

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.

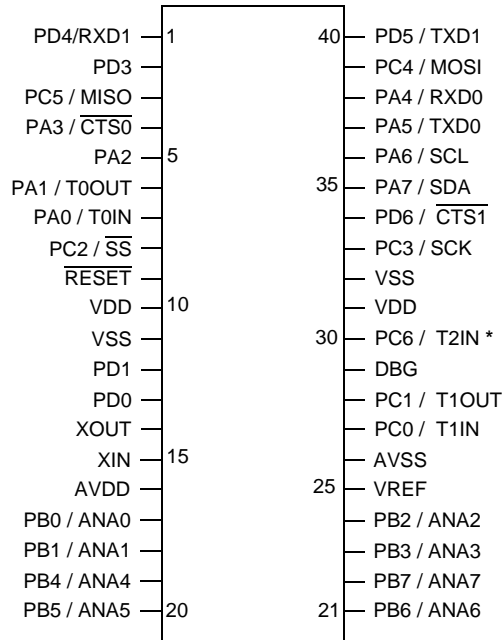
### Applications of "[Embedded - Microcontrollers](#)"

#### 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	31
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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f6401an020ec">https://www.e-xfl.com/product-detail/zilog/z8f6401an020ec</a>

## Pin Configurations

Figures 56 through 61 illustrate the pin configurations for all of the packages available in the Z8 Encore!® MCU family. Refer to Table 2 for a description of the signals.



**Note:** Timer 3 is not supported.

\* T2OUT is not supported.

**Figure 56. Z8Fxx01 in 40-Pin Dual Inline Package (DIP)**

## Power-On Reset

The Z8F640x family products contain an internal Power-On Reset (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 514 cycles of the Watch-Dog Timer oscillator. After the POR counter times out, the XTAL Counter is enabled to count a total of 16 system clock pulses. The Z8F640x family device is held in the Reset state until both the POR Counter and XTAL counter have timed out. After the device exits the Power-On Reset state, the eZ8 CPU fetches the Reset vector. Following Power-On Reset, the POR status bit in the Watch-Dog Timer Control (WDTCTL) register is set to 1.

Figure 62 illustrates Power-On Reset operation. Refer to the **Electrical Characteristics** chapter for the POR threshold voltage ( $V_{POR}$ ).

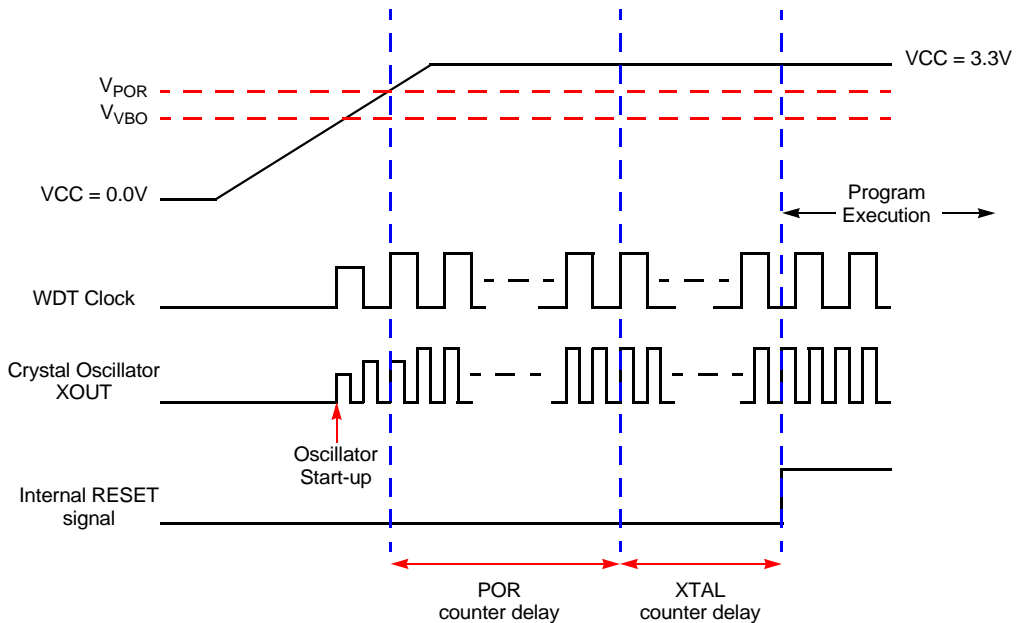


Figure 62. Power-On Reset Operation (not to scale)

