



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 <sup>2</sup> 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f6402ar020ec00tr">https://www.e-xfl.com/product-detail/zilog/z8f6402ar020ec00tr</a>



# *List of Figures*

Figure 1.	Z8 Encore!® Block Diagram . . . . .	3
Figure 2.	Z8Fxx01 in 40-Pin Dual Inline Package (DIP) . . . . .	7
Figure 3.	Z8Fxx01 in 44-Pin Plastic Leaded Chip Carrier (PLCC) . . . . .	8
Figure 4.	Z8Fxx01 in 44-Pin Low-Profile Quad Flat Package (LQFP) . . . . .	9
Figure 5.	Z8Fxx02 in 64-Pin Low-Profile Quad Flat Package (LQFP) . . . . .	10
Figure 6.	Z8Fxx02 in 68-Pin Plastic Leaded Chip Carrier (PLCC) . . . . .	11
Figure 7.	Z8Fxx03 in 80-Pin Quad Flat Package (QFP) . . . . .	12
Figure 8.	Power-On Reset Operation . . . . .	27
Figure 9.	Voltage Brown-Out Reset Operation . . . . .	28
Figure 10.	GPIO Port Pin Block Diagram . . . . .	34
Figure 11.	Interrupt Controller Block Diagram . . . . .	46
Figure 12.	Timer Block Diagram . . . . .	58
Figure 13.	UART Block Diagram . . . . .	79
Figure 14.	UART Asynchronous Data Format without Parity . . . . .	80
Figure 15.	UART Asynchronous Data Format with Parity . . . . .	80
Figure 16.	UART Asynchronous Multiprocessor (9-bit) Mode Data Format . . . . .	84
Figure 17.	Infrared Data Communication System Block Diagram . . . . .	95
Figure 18.	Infrared Data Transmission . . . . .	97
Figure 19.	Infrared Data Reception . . . . .	98
Figure 20.	SPI Configured as a Master in a Single Master, Single Slave System . . . . .	99
Figure 21.	SPI Configured as a Master in a Single Master, Multiple Slave System . . . . .	100
Figure 22.	SPI Configured as a Slave . . . . .	100
Figure 23.	SPI Timing When PHASE is 0 . . . . .	103
Figure 24.	SPI Timing When PHASE is 1 . . . . .	104
Figure 25.	7-Bit Addressed Slave Data Transfer Format . . . . .	113
Figure 26.	10-Bit Addressed Slave Data Transfer Format . . . . .	114
Figure 27.	Receive Data Transfer Format for a 7-Bit Addressed Slave . . . . .	115
Figure 28.	Receive Data Format for a 10-Bit Addressed Slave . . . . .	116
Figure 29.	Analog-to-Digital Converter Block Diagram . . . . .	133
Figure 30.	Flash Memory Arrangement . . . . .	139



Table 101. Absolute Maximum Ratings . . . . .	167
Table 102. DC Characteristics . . . . .	169
Table 103. AC Characteristics . . . . .	172
Table 104. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing . . . . .	173
Table 105. Flash Memory Electrical Characteristics and Timing . . . . .	173
Table 106. Watch-Dog Timer Electrical Characteristics and Timing . . . . .	174
Table 107. Analog-to-Digital Converter Electrical Characteristics and Timing . . . . .	174
Table 108. GPIO Port Input Timing . . . . .	176
Table 109. GPIO Port Output Timing . . . . .	177
Table 110. On-Chip Debugger Timing . . . . .	178
Table 111. SPI Master Mode Timing . . . . .	179
Table 112. SPI Slave Mode Timing . . . . .	180
Table 113. I2C Timing . . . . .	181
Table 114. Assembly Language Syntax Example 1 . . . . .	183
Table 115. Assembly Language Syntax Example 2 . . . . .	183
Table 116. Notational Shorthand . . . . .	184
Table 117. Additional Symbols . . . . .	185
Table 118. Condition Codes . . . . .	186
Table 119. Arithmetic Instructions . . . . .	187
Table 120. Bit Manipulation Instructions . . . . .	188
Table 121. Block Transfer Instructions . . . . .	188
Table 122. CPU Control Instructions . . . . .	189
Table 123. Load Instructions . . . . .	189
Table 124. Logical Instructions . . . . .	190
Table 125. Program Control Instructions . . . . .	190
Table 126. Rotate and Shift Instructions . . . . .	191
Table 127. eZ8 CPU Instruction Summary . . . . .	191
Table 128. Opcode Map Abbreviations . . . . .	203
Table 129. Ordering Information . . . . .	211

