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	31
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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/z8f6401pm020ec
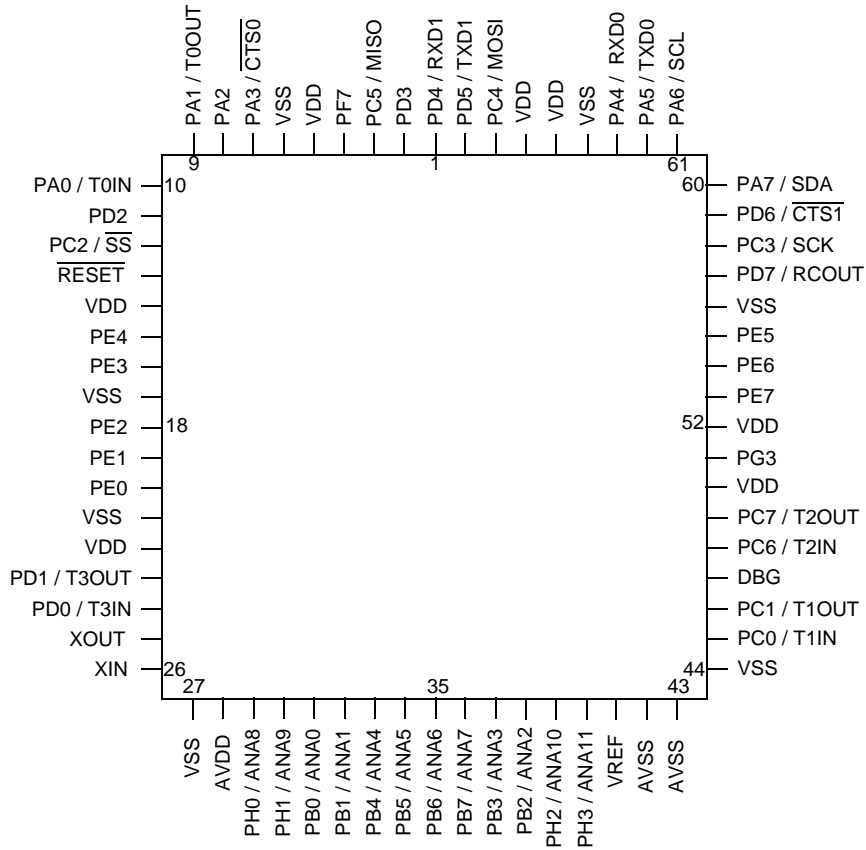
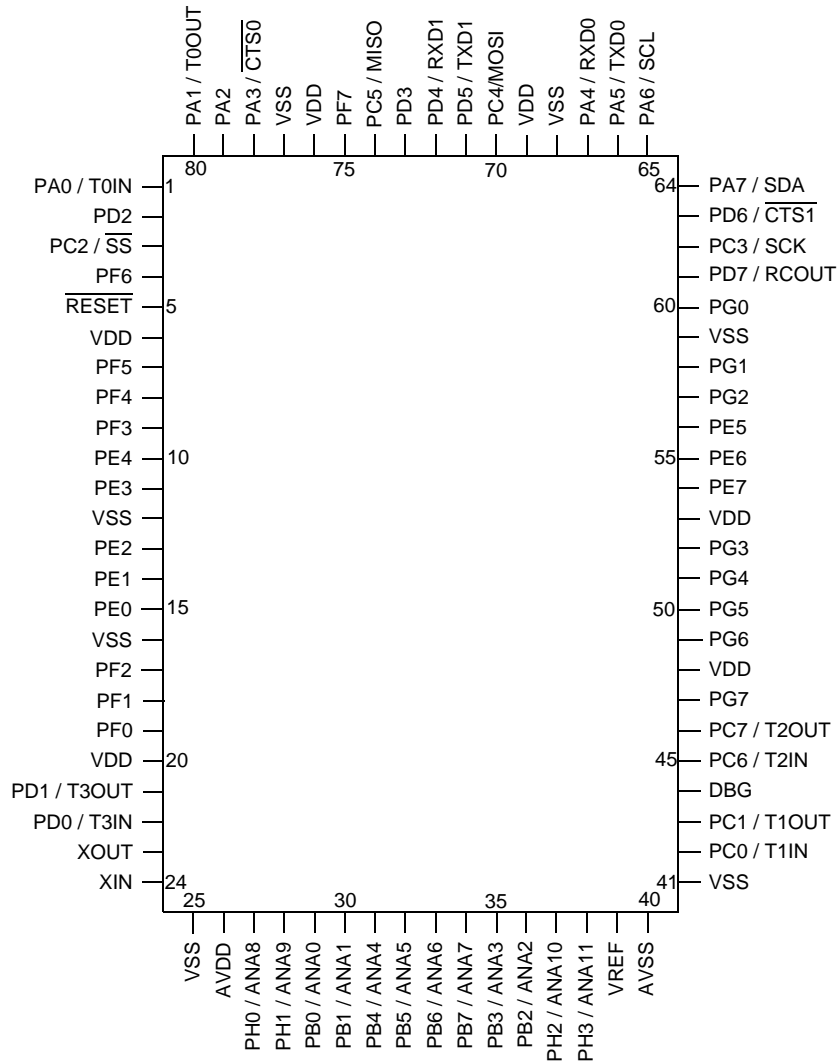**Figure 60. Z8Fxx02 in 68-Pin Plastic Leaded Chip Carrier (PLCC)**

