	19
Table 7.	Register File Address Map	20
Table 8.	Reset and STOP Mode Recovery Characteristics and Latency	25
Table 9.	Reset Sources and Resulting Reset Type	26
Table 10.	STOP Mode Recovery Sources and Resulting Action	29
Table 11.	Port Availability by Device and Package Type	33
Table 12.	Port Alternate Function Mapping	35
Table 13.	Port A-H GPIO Address Registers (PxADDR)	37
Table 14.	GPIO Port Registers and Sub-Registers	37
Table 15.	Port A-H Control Registers (PxCTL)	38
Table 16.	Port A-H Data Direction Sub-Registers	39
Table 17.	Port A-H Alternate Function Sub-Registers	39
Table 18.	Port A-H Output Control Sub-Registers	40
Table 19.	Port A-H High Drive Enable Sub-Registers	41
Table 20.	Port A-H Input Data Registers (PxIN)	42
Table 21.	Port A-H STOP Mode Recovery Source Enable Sub-Registers	42
Table 22.	Port A-H Output Data Register (PxOUT)	43
Table 23.	Interrupt Vectors in Order of Priority	45
Table 24.	Interrupt Request 0 Register (IRQ0)	48
Table 25.	Interrupt Request 1 Register (IRQ1)	49
Table 26.	Interrupt Request 2 Register (IRQ2)	50
Table 27.	IRQ0 Enable and Priority Encoding	51
Table 28.	IRQ0 Enable High Bit Register (IRQ0ENH)	51
Table 29.	IRQ0 Enable Low Bit Register (IRQ0ENL)	52
Table 30.	IRQ1 Enable and Priority Encoding	52
Table 31.	IRQ1 Enable Low Bit Register (IRQ1ENL)	53



### Use of All Uppercase Letters

The use of all uppercase letters designates the names of states and commands.

- Example 1: The bus is considered BUSY after the Start condition.
- Example 2: A START command triggers the processing of the initialization sequence.

### Bit Numbering

Bits are numbered from 0 to  $n-1$  where  $n$  indicates the total number of bits. For example, the 8 bits of a register are numbered from 0 to 7.

### Safeguards

It is important that all users understand the following safety terms, which are defined here.



**Caution:** Indicates a procedure or file may become corrupted if the user does not follow directions.

### Trademarks

ZiLOG, eZ8, Z8 Encore!, and Z8 are trademarks of [ZiLOG, Inc.](http://www.zilog.com) in the U.S.A. and other countries. All other trademarks are the property of their respective corporations.



**Table 2. Signal Descriptions (Continued)**

Signal Mnemonic	I/O	Description
<b>Reset</b>		
$\overline{\text{RESET}}$	I	RESET. Generates a Reset when asserted (driven Low).
<b>Power Supply</b>		
VDD	I	Power Supply.
AVDD	I	Analog Power Supply.
VSS	I	Ground.
AVSS	I	Analog Ground.

## Pin Characteristics

Table 3 provides detailed information on the characteristics for each pin available on the Z8F640x family products. Data in Table 3 is sorted alphabetically by the pin symbol mnemonic.

**Table 3. Pin Characteristics of the Z8F640x family**

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tri-State Output	Internal Pull-up or Pull-down	Schmitt Trigger Input	Open Drain Output
AVSS	N/A	N/A	N/A	N/A	No	No	N/A
AVDD	N/A	N/A	N/A	N/A	No	No	N/A
DBG	I/O	I	N/A	Yes	No	Yes	Yes
VSS	N/A	N/A	N/A	N/A	No	No	N/A
PA[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PB[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PC[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PD[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PE7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable

x represents integer 0, 1,... to indicate multiple pins with symbol mnemonics that differ only by the integer



Table 6. Register File Address Map (Continued)