## Program Memory

The eZ8 CPU supports 64KB of Program Memory address space. The Z8F640x family devices contain 16KB to 64KB of on-chip Flash memory in the Program Memory address space. Reading from Program Memory addresses outside the available Flash memory addresses returns FFH. Writing to these unemployments Program Memory addresses produces no effect. Table 4 describes the Program Memory Maps for the Z8F640x family products.

**Table 4. Z8F640x Family Program Memory Maps**

Program Memory Address (Hex)	Function
<b>Z8F160x Products</b>	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFFH	Program Memory
<b>Z8F240x Products</b>	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFFH	Program Memory
<b>Z8F320x Products</b>	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFFH	Program Memory
* See Table 22 on page 45 for a list of the interrupt vectors.	

**Table 4. Z8F640x Family Program Memory Maps (Continued)**

Program Memory Address (Hex)	Function
<b>Z8F480x Products</b>	
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
<b>Z8F640x Products</b>	
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.

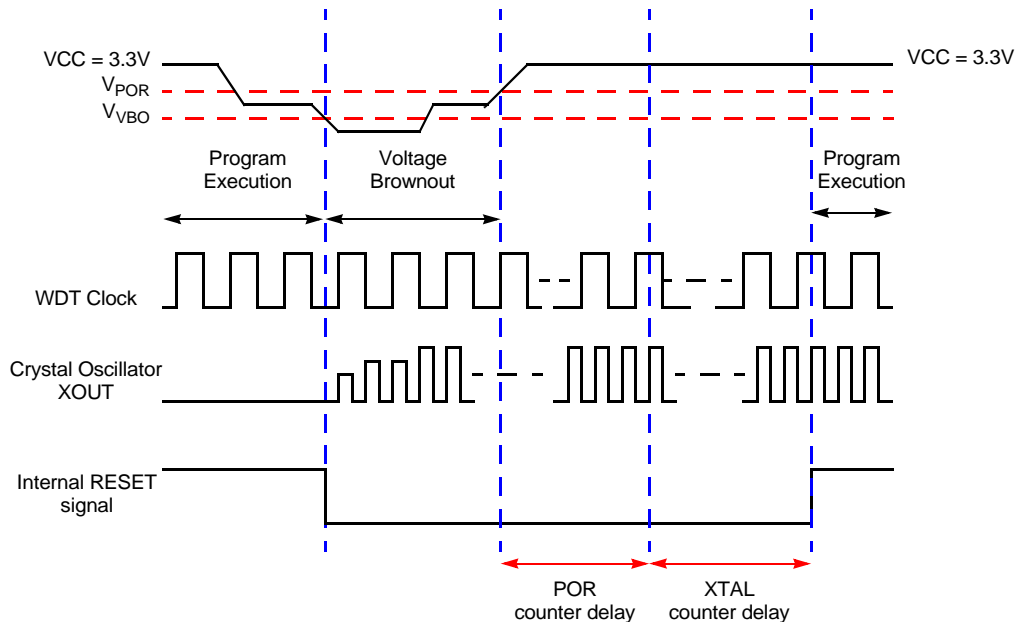
**Table 5. Z8F640x family Data Memory Maps**

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

threshold voltage,  $V_{VBO}$ ) and forces the device into the Reset state. While the supply voltage remains below the Power-On Reset voltage threshold ( $V_{POR}$ ), the VBO block holds the Z8F640x family device in the Reset state.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the Z8F640x family device progresses through a full System Reset sequence, as described in the Power-On Reset section. Following Power-On Reset, the POR status bit in the Watch-Dog Timer Control (WDTCTL) register is set to 1. Figure 63 illustrates Voltage Brown-Out operation. Refer to the **Electrical Characteristics** chapter for the VBO and POR threshold voltages ( $V_{VBO}$  and  $V_{POR}$ ).