Figure 61. Z8Fxx03 in 80-Pin Quad Flat Package (QFP)

Table 4. Z8F640x Family Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F480x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-BFFFH	Program Memory
Z8F640x Products	
0000-0001	Flash Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-FFFFH	Program Memory

* See Table 22 on page 45 for a list of the interrupt vectors.

Data Memory

The Z8F640x family devices contain 128 bytes of read-only memory at the top of the eZ8 CPU's 64KB Data Memory address space. The eZ8 CPU's LDE and LDEI instructions provide access to the Data Memory information. Table 5 describes the Z8F640x family's Data Memory Map.

Table 5. Z8F640x family Data Memory Maps

Data Memory Address (Hex)	Function
0000H-FFBFH	Reserved
FFC0H-FFD3H	Part Number 20-character ASCII alphanumeric code Left justified and filled with zeros
FFD4H-FFFFH	Reserved

System and Short Resets

During a System Reset, 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). A Short Reset differs from a System Reset only in the number of Watch-Dog Timer oscillator cycles required to exit Reset. A Short Reset requires only 66 Watch-Dog Timer oscillator cycles. Unless specifically stated otherwise, System Reset and Short Reset are referred to collectively as Reset.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watch-Dog Timer oscillator continue to run. The system clock begins operating following the Watch-Dog Timer oscillator cycle count. The eZ8 CPU and on-chip peripherals remain idle through the 16 cycles of the system clock.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. 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.

Reset Sources

Table 8 lists the reset sources and type of Reset as a function of the Z8F640x family device operating mode. The text following provides more detailed information on the individual Reset sources. Please note that Power-On Reset / Voltage Brown-Out events always have priority over all other possible reset sources to insure a full system reset occurs.

Table 8. Reset Sources and Resulting Reset Type

Operating Mode	Reset Source	Reset Type
Normal or Halt modes	Power-On Reset / Voltage Brown-Out	System Reset
	Watch-Dog Timer time-out when configured for Reset	Short Reset
	RESET pin assertion	Short Reset
	On-Chip Debugger initiated Reset (OCDCTL[1] set to 1)	System Reset except the On-Chip Debugger is unaffected by the reset
Stop mode	Power-On Reset / Voltage Brown-Out	System Reset
	RESET pin assertion	System Reset
	DBG pin driven Low	System Reset

Table 11. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description
Port D	PD0	T3IN	Timer 3 In (not available in 40- and 44-pin packages)
	PD1	T3OUT	Timer 3 Out (not available in 40- and 44-pin packages)
	PD2	N/A	No alternate function
	PD3	N/A	No alternate function
	PD4	RXD1 / IRRX1	UART 1 / IrDA 1 Receive Data
	PD5	TXD1 / IRTX1	UART 1 / IrDA 1 Transmit Data
	PD6	$\overline{CTS1}$	UART 1 Clear to Send
	PD7	RCOUT	Watch-Dog Timer RC Oscillator Output
Port E	PE[7:0]	N/A	No alternate functions
Port F	PF[7:0]	N/A	No alternate functions
Port G	PG[7:0]	N/A	No alternate functions
Port H	PH0	ANA8	ADC Analog Input 8
	PH1	ANA9	ADC Analog Input 9
	PH2	ANA10	ADC Analog Input 10
	PH3	ANA11	ADC Analog Input 11

GPIO Interrupts

Many of the GPIO port pins can be used as interrupt sources. Some port pins may be configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port pin interrupts generate an interrupt when any edge occurs (both rising and falling). Refer to the **Interrupt Controller** chapter for more information on interrupts using the GPIO pins.

