Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	60
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	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6403ft020ec00tr

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
FCE	Interrupt Port Select	IRQPS	00	55
FCF	Interrupt Control	IRQCTL	00	56
GPIO Port A				
FD0	Port A Address	PAADDR	00	37
FD1	Port A Control	PACTL	00	38
FD2	Port A Input Data	PAIN	XX	42
FD3	Port A Output Data	PAOUT	00	43
GPIO Port B				
FD4	Port B Address	PBADDR	00	37
FD5	Port B Control	PBCTL	00	38
FD6	Port B Input Data	PBIN	XX	42
FD7	Port B Output Data	PBOUT	00	43
GPIO Port C				
FD8	Port C Address	PCADDR	00	37
FD9	Port C Control	PCCTL	00	38
FDA	Port C Input Data	PCIN	XX	42
FDB	Port C Output Data	PCOUT	00	43
GPIO Port D				
FDC	Port D Address	PDADDR	00	37
FDD	Port D Control	PDCTL	00	38
FDE	Port D Input Data	PDIN	XX	42
FDF	Port D Output Data	PDOUT	00	43
GPIO Port E				
FE0	Port E Address	PEADDR	00	37
FE1	Port E Control	PECTL	00	38
FE2	Port E Input Data	PEIN	XX	42
FE3	Port E Output Data	PEOUT	00	43
GPIO Port F				
FE4	Port F Address	PFADDR	00	37
FE5	Port F Control	PFCTL	00	38
FE6	Port F Input Data	PFIN	XX	42
FE7	Port F Output Data	PFOUT	00	43
GPIO Port G				
FE8	Port G Address	PGADDR	00	37
FE9	Port G Control	PGCTL	00	38
FEA	Port G Input Data	PGIN	XX	42
FEB	Port G Output Data	PGOUT	00	43
GPIO Port H				
FEC	Port H Address	PHADDR	00	37

XX=Undefined

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}).

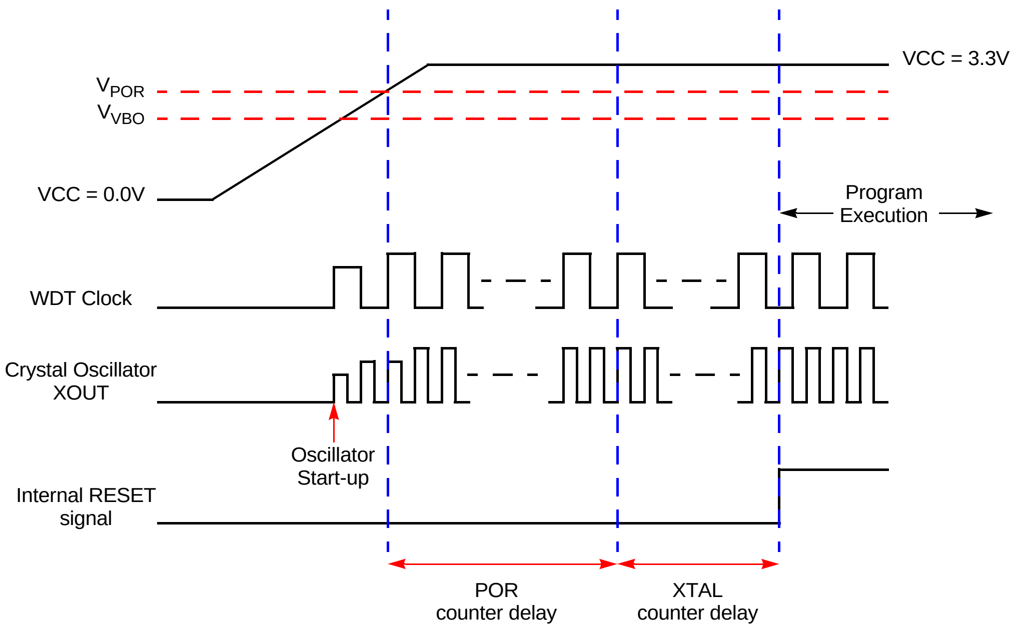


Figure 62. Power-On Reset Operation (not to scale)

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

Table 10. Port Availability by Device and Package Type (Continued)

Device	Packages	Port A	Port B	Port C	Port D	Port E	Port F	Port G	Port H
Z8F6401	44-pin	[7:0]	[7:0]	[7:0]	[6:0]	-	-	-	-
Z8F6402	64- and 68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]
Z8F6403	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]

Architecture

Figure 64 illustrates a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength are not illustrated.