Stop mode disables the Voltage Brown-Out detector.



**Figure 63. Voltage Brown-Out Reset Operation (not to scale)**

### Watch-Dog Timer Reset

If the device is in normal or Halt mode, the Watch-Dog Timer can initiate a System Reset at time-out if the WDT\_RES Option Bit is set to 1. This is the default (unprogrammed) setting of the WDT\_RES Option Bit. The WDT status bit in the WDT Control register is set to signify that the reset was initiated by the Watch-Dog Timer.

## 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)**

<b>BITS</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>FIELD</b>	T2I	T1I	T0I	U0RXI	U0TXI	I2CI	SPII	ADCI
<b>RESET</b>	0	0	0	0	0	0	0	0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>ADDR</b>	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.

Middle byte, Bits[15:8], of the 24-bit WDT reload value.

**Table 49. Watch-Dog Timer Reload Low Byte Register (WDTL)**

BITS	7	6	5	4	3	2	1	0
FIELD	WDTL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
ADDR	FF3H							
R/W* - Read returns the current WDT count value. Write sets the desired Reload Value.								

WDTL—WDT Reload Low

Least significant byte (LSB), Bits[7:0], of the 24-bit WDT reload value.





4. The I<sup>2</sup>C Controller loads the I<sup>2</sup>C Shift register with the contents of the I<sup>2</sup>C Data register.
5. After the first bit has been shifted out, a Transmit interrupt is asserted.
6. Software responds by writing eight bits of address to the I<sup>2</sup>C Data register.
7. The I<sup>2</sup>C Controller completes shifting of the two address bits and a 0 (write).
8. The I<sup>2</sup>C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL.
9. The I<sup>2</sup>C Controller loads the I<sup>2</sup>C Shift register with the contents of the I<sup>2</sup>C Data register.
10. The I<sup>2</sup>C Controller shifts out the next eight bits of address. After the first bits are shifted, the I<sup>2</sup>C Controller generates a Transmit interrupt.
11. Software responds by setting the START bit of the I<sup>2</sup>C Control register to generate a repeated START.
12. Software responds by writing 11110B followed by the 2-bit slave address and a 1 (read).
13. Software responds by setting the NAK bit of the I<sup>2</sup>C Control register, so that a Not Acknowledge is sent after the first byte of data has been read. If you want to read only one byte, software responds by setting the NAK bit of the I<sup>2</sup>C Control register.
14. After the I<sup>2</sup>C Controller shifts out the address bits mentioned in step 9, the I<sup>2</sup>C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL.
15. The I<sup>2</sup>C Controller sends the repeated START condition.
16. The I<sup>2</sup>C Controller loads the I<sup>2</sup>C Shift register with the contents of the I<sup>2</sup>C Data register.
17. The I<sup>2</sup>C Controller sends 11110B followed by the 2-bit slave read and a 1 (read).
18. The I<sup>2</sup>C slave sends an acknowledge by pulling the SDA signal Low during the next high period of SCL.
19. The I<sup>2</sup>C slave sends a byte of data.
20. A Receive interrupt is generated.
21. Software responds by reading the I<sup>2</sup>C Data register.
22. Software responds by setting the STOP bit of the I<sup>2</sup>C Control register.
23. A NAK condition is sent to the I<sup>2</sup>C slave.
24. A STOP condition is sent to the I<sup>2</sup>C slave.

## ADC Data High Byte Register

The ADC Data High Byte register contains the upper eight bits of the 10-bit ADC output. During a conversion, this value is invalid. Access to the ADC Data High Byte register is read-only. The full 10-bit ADC result is given by {ADCD\_H[7:0], ADCD\_L[7:6]}.