GPIO Control Register Definitions

Four registers for each Port provide access to GPIO control, input data, and output data. Table 12 lists these Port registers. Use the Port A-H Address and Control registers together to provide access to sub-registers for Port configuration and control.

AF[7:0]—Port Alternate Function enabled
0 = The port pin is in normal mode and the DDx bit in the Port A-H Data Direction sub-register determines the direction of the pin.
1 = The alternate function is selected. Port pin operation is controlled by the alternate function.

Port A-H Output Control Sub-Registers

The Port A-H Output Control sub-register (Table 17) is accessed through the Port A-H Control register by writing 03H to the Port A-H Address register. Setting the bits in the Port A-H Output Control sub-registers to 1 configures the specified port pins for open-drain operation. These sub-registers affect the pins directly and, as a result, alternate functions are also affected.

Table 17. Port A-H Output Control Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
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 03H in Port A-H Address Register, accessible via Port A-H Control Register							

POC[7:0]—Port Output Control
These bits function independently of the alternate function bit and disables the drains if set to 1.
0 = The drains are enabled for any output mode.
1 = The drain of the associated pin is disabled (open-drain mode).

UART Interrupts

The UART features separate interrupts for the transmitter and the receiver. In addition, when the UART primary functionality is disabled, the Baud Rate Generator can also function as a basic timer with interrupt capability.

Transmitter Interrupts

The transmitter generates an interrupt anytime the Transmit Data Register Empty bit (TDRE) is set to 1. This indicates that the transmitter is ready to accept new data for transmission. Writing to the UART Transmit Data register clears the UART Transmit interrupt.

Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte has been received and is available in the UART Receive Data register. This interrupt can be disabled independent of the other receiver interrupt sources.

- A break is received.

- An overrun is detected.

- A data framing error is detected.

Baud Rate Generator Interrupts

If the Baud Rate Generator (BRG) interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UARTx Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

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

When the UART is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 register to 0.

2. Load the desired 16-bit count value into the UART Baud Rate High and Low Byte registers.

3. Enable the Baud Rate Generator timer function and associated interrupt by setting the
BIRQ bit in the UARTx Control 1 register to 1.

UART Control Register Definitions

The UART control registers support both the UARTs and the associated Infrared Encoder/
Decoders. For more information on the infrared operation, refer to the **Infrared Encoder/
Decoder** chapter on page 95.

UARTx Transmit Data Register

Data bytes written to the UARTx Transmit Data register (Table 50) are shifted out on the
TXDx pin. The Write-only UARTx Transmit Data register shares a Register File address
with the Read-only UARTx Receive Data register.

Table 50. UARTx Transmit Data Register (UxTXD)

BITS	7	6	5	4	3	2	1	0
FIELD	TXD							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
ADDR	F40H and F48H							

TXD—Transmit Data
UART transmitter data byte to be shifted out through the TXDx pin.

UARTx Receive Data Register

Data bytes received through the RXDx pin are stored in the UARTx Receive Data register (Table 51). The Read-only UARTx Receive Data register shares a Register File address with the Write-only UARTx Transmit Data register.

Table 51. UARTx Receive Data Register (UxRXD)

BITS	7	6	5	4	3	2	1	0
FIELD	RXD							
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	F40H and F48H							

RXD—Receive Data
UART receiver data byte from the RXDx pin

UARTx Status 0 and Status 1 Registers

The UARTx Status 0 and Status 1 registers (Table 52 and 53) identify the current UART operating configuration and status.

Table 52. UARTx Status 0 Register (UxSTAT0)

BITS	7	6	5	4	3	2	1	0
FIELD	RDA	PE	OE	FE	BRKD	TDRE	TXE	CTS
RESET	0	0	0	0	0	1	1	X
R/W	R	R	R	R	R	R	R	R
ADDR	F41H and F49H							

RDA—Receive Data Available
This bit indicates that the UART Receive Data register has received data. Reading the UART Receive Data register clears this bit.
0 = The UART Receive Data register is empty.
1 = There is a byte in the UART Receive Data register.

