



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 ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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	https://www.e-xfl.com/product-detail/zilog/z8f3201an020ec00tr

Table 4. Z8F640x Family Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F480x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-BFFFH	Program Memory
Z8F640x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-FFFFH	Program Memory
* See Table 22 on page 45 for a list of the interrupt vectors.	

Data Memory

The Z8F640x family devices contain 128 bytes of read-only memory at the top of the eZ8 CPU's 64KB Data Memory address space. The eZ8 CPU's LDE and LDEI instructions provide access to the Data Memory information. Table 5 describes the Z8F640x family's Data Memory Map.

Table 5. Z8F640x family Data Memory Maps

Data Memory Address (Hex)	Function
0000H-FFBFH	Reserved
FFC0H-FFD3H	Part Number 20-character ASCII alphanumeric code Left justified and filled with zeros
FFD4H-FFFFH	Reserved

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 15).

Table 15. 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	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	If 01H in Port A-H Address Register, accessible via 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 16) 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. Refer to the **GPIO Alternate Functions** section to determine the alternate function associated with each port pin.



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 16. 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	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	If 02H in Port A-H Address Register, accessible via Port A-H Control Register							

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) 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 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 25. 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	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	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.

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.

where x indicates the specific GPIO Port C pin number (0 through 3).

IRQ0 Enable High and Low Bit Registers

The IRQ0 Enable High and Low Bit registers (Tables 27 and 28) form a priority encoded enabling for interrupts in the Interrupt Request 0 register. Priority is generated by setting bits in each register. Table 26 describes the priority control for IRQ0.

Table 26. IRQ0 Enable and Priority Encoding

IRQ0ENH[x]	IRQ0ENL[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.

Table 27. IRQ0 Enable High Bit Register (IRQ0ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	T2ENH	T1ENH	T0ENH	U0RENH	U0TENH	I2CENH	SPIENH	ADCENH
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	FC1H							

T2ENH—Timer 2 Interrupt Request Enable High Bit

T1ENH—Timer 1 Interrupt Request Enable High Bit

T0ENH—Timer 0 Interrupt Request Enable High Bit

U0RENH—UART 0 Receive Interrupt Request Enable High Bit

U0TENH—UART 0 Transmit Interrupt Request Enable High Bit

I2CENH—I²C Interrupt Request Enable High Bit

SPIENH—SPI Interrupt Request Enable High Bit

ADCENH—ADC Interrupt Request Enable High Bit



If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting.

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

1. Write to the Timer Control register to:
 - Disable the timer
 - Configure the timer for Compare mode.
 - Set the prescale value.
 - Set the initial logic level (High or Low) for the Timer Output alternate function, if desired.
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 Compare 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 Compare mode, the system clock always provides the timer input. The Compare time is given by the following equation:

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

Gated Mode

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

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

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

1. Write to the Timer Control register to:
 - Disable the timer



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

Capture/Compare Mode

In Capture/Compare mode, the timer begins counting on the *first* external Timer Input transition. The desired transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent desired transition (after the first) of the Timer Input signal captures the current count value. The Capture value is written to the Timer PWM High and Low Byte Registers. When the Capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H, and counting resumes.

If no Capture event occurs, the timer counts up to the 16-bit Compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the Compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

The steps for configuring a timer for Capture/Compare mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer
 - Configure the timer for Capture/Compare mode.
 - Set the prescale value.
 - Set the Capture edge (rising or falling) for the Timer Input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the Compare value.
4. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.



Watch-Dog Timer

Overview

The Watch-Dog Timer (WDT) helps protect against corrupt or unreliable software, power faults, and other system-level problems which may place the Z8 Encore!® into unsuitable operating states. The Watch-Dog Timer includes the following features:

- On-chip RC oscillator
- A selectable time-out response: Short Reset or interrupt
- 24-bit programmable time-out value

Operation

The Watch-Dog Timer (WDT) is a retriggerable one-shot timer that resets or interrupts the Z8F640x family device when the WDT reaches its terminal count. The Watch-Dog Timer uses its own dedicated on-chip RC oscillator as its clock source. The Watch-Dog Timer has only two modes of operation—on and off. Once enabled, it always counts and must be refreshed to prevent a time-out. An enable can be performed by executing the WDT instruction or by setting the WDT_AO Option Bit. The WDT_AO bit enables the Watch-Dog Timer to operate all the time, even if a WDT instruction has not been executed.

The Watch-Dog Timer is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is given by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{50}$$

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTM[7:0], WDTL[7:0]} and the typical Watch-Dog Timer RC oscillator frequency is 50kHz. The Watch-Dog Timer cannot be refreshed once it reaches 000002H. The WDT Reload Value must not be set to values below 000004H. Table 45 provides

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.