**Table 81. ADC Data High Byte Register (ADCD\_H)**

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_H							
RESET	X							
R/W	R							
ADDR	F72H							

ADCD\_H—ADC Data High Byte

This byte contains the upper eight bits of the 10-bit ADC output. These bits are not valid during a conversion. These bits are undefined after a Reset.

## ADC Data Low Bits Register

The ADC Data Low Bits register contains the lower two bits of the conversion value. During a conversion this value is invalid. Access to the ADC Data Low Bits register is read-only. The full 10-bit ADC result is given by {ADCD\_H[7:0], ADCD\_L[7:6]}.

**Table 82. ADC Data Low Bits Register (ADCD\_L)**

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_L		Reserved					
RESET	X		X					
R/W	R		R					
ADDR	F73H							

ADCD\_L—ADC Data Low Bits

These are the least significant two bits of the 10-bit ADC output. During a conversion, this value is invalid. These bits are undefined after a Reset.

Reserved

These bits are reserved and are always undefined.



## Flash Memory

### Overview

The Z8F640x family features up to 64KB (65,536 bytes) of non-volatile Flash memory with read/write/erase capability. The Flash Memory can be programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash memory array is arranged in pages with 512 bytes per page. The 512-byte page is the minimum Flash block size that can be erased. Each page is divided into 8 rows of 64 bytes. The Flash memory also contains a High Sector that can be enabled for writes and erase separately from the rest of the Flash array. The first 2 bytes of the Flash Program memory are used as Option Bits. Refer to the **Option Bits** chapter for more information on their operation.

Table 83 describes the Flash memory configuration for each device in the Z8F640x family. Figure 84 illustrates the Flash memory arrangement.

**Table 83. Z8F640x family Flash Memory Configurations**

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash High Sector Size KB (Bytes)	High Sector Addresses
Z8F160x	16 (16,384)	32	0000H - 3FFFH	1 (1024)	3C00H - 3FFFH
Z8F240x	24 (24,576)	48	0000H - 5FFFH	2 (2048)	5800H - 5FFFH
Z8F320x	32 (32,768)	64	0000H - 7FFFH	2 (2048)	7800H - 7FFFH
Z8F480x	48 (49,152)	96	0000H - BFFFH	4 (4096)	B000H - BFFFH
Z8F640x	64 (65,536)	128	0000H - FFFFH	8 (8192)	E000H - FFFFH



## Flash Status Register

The Flash Status register indicates the current state of the Flash Controller. This register can be read at any time. The Read-only Flash Status Register shares its Register File address with the Write-only Flash Control Register.

**Table 86. Flash Status Register (FSTAT)**

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved		FSTAT					
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
ADDR	FF8H							

Reserved

These bits are reserved and must be 0.

FSTAT—Flash Controller Status

000000 = Flash Controller locked.

000001 = First unlock command received.

000010 = Flash Controller unlocked (second unlock command received).

001xxx = Program operation in progress.

010xxx = Page erase operation in progress.

100xxx = Mass erase operation in progress.

Table 93. On-Chip Debugger Commands

Debug Command	Command Byte	Enabled when NOT in Debug mode?	Disabled by Read Protect Option Bit
Write Program Memory	0AH	-	Disabled
Read Program Memory	0BH	-	Disabled
Write Data Memory	0CH	-	Yes
Read Data Memory	0DH	-	-
Read Program Memory CRC	0EH	-	-
Reserved	0FH	-	-
Step Instruction	10H	-	Disabled
Stuff Instruction	11H	-	Disabled
Execute Instruction	12H	-	Disabled
Reserved	13H - 1FH	-	-
Write Watchpoint	20H	-	Disabled
Read Watchpoint	21H	-	-
Reserved	22H - FFH	-	-

In the following bulleted list of OCD Commands, data and commands sent from the host to the On-Chip Debugger are identified by 'DBG <-- Command/Data'. Data sent from the On-Chip Debugger back to the host is identified by 'DBG --> Data'

