



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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	24KB (24K 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	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.620", 15.75mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2401pm020sc

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



Signal Descriptions

Table 2 describes the Z8F640x family signals. Refer to the section **Pin Configurations on page 7** to determine the signals available for the specific package styles.

Signal Mnemonic	I/O	Description
General-Purpose I/O	Ports A	н
PA[7:0]	I/O	Port A[7:0]. These pins are used for general-purpose I/O.
PB[7:0]	I/O	Port B[7:0]. These pins are used for general-purpose I/O.
PC[7:0]	I/O	Port C[7:0]. These pins are used for general-purpose I/O.
PD[7:0]	I/O	Port D[7:0]. These pins are used for general-purpose I/O.
PE[7:0]	I/O	Port E[7:0]. These pins are used for general-purpose I/O.
PF[7:0]	I/O	Port F[7:0]. These pins are used for general-purpose I/O.
PG[7:0]	I/O	Port G[7:0]. These pins are used for general-purpose I/O.
PH[3:0]	I/O	Port H[3:0]. These pins are used for general-purpose I/O.
I ² C Controller		
SCL	0	Serial Clock. This is the output clock for the I^2C . This pin is multiplexed with a general-purpose I/O pin. When the general-purpose I/O pin is configured for alternate function to enable the SCL function, this pin is open-drain.
SDA	I/O	Serial Data. This open-drain pin is used to transfer data between the I^2C and a slave. This pin is multiplexed with a general-purpose I/O pin. When the general-purpose I/O pin is configured for alternate function to enable the SDA function, this pin is open-drain.
SPI Controller		
SS	I/O	Slave Select. This signal can be an output or an input. If the Z8 Encore! is the SPI master, this pin may be configured as the Slave Select output. If the Z8 Encore! is the SPI slave, this pin is the input slave select. It is multiplexed with a general-purpose I/O pin.
SCK	I/O	SPI Serial Clock. The SPI master supplies this pin. If the Z8 Encore! is the SPI master, this pin is an output. If the Z8 Encore! is the SPI slave, this pin is an input. It is multiplexed with a general-purpose I/O pin.
MOSI	I/O	Master Out Slave In. This signal is the data output from the SPI master device and the data input to the SPI slave device. It is multiplexed with a general-purpose I/O pin.
MISO	I/O	Master In Slave Out. This pin is the data input to the SPI master device and the data output from the SPI slave device. It is multiplexed with a general-purpose I/O pin.

Table 2. Signal Descriptions



External Pin Reset

The $\overline{\text{RESET}}$ pin has a Schmitt-triggered input and an internal pull-up. Once the $\overline{\text{RESET}}$ pin is asserted, the device progresses through the Short Reset sequence. While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8F640x family device continues to be held in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the Short Reset time-out, the device exits the Reset state immediately following $\overline{\text{RESET}}$ pin deassertion. Following a Short Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the Watch-Dog Timer Control (WDTCTL) register is set to 1.

Stop Mode Recovery

Stop mode is entered by execution of a STOP instruction by the eZ8 CPU. Refer to the **Low-Power Modes** chapter for detailed Stop mode information. During Stop Mode Recovery, the Z8F640x family device is held in reset for 514 cycles of the Watch-Dog Timer oscillator followed by 16 cycles of the system clock (crystal oscillator). Stop Mode Recovery does not affect any values in the Register File, including the Stack Pointer, Register Pointer, Flags and general-purpose RAM.

The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the STOP bit in the Watch-Dog Timer Control Register is set to 1. Table 9 lists the Stop Mode Recovery sources and resulting actions. The text following provides more detailed information on each of the Stop Mode Recovery sources.

Operating Mode	Stop Mode Recovery Source	Action		
Stop mode	Watch-Dog Timer time-out when configured for Reset	Stop Mode Recovery		
	Watch-Dog Timer time-out when configured for interrupt	Stop Mode Recovery followed by interrupt (in interrupts are enabled)		
	Data transition on any GPIO Port pin enabled as a Stop Mode Recovery source	Stop Mode Recovery		

Table	9.	Stop	Mode	Recovery	Sources	and	Resulting	Action
	- •	~~~p	1.10000		D'our eeb			

Stop Mode Recovery Using Watch-Dog Timer Time-Out

If the Watch-Dog Timer times out during Stop mode, the Z8F640x family device undergoes a STOP Mode Recovery sequence. In the Watch-Dog Timer Control register, the WDT and STOP bits are set to 1. If the Watch-Dog Timer is configured to generate an interrupt upon time-out and the device is configured to respond to interrupts, the Z8F640x family device services the Watch-Dog Timer interrupt request following the normal Stop Mode Recovery sequence.