CTSE—CTS Enable

0 = The $\overline{\text{CTS}}$ signal has no effect on the transmitter.

1 = The UART recognizes the $\overline{\text{CTS}}$ signal as an enable control from the transmitter.

PEN—Parity Enable

This bit enables or disables parity. Even or odd is determined by the PSEL bit.

0 = Parity is disabled.

1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.

PSEL—Parity Select

0 = Even parity is transmitted and expected on all received data.

1 = Odd parity is transmitted and expected on all received data.

SBRK—Send Break

This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so insure that the transmitter has finished sending data before setting this bit.

0 = No break is sent.

1 = The output of the transmitter is zero.

STOP—Stop Bit Select

0 = The transmitter sends one stop bit.

1 = The transmitter sends two stop bits.

LBEN—Loop Back Enable

0 = Normal operation.

1 = All transmitted data is looped back to the receiver.

Table 55. UARTx Control 1 Register (UxCTL1)

BITS	7	6	5	4	3	2	1	0
FIELD	BIRQ	MPM	MPE	MPBT	Reserved		$\overline{\text{RDAIRQ}}$	IREN
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	F43H and F4BH							

BIRQ—Baud Rate Generator Interrupt Request

This bit sets an interrupt request when the Baud Rate Generator times out and is only set if a UART is not enabled. The is bit produces no effect when the UART is enabled.

0 = Interrupts behave as set by UART control.

1 = The Baud Rate Generator generates a receive interrupt when it counts down to zero.

MPM—Multiprocessor (9-bit) mode Select

This bit is used to enable Multiprocessor (9-bit) mode.

Infrared Encoder/Decoder

Overview

The Z8F640x family products contain two fully-functional, high-performance UART to Infrared Encoder/Decoders (Endecs). Each Infrared Endec is integrated with an on-chip UART to allow easy communication between the Z8F640x family device and IrDA Physical Layer Specification Version 1.3-compliant infrared transceivers. Infrared communication provides secure, reliable, low-cost, point-to-point communication between PCs, PDAs, cell phones, printers and other infrared enabled devices.

Architecture

Figure 71 illustrates the architecture of the Infrared Endec.

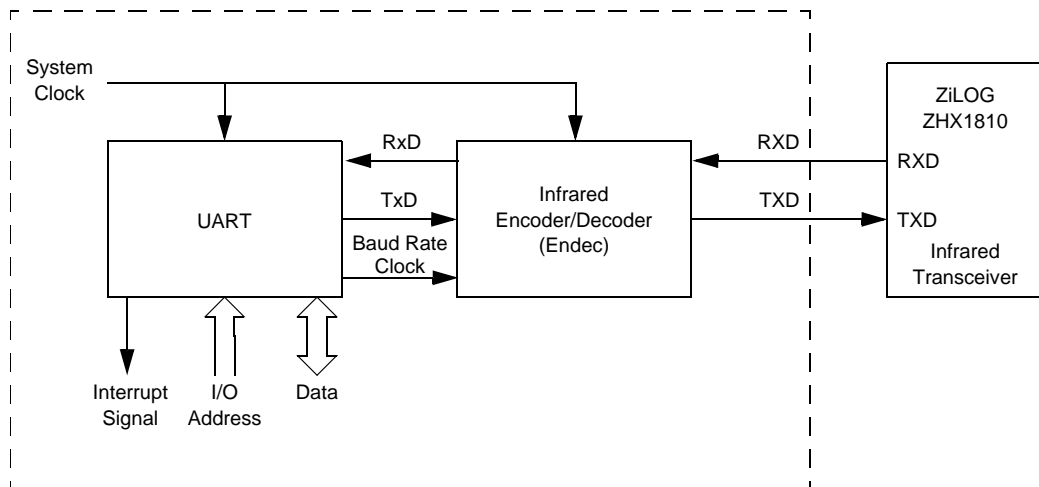


Figure 71. Infrared Data Communication System Block Diagram

SPI Control Register

The SPI Control register configures the SPI for transmit and receive operations.

Table 61. SPI Control Register (SPICTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN
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	F61H							

IRQE—Interrupt Request Enable

0 = SPI interrupts are disabled. No interrupt requests are sent to the Interrupt Controller.

1 = SPI interrupts are enabled. Interrupt requests are sent to the Interrupt Controller.

STR—Start an SPI Interrupt Request

0 = No effect.

