



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	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/z8f6402ar020sc00tr



Option Bits	148
Overview	148
Operation	148
Option Bit Configuration By Reset	148
Option Bit Address Space	148
Program Memory Address 0000H	149
Program Memory Address 0001H	150
On-Chip Debugger	151
Overview	151
Architecture	151
Operation	152
OCD Interface	152
Debug Mode	153
OCD Data Format	154
OCD Auto-Baud Detector/Generator	154
OCD Serial Errors	155
Breakpoints	155
Watchpoints	155
Runtime Counter	156
On-Chip Debugger Commands	156
On-Chip Debugger Control Register Definitions	161
OCD Control Register	161
OCD Status Register	162
OCD Watchpoint Control Register	163
OCD Watchpoint Address Register	164
OCD Watchpoint Data Register	164
On-Chip Oscillator	165
20MHz Crystal Oscillator Operation	165
Electrical Characteristics	167
Absolute Maximum Ratings	167
DC Characteristics	169
AC Characteristics	172
On-Chip Peripheral AC and DC Electrical Characteristics	173
General Purpose I/O Port Input Data Sample Timing	176
General Purpose I/O Port Output Timing	177
On-Chip Debugger Timing	178
SPI Master Mode Timing	179
SPI Slave Mode Timing	180
I2C Timing	181
eZ8 CPU Instruction Set	182
Assembly Language Programming Introduction	182



Assembly Language Syntax	183
eZ8 CPU Instruction Notation	183
Condition Codes	186
eZ8 CPU Instruction Classes	187
eZ8 CPU Instruction Summary	191
Flags Register	201
Opcode Maps	202
Packaging	206
Ordering Information	211
Part Number Description	214
Precharacterization Product	215
Document Information	215
Document Number Description	215
Customer Feedback Form	216



Manual Objectives

This Product Specification provides detailed operating information for the Z8F640x, Z8F480x, Z8F320x, Z8F240x, and Z8F160x devices within the Z8 Encore!™ Microcontroller (MCU) family of products. Within this document, the Z8F640x, Z8F480x, Z8F320x, Z8F240x, and Z8F160x are referred to collectively as Z8 Encore!™ or the Z8F640x family unless specifically stated otherwise.

About This Manual

ZiLOG recommends that the user 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.

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



Interrupt Controller

Overview

The interrupt controller on the Z8F640x family device prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of the interrupt controller on the Z8F640x family device include the following:

- 24 unique interrupt vectors:
 - 12 GPIO port pin interrupt sources
 - 12 on-chip peripheral interrupt sources
- Flexible GPIO interrupts
 - 8 selectable rising and falling edge GPIO interrupts
 - 4 dual-edge interrupts
- 3 levels of individually programmable interrupt priority
- Watch-Dog Timer can be configured to generate an interrupt

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt control has no effect on operation. Refer to the *eZ8 CPU User Manual* for more information regarding interrupt servicing by the eZ8 CPU. The *eZ8 CPU User Manual* is available for download at www.zilog.com.

Interrupt Vector Listing

Table 22 lists all of the interrupts available on the Z8F640x family device in order of priority. The interrupt vector is stored with the most significant byte (MSB) at the even Program Memory address and the least significant byte (LSB) at the following odd Program Memory address.

