



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	l²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
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 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2422ar020ec

Email: info@E-XFL.COM

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

zilog



LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2007 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.



Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

Z8 Encore! XP[®] 64K Series Flash Microcontrollers **Product Specification**



Master Interrupt Enable 69
Interrupt Vectors and Priority 70
Interrupt Assertion
Software Interrupt Assertion
Interrupt Control Register Definitions
Interrupt Request 0 Register 71
Interrupt Request 1 Register 72
Interrupt Request 2 Register 73
IRQ0 Enable High and Low Bit Registers
IRQ1 Enable High and Low Bit Registers
IRQ2 Enable High and Low Bit Registers
Interrupt Edge Select Register
Interrupt Port Select Register
Interrupt Control Register
Timers
Overview 81
Architecture 81
Timer Operating Modes
Peading the Timer Count Values
Timer Control Register Definitions
Timer 0.3 High and Low Byte Degisters
Timer Deload High and Low Byte Degisters
Timer 0.3 DW/M High and Low Byte Registers
Timer 0.3 Control 0 Pegisters
Timer 0.3 Control 1 Registers
Watchdog Timer
Overview
Operation
Watchdog Timer Refresh 98
Watchdog Timer Time-Out Response 98
Watchdog Timer Reload Unlock Sequence
Watchdog Timer Control Register Definitions
Watchdog Timer Control Register 100
Watchdog Timer Reload Upper, High and Low Byte Registers 101

Z8 Encore! XP[®] 64K Series Flash Microcontrollers Product Specification





Data Register

PS019919-1207

zilog

On-Chip Debugger Initiated Reset

A Power-On Reset can be initiated using the On-Chip Debugger by setting the RST bit in the OCD Control register. The On-Chip Debugger block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the system reset. Following the system reset the POR bit in the WDT Control register is set.

Stop Mode Recovery

STOP mode is entered by the eZ8 executing a STOP instruction. For detailed STOP mode information, see Low-Power Modes on page 47. During Stop Mode Recovery, the devices are held in reset for 66 cycles of the Watchdog Timer oscillator followed by 16 cycles of the system clock. Stop Mode Recovery only affects the contents of the Watchdog Timer Control register. Stop Mode Recovery does not affect any other values in the Register File, including the Stack Pointer, Register Pointer, Flags, peripheral control registers, 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 Watchdog Timer Control Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions.

Table 10. Stop Mode Re	covery Sources and	Resulting Action
------------------------	--------------------	-------------------------

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

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and the 64K Series devices are configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.



BITS	7	6	5	4	3	2	1	0
FIELD	PAD7ENH	PAD6ENH	PAD5ENH	PAD4ENH	PAD3ENH	PAD2ENH	PAD1ENH	PAD0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FC4H							

Table 31. IRQ1 Enable High Bit Register (IRQ1ENH)

PADxENH—Port A or Port D Bit[x] Interrupt Request Enable High Bit. For selection of either Port A or Port D as the interrupt source, see Interrupt Port Select Register on page 78.

Table 32. IRQ1 Enable Low Bit Register (IRQ1ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	PAD7ENL	PAD6ENL	PAD5ENL	PAD4ENL	PAD3ENL	PAD2ENL	PAD1ENL	PAD0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FC5H							

PADxENL—Port A or Port D Bit[x] Interrupt Request Enable Low Bit For selection of either Port A or Port D as the interrupt source, see Interrupt Port Select Register on page 78.

IRQ2 Enable High and Low Bit Registers

The IRQ2 Enable High and Low Bit registers (see Table 34 and Table 35 on page 77) form a priority encoded enabling for interrupts in the Interrupt Request 2 register. Priority is generated by setting bits in each register. Table 33 describes the priority control for IRQ2.