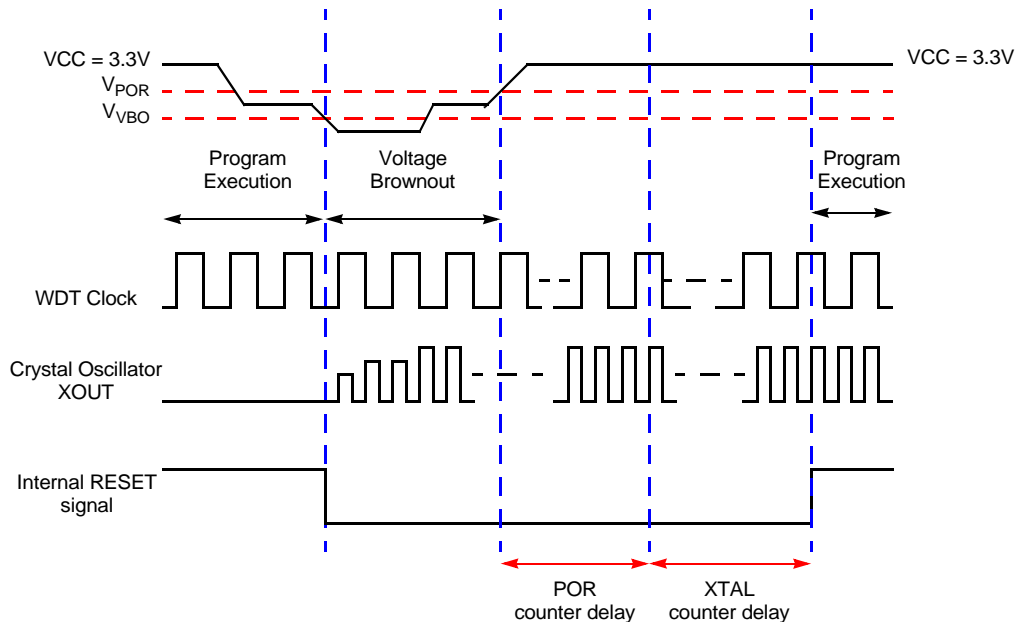
Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
F62	SPI Status	SPISTAT	01	108
F63	SPI Mode	SPIMODE	00	109
F64-F65	Reserved	—	XX	
F66	SPI Baud Rate High Byte	SPIBRH	FF	110
F67	SPI Baud Rate Low Byte	SPIBRL	FF	110
F68-F69	Reserved	—	XX	
<b>Analog-to-Digital Converter (ADC)</b>				
F70	ADC Control	ADCCTL	20	135
F71	Reserved	—	XX	
F72	ADC Data High Byte	ADCD_H	XX	137
F73	ADC Data Low Bits	ADCD_L	XX	137
F74-FAF	Reserved	—	XX	
<b>DMA 0</b>				
FB0	DMA0 Control	DMA0CTL	00	124
FB1	DMA0 I/O Address	DMA0IO	XX	125
FB2	DMA0 End/Start Address High Nibble	DMA0H	XX	126
FB3	DMA0 Start Address Low Byte	DMA0START	XX	127
FB4	DMA0 End Address Low Byte	DMA0END	XX	128
<b>DMA 1</b>				
FB8	DMA1 Control	DMA1CTL	00	124
FB9	DMA1 I/O Address	DMA1IO	XX	125
FBA	DMA1 End/Start Address High Nibble	DMA1H	XX	126
FBB	DMA1 Start Address Low Byte	DMA1START	XX	127
FBC	DMA1 End Address Low Byte	DMA1END	XX	128
<b>DMA ADC</b>				
FBD	DMA_ADC Address	DMAA_ADDR	XX	128
FBE	DMA_ADC Control	DMAACTL	00	130
FBF	DMA_ADC Status	DMAASTAT	00	131
<b>Interrupt Controller</b>				
FC0	Interrupt Request 0	IRQ0	00	48
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	51
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	51
FC3	Interrupt Request 1	IRQ1	00	49
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	52
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	52
FC6	Interrupt Request 2	IRQ2	00	50
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	53
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	53
FC9-FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	54

XX=Undefined

threshold voltage,  $V_{VBO}$ ) and forces the device into the Reset state. While the supply voltage remains below the Power-On Reset voltage threshold ( $V_{POR}$ ), the VBO block holds the Z8F640x family device in the Reset state.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the Z8F640x family device progresses through a full System Reset sequence, as described in the Power-On Reset section. Following Power-On Reset, the POR status bit in the Watch-Dog Timer Control (WDTCTL) register is set to 1. Figure 63 illustrates Voltage Brown-Out operation. Refer to the **Electrical Characteristics** chapter for the VBO and POR threshold voltages ( $V_{VBO}$  and  $V_{POR}$ ).

Stop mode disables the Voltage Brown-Out detector.



**Figure 63. Voltage Brown-Out Reset Operation (not to scale)**

### Watch-Dog Timer Reset

If the device is in normal or Halt mode, the Watch-Dog Timer can initiate a System Reset at time-out if the WDT\_RES Option Bit is set to 1. This is the default (unprogrammed) setting of the WDT\_RES Option Bit. The WDT status bit in the WDT Control register is set to signify that the reset was initiated by the Watch-Dog Timer.

**Table 33. 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	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	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 34. 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	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	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  
C1ENL—Port C1 Interrupt Request Enable Low Bit  
C0ENL—Port C0 Interrupt Request Enable Low Bit

## Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 35) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port input pin. The



## Interrupt Control Register

The Interrupt Control (IRQCTL) register (Table 37) contains the master enable bit for all interrupts.