Table 22. Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt Source	Interrupt Assertion Type
Highest	0002h	Reset (not an interrupt)	Not applicable
	0004h	Watch-Dog Timer	Continuous assertion
	0006h	Illegal Instruction Trap (not an interrupt)	Not applicable
	0008h	Timer 2	Single assertion (pulse)
	000Ah	Timer 1	Single assertion (pulse)
	000Ch	Timer 0	Single assertion (pulse)
	000Eh	UART 0 receiver	Continuous assertion
	0010h	UART 0 transmitter	Continuous assertion
	0012h	I ² C	Continuous assertion
	0014h	SPI	Continuous assertion
	0016h	ADC	Single assertion (pulse)
	0018h	Port A7 or Port D7, rising or falling input edge	Single assertion (pulse)
	001Ah	Port A6 or Port D6, rising or falling input edge	Single assertion (pulse)
	001Ch	Port A5 or Port D5, rising or falling input edge	Single assertion (pulse)
	001Eh	Port A4 or Port D4, rising or falling input edge	Single assertion (pulse)
	0020h	Port A3 or Port D3, rising or falling input edge	Single assertion (pulse)
	0022h	Port A2 or Port D2, rising or falling input edge	Single assertion (pulse)
	0024h	Port A1 or Port D1, rising or falling input edge	Single assertion (pulse)
	0026h	Port A0 or Port D0, rising or falling input edge	Single assertion (pulse)
	0028h	Timer 3 (<i>not available in 40/44-pin packages</i>)	Single assertion (pulse)
	002Ah	UART 1 receiver	Continuous assertion
	002Ch	UART 1 transmitter	Continuous assertion
	002Eh	DMA	Single assertion (pulse)
	0030h	Port C3, both input edges	Single assertion (pulse)
	0032h	Port C2, both input edges	Single assertion (pulse)
	0034h	Port C1, both input edges	Single assertion (pulse)
Lowest	0036h	Port C0, both input edges	Single assertion (pulse)

- Execution of a Trap instruction
- Illegal instruction trap

Interrupt Vectors and Priority

The Z8F640x family device interrupt controller supports three levels of interrupt priority. Level 3 is the highest priority, Level 2 is the second highest priority, and Level 1 is the lowest priority. If all of the interrupts were enabled with identical interrupt priority (all as Level 2 interrupts, for example), then interrupt priority would be assigned from highest to lowest as specified in Table 22. Level 3 interrupts always have higher priority than Level 2 interrupts which, in turn, always have higher priority than Level 1 interrupts. Within each interrupt priority level (Level 1, Level 2, or Level 3), priority is assigned as specified in Table 22.

Reset, Watch-Dog Timer interrupt (if enabled), and Illegal Instruction Trap always have highest (Level 3) priority.

Interrupt Assertion Types

Two types of interrupt assertion - single assertion (pulse) and continuous assertion - are used within the Z8F640x family device. The type of interrupt assertion for each interrupt source is listed in Table 22.

Single Assertion (Pulse) Interrupt Sources

Some interrupt sources assert their interrupt requests for only a single system clock period (single pulse). When the interrupt request is acknowledged by the eZ8 CPU, the corresponding bit in the Interrupt Request register is cleared until the next interrupt occurs. Writing a 0 to the corresponding bit in the Interrupt Request register likewise clears the interrupt request.

Continuous Assertion Interrupt Sources

Other interrupt sources continuously assert their interrupt requests until cleared at the source. For these continuous assertion interrupt sources, interrupt acknowledgement by the eZ8 CPU does not clear the corresponding bit in the Interrupt Request register. Writing a 0 to the corresponding bit in the Interrupt Request register only clears the interrupt for a single clock cycle. Since the source is continuously asserting the interrupt request, the interrupt request bit is set to 1 again during the next clock cycle.

The only way to clear continuous assertion interrupts is at the source of the interrupt (for example, in the UART or SPI peripherals). The source of the interrupt must be cleared first. After the interrupt is cleared at the source, the corresponding bit in the Interrupt Request register must also be cleared to 0. Both the interrupt source and the IRQ register must be cleared.

Interrupt Control Register Definitions

For all interrupts other than the Watch-Dog Timer interrupt, the interrupt control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) register (Table 23) stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 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 0 register to determine if any interrupt requests are pending

Table 23. Interrupt Request 0 Register (IRQ0)

BITS	7	6	5	4	3	2	1	0
FIELD	T2I	T1I	T0I	U0RXI	U0TXI	I2CI	SPII	ADCI
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	FC0H							

T2I—Timer 2 Interrupt Request

0 = No interrupt request is pending for Timer 2.

1 = An interrupt request from Timer 2 is awaiting service.

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.

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.

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

Figures 68 and 69 illustrates the asynchronous data format employed by the UART without parity and with parity, respectively.