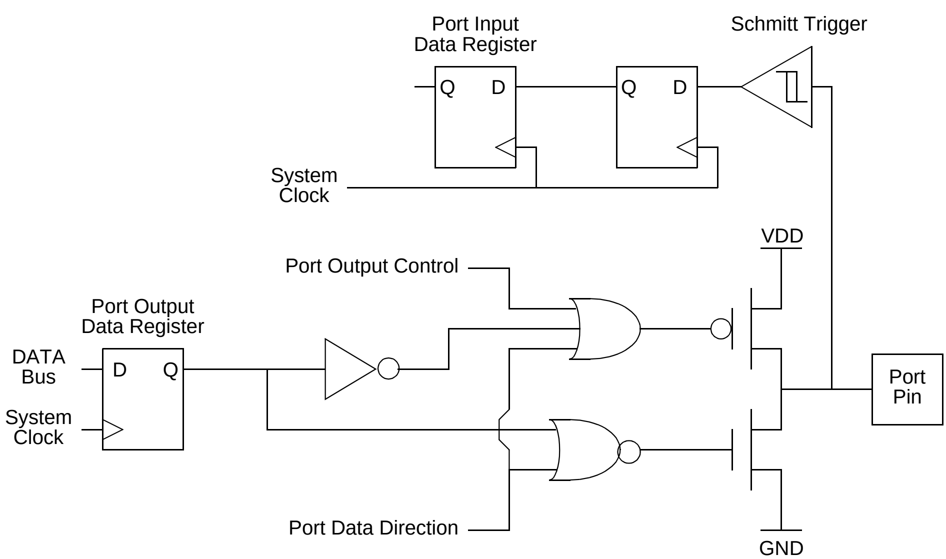


Figure 64. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins can be used as both general-purpose I/O and to provide access to on-chip peripheral functions such as the timers and serial communication devices. The Port A-H Alternate Function sub-registers configure these pins for either general-purpose I/O or alternate function operation. When a pin is configured for alternate function, control

Table 12. GPIO Port Registers and Sub-Registers

Port Register Mnemonic	Port Register Name
P*x*ADDR	Port A-H Address Register (Selects sub-registers)
P*x*CTL	Port A-H Control Register (Provides access to sub-registers)
P*x*IN	Port A-H Input Data Register
P*x*OUT	Port A-H Output Data Register
Port Sub-Register Mnemonic	**Port Register Name**
P*x*DD	Data Direction
P*x*AF	Alternate Function
P*x*OC	Output Control (Open-Drain)
P*x*HDE	High Drive Enable
P*x*SMRE	STOP Mode Recovery Source Enable

Port A-H Address Registers

The Port A-H Address registers select the GPIO Port functionality accessible through the Port A-H Control registers. The Port A-H Address and Control registers combine to provide access to all GPIO Port control (Table 13).

Table 13. Port A-H GPIO Address Registers (P*x*ADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	PADDR[7:0]							
RESET	00H							
R/W	R/W							
ADDR	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

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^2C	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 *(not available in 40/44-pin packages)*	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)

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:

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

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

mode. Refer to the **Reset and Stop Mode Recovery** chapter for more information on STOP Mode Recovery.

If interrupts are enabled, following completion of the Stop Mode Recovery the eZ8 CPU responds to the interrupt request by fetching the Watch-Dog Timer interrupt vector and executing code from the vector address.

WDT Reset in Normal Operation

If configured to generate a Reset when a time-out occurs, the Watch-Dog Timer forces the Z8F640x family device into the Short Reset state. The WDT status bit in the Watch-Dog Timer Control register is set to 1. Refer to the **Reset and Stop Mode Recovery** chapter for more information on Short Reset.