**Table 37. Interrupt Control Register (IRQCTL)**

BITS	7	6	5	4	3	2	1	0
FIELD	IRQE	Reserved						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
ADDR	FCFH							

**IRQE—Interrupt Request Enable**

This bit is set to 1 by execution of an EI (Enable Interrupts) or IRET (Interrupt Return) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, or Reset.

0 = Interrupts are disabled.

1 = Interrupts are enabled.

**Reserved**

These bits must be 0.



# *Timers*

## Overview

The Z8F640x family products contain three to four 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse-width modulated (PWM) signals. The timers' features include:

- 16-bit reload counter
- Programmable prescaler with prescale values from 1 to 128
- PWM output generation
- Capture and compare capability
- External input pin for timer input, clock gating, or capture signal. External input pin signal frequency is limited to a maximum of one-fourth the system clock frequency.
- Timer output pin
- Timer interrupt

In addition to the timers described in this chapter, the Baud Rate Generators for any unused UART, SPI, or I<sup>2</sup>C peripherals may also be used to provide basic timing functionality. Refer to the respective serial communication peripheral chapters for information on using the Baud Rate Generators as timers. Timer 3 is unavailable in the 40- and 44-pin packages.

## Architecture

Figure 66 illustrates the architecture of the timers.

- 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 (usually 0001H). This only affects the first pass in Continuous mode. After the first timer Reload in Continuous 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. 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 Continuous mode, the system clock always provides the timer input. The timer period is given by the following equation:

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

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the One-Shot mode equation must be used to determine the first time-out period.

### Counter Mode

In Counter mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO Port pin Timer Input alternate function. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the Timer Input signal. In Counter mode, the prescaler is disabled.



**Caution:** The input frequency of the Timer Input signal must not exceed one-fourth the system clock frequency.

Upon reaching the Reload value stored in the Timer Reload High and Low Byte registers, 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) at timer Reload.

The steps for configuring a timer for Counter mode and initiating the count are as follows:

1. Write to the Timer Control register to:
  - Disable the timer
  - Configure the timer for Counter mode.



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. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

In Capture/Compare mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

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

## Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte register is read, the contents of the Timer Low Byte register are placed in a holding register. A subsequent read from the Timer Low Byte register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte register returns the actual value in the counter.

## Timer Output Signal Operation

Timer Output is a GPIO Port pin alternate function. Generally, the Timer Output is toggled every time the counter is reloaded.

## Timer Control Register Definitions

Timers 0–2 are available in all packages. Timer 3 is available only in the 64-, 68- and 80-pin packages.

## Timer 0-3 High and Low Byte Registers

The Timer 0-3 High and Low Byte (TxH and TxL) registers (Tables 38 and 39) contain the current 16-bit timer count value. When the timer is enabled, a read from TxH causes the value in TxL to be stored in a temporary holding register. A read from TMRL always returns this temporary register when the timers are enabled. When the timer is disabled, reads from the TMRL reads the register directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations, so simultaneous 16-bit writes are not possible. If either the Timer High or Low Byte registers are

written during counting, the 8-bit written value is placed in the counter (High or Low Byte) at the next clock edge. The counter continues counting from the new value.

**Table 38. Timer 0-3 High Byte Register (TxH)**

BITS	7	6	5	4	3	2	1	0
FIELD	TH							
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	F00H, F08H, F10H, F18H							

**Table 39>. Timer 0-3 Low Byte Register (TxL)**

BITS	7	6	5	4	3	2	1	0
FIELD	TL							
RESET	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F01H, F09H, F11H, F19H							

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 0-3 Reload High and Low Byte (TxRH and TxRL) registers (Tables 40 and 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.

In single-byte DMA transactions to the Timer Reload High Byte register, the temporary holding register is bypassed and the value is written directly to the register. If the DMA is

Middle byte, Bits[15:8], of the 24-bit WDT reload value.

**Table 49. Watch-Dog Timer Reload Low Byte Register (WDTL)**

BITS	7	6	5	4	3	2	1	0
FIELD	WDTL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
ADDR	FF3H							
R/W* - Read returns the current WDT count value. Write sets the desired Reload Value.								

WDTL—WDT Reload Low

Least significant byte (LSB), Bits[7:0], of the 24-bit WDT reload value.