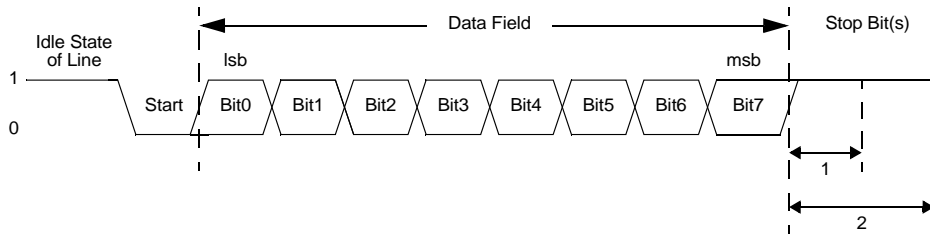


Figure 68. UART Asynchronous Data Format without Parity

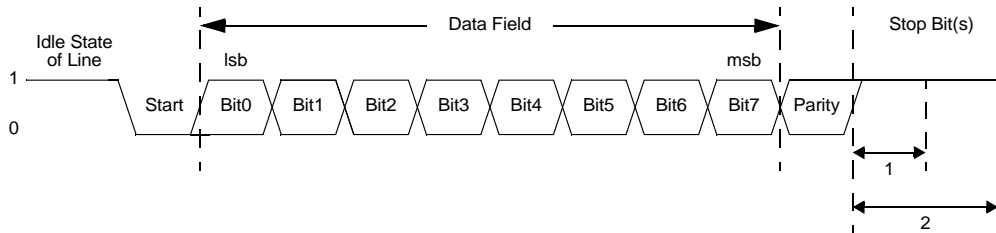


Figure 69. UART Asynchronous Data Format with Parity

Transmitting Data using the Polled Method

Follow these steps to transmit data using the polled method of operation:

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. Write to the UART Control 1 register to enable Multiprocessor (9-bit) mode functions, if desired.
4. 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 using the CTS pin.

Table 76 provides an example of the Register File addresses if the DMA_ADC Address register contains the value 72H.

Table 76. DMA_ADC Register File Address Example

ADC Analog Input	Register File Address (Hex) ¹
0	720H-721H
1	722H-723H
2	724H-725H
3	726H-727H
4	728H-729H
5	72AH-72BH
6	72CH-72DH
7	72EH-72FH
8	730H-731H
9	732H-733H
10	734H-735H
11	736H-737H

¹ DMAA_ADDR set to 72H.

Table 77. DMA_ADC Address Register (DMAA_ADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	DMAA_ADDR							Reserved
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	FBDH							

DMAA_ADDR—DMA_ADC Address

These bits specify the seven most-significant bits of the 12-bit Register File addresses used for storing the ADC output data. The ADC Analog Input Number defines the five least-significant bits of the Register File address. Full 12-bit address is {DMAA_ADDR[7:1], 4-bit ADC Analog Input Number, 0}.

Reserved

This bit is reserved and must be 0.



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.



Caution: The byte at each address of the Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs.

Page Erase

The Flash memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Flash Page Select register identifies the page to be erased. With the Flash Controller unlocked, writing the value 95H to the Flash Control register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. If the Page Erase operation is performed through the On-Chip Debugger, poll the Flash Status register to determine when the Page Erase operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state.

Mass Erase

The Flash memory can also be Mass Erased using the Flash Controller. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked, writing the value 63H to the Flash Control register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Typically, the Flash Memory is Mass Erased using the On-Chip Debugger. Via the On-Chip Debugger, poll the Flash Status register to determine when the Mass Erase operation is complete. Although the Flash can be Mass Erased by user program code, when the Mass Erase is complete the user program code is completely erased. When the Mass Erase is complete, the Flash Controller returns to its locked state.

Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Row Programming algorithms by controlling the Flash programming signals directly.

Row programming is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of the Flash memory. Mass Erase and Page Erase operations are also supported when the Flash Controller is bypassed.

Please refer to the document entitled *Third-Party Flash Programming Support for Z8 Encore![™]* for more information on bypassing the Flash Controller. This document is available for download at www.zilog.com.