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.

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								

Table 16. Port A-H Alternate Function Sub-Registers



- 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.



BITS	7	6	5	4	3	2	1	0
FIELD	T3ENH	U1RENH	UITENH	DMAENH	C3ENH	C2ENH	C1ENH	C0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR				FC	7H			

 Table 33. IRQ2 Enable High Bit Register (IRQ2ENH)

T3ENH—Timer 3 Interrupt Request Enable High Bit U1RENH—UART 1 Receive Interrupt Request Enable High Bit U1TENH—UART 1 Transmit Interrupt Request Enable High Bit DMAENH—DMA Interrupt Request Enable High Bit C3ENH—Port C3 Interrupt Request Enable High Bit C2ENH—Port C2 Interrupt Request Enable High Bit C1ENH—Port C1 Interrupt Request Enable High Bit C0ENH—Port C0 Interrupt Request Enable High Bit

 Table 34. IRQ2 Enable Low Bit Register (IRQ2ENL)

BITS	7	6	5	4	3	2	1	0	
FIELD	T3ENL	U1RENL	UITENL	DMAENL	C3ENL	C2ENL	C1ENL	C0ENL	
RESET	0	0	0	0	0	0	0	0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ADDR	FC8H								

T3ENL—Timer 3 Interrupt Request Enable Low Bit

U1RENL-UART 1 Receive Interrupt Request Enable Low Bit

U1TENL-UART 1 Transmit Interrupt Request Enable Low Bit

DMAENL—DMA Interrupt Request Enable Low Bit

C3ENL—Port C3 Interrupt Request Enable Low Bit

C2ENL—Port C2 Interrupt Request Enable Low Bit

C1ENL—Port C1 Interrupt Request Enable Low Bit

COENL-Port CO Interrupt Request Enable Low Bit

Interrupt Edge Select Register

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



Interrupt Control Register

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

Table 37. Interrupt Control Register (IRQCTL)

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

IRQE—Interrupt Request Enable

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

0 = Interrupts are disabled.

1 = Interrupts are enabled.

Reserved

These bits must be 0.



- Select either the rising edge or falling edge of the Timer Input signal for the count. This also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function does not have to be enabled.
- 2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in Counter mode. After the first timer Reload in Counter mode, counting always begins at the reset value of 0001H. Generally, in Counter mode the Timer High and Low Byte registers must be written with the value 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. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
- 7. Write to the Timer Control register to enable the timer.

In Counter mode, the number of Timer Input transitions since the timer start is given by the following equation:

Counter Mode Timer Input Transitions = Current Count Value – Start Value

PWM Mode

In PWM mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO Port pin. The timer input is the system clock. The timer first counts up to the 16bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the Reload value stored in the Timer Reload High and Low Byte registers. Upon 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.

If the TPOL bit in the Timer Control register is set to 1, the Timer Output signal begins as a High (1) and then transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the Reload value and is reset to 0001H.

If the TPOL bit in the Timer Control register is set to 0, the Timer Output signal begins as a Low (0) and then transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the Reload value and is reset to 0001H.

The steps for configuring a timer for PWM mode and initiating the PWM operation are as follows:

1. Write to the Timer Control register to:



- Disable the timer
- Configure the timer for PWM mode.
- Set the prescale value.
- Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function.
- 2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
- 3. Write to the PWM High and Low Byte registers to set the PWM value.
- 4. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
- 5. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 6. Configure the associated GPIO port pin for the Timer Output alternate function.
- 7. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is given by the following equation:

PWM Period (s) = Reload Value × Prescale System Clock Frequency (Hz)

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

If TPOL is set to 0, the ratio of the PWM output High time to the total period is given by:

PWM Output High Time Ratio (%) = $\frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is given by:

PWM Output High Time Ratio (%) =
$$\frac{PWM Value}{Reload Value} \times 100$$

Capture Mode

In Capture mode, the current timer count value is recorded when the desired external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control register determines if the Capture occurs on a rising edge or a falling edge of the





Figure 73. Infrared Data Reception

Jitter

Because of the inherent sampling of the received IR_RXD signal by the bit rate clock, some jitter can be expected on the first bit in any sequence of data. All subsequent bits in the received data stream are a fixed 16-clock periods wide.

Infrared Encoder/Decoder Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined beginning on page 86.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART*x* Control 1 register to 1 to enable the Infrared Encoder/ Decoder *before* enabling the GPIO Port alternate function for the corresponding pin.