# *UART*

## Overview

The Universal Asynchronous Receiver/Transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The Z8F640x family device contains two fully independent UARTs. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two Stop bits
- Separate transmit and receive interrupts
- Framing, parity, overrun and break detection
- Separate transmit and receive enables
- Selectable 9-bit multiprocessor (9-bit) mode
- 16-bit Baud Rate Generator (BRG)

## Architecture

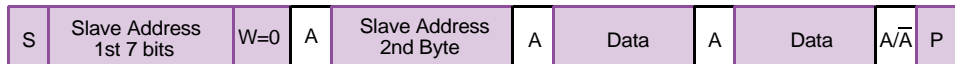
The UART consists of three primary functional blocks: transmitter, receiver, and baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. Figure 67 illustrates the UART architecture.

14. Software responds by setting the STOP bit of the I<sup>2</sup>C Control register.
15. If no new data is to be sent or address is to be sent, software responds by clearing the TXI bit of the I<sup>2</sup>C Control register.
16. The I<sup>2</sup>C Controller completes transmission of the data on the SDA signal.
17. The I<sup>2</sup>C Controller sends the STOP condition to the I<sup>2</sup>C bus.

### Writing a Transaction with a 10-Bit Address

1. The I<sup>2</sup>C Controller shifts the I<sup>2</sup>C Shift register out onto SDA signal.
2. The I<sup>2</sup>C 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 I<sup>2</sup>C Controller sends a Stop signal.
3. If the slave needs to service an interrupt, it pulls the SCL signal low, which halts I<sup>2</sup>C operation.
4. If there is no other data in the I<sup>2</sup>C Data register or the STOP bit in the I<sup>2</sup>C Control register is set by software, then the Stop signal is sent.

The data transfer format for a 10-bit addressed slave is illustrated in the figure below. Shaded regions indicate data transferred from the I<sup>2</sup>C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I<sup>2</sup>C Controller.



**Figure 80. 10-Bit Addressed Slave Data Transfer Format**

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 transmit operation is carried out in the same manner as 7-bit addressing.

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

1. Software asserts the IEN bit in the I<sup>2</sup>C Control register.
2. Software asserts the TXI bit of the I<sup>2</sup>C Control register to enable Transmit interrupts.
3. The I<sup>2</sup>C interrupt asserts because the I<sup>2</sup>C Data register is empty.
4. Software responds to the TDRE bit by writing the first slave address byte. The least-significant bit must be 0 for the write operation.
5. Software asserts the START bit of the I<sup>2</sup>C Control register.
6. The I<sup>2</sup>C Controller sends the START condition to the I<sup>2</sup>C slave.



When the DMA is configured for two-byte word transfers, the DMAx I/O Address register must contain an even numbered address.

**Table 72. DMAx I/O Address Register (DMAxIO)**

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_IO							
RESET	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FB1H, FB9H							

DMA\_IO—DMA on-chip peripheral control register address

This byte sets the low byte of the on-chip peripheral control register address on Register File Page FH (addresses F00H to FFFH).

### DMAx Address High Nibble Register

The DMAx Address High register specifies the upper four bits of address for the Start/Current and End Addresses of DMAx.

**Table 73. DMAx Address High Nibble Register (DMAxH)**

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_END_H				DMA_START_H			
RESET	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FB2H, FHAH							

DMA\_END\_H—DMAx End Address High Nibble

These bits, used with the DMAx End Address Low register, form a 12-bit End Address. The full 12-bit address is given by {DMA\_END\_H[3:0], DMA\_END[7:0]}.

DMA\_START\_H—DMAx Start/Current Address High Nibble

These bits, used with the DMAx Start/Current Address Low register, form a 12-bit Start/Current Address. The full 12-bit address is given by {DMA\_START\_H[3:0], DMA\_START[7:0]}.

1. Enable the desired analog inputs by configuring the general-purpose I/O pins for alternate function. This configuration disables the digital input and output drivers.
2. Write to the ADC Control register to configure the ADC and begin the conversion. The bit fields in the ADC Control register can be written simultaneously:
  - Write to `ANAIN[3 : 0]` to select one of the 12 analog input sources.
  - Clear `CONT` to 0 to select a single-shot conversion.
  - Write to `VREF` to enable or disable the internal voltage reference generator.
  - Set `CEN` to 1 to start the conversion.
3. `CEN` remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power-up before beginning the 5129 cycle conversion.
4. When the conversion is complete, the ADC control logic performs the following operations:
  - 10-bit data result written to `{ADCD_H[7:0], ADCD_L[7:6]}`.
  - `CEN` resets to 0 to indicate the conversion is complete.
  - An interrupt request is sent to the Interrupt Controller.
5. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

## Continuous Conversion

When configured for continuous conversion, the ADC continuously performs an analog-to-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated only at the end of the first conversion after enabling.



**Caution:** In Continuous mode, users must be aware that ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not seen at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.