- Read OCD Revision (00H)**—The Read OCD Revision command is used to determine the version of the On-Chip Debugger. If OCD commands are added, removed, or changed, this revision number changes.
 

```
DBG <-- 00H
DBG --> OCDREV[15:8] (Major revision number)
DBG --> OCDREV[7:0] (Minor revision number)
```
- Read OCD Status Register (02H)**—The Read OCD Status Register command is used to read the OCDSTAT register.
 

```
DBG <-- 02H
DBG --> OCDSTAT[7:0]
```
- Read Runtime Counter (03H)**—The Runtime Counter is used to count Z8 Encore! system clock cycles in between Breakpoints. The 16-bit Runtime Counter counts up from 0000H and stops at the maximum count of FFFFH. The Runtime Counter is overwritten during the Write Memory, Read Memory, Write Register, Read Register, Read Memory CRC, Step Instruction, Stuff Instruction, and Execute Instruction commands.

```
DBG <-- 03H
DBG --> RuntimeCounter[15:8]
DBG --> RuntimeCounter[7:0]
```

- **Write OCD Control Register (04H)**—The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of putting the Z8F640x family device back into normal operating mode is to reset the device.

```
DBG <-- 04H
DBG <-- OCDCTL[7:0]
```

- **Read OCD Control Register (05H)**—The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG <-- 05H
DBG --> OCDCTL[7:0]
```

- **Write Program Counter (06H)**—The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the Z8F640x family device is not in Debug mode or if the Read Protect Option Bit is enabled, the Program Counter (PC) values are discarded.

```
DBG <-- 06H
DBG <-- ProgramCounter[15:8]
DBG <-- ProgramCounter[7:0]
```

- **Read Program Counter (07H)**—The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the Z8F640x family device is not in Debug mode or if the Read Protect Option Bit is enabled, this command returns FFFFH.

```
DBG <-- 07H
DBG --> ProgramCounter[15:8]
DBG --> ProgramCounter[7:0]
```

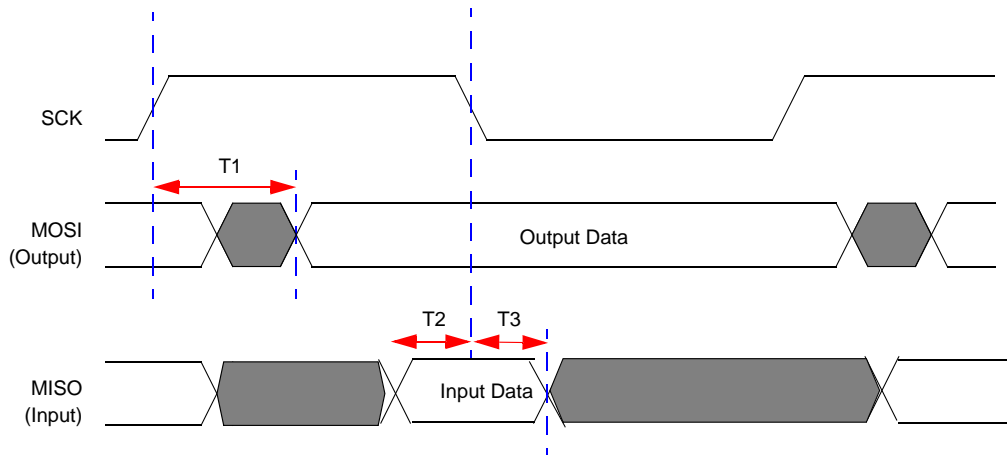
- **Write Register (08H)**—The Write Register command writes data to the Register File. Data can be written 1-256 bytes at a time (256 bytes can be written by setting size to zero). If the Z8F640x family device is not in Debug mode, the address and data values are discarded. If the Read Protect Option Bit is enabled, then only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG <-- 08H
DBG <-- {4'h0, Register Address[11:8]}
DBG <-- Register Address[7:0]
DBG <-- Size[7:0]
DBG <-- 1-256 data bytes
```

- **Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1-256 bytes at a time (256 bytes can be read by setting size to

## 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**

**Table 110. SPI Master Mode Timing**

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



# *eZ8 CPU Instruction Set*

## Assembly Language Programming Introduction

The eZ8 CPU assembly language provides a means for writing an application program without having to be concerned with actual memory addresses or machine instruction formats. A program written in assembly language is called a source program. Assembly language allows the use of symbolic addresses to identify memory locations. It also allows mnemonic codes (opcodes and operands) to represent the instructions themselves. The opcodes identify the instruction while the operands represent memory locations, registers, or immediate data values.