SPI Control Register

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

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								

Table 61. SPI Control Register (SPICTL)

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.



128

BITS	7	6	5	4	3	2	1	0			
FIELD		DMA_START									
RESET	Х	Х	Х	Х	Х	Х	Х	Х			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			
ADDR	FB3H, FHBH										

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

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.

BITS	7	6	5	4	3	2	1	0		
FIELD		DMA_END								
RESET	Х	Х	Х	Х	Х	Х	Х	Х		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ADDR		FB4H, FBCH								

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

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.



On-Chip Debugger

Overview

The Z8F640x family devices have an integrated On-Chip Debugger (OCD) that provides advanced debugging features including:

- Reading and writing of the Register File
- Reading and writing of Program and Data Memory
- Setting of Breakpoints and Watchpoints
- Execution of eZ8 CPU instructions.

Architecture

The On-Chip Debugger consists of four primary functional blocks: transmitter, receiver, auto-baud generator, and debug controller. Figure 86 illustrates the architecture of the On-Chip Debugger



Figure 86. On-Chip Debugger Block Diagram



BRKEN-Breakpoint Enable

This bit controls the behavior of the BRK instruction (opcode 00H). By default, Breakpoints are disabled and the BRK instruction behaves like a NOP. If this bit is set to 1, when a BRK instruction is decoded, the DBGMODE bit of the OCDCTL register is automatically set to one.

0 = Breakpoints are disabled.

1 = Breakpoints are enabled.

DBGACK—Debug Acknowledge

This bit enables the debug acknowledge feature. If this bit is set to 1, then the OCD sends an Debug Acknowledge character (FFH) to the host when a Breakpoint or Watchpoint occurs.

0 = Debug Acknowledge is disabled.

1 = Debug Acknowledge is enabled.

Reserved

These bits are reserved and must be 0.

RST-Reset

Setting this bit to 1 resets the Z8F640x family device. The device goes through a normal Power-On Reset sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 when the reset finishes.

0 = No effect.

1 = Reset Z8F640x family device.

OCD Status Register

The OCD Status register reports status information about the current state of the debugger and the Z8F640x family device.

Table 95. OCD Status Register (OCDSTAT)

BITS	7	6	5	4	3	2	1	0			
FIELD	DBG	HALT	RPEN	Reserved							
RESET	0	0	0	0	0	0	0	0			
R/W	R	R	R	R	R	R	R	R			

DBG—Debug Status

0 = The Z8F640x family device is operating in normal mode.

1 = The Z8F640x family device is in Debug mode.

HALT-Halt Mode

0 = The Z8F640x family device is not in Halt mode.

1 = The Z8F640x family device is in Halt mode.



179

SPI Master Mode Timing

Figure 96 and Table 110 provide timing information for SPI Master mode pins. Timing is shown with SCK rising edge used to source MOSI output data, SCK falling edge used to sample MISO input data. Timing on the SS output pin(s) is controlled by software.



Figure 96. SPI Master Mode Timing

Fable 110	SPI	Master	Mode	Timing
-----------	-----	--------	------	--------

		Delay (ns)					
Parameter	Abbreviation	Minimum	Maximum				
T ₁	SCK Rise to MOSI output Valid Delay	-5	+5				
T ₂	MISO input to SCK (receive edge) Setup Time	20					
T ₃	MISO input to SCK (receive edge) Hold Time	0					



180

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

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



Table 116. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
В	Binary Number Suffix
%	Hexadecimal Number Prefix
Н	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow. For example,

 $dst \leftarrow dst + src$

indicates the source data is added to the destination data and the result is stored in the destination location.