1 = Setting this bit to 1 also sets the **IRQ** bit in the SPI Status register to 1. Setting this bit forces the SPI to send an interrupt request to the Interrupt Control. This bit can be used by software for a function similar to transmit buffer empty in a UART.

BIRQ—BRG Timer Interrupt Request

If the SPI is enabled, this bit has no effect. If the SPI is disabled:

0 = The Baud Rate Generator timer function is disabled.

1 = The Baud Rate Generator timer function and time-out interrupt are enabled.

PHASE—Phase Select

Sets the phase relationship of the data to the clock. Refer to the **SPI Clock Phase and Polarity Control** section for more information on operation of the **PHASE** bit.

CLKPOL—Clock Polarity

0 = SCK idles Low (0).

1 = SCK idle High (1).

WOR—Wire-OR (Open-Drain) Mode Enabled

0 = SPI signal pins not configured for open-drain.

1 = All four SPI signal pins (SCK, \overline{SS} , MISO, MOSI) configured for open-drain function. This setting is typically used for multi-master and/or multi-slave configurations.

MMEN—SPI Master Mode Enable

0 = SPI configured in Slave mode.

1 = SPI configured in Master mode.

SPI Baud Rate High and Low Byte Registers

The SPI Baud Rate High and Low Byte registers combine to form a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator. The reload value must be greater than or equal to 0002H for proper SPI operation (maximum baud rate is system clock frequency divided by 4). The SPI baud rate is calculated using the following equation:

$$\text{SPI Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{2 \times \text{BRG}[15:0]}$$

Table 64. SPI Baud Rate High Byte Register (SPIBRH)

BITS	7	6	5	4	3	2	1	0
FIELD	BRH							
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	F66H							

BRH = SPI Baud Rate High Byte

Most significant byte, BRG[15:8], of the SPI Baud Rate Generator's reload value.

Table 65. SPI Baud Rate Low Byte Register (SPIBRL)

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
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	F67H							

BRL = SPI Baud Rate Low Byte

Least significant byte, BRG[7:0], of the SPI Baud Rate Generator's reload value.



Direct Memory Access Controller

Overview

The Z8F640x family device's Direct Memory Access (DMA) Controller provides three independent Direct Memory Access channels. Two of the channels (DMA0 and DMA1) transfer data between the on-chip peripherals and the Register File. The third channel (DMA_ADC) controls the Analog-to-Digital Converter (ADC) operation and transfers the Single-Shot mode ADC output data to the Register File.

Operation

DMA0 and DMA1 Operation

DMA0 and DMA1, referred to collectively as DMA_x, transfer data either from the on-chip peripheral control registers to the Register File, or from the Register File to the on-chip peripheral control registers. The sequence of operations in a DMA_x data transfer is:

1. DMA_x trigger source requests a DMA data transfer.
2. DMA_x requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA_x transfers either a single byte or a two-byte word (depending upon configuration) and then returns system bus control back to the eZ8 CPU.
4. If Current Address equals End Address:
 - DMA_x reloads the original Start Address
 - If configured to generate an interrupt, DMA_x sends an interrupt request to the Interrupt Controller
 - If configured for single-pass operation, DMA_x resets the DEN bit in the DMA_x Control register to 0 and the DMA is disabled.

If Current Address does not equal End Address, the Current Address increments by 1 (single-byte transfer) or 2 (two-byte word transfer).

Table 74. DMAx Start/Current Address Low Byte Register (DMAxSTART)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_START							
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	FB3H, FHBH							

DMA_START—DMAx Start/Current Address Low

These bits, with the four lower bits of the DMAx_H register, form the 12-bit Start/Current address. The full 12-bit address is given by {DMA_START_H[3:0], DMA_START[7:0]}.

DMAx End Address Low Byte Register

The DMAx End Address Low Byte register, in conjunction with the DMAx_H register, forms a 12-bit End Address.

Table 75. DMAx End Address Low Byte Register (DMAxEND)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_END							
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	FB4H, FBCH							

DMA_END—DMAx End Address Low

These bits, with the four upper bits of the DMAx_H register, form a 12-bit address. This address is the ending location of the DMAx transfer. The full 12-bit address is given by {DMA_END_H[3:0], DMA_END[7:0]}.

DMA_ADC Address Register

The DMA_ADC Address register points to a block of the Register File to store ADC conversion values as illustrated in Table 76. This register contains the seven most-significant bits of the 12-bit Register File addresses. The five least-significant bits are calculated from the ADC Analog Input number (5-bit base address is equal to twice the ADC Analog Input number). The 10-bit ADC conversion data is stored as two bytes with the most significant byte of the ADC data stored at the even numbered Register File address.



this bit to 0 when a conversion has been completed.