## Voltage Brown-Out Reset

The devices in the Z8F640x family provide low Voltage Brown-Out (VBO) protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO

## Port A-H Output Data Register

The Port A-H Output Data register (Table 21) writes output data to the pins.

**Table 21. Port A-H Output Data Register (PxOUT)**

<b>BITS</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>FIELD</b>	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
<b>RESET</b>	0	0	0	0	0	0	0	0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>ADDR</b>	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

**POUT[7:0]**—Port Output Data

These bits contain the data to be driven out from the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation.

0 = Drive a logical 0 (Low).

1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control register bit to 1.

Table 30. IRQ1 Enable High Bit Register (IRQ1ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	PAD7ENH	PAD6ENH	PAD5ENH	PAD4ENH	PAD3ENH	PAD2ENH	PAD1ENH	PAD0ENH
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	FC4H							

PADxENH—Port A or Port D Bit[x] Interrupt Request Enable High Bit  
Refer to the Interrupt Port Select register for selection of either Port A or Port D as the interrupt source.

Table 31. IRQ1 Enable Low Bit Register (IRQ1ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	PAD7ENL	PAD6ENL	PAD5ENL	PAD4ENL	PAD3ENL	PAD2ENL	PAD1ENL	PAD0ENL
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	FC5H							

PADxENL—Port A or Port D Bit[x] Interrupt Request Enable Low Bit  
Refer to the Interrupt Port Select register for selection of either Port A or Port D as the interrupt source.

## IRQ2 Enable High and Low Bit Registers

The IRQ2 Enable High and Low Bit registers (Tables 33 and 34) form a priority encoded enabling for interrupts in the Interrupt Request 2 register. Priority is generated by setting bits in each register. Table 32 describes the priority control for IRQ2.

Table 32. IRQ2 Enable and Priority Encoding

IRQ2ENH[x]	IRQ2ENL[x]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

where  $x$  indicates the register bits from 0 through 7.

## 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.

## Timer 0-3 PWM High and Low Byte Registers

The Timer 0-3 PWM High and Low Byte (TxPWMH and TxPWML) registers (Tables 42 and 43) are used for Pulse-Width Modulator (PWM) operations. These registers also store the Capture values for the Capture and Capture/Compare modes.

**Table 42. Timer 0-3 PWM High Byte Register (TxPWMH)**

BITS	7	6	5	4	3	2	1	0
FIELD	PWMH							
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	F04H, F0CH, F14H, F1CH							

**Table 43. Timer 0-3 PWM Low Byte Register (TxPWML)**

BITS	7	6	5	4	3	2	1	0
FIELD	PWML							
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	F05H, F0DH, F15H, F1DH							

**PWMH and PWML—Pulse-Width Modulator High and Low Bytes**

These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control Register (TxCTL) register.

The TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in Capture or Capture/Compare modes.



5. Check the TDRE bit in the UART Status 0 register to determine if the Transmit Data register is empty (indicated by a 1). If empty, continue to Step 6. If the Transmit Data register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data register becomes available to receive new data.
6. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmit the data.
7. To transmit additional bits, return to Step 5.

### Transmitting Data using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data register to accept new data for transmission. Follow these steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the desired baud rate.
2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the desired priority.
5. Write to the UART Control 1 register to enable Multiprocessor (9-bit) mode functions, if desired.
6. Write to the UART Control 0 register to:
  - Set the transmit enable bit (TEN) to enable the UART for data transmission
  - Enable parity, if desired, and select either even or odd parity.
  - Set or clear the CTSE bit to enable or disable control from the receiver via the CTS pin.
7. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data transmission. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) should perform the following:

8. Write the data byte to the UART Transmit Data register. The transmitter will automatically transfer the data to the Transmit Shift register and transmit the data.
9. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
10. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.



mitter and receiver sections, a Baud Rate (clock) Generator and a control unit. The transmitter and receiver sections use the same clock.