Each assembly language program consists of a series of symbolic commands called statements. Each statement can contain labels, operations, operands and comments.

Labels can be assigned to a particular instruction step in a source program. The label identifies that step in the program as an entry point for use by other instructions.

The assembly language also includes assembler directives that supplement the machine instruction. The assembler directives, or pseudo-ops, are not translated into a machine instruction. Rather, the pseudo-ops are interpreted as directives that control or assist the assembly process.

The source program is processed (assembled) by the assembler to obtain a machine language program called the object code. The object code is executed by the eZ8 CPU. An example segment of an assembly language program is detailed in the following example.

### Assembly Language Source Program Example

```
JP START      ; Everything after the semicolon is a comment.

START:        ; A label called "START". The first instruction (JP START) in this
              ; example causes program execution to jump to the point within the
              ; program where the START label occurs.

LD R4, R7     ; A Load (LD) instruction with two operands. The first operand,
              ; Working Register R4, is the destination. The second operand,
              ; Working Register R7, is the source. The contents of R7 is
              ; written into R4.

LD 234H, #01  ; Another Load (LD) instruction with two operands.
              ; The first operand, Extended Mode Register Address 234H,
              ; identifies the destination. The second operand, Immediate Data
```



**Table 121. CPU Control Instructions**

Mnemonic	Operands	Instruction
CCF	—	Complement Carry Flag
DI	—	Disable Interrupts
EI	—	Enable Interrupts
HALT	—	Halt Mode
NOP	—	No Operation
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
SRP	src	Set Register Pointer
STOP	—	Stop Mode
WDT	—	Watch-Dog Timer Refresh

**Table 122. 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
LDX	dst, src	Load using Extended Addressing
LEA	dst, X(src)	Load Effective Address
POP	dst	Pop
POPX	dst	Pop using Extended Addressing
PUSH	src	Push
PUSHX	src	Push using Extended Addressing



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		
ADD dst, src	$\text{dst} \leftarrow \text{dst} + \text{src}$	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	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, src	$\text{dst} \leftarrow \text{dst} + \text{src}$	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3
AND dst, src	$\text{dst} \leftarrow \text{dst AND src}$	r	r	52	-	*	*	0	-	-	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	$\text{dst} \leftarrow \text{dst AND src}$	ER	ER	58	-	*	*	0	-	-	4	3
		ER	IM	59							4	3
BCLR bit, dst	$\text{dst}[\text{bit}] \leftarrow 0$	r		E2	-	*	*	0	-	-	2	2
BIT p, bit, dst	$\text{dst}[\text{bit}] \leftarrow \text{p}$	r		E2	-	*	*	0	-	-	2	2
BRK	Debugger Break			00	-	-	-	-	-	-	1	1
BSET bit, dst	$\text{dst}[\text{bit}] \leftarrow 1$	r		E2	-	*	*	0	-	-	2	2
BSWAP dst	$\text{dst}[7:0] \leftarrow \text{dst}[0:7]$	R		D5	X	*	*	0	-	-	2	2
BTJ p, bit, src, dst	if $\text{src}[\text{bit}] = \text{p}$ $\text{PC} \leftarrow \text{PC} + \text{X}$	r		F6	-	-	-	-	-	-	3	3
		Ir		F7							3	4
BTJNZ bit, src, dst	if $\text{src}[\text{bit}] = 1$ $\text{PC} \leftarrow \text{PC} + \text{X}$	r		F6	-	-	-	-	-	-	3	3
		Ir		F7							3	4
Flags Notation:	* = Value is a function of the result of the operation. - = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							



Table 128. Ordering Information (Continued)