Table 33.	IRQ2	Enable	and	Priority	/ Encoding
-----------	------	--------	-----	----------	------------

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



UART

Overview

The Universal Asynchronous Receiver/Transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two Stop bits
- Separate transmit and receive interrupts
- Framing, parity, overrun and break detection
- Separate transmit and receive enables
- 16-bit Baud Rate Generator (BRG)
- Selectable MULTIPROCESSOR (9-bit) mode with three configurable interrupt schemes
- Baud Rate Generator timer mode
- Driver Enable output for external bus transceivers

Architecture

The UART consists of three primary functional blocks: Transmitter, Receiver, and Baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. Figure 13 on page 104 displays the UART architecture.

zilog

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

SPI Signals

The four basic SPI signals are:

- Master-In/Slave-Out
- Master-Out/Slave-In
- Serial Clock
- Slave Select

Each signal is described in both Master and Slave modes.

Master-In/Slave-Out

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

Master-Out/Slave-In

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

Serial Clock

The Serial Clock (SCK) synchronizes data movement both in and out of the device through its MOSI and MISO pins. In MASTER mode, the SPI's Baud Rate Generator creates the serial clock. The Master drives the serial clock out its own SCK pin to the Slave's SCK pin. When the SPI is configured as a Slave, the SCK pin is an input and the clock signal from the Master synchronizes the data transfer between the Master and Slave devices. Slave devices ignore the SCK signal, unless the \overline{SS} pin is asserted. When configured as a slave, the SPI block requires a minimum SCK period of greater than or equal to 8 times the system (XIN) clock period.



140

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

SPI Mode Register

The SPI Mode register (Table 66) configures the character bit width and the direction and value of the \overline{SS} pin.

Table	66.	SPI	Mode	Register	(SPIMODE)	۱
IUNIC	UU .	U I I	mouc	Register		,

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

Reserved—Must be 0.

DIAG-Diagnostic Mode Control bit

This bit is for SPI diagnostics. Setting this bit allows the Baud Rate Generator value to be read using the SPIBRH and SPIBRL register locations.

- 0 = Reading SPIBRH, SPIBRL returns the value in the SPIBRH and SPIBRL registers
- 1 = Reading SPIBRH returns bits [15:8] of the SPI Baud Rate Generator; and reading SPIBRL returns bits [7:0] of the SPI Baud Rate Counter. The Baud Rate Counter High and Low byte values are not buffered.

Caution: *Exercise caution if reading the values while the BRG is counting.*

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

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

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



In order for a receive (read) DMA transaction to send a Not Acknowledge on the last byte, the receive DMA must be set up to receive n-1 bytes, then software must set the NAK bit and receive the last (nth) byte directly.

Start and Stop Conditions

The master (I^2C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I²C Controller generates a START condition by pulling the SDA signal Low while SCL is High. To complete a transaction, the I²C Controller generates a Stop condition by creating a low-to-high transition of the SDA signal while the SCL signal is high. The START and STOP bits in the I²C Control register control the sending of the Start and Stop conditions. A master is also allowed to end one transaction and begin a new one by issuing a Restart. This is accomplished by setting the START bit at the end of a transaction, rather than the STOP bit. Note that the Start condition not sent until the START bit is set and data has been written to the I²C Data register.

Master Write and Read Transactions

The following sections provide a recommended procedure for performing I²C write and read transactions from the I²C Controller (master) to slave I²C devices. In general software should rely on the TDRE, RDRF and NCKI bits of the status register (these bits generate interrupts) to initiate software actions. When using interrupts or DMA, the TXI bit is set to start each transaction and cleared at the end of each transaction to eliminate a 'trailing' Transmit interrupt.

Caution should be used in using the ACK status bit within a transaction because it is difficult for software to tell when it is updated by hardware.

When writing data to a slave, the I²C pauses at the beginning of the Acknowledge cycle if the data register has not been written with the next value to be sent (TDRE bit in the I²C Status register = 1). In this scenario where software is not keeping up with the I²C bus (TDRE asserted longer than one byte time), the Acknowledge clock cycle for byte n is delayed until the Data register is written with byte n + 1, and appears to be grouped with the data clock cycles for byte n+1. If either the START or STOP bit is set, the I²C does not pause prior to the Acknowledge cycle because no additional data is sent.

When a Not Acknowledge condition is received during a write (either during the address or data phases), the I²C Controller generates the Not Acknowledge interrupt (NCKI = 1) and pause until either the STOP or START bit is set. Unless the Not Acknowledge was received on the last byte, the Data register will already have been written with the next address or data byte to send. In this case the FLUSH bit of the Control register should be set at the same time the STOP or START bit is set to remove the stale transmit data and enable subsequent Transmit interrupts.