During an SPI transfer, data is sent and received simultaneously by both the Master and the Slave SPI devices. Separate signals are required for data and the serial clock. When an SPI transfer occurs, a multi-bit (typically 8-bit) character is shifted out one data pin and a multi-bit character is simultaneously shifted in on a second data pin. An 8-bit shift register in the Master and another 8-bit shift register in the Slave are connected as a circular buffer. The SPI shift register is single-buffered in the transmit and receive directions. New data to be transmitted cannot be written into the shift register until the previous transmission is complete and receive data (if valid) has been read.

## SPI Signals

The four basic SPI signals are:

- MISO (Master-In, Slave-Out)
- MOSI (Master-Out, Slave-In)
- SCK (SPI Serial Clock)
- $\overline{SS}$  (Slave Select)

The following paragraphs discuss these SPI signals. Each signal is described in both Master and Slave modes.

### Master-In, Slave-Out

The Master-In, Slave-Out (MISO) pin is configured as an input in a Master device and as an output in a Slave device. It is one of the two lines that transfer serial data, with the most significant bit sent first. The MISO pin of a Slave device is placed in a high-impedance state if the Slave is not selected. When the SPI is not enabled, this signal is in a high-impedance state.

### Master-Out, Slave-In

The Master-Out, Slave-In (MOSI) pin is configured as an output in a Master device and as an input in a Slave device. It is one of the two lines that transfer serial data, with the most significant bit sent first. When the SPI is not enabled, this signal is in a high-impedance state.

### Serial Clock

The Serial Clock (SCK) is used to synchronize data movement both in and out of the device through its MOSI and MISO pins. In Master mode, the SPI's Baud Rate Generator creates the serial clock. The Master drives the serial clock out its own SCK pin to the Slave's SCK pin. When the SPI is configured as a Slave, the SCK pin is an input and the clock signal from the Master synchronizes the data transfer between the Master and Slave devices. Slave devices ignore the SCK signal, unless the  $\overline{SS}$  pin is asserted.

## SPI Mode Register

The SPI Mode register configures the character bit width and the direction and value of the  $\overline{SS}$  pin.

**Table 63. SPI Mode Register (SPIMODE)**

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved			NUMBITS[2:0]			SSIO	SSV
RESET	0			0	0	0	0	0
R/W	R			R/W	R/W	R/W	R/W	R/W
ADDR	F63H							

Reserved

These bits are reserved and must be 0.

NUMBITS[2:0]—Number of Data Bits Per Character to Transfer

This field contains the number of bits to shift for each character transfer. Refer to the SPI Data Register description for information on valid bit positions when the character length is less than 8-bits.

000 = 8 bits  
001 = 1 bit  
010 = 2 bits  
011 = 3 bits  
100 = 4 bits  
101 = 5 bits  
110 = 6 bits  
111 = 7 bits.

SSIO—Slave Select I/O

0 =  $\overline{SS}$  pin configured as an input.

1 =  $\overline{SS}$  pin configured as an output (Master mode only).

SSV—Slave Select Value

If SSIO = 1 and SPI configured as a Master:

0 =  $\overline{SS}$  pin driven Low (0).

1 =  $\overline{SS}$  pin driven High (1).

This bit has no effect if SSIO = 0 or SPI configured as a Slave.