WDT Reset in Stop Mode

If configured to generate a Reset when a time-out occurs and the Z8F640x family device is in STOP mode, the Watch-Dog Timer initiates a Stop Mode Recovery. Both the `WDT` status bit and the `STOP` bit in the Watch-Dog Timer Control register are set to 1 following WDT time-out in STOP mode. Refer to the **Reset and Stop Mode Recovery** chapter for more information.

Watch-Dog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watch-Dog Timer Control register (WDTCTL) unlocks the three Watch-Dog Timer Reload Byte registers (WDTU, WDTH, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL register address produce no effect on the bits in the WDTCTL register. The locking mechanism prevents spurious writes to the Reload registers. The follow sequence is required to unlock the Watch-Dog Timer Reload Byte registers (WDTU, WDTH, and WDTL) for write access.

1. Write 55H to the Watch-Dog Timer Control register (WDTCTL)
2. Write AAH to the Watch-Dog Timer Control register (WDTCTL)
3. Write the Watch-Dog Timer Reload Upper Byte register (WDTU)
4. Write the Watch-Dog Timer Reload High Byte register (WDTH)
5. Write the Watch-Dog Timer Reload Low Byte register (WDTL)

All three Watch-Dog Timer Reload registers must be written in the order just listed. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes can occur, unless the sequence is restarted. The value in the Watch-Dog Timer Reload registers is loaded into the counter when the Watch-Dog Timer is first enabled and every time a WDT instruction is executed.

Table 57. UART*x* Baud Rate Low Byte Register (U*x*BRL)

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	F47H and F4FH							

The UART data rate is calculated using the following equation:

$$\textbf{UART Baud Rate (bits/s)} = \frac{\textbf{System Clock Frequency (Hz)}}{16 \times \textbf{UART Baud Rate Divisor Value}}$$

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

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

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

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

For reliable communication, the UART baud rate error must never exceed 5 percent. Table 58 provides information on data rate errors for popular baud rates and commonly used crystal oscillator frequencies.

Table 58. UART Baud Rates

20.0 MHz System Clock

Desired Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	1	1250.0	0.00
625.0	2	625.0	0.00
250.0	5	250.0	0.00
115.2	11	113.6	-1.36
57.6	22	56.8	-1.36
38.4	33	37.9	-1.36
19.2	65	19.2	0.16
9.60	130	9.62	0.16
4.80	260	4.81	0.16
2.40	521	2.40	-0.03
1.20	1042	1.20	-0.03
0.60	2083	0.60	0.02
0.30	4167	0.30	-0.01

18.432 MHz System Clock

Desired Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	1	1152.0	-7.84%
625.0	2	576.0	-7.84%
250.0	5	230.4	-7.84%
115.2	10	115.2	0.00
57.6	20	57.6	0.00
38.4	30	38.4	0.00
19.2	60	19.2	0.00
9.60	120	9.60	0.00
4.80	240	4.80	0.00
2.40	480	2.40	0.00
1.20	960	1.20	0.00
0.60	1920	0.60	0.00
0.30	3840	0.30	0.00

16.667 MHz System Clock

Desired Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	1	1041.69	-16.67
625.0	2	520.8	-16.67
250.0	4	260.4	4.17
115.2	9	115.7	0.47
57.6	18	57.87	0.47
38.4	27	38.6	0.47
19.2	54	19.3	0.47
9.60	109	9.56	-0.45
4.80	217	4.80	-0.83
2.40	434	2.40	0.01
1.20	868	1.20	0.01
0.60	1736	0.60	0.01
0.30	3472	0.30	0.01

11.0592 MHz System Clock

Desired Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	N/A	N/A	N/A
625.0	1	691.2	10.59
250.0	3	230.4	-7.84
115.2	6	115.2	0.00
57.6	12	57.6	0.00
38.4	18	38.4	0.00
19.2	36	19.2	0.00
9.60	72	9.60	0.00
4.80	144	4.80	0.00
2.40	288	2.40	0.00
1.20	576	1.20	0.00
0.60	1152	0.60	0.00
0.30	2304	0.30	0.00

SPIEN—SPI Enable
0 = SPI disabled.
1 = SPI enabled.

SPI Status Register

The SPI Status register indicates the current state of the SPI.