1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.

Reserved

This bit is reserved and must be 0.

VREF

0 = Internal voltage reference generator enabled. The VREF pin should be left unconnected (or capacitively coupled to analog ground).

1 = Internal voltage reference generator disabled. An external voltage reference must be provided through the VREF pin.

CONT

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.

ANAIN—Analog Input Select

These bits select the analog input for conversion. Not all Port pins in this list are available in all packages for the Z8F640x family of products. Refer to the **Signal and Pin Descriptions** chapter for information regarding the Port pins available with each package style.

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.

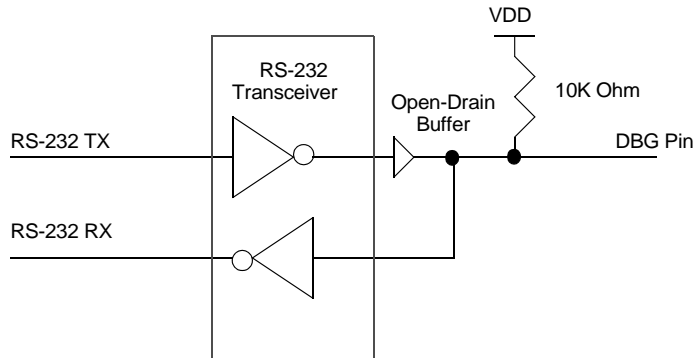


Figure 88. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

Debug Mode

The operating characteristics of the Z8F640x family devices in Debug mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions.
- The system clock operates unless in Stop mode.
- All enabled on-chip peripherals operate unless in Stop mode.
- Automatically exits Halt mode.
- Constantly refreshes the Watch-Dog Timer, if enabled.

Entering Debug Mode

The Z8F640x family device enters Debug mode following any of the following operations:

- Writing the DBGMODE bit in the OCD Control Register to 1 using the OCD interface.
- eZ8 CPU execution of a BRK (Breakpoint) instruction (when enabled).
- Break upon a Watchpoint match.
- If the DBG pin is Low when the Z8F640x family device exits Reset, the On-Chip Debugger automatically puts the device into Debug mode.

Exiting Debug Mode

The device exits Debug mode following any of the following operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0.

General Purpose I/O Port Output Timing

Figure 94 and Table 108 provide timing information for GPIO Port pins.

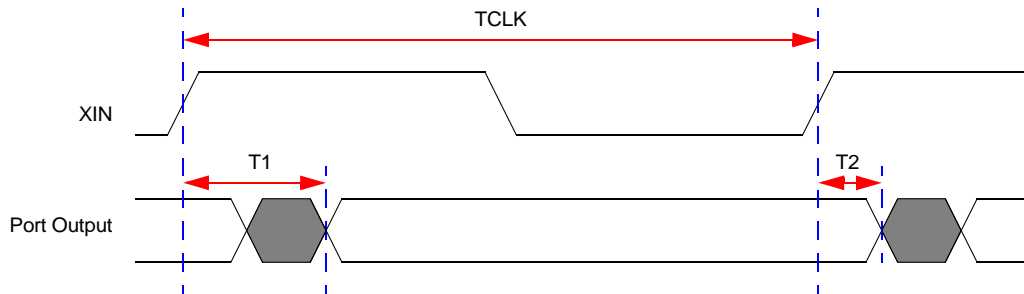


Figure 94. GPIO Port Output Timing

Table 108. GPIO Port Output Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
T ₁	XIN Rise to Port Output Valid Delay	–	15
T ₂	XIN Rise to Port Output Hold Time	2	–

On-Chip Debugger Timing

Figure 95 and Table 109 provide timing information for DBG pins. The timing specifications presume a rise and fall time on DBG of less than 4μs.