PE—Parity Error
This bit indicates that a parity error has occurred. Reading the UART Receive Data register clears this bit.

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

DLE—DMA*x* Loop Enable
0 = DMA*x* reloads the original Start Address and is then disabled after the End Address data is transferred.
1 = DMA*x*, after the End Address data is transferred, reloads the original Start Address and continues operating.

DDIR—DMA*x* Data Transfer Direction
0 = Register File → on-chip peripheral control register.
1 = on-chip peripheral control register → Register File.

IRQEN—DMA*x* Interrupt Enable
0 = DMA*x* does not generate any interrupts.
1 = DMA*x* generates an interrupt when the End Address data is transferred.

WSEL—Word Select
0 = DMA*x* 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 request 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: I^{2}C Receiver Interrupt. DMA1 Control register: I^{2}C Transmitter Interrupt register empty.
111 = Reserved.

DMA*x* I/O Address Register

The DMA*x* I/O Address register contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is given by {FH, DMA*x*_IO[7:0]}.

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
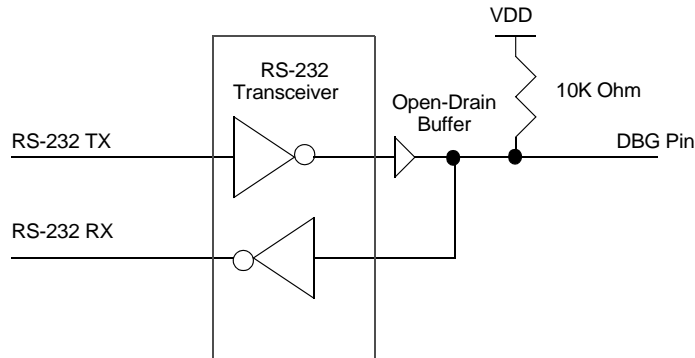These bits are reserved and are always undefined.

Figure 88. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

Debug Mode

The operating characteristics of the Z8F640x family devices in Debug mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions.

- The system clock operates unless in Stop mode.

- All enabled on-chip peripherals operate unless in Stop mode.

- Automatically exits Halt mode.

- Constantly refreshes the Watch-Dog Timer, if enabled.

Entering Debug Mode

The Z8F640x family device enters Debug mode following any of the following operations:

- Writing the DBGMODE bit in the OCD Control Register to 1 using the OCD interface.

- eZ8 CPU execution of a BRK (Breakpoint) instruction (when enabled).

- Break upon a Watchpoint match.

- If the DBG pin is Low when the Z8F640x family device exits Reset, the On-Chip Debugger automatically puts the device into Debug mode.

Exiting Debug Mode

The device exits Debug mode following any of the following operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0.
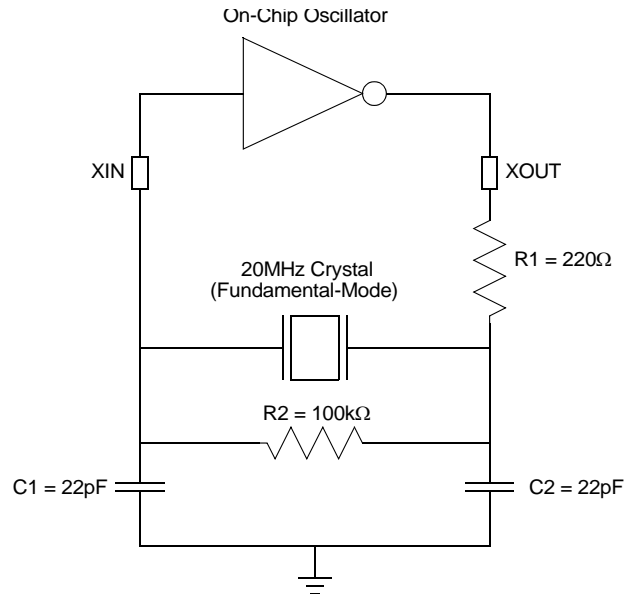
Figure 90. Recommended Crystal Oscillator Configuration (20MHz operation)

Table 99. Recommended Crystal Oscillator Specifications (20MHz Operation)

Parameter	Value	Units	Comments
Frequency	20	MHz	
Resonance	Parallel		
Mode	Fundamental		
Series Resistance (R_S)	25	Ω	Maximum
Load Capacitance (C_L)	20	pF	Maximum
Shunt Capacitance (C_0)	7	pF	Maximum
Drive Level	1	mW	Maximum

; value 01H, is the source. The value 01H is written into the
; Register at address 234H.