Part	Flash KB (Bytes)	RAM KB (Bytes)	Max. Speed (MHz)	Temp (°C)	Voltage (V)	Package	Part Number
<b>Z8 Encore!<sup>®</sup> with 64KB Flash, Standard Temperature</b>							
Z8 Encore! <sup>®</sup>	64 (65,536)	4 (4096)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F6401PM020SC
Z8 Encore! <sup>®</sup>	64 (65,536)	4 (4096)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F6401AN020SC
Z8 Encore! <sup>®</sup>	64 (65,536)	4 (4096)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F6401VN020SC
Z8 Encore! <sup>®</sup>	64 (65,536)	4 (4096)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F6402AR020SC
Z8 Encore! <sup>®</sup>	64 (65,536)	4 (4096)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F6402VS020SC
Z8 Encore! <sup>®</sup>	64 (65,536)	4 (4096)	20	0 to +70	3.0 - 3.6	QFP-80	Z8F6403FT020SC
<b>Z8 Encore!<sup>®</sup> with 16KB Flash, Extended Temperature</b>							
Z8 Encore! <sup>®</sup>	16 (16,384)	2 (2048)	20	-40 to +105	3.0 - 3.6	PDIP-40	Z8F1601PM020EC
Z8 Encore! <sup>®</sup>	16 (16,384)	2 (2048)	20	-40 to +105	3.0 - 3.6	LQFP-44	Z8F1601AN020EC
Z8 Encore! <sup>®</sup>	16 (16,384)	2 (2048)	20	-40 to +105	3.0 - 3.6	PLCC-44	Z8F1601VN020EC
Z8 Encore! <sup>®</sup>	16 (16,384)	2 (2048)	20	-40 to +105	3.0 - 3.6	LQFP-64	Z8F1602AR020EC
Z8 Encore! <sup>®</sup>	16 (16,384)	2 (2048)	20	-40 to +105	3.0 - 3.6	PLCC-68	Z8F1602VS020EC
<b>Z8 Encore!<sup>®</sup> with 24KB Flash, Extended Temperature</b>							
Z8 Encore! <sup>®</sup>	24 (24,576)	2 (2048)	20	-40 to +105	3.0 - 3.6	PDIP-40	Z8F2401PM020EC
Z8 Encore! <sup>®</sup>	24 (24,576)	2 (2048)	20	-40 to +105	3.0 - 3.6	LQFP-44	Z8F2401AN020EC
Z8 Encore! <sup>®</sup>	24 (24,576)	2 (2048)	20	-40 to +105	3.0 - 3.6	PLCC-44	Z8F2401VN020EC
Z8 Encore! <sup>®</sup>	24 (24,576)	2 (2048)	20	-40 to +105	3.0 - 3.6	LQFP-64	Z8F2402AR020EC
Z8 Encore! <sup>®</sup>	24 (24,576)	2 (2048)	20	-40 to +105	3.0 - 3.6	PLCC-68	Z8F2402VS020EC
<b>Z8 Encore!<sup>®</sup> with 32KB Flash, Extended Temperature</b>							
Z8 Encore! <sup>®</sup>	32 (32,768)	2 (2048)	20	-40 to +105	3.0 - 3.6	PDIP-40	Z8F3201PM020EC
Z8 Encore! <sup>®</sup>	32 (32,768)	2 (2048)	20	-40 to +105	3.0 - 3.6	LQFP-44	Z8F3201AN020EC
Z8 Encore! <sup>®</sup>	32 (32,768)	2 (2048)	20	-40 to +105	3.0 - 3.6	PLCC-44	Z8F3201VN020EC
Z8 Encore! <sup>®</sup>	32 (32,768)	2 (2048)	20	-40 to +105	3.0 - 3.6	LQFP-64	Z8F3202AR020EC
Z8 Encore! <sup>®</sup>	32 (32,768)	2 (2048)	20	-40 to +105	3.0 - 3.6	PLCC-68	Z8F3202VS020EC