Table 62. SPI Status Register (SPISTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQ	OVR	COL	Reserved			TXST	SLAS
RESET	0	0	0	0			0	1
R/W	R/W*	R/W*	R/W*	R			R	R
ADDR	F62H							

R/W* = Read access. Write a 1 to clear the bit to 0.

IRQ—Interrupt Request
0 = No SPI interrupt request pending.
1 = SPI interrupt request is pending.

OVR—Overrun
0 = An overrun error has not occurred.
1 = An overrun error has been detected.

COL—Collision
0 = A multi-master collision (mode fault) has not occurred.
1 = A multi-master collision (mode fault) has been detected.

Reserved
These bits are reserved and must be 0.

TXST—Transmit Status
0 = No data transmission currently in progress.
1 = Data transmission currently in progress.

SLAS—Slave Select
If SPI enabled as a Slave,
0 = $\overline{SS}$ input pin is asserted (Low)
1 = $\overline{SS}$ input is not asserted (High).
If SPI enabled as a Master, this bit is not applicable.

received a byte of data. When active, this bit causes the I^2C Controller to generate an interrupt. This bit is cleared by reading the I^2C Data register.

ACK—Acknowledge
This bit indicates the status of the Acknowledge for the last byte transmitted or received. When set, this bit indicates that an Acknowledge was received for the last byte transmitted or received.

10B—10-Bit Address
This bit indicates whether a 10- or 7-bit address is being transmitted. After the `START` bit is set, if the five most-significant bits of the address are `11110B`, this bit is set. When set, it is reset once the first byte of the address has been sent.

RD—Read
This bit indicates the direction of transfer of the data. It is active high during a read. The status of this bit is determined by the least-significant bit of the I^2C Shift register after the `START` bit is set.

TAS—Transmit Address State
This bit is active high while the address is being shifted out of the I^2C Shift register.

DSS—Data Shift State
This bit is active high while data is being transmitted to or from the I^2C Shift register.

NCKI—NACK Interrupt
This bit is set high when a Not Acknowledge condition is received or sent and neither the START nor the STOP bit is active. When set, this bit generates an interrupt that can only be cleared by setting the `START` or `STOP` bit, allowing the user to specify whether he wants to perform a `STOP` or a repeated `START`.

I^2C Control Register

The I^2C Control register enables the I^2C operation.

Table 68. I^2C Control Register (I2CCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IEN	START	STOP	BIRQ	TXI	NAK	FLUSH	FILTEN
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	F52H							

IEN—I^2C Enable
This bit enables the I^2C transmitter and receiver.

Table 123. Logical Instructions

Mnemonic	Operands	Instruction
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using Extended Addressing
COM	dst	Complement
OR	dst, src	Logical OR
ORX	dst, src	Logical OR using Extended Addressing
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using Extended Addressing

Table 124. Program Control Instructions

Mnemonic	Operands	Instruction
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap

Table 126. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	-	-	-	-	-	-	3	3
			Ir	F7							3	4
CALL dst	SP ← SP -2 @SP ← PC PC ← dst	IRR		D4	-	-	-	-	-	-	2	6
		DA		D6							3	3
CCF	C ← ~C			EF	*	-	-	-	-	-	1	2
CLR dst	dst ← 00H	R		B0	-	-	-	-	-	-	2	2
		IR		B1							2	3
COM dst	dst ← ~dst	R		60	-	*	*	0	-	-	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	-	-	2	3
		r	Ir	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	-	-	3	3
		r	Ir	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	-	-	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	-	-	4	3
		ER	IM	A9							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 dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
DA dst	dst ← DA(dst)	R		40	*	*	*	X	-	-	2	2
		IR		41							2	3
DEC dst	dst ← dst - 1	R		30	-	*	*	*	-	-	2	2
		IR		31							2	3
DECW dst	dst ← dst - 1	RR		80	-	*	*	*	-	-	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	-	-	-	-	-	-	1	2
DJNZ dst, RA	dst ← dst – 1 if dst ≠ 0 PC ← PC + X	r		0A-FA	-	-	-	-	-	-	2	3
EI	IRQCTL[7] ← 1			9F	-	-	-	-	-	-	1	2
HALT	Halt Mode			7F	-	-	-	-	-	-	1	2
INC dst	dst ← dst + 1	R		20	-	*	*	*	-	-	2	2
		IR		21							2	3
		r		0E-FE							1	2
INCW dst	dst ← dst + 1	RR		A0	-	*	*	*	-	-	2	5
		IRR		A1							2	6
IRET	FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1			BF	*	*	*	*	*	*	1	5
JP dst	PC ← dst	DA		8D	-	-	-	-	-	-	3	2
		IRR		C4							2	3
JP cc, dst	if cc is true PC ← dst	DA		0D-FD	-	-	-	-	-	-	3	2
JR dst	PC ← PC + X	DA		8B	-	-	-	-	-	-	2	2
JR cc, dst	if cc is true PC ← PC + X	DA		0B-FB	-	-	-	-	-	-	2	2