OCD Watchpoint Address Register

The OCD Watchpoint Address register specifies the lower 8 bits of the Register File address bus to match when generating Watchpoint Debug Breaks. The full 12-bit Register File address is given by {WPTCTL3:0], WPTADDR[7:0]}.

Table 97. OCD Watchpoint Address (WPTADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	WPTADDR[7:0]							
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

WPTADDR[7:0]—Watchpoint Register File Address

These bits specify the lower eight bits of the register address to match when generating a Watchpoint Debug Break.

OCD Watchpoint Data Register

The OCD Watchpoint Data register specifies the data to match if Watchpoint data match is enabled.

Table 98. OCD Watchpoint Data (WPTDATA)

BITS	7	6	5	4	3	2	1	0
FIELD	WPTDATA[7:0]							
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

WPTDATA[7:0]—Watchpoint Register File Data

These bits specify the Register File data to match when generating Watchpoint Debug Breaks with the WPDM bit (WPTCTL[5]) is set to 1.

General Purpose I/O Port Input Data Sample Timing

Figure 93 illustrates timing of the GPIO Port input sampling. The input value on a GPIO Port pin is sampled on the rising edge of the system clock. The Port value is then available to the eZ8 CPU on the second rising clock edge following the change of the Port value.

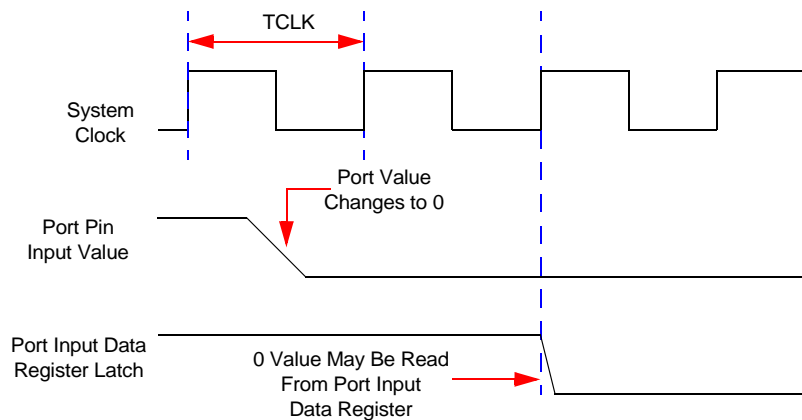


Figure 93. Port Input Sample Timing

Table 107. GPIO Port Input Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
T _{S_PORT}	Port Input Transition to XIN Rise Setup Time (Not pictured)	5	–
T _{H_PORT}	XIN Rise to Port Input Transition Hold Time (Not pictured)	5	–
T _{SMR}	GPIO Port Pin Pulse Width to Insure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources)	1μs	



Table 125. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

eZ8 CPU Instruction Summary

Table 126 summarizes the eZ8 CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch, and the number of CPU clock cycles required for the instruction execution.

Table 126. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	13							2	4
		R	R	14							3	3
		R	IR	15							3	4
		R	IM	16							3	3
		IR	IM	17							3	4
ADCX dst, src	dst ← dst + src + C	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3
Flags Notation:	* = Value is a function of the result of the operation. - = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							



Table 126. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
LD dst, rc	dst ← src	r	IM	0C-FC	-	-	-	-	-	-	2	2
		r	X(r)	C7							3	3
		X(r)	r	D7							3	4
		r	Ir	E3							2	3
		R	R	E4							3	2
		R	IR	E5							3	3
		R	IM	E6							3	3
		IR	IM	E7							3	3
		Ir	r	F3							2	3
		IR	R	F5							3	3
LDC dst, src	dst ← src	r	Irr	C2	-	-	-	-	-	-	2	5
		Ir	Irr	C5							2	9
		Irr	r	D2							2	5
LDCI dst, src	dst ← src	Ir	Irr	C3	-	-	-	-	-	-	2	9
	r ← r + 1 rr ← rr + 1	Irr	Ir	D3							2	9
LDE dst, src	dst ← src	r	Irr	82	-	-	-	-	-	-	2	5
		Irr	r	92							2	5
LDEI dst, src	dst ← src	Ir	Irr	83	-	-	-	-	-	-	2	9
	r ← r + 1 rr ← rr + 1	Irr	Ir	93							2	9
Flags Notation:		* = Value is a function of the result of the operation.			0 = Reset to 0							
		- = Unaffected			1 = Set to 1							
		X = Undefined										

Figure 105 illustrates the 64-pin LQFP (low-profile quad flat package) available for the Z8F1602, Z8F2402, Z8F3202, Z8F4802, and Z8F6402 devices.