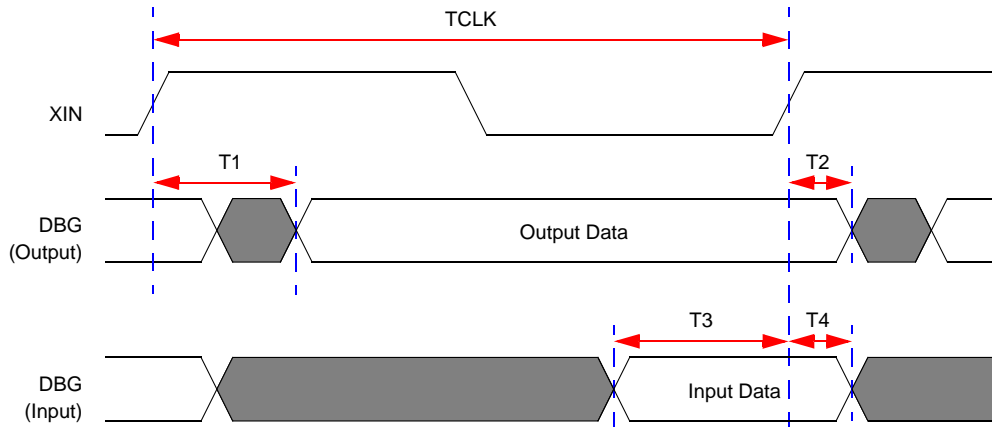


Figure 95. On-Chip Debugger Timing

Table 109. On-Chip Debugger Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
DBG			
T ₁	XIN Rise to DBG Valid Delay	–	15
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



		Lower Nibble (Hex)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Upper Nibble (Hex)	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																
	8																
	9																
	A			3.3 CPC r1,r2	3.4 CPC r1,lr2	4.3 CPC R2,R1	4.4 CPC IR2,R1	4.3 CPC R1,IM	4.4 CPC IR1,IM	5.3 CPCX ER2,ER1	5.3 CPCX IM,ER1						
	B																
	C	3.2 SRL R1	3.3 SRL IR1														
	D																
	E																
	F																

Figure 102. Second Opcode Map after 1FH

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

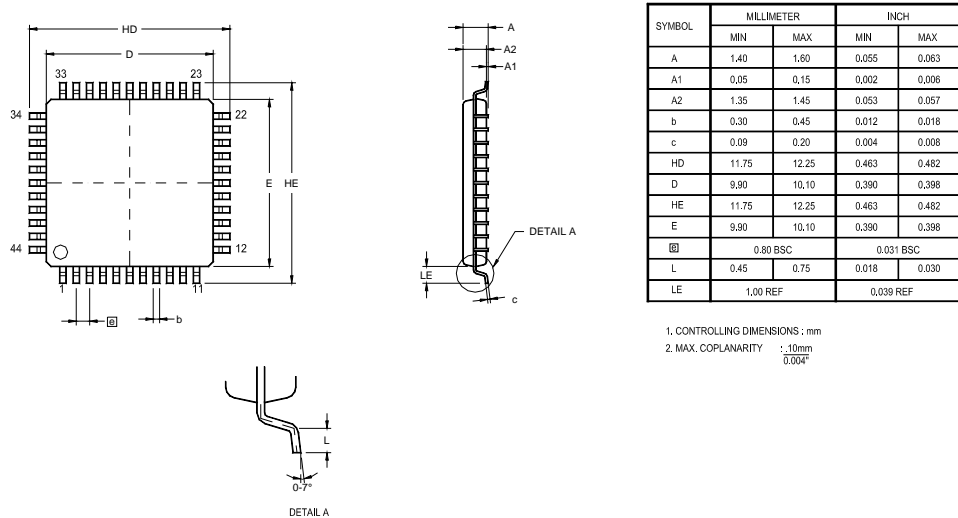


Figure 104. 44-Lead Low-Profile Quad Flat Package (LQFP)

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

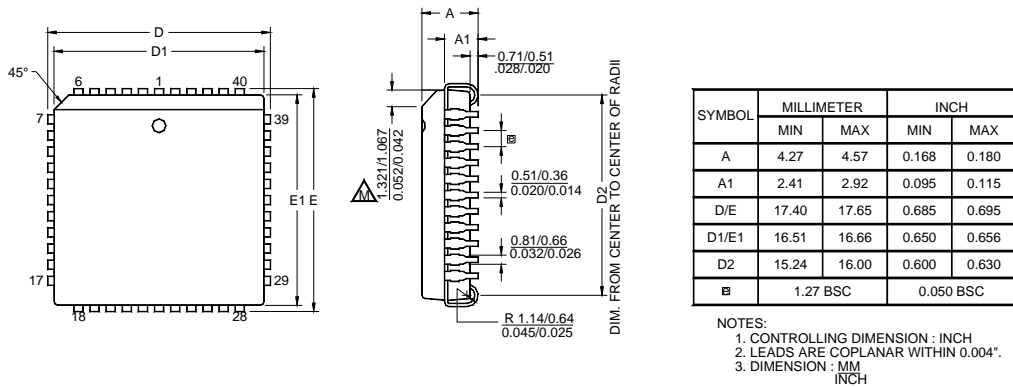


Figure 105. 44-Lead Plastic Lead Chip Carrier Package (PLCC)