4. The I<sup>2</sup>C Controller loads the I<sup>2</sup>C Shift register with the contents of the I<sup>2</sup>C Data register.
5. After the first bit has been shifted out, a Transmit interrupt is asserted.
6. Software responds by writing eight bits of address to the I<sup>2</sup>C Data register.
7. The I<sup>2</sup>C Controller completes shifting of the two address bits and a 0 (write).
8. The I<sup>2</sup>C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL.
9. The I<sup>2</sup>C Controller loads the I<sup>2</sup>C Shift register with the contents of the I<sup>2</sup>C Data register.
10. The I<sup>2</sup>C Controller shifts out the next eight bits of address. After the first bits are shifted, the I<sup>2</sup>C Controller generates a Transmit interrupt.
11. Software responds by setting the START bit of the I<sup>2</sup>C Control register to generate a repeated START.
12. Software responds by writing 11110B followed by the 2-bit slave address and a 1 (read).
13. Software responds by setting the NAK bit of the I<sup>2</sup>C Control register, so that a Not Acknowledge is sent after the first byte of data has been read. If you want to read only one byte, software responds by setting the NAK bit of the I<sup>2</sup>C Control register.
14. After the I<sup>2</sup>C Controller shifts out the address bits mentioned in step 9, the I<sup>2</sup>C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL.
15. The I<sup>2</sup>C Controller sends the repeated START condition.
16. The I<sup>2</sup>C Controller loads the I<sup>2</sup>C Shift register with the contents of the I<sup>2</sup>C Data register.
17. The I<sup>2</sup>C Controller sends 11110B followed by the 2-bit slave read and a 1 (read).
18. The I<sup>2</sup>C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL.
19. The I<sup>2</sup>C slave sends a byte of data.
20. A Receive interrupt is generated.
21. Software responds by reading the I<sup>2</sup>C Data register.
22. Software responds by setting the STOP bit of the I<sup>2</sup>C Control register.
23. A NAK condition is sent to the I<sup>2</sup>C slave.
24. A STOP condition is sent 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]}.

## ADC Data High Byte Register

The ADC Data High Byte register contains the upper eight bits of the 10-bit ADC output. During a conversion, this value is invalid. Access to the ADC Data High Byte register is read-only. The full 10-bit ADC result is given by {ADCD\_H[7:0], ADCD\_L[7:6]}.

**Table 81. ADC Data High Byte Register (ADCD\_H)**

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_H							
RESET	X							
R/W	R							
ADDR	F72H							

ADCD\_H—ADC Data High Byte

This byte contains the upper eight bits of the 10-bit ADC output. These bits are not valid during a conversion. These bits are undefined after a Reset.

## ADC Data Low Bits Register

The ADC Data Low Bits register contains the lower two bits of the conversion value. During a conversion this value is invalid. Access to the ADC Data Low Bits register is read-only. The full 10-bit ADC result is given by {ADCD\_H[7:0], ADCD\_L[7:6]}.

**Table 82. ADC Data Low Bits Register (ADCD\_L)**

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_L		Reserved					
RESET	X		X					
R/W	R		R					
ADDR	F73H							

ADCD\_L—ADC Data Low Bits

These are the least significant two bits of the 10-bit ADC output. During a conversion, this value is invalid. These bits are undefined after a Reset.

Reserved

These bits are reserved and are always undefined.