- cc 184
- CCF 189
- characteristics, electrical 167
- clear 189
- clock phase (SPI) 102
- CLR 189
- COM 190
- compare 71
- compare - extended addressing 187
- compare mode 71
- compare with carry 187
- compare with carry - extended addressing 187
- complement 190
- complement carry flag 188, 189
- condition code 184
- continuous assertion interrupt sources 47
- continuous conversion (ADC) 134
- continuous mode 70
- control register definition, UART 86
- control register, I2C 119
- counter modes 70
- CP 187
- CPC 187
- CPCX 187
- CPU and peripheral overview 3
- CPU control instructions 189
- CPX 187
- customer feedback form 216
- customer information 216
- customer service 213

## D

- DA 184, 187
- data memory 19
- data register, I2C 118
- DC characteristics 169
- debugger, on-chip 151
- DEC 187
- decimal adjust 187
- decrement 187
- decrement and jump non-zero 190
- decrement word 187
- DECW 187

- destination operand 185
- device, port availability 33
- DI 189
- direct address 184
- direct memory access controller 122
- disable interrupts 189
- DJNZ 190
- DMA
  - address high nibble register 126
  - configuring for DMA\_ADC data transfer 124
  - configuring DMA0-1 data transfer 123
  - control of ADC 135
  - control register 124
  - control register definitions 124
  - controller 5
  - DMA\_ADC address register 128
  - DMA\_ADC control register 130
  - DMA\_ADC operation 123
  - end address low byte register 128
  - I/O address register 125
  - operation 122
  - start/current address low byte register 127
  - status register 131
- DMAA\_STAT register 131
- DMACTL register 130
- DMAxCTL register 124
- DMAxEND register 128
- DMAxH register 126
- DMAxI/O address (DMAxIO) 126
- DMAxIO register 126
- DMAxSTART register 128
- document number description 215
- dst 185

## E

- EI 189
- electrical characteristics 167
  - ADC 174
  - flash memory and timing 173
  - GPIO input data sample timing 176
  - watch-dog timer 174
- enable interrupt 189
- ER 184



- gated mode 64, 71
- one-shot mode 58, 70
- operating mode 58
- PWM mode 61, 70
- reading the timer count values 66
- reload high and low byte registers 67
- timer control register definitions 66
- timer output signal operation 66
- timers 0-3
  - control registers 70
  - high and low byte registers 66, 69
- TM, TMX 188
- tools, hardware and software 214
- transmit
  - IrDA data 96
- transmit interrupt 112
- transmitting UART data-pollled method 80
- transmitting UART data-interrupt-driven method 81
- TRAP 190

## U

- UART 4
  - architecture 78
  - asynchronous data format without/with parity 80
  - baud rate generator 85
  - baud rates table 93
  - control register definitions 86
  - controller signals 14
  - data format 79
  - interrupts 85
  - multiprocessor mode 84
  - receiving data using DMA controller 83
  - receiving data using interrupt-driven method 82
  - receiving data using the polled method 82
  - transmitting data using the interrupt-driven method 81
  - transmitting data using the polled method 80
  - x baud rate high and low registers 91
  - x control 0 and control 1 registers 89
  - x status 0 and status 1 registers 87
- UxBRH register 91

- UxBRL register 92
- UxCTL0 register 89
- UxCTL1 register 90
- UxRXD register 87
- UxSTAT0 register 87
- UxSTAT1 register 89
- UxTXD register 86

## V

- vector 184
- voltage brown-out reset (VBR) 27

## W

- watch-dog timer
  - approximate time-out delays 72, 73
  - CNTL 28
  - control register 75
  - electrical characteristics and timing 174
  - interrupt in normal operation 73
  - interrupt in stop mode 73
  - operation 72
  - refresh 73, 189
  - reload unlock sequence 74
  - reload upper, high and low registers 76
  - reset 28
  - reset in normal operation 74
  - reset in stop mode 74
  - time-out response 73
- WDTCTL register 75
- WDTH register 76
- WDTL register 77
- working register 184
- working register pair 184
- WTDU register 76

## X

- X 184
- XOR 190
- XORX 190