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed by users that prefer manual program coding or intend to implement their own assembler.

Example 1: If the contents of Registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 113. Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2: In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0 - 255 or, using Escaped Mode Addressing, a Working Register R0 - R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 114. Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

See the device-specific Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status flags, and address modes are represented by a notational shorthand that is described in Table 115

eZ8 CPU Instruction Classes

eZ8 CPU instructions can be divided functionally into the following groups:

- Arithmetic
- Bit Manipulation
- Block Transfer
- CPU Control
- Load
- Logical
- Program Control
- Rotate and Shift

Tables 118 through 125 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instruction can be considered as a subset of more than one category. Within these tables, the source operand is identified as 'src', the destination operand is 'dst' and a condition code is 'cc'.

Table 118. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using Extended Addressing
ADD	dst, src	Add
ADDX	dst, src	Add using Extended Addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
CPX	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
MULT	dst	Multiply

Table 125. Rotate and Shift Instructions

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

eZ8 CPU Instruction Summary

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

Table 126. eZ8 CPU Instruction Summary
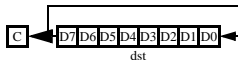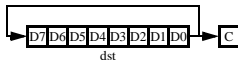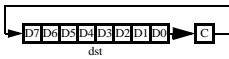
Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	13							2	4
		R	R	14							3	3
		R	IR	15							3	4
		R	IM	16							3	3
		IR	IM	17							3	4
ADCX dst, src	dst ← dst + src + C	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3
Flags Notation:	* = Value is a function of the result of the operation. - = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

Table 126. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	src	Opcode(s) (Hex)	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)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
POP dst	dst ← @SP SP ← SP + 1	R		50	-	-	-	-	-	-	2	2
		IR		51							2	3
POPX dst	dst ← @SP SP ← SP + 1	ER		D8	-	-	-	-	-	-	3	2
PUSH src	SP ← SP − 1 @SP ← src	R		70	-	-	-	-	-	-	2	2
		IR		71							2	3
PUSHX src	SP ← SP − 1 @SP ← src	ER		C8	-	-	-	-	-	-	3	2
RCF	C ← 0			CF	0	-	-	-	-	-	1	2
RET	PC ← @SP SP ← SP + 2			AF	-	-	-	-	-	-	1	4
RL dst		R		90	*	*	*	*	-	-	2	2
		IR		91							2	3
RLC dst		R		10	*	*	*	*	-	-	2	2
		IR		11							2	3
RR dst		R		E0	*	*	*	*	-	-	2	2
		IR		E1							2	3
RRC dst		R		C0	*	*	*	*	-	-	2	2
		IR		C1							2	3

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

Ordering Information

Table 128. Ordering Information

Part	Flash KB (Bytes)	RAM KB (Bytes)	Max. Speed (MHz)	Temp (^{0}C)	Voltage (V)	Package	Part Number
Z8 Encore!® with 16KB Flash, Standard Temperature							
Z8 Encore!®	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F1601PM020SC
Z8 Encore!®	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F1601AN020SC
Z8 Encore!®	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F1601VN020SC
Z8 Encore!®	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F1602AR020SC
Z8 Encore!®	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F1602VS020SC
Z8 Encore!® with 24KB Flash, Standard Temperature							
Z8 Encore!®	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F2401PM020SC
Z8 Encore!®	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F2401AN020SC
Z8 Encore!®	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F2401VN020SC
Z8 Encore!®	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F2402AR020SC
Z8 Encore!®	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F2402VS020SC
Z8 Encore!® with 32KB Flash, Standard Temperature							
Z8 Encore!®	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F3201PM020SC
Z8 Encore!®	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F3201AN020SC
Z8 Encore!®	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F3201VN020SC
Z8 Encore!®	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F3202AR020SC
Z8 Encore!®	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F3202VS020SC
Z8 Encore!®with 48KB Flash, Standard Temperature							
Z8 Encore!®	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F4801PM020SC
Z8 Encore!®	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F4801AN020SC
Z8 Encore!®	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F4801VN020SC
Z8 Encore!®	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F4802AR020SC
Z8 Encore!®	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F4802VS020SC
Z8 Encore!®	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	QFP-80	Z8F4803FT020SC

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.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____