- **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
```

RPEN—Read Protect Option Bit Enabled

0 = The Read Protect Option Bit is disabled (1).

0 = The Read Protect Option Bit is enabled (0), disabling many OCD commands.

Reserved

These bits are always 0.

## OCD Watchpoint Control Register

The OCD Watchpoint Control register is used to configure the debug Watchpoint.

**Table 96. OCD Watchpoint Control/Address (WPTCTL)**

BITS	7	6	5	4	3	2	1	0
FIELD	WPW	WPR	WPDM	Reserved	WPTADDR[11:8]			
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

WPW—Watchpoint Break on Write

This bit cannot be set if the Read Protect Option Bit is enabled.

0 = Watchpoint Break on Register File write is disabled.

1 = Watchpoint Break on Register File write is enabled.

WPR—Watchpoint Break on Read

This bit cannot be set if the Read Protect Option Bit is enabled.

0 = Watchpoint Break on Register File read is disabled.

1 = Watchpoint Break on Register File write is enabled.

WPDM—Watchpoint Data Match

If this bit is set, then the Watchpoint only generates a Debug Break if the data being read or written matches the specified Watchpoint data. Either the WPR and/or WPW bits must also be set for this bit to affect operation. This bit cannot be set if the Read Protect Option Bit is enabled.