The steps for setting up the ADC and initiating continuous conversion are as follows:

1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This disables the digital input and output driver.
2. Write to the ADC Control register to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
  - Write to `ANAIN[3 : 0]` to select one of the 12 analog input sources.



ister. When the Watchpoint event occurs, the Z8F640x family device enters Debug mode and the DBGMODE bit in the OCDCTL register becomes 1.

## Runtime Counter

The On-Chip Debugger contains a 16-bit Runtime Counter. It counts system clock cycles between Breakpoints. The counter starts counting when the On-Chip Debugger leaves Debug mode and stops counting when it enters Debug mode again or when it reaches the maximum count of FFFFH.

## On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation of the Z8F640x family device, 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 Z8F640x family device. When this option is enabled, several of the OCD commands are disabled. Table 93 contains a summary of the On-Chip Debugger commands. Each OCD command is described in further detail in the bulleted list following Table 93. Table 93 indicates those commands that operate when the Z8F640x family device is not in Debug mode (normal operation) and those commands that are disabled by programming the Read Protect Option Bit.

**Table 93. On-Chip Debugger Commands**

Debug Command	Command Byte	Enabled when NOT in Debug mode?	Disabled by Read Protect Option Bit
Read OCD Revision	00H	Yes	-
Reserved	01H	-	-
Read OCD Status Register	02H	Yes	-
Read Runtime Counter	03H	-	-
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit
Read OCD Control Register	05H	Yes	-
Write Program Counter	06H	-	Disabled
Read Program Counter	07H	-	Disabled
Write Register	08H	-	Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control register.
Read Register	09H	-	Disabled



- **Read Data Memory (0DH)**—The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the Z8F640x family device is not in Debug mode, this command returns FFH for the data.

```
DBG <-- 0DH
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
```

- **Read Program Memory CRC (0EH)**—The Read Program Memory CRC command computes and returns the CRC (cyclic redundancy check) of Program Memory using the 16-bit CRC-CCITT polynomial. If the Z8F640x family device is not in Debug mode, this command returns FFFFH for the CRC value. Unlike most other OCD Read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads the Program Memory, calculates the CRC value, and returns the result. The delay is a function of the Program Memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Program Memory.

```
DBG <-- 0EH
DBG --> CRC [15:8]
DBG --> CRC [7:0]
```

- **Step Instruction (10H)**—The Step Instruction command steps one assembly instruction at the current Program Counter (PC) location. If the Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, the OCD ignores this command.

```
DBG <-- 10H
```

- **Stuff Instruction (11H)**—The Stuff Instruction command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0-4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, the OCD ignores this command.

```
DBG <-- 11H
DBG <-- opcode [7:0]
```

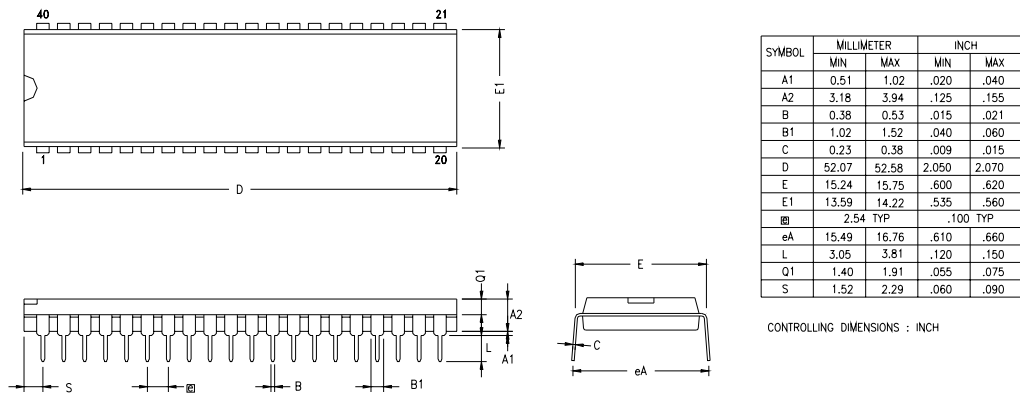
- **Execute Instruction (12H)**—The Execute Instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over Breakpoints. The number of bytes to send for the instruction depends on the opcode. If the Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, this command reads and discards one byte.

```
DBG <-- 12H
DBG <-- 1-5 byte opcode
```



## Packaging

Figure 103 illustrates the 40-pin PDIP (plastic dual-inline package) available for the Z8F1601, Z8F2401, Z8F3201, Z8F4801, and Z8F6401 devices.



**Figure 103. 40-Lead Plastic Dual-Inline Package (PDIP)**