When reading data from the slave, the I²C pauses after the data Acknowledge cycle until the receive interrupt is serviced and the RDRF bit of the status register is cleared by



162

TXRXSTATE	State Description
0_000	Idle State
0_0001	START State
0_0010	Send/Receive data bit 7
0_0011	Send/Receive data bit 6
0_0100	Send/Receive data bit 5
0_0101	Send/Receive data bit 4
0_0110	Send/Receive data bit 3
0_0111	Send/Receive data bit 2
0_1000	Send/Receive data bit 1
0_1001	Send/Receive data bit 0
0_1010	Data Acknowledge State
0_1011	Second half of data Acknowledge State used only for not acknowledge
0_1100	First part of STOP state
0_1101	Second part of STOP state
0_1110	10-bit addressing: Acknowledge State for 2nd address byte 7-bit addressing: Address Acknowledge State
0_1111	10-bit address: Bit 0 (Least significant bit) of 2nd address byte 7-bit address: Bit 0 (Least significant bit) (R/W) of address byte
1_0000	10-bit addressing: Bit 7 (Most significant bit) of 1st address byte
1_0001	10-bit addressing: Bit 6 of 1st address byte
1_0010	10-bit addressing: Bit 5 of 1st address byte
1_0011	10-bit addressing: Bit 4 of 1st address byte
1_0100	10-bit addressing: Bit 3 of 1st address byte
1_0101	10-bit addressing: Bit 2 of 1st address byte
1_0110	10-bit addressing: Bit 1 of 1st address byte
1_0111	10-bit addressing: Bit 0 (R/W) of 1st address byte
1_1000	10-bit addressing: Acknowledge state for 1st address byte
1_1001	10-bit addressing: Bit 7 of 2nd address byte 7-bit addressing: Bit 7 of address byte
1_1010	10-bit addressing: Bit 6 of 2nd address byte 7-bit addressing: Bit 6 of address byte
1_1011	10-bit addressing: Bit 5 of 2nd address byte 7-bit addressing: Bit 5 of address byte
1_1100	10-bit addressing: Bit 4 of 2nd address byte 7-bit addressing: Bit 4 of address byte



TXRXSTATE	State Description
1_1101	10-bit addressing: Bit 3 of 2nd address byte 7-bit addressing: Bit 3 of address byte
1_1110	10-bit addressing: Bit 2 of 2nd address byte 7-bit addressing: Bit 2 of address byte
1_1111	10-bit addressing: Bit 1 of 2nd address byte 7-bit addressing: Bit 1 of address byte

I²C Diagnostic Control Register

The I²C Diagnostic register (Table 76) provides control over diagnostic modes. This register is a read/write register used for I²C diagnostics.

Table 76. I²C Diagnostic Control Register (I2CDIAG)

BITS	7	6	5	4	3	2	1	0			
FIELD	Reserved										
RESET	0										
R/W	R										
ADDR	F56H										

DIAG = Diagnostic Control Bit - Selects read back value of the Baud Rate Reload registers.

- 0 = NORMAL mode. Reading the Baud Rate High and Low Byte registers returns the baud rate reload value.
- 1 = DIAGNOSTIC mode. Reading the Baud Rate High and Low Byte registers returns the baud rate counter value.



1 = DMAx is enabled and initiates a data transfer upon receipt of a request from the trigger source.

DLE—DMAx Loop Enable

- 0 = DMAx reloads the original Start Address and is then disabled after the End Address data is transferred.
- 1 = DMAx, after the End Address data is transferred, reloads the original Start Address and continues operating.

DDIR—DMAx Data Transfer Direction

- $0 = \text{Register File} \rightarrow \text{on-chip peripheral control register.}$
- 1 = on-chip peripheral control register \rightarrow Register File.

IRQEN—DMAx Interrupt Enable

0 = DMAx does not generate any interrupts.

1 = DMAx generates an interrupt when the End Address data is transferred.

WSEL—Word Select

- 0 = DMAx transfers a single byte per request.
- 1 = DMA*x* transfers a two-byte word per request. The address for the on-chip peripheral control register must be an even address.