0 = Watchpoint Break on read and/or write does not require a data match.

1 = Watchpoint Break on read and/or write requires a data match.

Reserved

This bit is reserved and must be 0.

RADDR[11:8]—Register address

These bits specify the upper 4 bits of the Register File address to match when generating a Watchpoint Debug Break. The full 12-bit Register File address is given by {WPTCTL3:0, WPTADDR[7:0]}.

## SPI Slave Mode Timing

Figure 97 and Table 111 provide timing information for the SPI slave mode pins. Timing is shown with SCK rising edge used to source MISO output data, SCK falling edge used to sample MOSI input data.

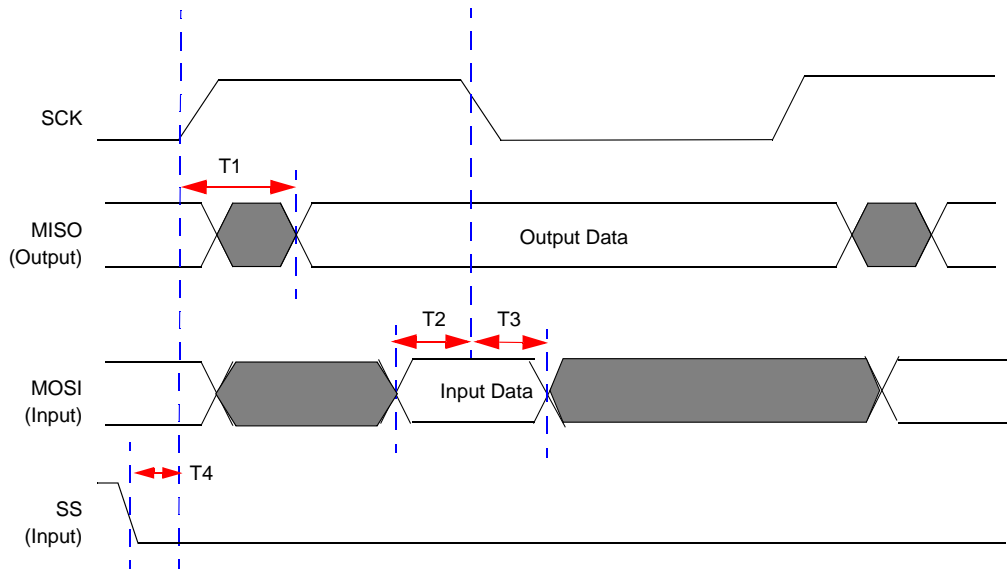


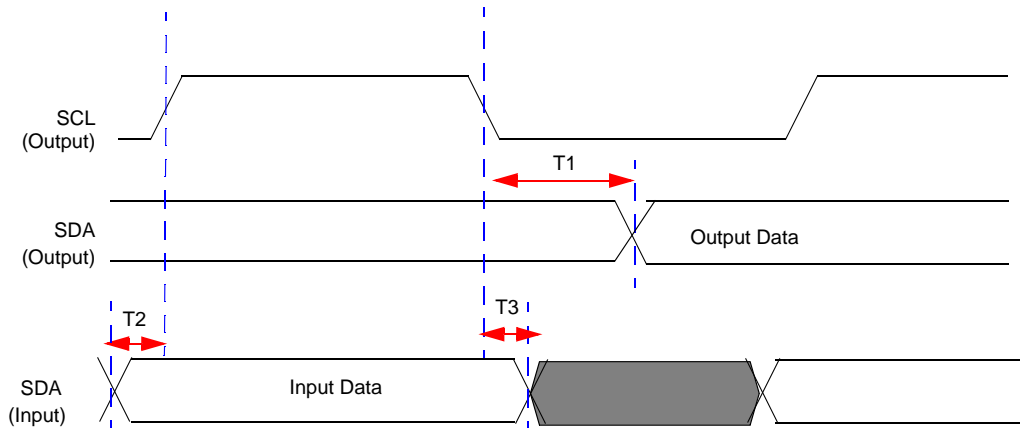
Figure 97. SPI Slave Mode Timing

Table 111. SPI Slave Mode Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
T <sub>1</sub>	SCK (transmit edge) to MISO output Valid Delay	2 * X <sub>in</sub> period	3 * X <sub>in</sub> period + 20 nsec
T <sub>2</sub>	MOSI input to SCK (receive edge) Setup Time	0	
T <sub>3</sub>	MOSI input to SCK (receive edge) Hold Time	3 * X <sub>in</sub> period	
T <sub>4</sub>	SS input assertion to SCK setup	1 * X <sub>in</sub> period	

## I<sup>2</sup>C Timing

Figure 98 and Table 112 provide timing information for I<sup>2</sup>C pins.