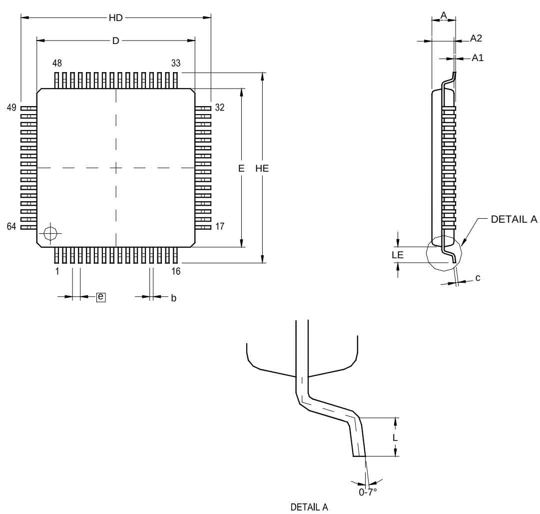
Flags Notation: * = Value is a function of the result of the operation.
- = Unaffected
X = Undefined
0 = Reset to 0
1 = Set to 1

Table 126. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
LDX dst, src	dst ← src	r	ER	84	-	-	-	-	-	-	3	2
		Ir	ER	85							3	3
		R	IRR	86							3	4
		IR	IRR	87							3	5
		r	X(rr)	88							3	4
		X(rr)	r	89							3	4
		ER	r	94							3	2
		ER	Ir	95							3	3
		IRR	R	96							3	4
		IRR	IR	97							3	5
		ER	ER	E8							4	2
		ER	IM	E9							4	2
LEA dst, X(src)	dst ← src + X	r	X(r)	98	-	-	-	-	-	-	3	3
		rr	X(rr)	99							3	5
MULT dst	dst[15:0] ← dst[15:8] * dst[7:0]	RR		F4	-	-	-	-	-	-	2	8
NOP	No operation			0F	-	-	-	-	-	-	1	2
OR dst, src	dst ← dst OR src	r	r	42	-	*	*	0	-	-	2	3
		r	Ir	43							2	4
		R	R	44							3	3
		R	IR	45							3	4
		R	IM	46							3	3
		IR	IM	47							3	4
ORX dst, src	dst ← dst OR src	ER	ER	48	-	*	*	0	-	-	4	3
		ER	IM	49							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

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



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	1.40	1.60	0.055	0.063
A1	0.05	0.15	0.002	0.006
A2	1.35	1.45	0.053	0.057
b	0.17	0.27	0.007	0.011
c	0.09	0.20	0.004	0.008
HD	11.75	12.25	0.463	0.482
D	9.90	10.10	0.390	0.398
HE	11.75	12.25	0.463	0.482
E	9.90	10.10	0.390	0.398
e	0.50 BSC		0.0197 BSC	
L	0.45	0.75	0.018	0.030
LE	1.00 REF		0.039 REF	

1. CONTROLLING DIMENSIONS : mm
2. MAX. COPLANARITY : 0.10mm / 0.004"

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

Problem Description or Suggestion

Provide a complete description of the problem or your suggestion. If you are reporting a specific problem, include all steps leading up to the occurrence of the problem. Attach additional pages as necessary.

M

N

O

U

V

W

X