RSS—Request Trigger Source Select

The Request Trigger Source Select field determines the peripheral that can initiate a DMA transfer. The corresponding interrupts do not need to be enabled within the Interrupt Controller to initiate a DMA transfer. However, if the Request Trigger Source can enable or disable the interrupt request sent to the Interrupt Controller, the interrupt request must be enabled within the Request Trigger Source block.

- 000 = Timer 0.
- 001 = Timer 1.
- 010 = Timer 2.
- 011 = Timer 3.

100 = DMA0 Control register: UART0 Received Data register contains valid data. DMA1 Control register: UART0 Transmit Data register empty.

101 = DMA0 Control register: UART1 Received Data register contains valid data. DMA1 Control register: UART1 Transmit Data register empty.

- 110 = DMA0 Control register: I²C Receiver Interrupt. DMA1 Control register: I²C
- Transmitter Interrupt register empty.
- 111 = Reserved.

DMAx I/O Address Register

The DMAx I/O Address register (Table 78) contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is given by {FH,

168

zilog

Information Area

Table 91 describes the 64K Series Information Area. This 512-byte Information Area is accessed by setting bit 7 of the Page Select Register to 1. When access is enabled, the Information Area is mapped into Flash Memory and overlays the 512 bytes at addresses FE00H to FFFFH. When the Information Area access is enabled, LDC instructions return data from the Information Area. CPU instruction fetches always comes from Flash Memory regardless of the Information Area access bit. Access to the Information Area is read-only.

Flash Memory Address (Hex)	Function
FE00H-FE3FH	Reserved
FE40H-FE53H	Part Number 20-character ASCII alphanumeric code Left justified and filled with zeros
FE54H-FFFFH	Reserved

Table 91. Z8 Encore! XP 64K Series Flash Microcontrollers Information Area Map

Operation

The Flash Controller provides the proper signals and timing for Byte Programming, Page Erase, and Mass Erase of the Flash memory. The Flash Controller contains a protection mechanism, via the Flash Control register (FCTL), to prevent accidental programming or erasure. The following subsections provide details on the various operations (Lock, Unlock, Sector Protect, Byte Programming, Page Erase, and Mass Erase).



Option Bits

Overview

Option Bits allow user configuration of certain aspects of the 64K Series operation. The feature configuration data is stored in the Flash Memory and read during Reset. The features available for control via the Option Bits are:

- Watchdog Timer time-out response selection-interrupt or Reset.
- Watchdog Timer enabled at Reset.
- The ability to prevent unwanted read access to user code in Flash Memory.
- The ability to prevent accidental programming and erasure of the user code in Flash Memory.
- Voltage Brownout configuration-always enabled or disabled during STOP mode to reduce STOP mode power consumption.
- Oscillator mode selection-for high, medium, and low power crystal oscillators, or external RC oscillator.

Operation

Option Bit Configuration By Reset

Each time the Option Bits are programmed or erased, the device must be Reset for the change to take place. During any reset operation (System Reset, Reset, or Stop Mode Recovery), the Option Bits are automatically read from the Flash Memory and written to Option Configuration registers. The Option Configuration registers control operation of the devices within the 64K Series. Option Bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access.

Option Bit Address Space

The first two bytes of Flash Memory at addresses 0000H (see Table 98 on page 196) and 0001H (see Table 99 on page 197) are reserved for the user Option Bits. The byte at Flash Memory address 0000H configures user options. The byte at Flash Memory address 0001H is reserved for future use and must remain unprogrammed.



210

eZ8 CPU loops on the BRK instruction. 0 = BRK instruction sets DBGMODE to 1. 1 = eZ8 CPU loops on BRK instruction.

Reserved

These bits are reserved and must be 0.

RST—Reset

Setting this bit to 1 resets the 64K Series devices. The devices go 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 the 64K Series device

OCD Status Register

The OCD Status register (Table 103) reports status information about the current state of the debugger and the system.

BITS	7	6	5	4	3	2	1	0					
FIELD	IDLE	HALT	RPEN	Reserved									
RESET		0											
R/W				F	२								

IDLE—CPU idling

This bit is set if the part is in DEBUG mode (DBGMODE is 1), or if a BRK instruction occurred since the last time OCDCTL was written. This can be used to determine if the CPU is running or if it is idling.