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						



198

A		Address Mode			Flags						E-4-b	Transfor
Assembly Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Z	S	V	D	Н	- Fetch Cycles	Cycles
SBC dst, src	$dst \leftarrow dst - src - C$	r	r	32	*	*	*	*	1	*	2	3
		r	Ir	33	-						2	4
		R	R	34	-						3	3
		R	IR	35	-						3	4
		R	IM	36	-						3	3
		IR	IM	37	-						3	4
SBCX dst, src	$dst \leftarrow dst - src - C$	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39	-						4	3
SCF	C ← 1			DF	1	-	-	-	-	-	1	2
SRA dst	T ¥ ¥	R		D0	*	*	*	0	-	-	2	2
	D7 D6 D5 D4 D3 D2 D1 D0 C	IR		D1	-						2	3
SRL dst	0 - D7 D6 D5 D4 D3 D2 D1 D0 C	R		1F C0	*	*	0	*	-	-	3	2
	dst	IR		1F C1	-						3	3
SRP src	$RP \leftarrow src$		IM	01	-	-	-	-	-	-	2	2
STOP	Stop Mode			6F	-	-	-	-	-	-	1	2
SUB dst, src	$dst \leftarrow dst - src$	r	r	22	*	*	*	*	1	*	2	3
		r	Ir	23	-						2	4
		R	R	24	-						3	3
		R	IR	25	-						3	4
		R	IM	26	-						3	3
		IR	IM	27	-						3	4
SUBX dst, src	$dst \leftarrow dst - src$	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29	-						4	3
SWAP dst	$dst[7:4] \leftrightarrow dst[3:0]$	R		F0	Х	*	*	Х	-	-	2	2
		IR		F1	-						2	3
Flags Notation:	* = Value is a function of - = Unaffected X = Undefined	the resu	lt of the o	operation.		0 = 1 =	Res Set	et to to 1	0			

Table 126. eZ8 CPU Instruction Summary (Continued)



logical exclusive OR/extended addressing 190 logical instructions 190 logical OR 190 logical OR/extended addressing 190 low power modes 31 LQFP 44 lead 207 64 lead 208

Μ

master interrupt enable 46 master-in, slave-out and-in 101 memory data 19 program 18 **MISO 101** mode capture 71 capture/compare 71 continuous 70 counter 70 gated 71 one-shot 70 **PWM 70** modes 71 **MOSI 101 MULT 187** multiply 187 multiprocessor mode, UART 84

N

NOP (no operation) 189 not acknowledge interrupt 112 notation b 184 cc 184 DA 184 ER 184 IM 184 IR 184 IR 184 IRR 184 Irr 184 p 184 R 184 r 184 RA 184 RR 184 rr 184 vector 184 X 184 notational shorthand 184

0

OCD architecture 151 auto-baud detector/generator 154 baud rate limits 154 block diagram 151 breakpoints 155 commands 156 control register 161 data format 154 DBG pin to RS-232 Interface 152 debug mode 153 debugger break 190 interface 152 serial errors 155 status register 162 timing 178 watchpoint address register 164 watchpoint control register 163 watchpoint data register 164 watchpoints 155 OCD commands execute instruction (12H) 160 read data memory (0DH) 160 read OCD control register (05H) 158 read OCD revision (00H) 157 read OCD status register (02H) 157 read program counter (07H) 158 read program memory (0BH) 159 read program memory CRC (0EH) 160 read register (09H) 158 read runtime counter (03H) 157



SDA and SCL (IrDA) signals 111 second opcode map after 1FH 205 serial clock 101 serial peripheral interface (SPI) 99 set carry flag 188, 189 set register pointer 189 shift right arithmetic 191 shift right logical 191 signal descriptions 13 single assertion (pulse) interrupt sources 47 single-shot conversion (ADC) 133 SIO 5 slave data transfer formats (I2C) 114 slave select 102 software trap 190 source operand 185 SP 185 SPI architecture 99 baud rate generator 105 baud rate high and low byte register 110 clock phase 102 configured as slave 100 control register 107 control register definitions 106 data register 106 error detection 105 interrupts 105 mode fault error 105 mode register 109 multi-master operation 104 operation 100 overrun error 105 signals 101 single master, multiple slave system 100 single master, single slave system 99 status register 108 timing, PHASE = 0.103timing, PHASE=1 104 SPI controller signals 13 SPI mode (SPIMODE) 109 SPIBRH register 110 SPIBRL register 110 SPICTL register 107

SPIDATA register 106 SPIMODE register 109 SPISTAT register 108 SRA 191 src 185 SRL 191 **SRP 189** SS, SPI signal 101 stack pointer 185 status register, I2C 118 **STOP 189** stop mode 31, 189 stop mode recovery sources 29 using a GPIO port pin transition 30 using watch-dog timer time-out 29 **SUB 188** subtract 188 subtract - extended addressing 188 subtract with carry 188 subtract with carry - extended addressing 188 **SUBX 188 SWAP 191** swap nibbles 191 symbols, additional 185 system and short resets 26

Т

TCM 188 TCMX 188 technical support 213 test complement under mask 188 test under mask 188 timer signals 14 timers 5, 57 architecture 57 block diagram 58 capture mode 62, 71 compare mode 63, 71 compare mode 63, 71 continuous mode 59, 70 counter mode 60 counter modes 70