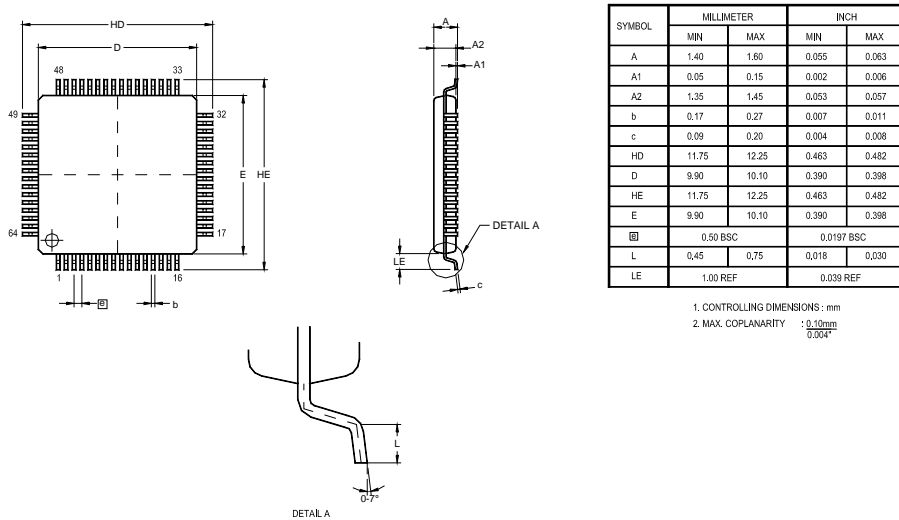


Figure 106. 64-Lead Low-Profile Quad Flat Package (LQFP)

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

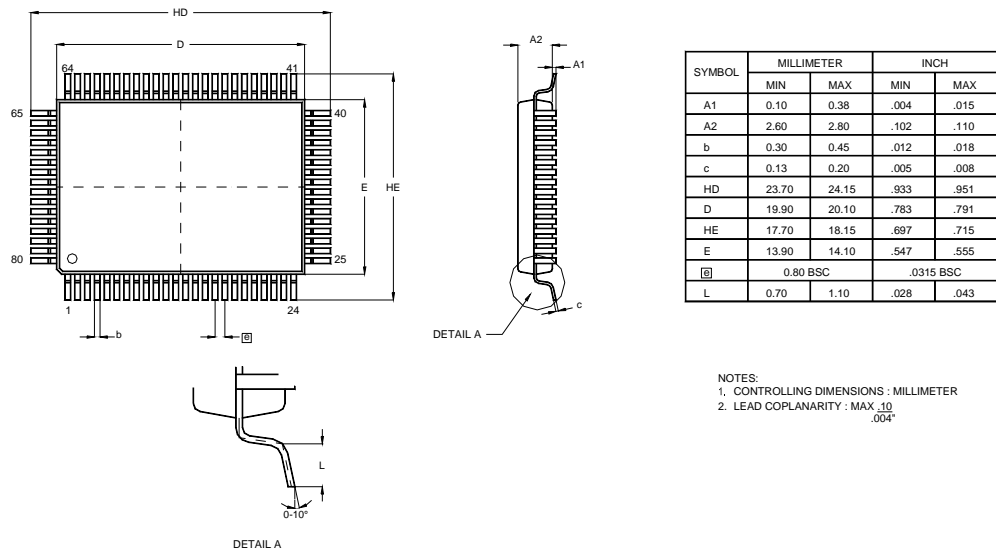


Figure 108. 80-Lead Quad-Flat Package (QFP)