**Figure 98. I<sup>2</sup>C Timing**

**Table 112. I<sup>2</sup>C Timing**

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
T <sub>1</sub>	SCL Fall to SDA output delay		SCL period/4
T <sub>2</sub>	SDA Input to SCL rising edge Setup Time	0	
T <sub>3</sub>	SDA Input to SCL falling edge Hold Time	0	

## 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 is used to decide if the conditional jump is executed.

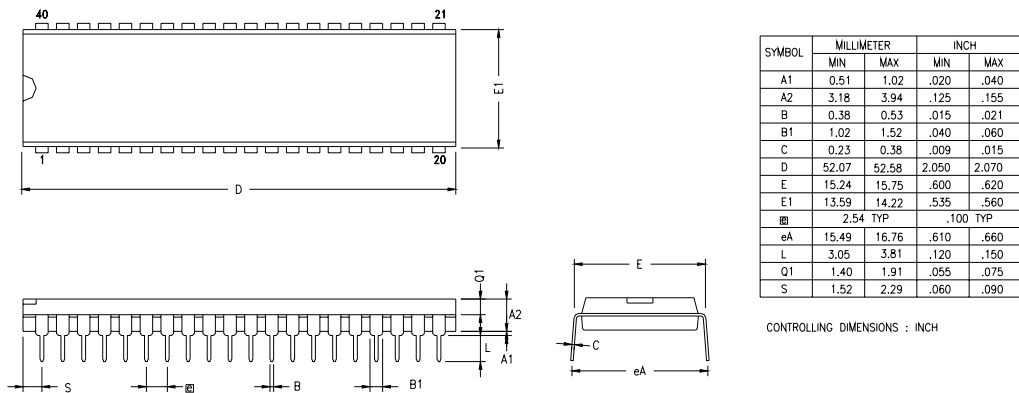
**Table 117. Condition Codes**

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



## 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)**

Figure 104 illustrates the 44-pin LQFP (low profile quad flat package) available for the Z8F1601, Z8F2401, Z8F3201, Z8F4801, and Z8F6401 devices.

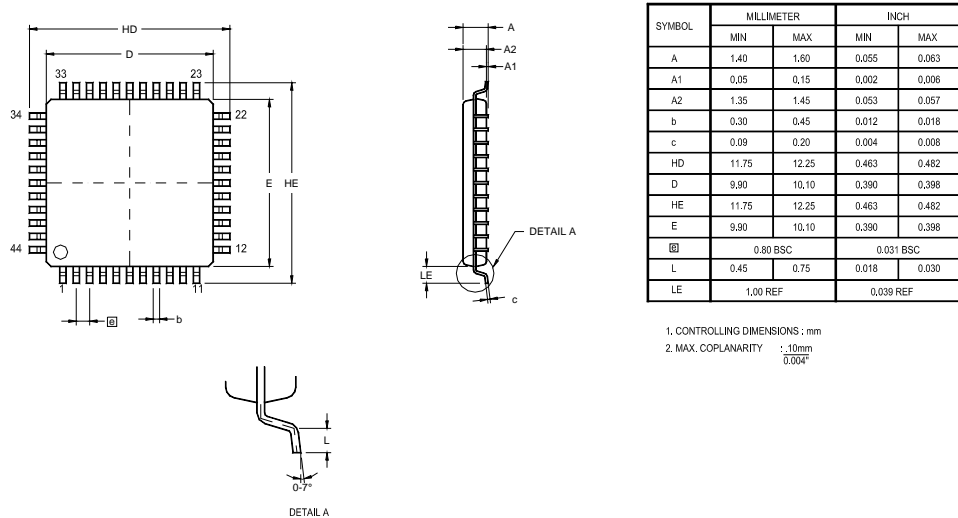


Figure 104. 44-Lead Low-Profile Quad Flat Package (LQFP)

Figure 105 illustrates the 44-pin PLCC (plastic lead chip carrier) package available for the Z8F1601, Z8F2401, Z8F3201, Z8F4801, and Z8F6401 devices.

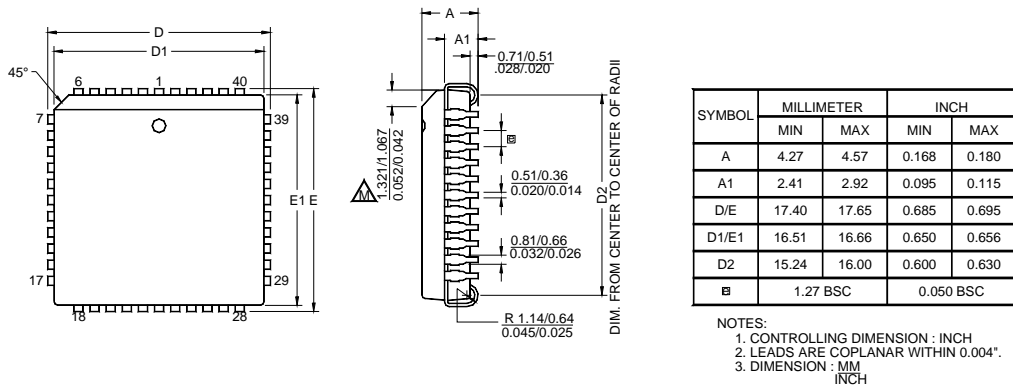
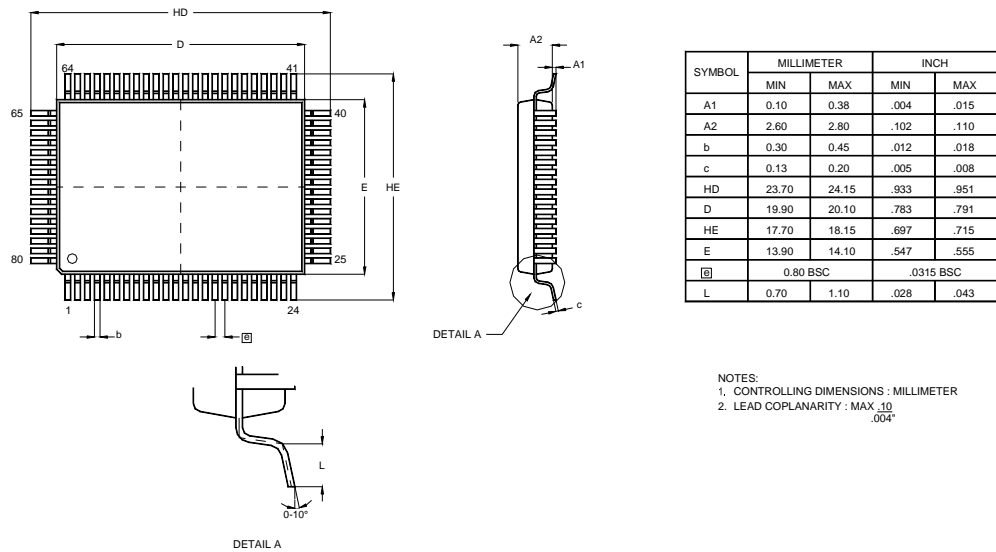


Figure 105. 44-Lead Plastic Lead Chip Carrier Package (PLCC)

Figure 108 illustrates the 80-pin QFP (quad flat package) available for the Z8F4803 and Z8F6403 devices.



**Figure 108. 80-Lead Quad-Flat Package (QFP)**