0 = The eZ8 CPU is running.

1 = The eZ8 CPU is either stopped or looping on a BRK instruction.

HALT—HALT Mode

0 = The device is not in HALT mode.

1 = The device is in HALT mode.

RPEN—Read Protect Option Bit Enabled

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

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

Reserved

These bits are always 0.

Z8 Encore! XP[®] 64K Series Flash Microcontrollers Product Specification





Table 122. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B).
сс	Condition Code	—	Refer to Condition Codes overview in the eZ8 CPU User Manual.
DA	Direct Address	Addrs	Addrs. represents a number in the range of 0000H to FFFFH
ER	Extended Addressing Register	Reg	Reg. represents a number in the range of 000H to FFFH
IM	Immediate Data	#Data	Data is a number between 00H to FFH
lr	Indirect Working Register	@Rn	n = 0 –15
IR	Indirect Register	@Reg	Reg. represents a number in the range of 00H to FFH
Irr	Indirect Working Register Pair	@RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14
IRR	Indirect Register Pair	@Reg	Reg. represents an even number in the range 00H to FEH
р	Polarity	р	Polarity is a single bit binary value of either 0B or 1B.
r	Working Register	Rn	n = 0 – 15
R	Register	Reg	Reg. represents a number in the range of 00H to FFH
RA	Relative Address	Х	X represents an index in the range of +127 to -128 which is an offset relative to the address of the next instruction
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to -128 range.

Table 123 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.



248

Table 128. CPU Control Instructions

Mnemonic	Operands	Instruction
STOP	_	STOP Mode
WDT	_	Watchdog Timer Refresh

Table 129. Load Instructions

Mnemonic	Operands	Instruction
CLR	dst	Clear
LD	dst, src	Load
LDC	dst, src	Load Constant to/from Program Memory
LDCI	dst, src	Load Constant to/from Program Memory and Auto-Increment Addresses
LDE	dst, src	Load External Data to/from Data Memory
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment Addresses
LDWX	dst, src	Load Word using Extended Addressing
LDX	dst, src	Load using Extended Addressing
LEA	dst, X(src)	Load Effective Address
POP	dst	Рор
POPX	dst	Pop using Extended Addressing
PUSH	SrC	Push
PUSHX	src	Push using Extended Addressing

Table 130. Logical Instructions

Mnemonic	Operands	Instruction
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using Extended Addressing
СОМ	dst	Complement
OR	dst, src	Logical OR
ORX	dst, src	Logical OR using Extended Addressing



250

Table 132. Rotate and Shift Instructions (Continued)

Mnemonic	Operands	Instruction					
SRL dst		Shift Right Logical					
SWAP	dst	Swap Nibbles					

eZ8 CPU Instruction Summary

Table 133 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.

Assembly		Address Mode		- Oncode(s)	Flags						Fetch	Instr
Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Ζ	S	V	D	н	Cycles	Cycles
ADC dst, src	$dst \leftarrow dst + src + C$	r	r	12	*	*	*	*	0	*	2	3
	-	r	lr	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, sro	$dst \leftarrow dst + src + C$	ER	ER	18	*	*	*	*	0	*	4	3
	-	ER	IM	19	•						4	3
ADD dst, src	$dst \gets dst + src$	r	r	02	*	*	*	*	0	*	2	3
	-	r	lr	03	•						2	4
	-	R	R	04	•						3	3
	-	R	IR	05	•						3	4
	-	R	IM	06	•						3	3
		IR	IM	07	•						3	4
ADDX dst, sro	$dst \leftarrow dst + src$	ER	ER	08	*	*	*	*	0	*	4	3
	-	ER	IM	09	•						4	3

Table 133. eZ8 CPU Instruction Summary

Z8 Encore! XP[®] 64K Series Flash Microcontrollers Product Specification



Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 59 and Table 134 on page 262. Figure 60 on page 263 and Figure 61 on page 264 provide information on each of the $eZ8^{TM}$ CPU instructions.



Figure 59. Opcode Map Cell Description