Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f041ahh020sc00tr

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	60
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	63
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	63
FC3	Interrupt Request 1	IRQ1	00	61
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	64
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	64
FC6	Interrupt Request 2	IRQ2	00	62
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	65
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	65
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	67
FCE	Shared Interrupt Select	IRQSS	00	67
FCF	Interrupt Control	IRQCTL	00	67
GPIO Port A				
FD0	Port A Address	PAADDR	00	45
FD1	Port A Control	PACTL	00	47
FD2	Port A Input Data	PAIN	XX	47
FD3	Port A Output Data	PAOUT	00	47
GPIO Port B				
FD4	Port B Address	PBADDR	00	45
FD5	Port B Control	PBCTL	00	47
FD6	Port B Input Data	PBIN	XX	47
FD7	Port B Output Data	PBOUT	00	47
GPIO Port C				
FD8	Port C Address	PCADDR	00	45
FD9	Port C Control	PCCTL	00	47
FDA	Port C Input Data	PCIN	XX	47
FDB	Port C Output Data	PCOUT	00	47
GPIO Port D				
FDC	Port D Address	PDADDR	00	45
FDD	Port D Control	PDCTL	00	47
FDE	Reserved	—	XX	

XX=Undefined

Table 14. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B	PB0	Reserved		AFS1[0]: 0
		ANA0/AMPOUT	ADC Analog Input/LPO Output	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1/AMPINN	ADC Analog Input/LPO Input (N)	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2/AMPINP	ADC Analog Input/LPO Input (P)	AFS1[2]: 1
	PB3	CLKIN	External Clock Input	AFS1[3]: 0
		ANA3	ADC Analog Input	AFS1[3]: 1
	PB4	Reserved		AFS1[4]: 0
		ANA7	ADC Analog Input	AFS1[4]: 1
	PB5	Reserved		AFS1[5]: 0
		VREF*	ADC Voltage Reference	AFS1[5]: 1
	PB6	Reserved		AFS1[6]: 0
		Reserved		AFS1[6]: 1
	PB7	Reserved		AFS1[7]: 0
		Reserved		AFS1[7]: 1

Note: Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is not used to select the function. Also, alternate function selection as described in Port A–D Alternate Function Sub-Registers on page 47 must also be enabled.

* VREF is available on PB5 in 28-pin products only.

Reserved—Must be 0.

C3ENL—Port C3 Interrupt Request Enable Low Bit
C2ENL—Port C2 Interrupt Request Enable Low Bit
C1ENL—Port C1 Interrupt Request Enable Low Bit
C0ENL—Port C0 Interrupt Request Enable Low Bit

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 45) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

Table 45. Interrupt Edge Select Register (IRQES)

BITS	7	6	5	4	3	2	1	0
FIELD	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
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	FCDH							

IES*x*—Interrupt Edge Select *x*
0 = An interrupt request is generated on the falling edge of the PA*x* input.
1 = An interrupt request is generated on the rising edge of the PA*x* input.
where *x* indicates the specific GPIO Port pin number (0 through 7).

Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) register (Table 46) determines the source of the PADxS interrupts. The Shared Interrupt Select register selects between Port A and alternate sources for the individual interrupts.

Because these shared interrupts are edge-triggered, it is possible to generate an interrupt just by switching from one shared source to another. For this reason, an interrupt must be disabled before switching between sources.

0001H and counting resumes. The INPCAP bit in TxCTL0 register is cleared to indicate the timer interrupt is not caused by an input capture event.

Follow the steps below for configuring a timer for CAPTURE RESTART mode and initiating the count:

1. Write to the Timer Control register to:
   - Disable the timer.
   - Configure the timer for CAPTURE RESTART mode by writing the TMODE bits in the TxCTL1 register and the TMODEHI bit in TxCTL0 register.
   - Set the prescale value.
   - Set the Capture edge (rising or falling) for the Timer Input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. This allows the software to determine if interrupts were generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt was generated by a Reload.
5. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL0 register.
6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control register to enable the timer and initiate counting.

In CAPTURE mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

COMPARE Mode

In COMPARE mode, the timer counts up to the 16-bit maximum Compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Compare value, the timer generates an interrupt and counting continues (the timer value is not reset to 0001H). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon Compare.

If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting.

0 = No framing error occurred.
1 = A framing error occurred.

BRKD—Break Detect
This bit indicates that a break occurred. If the data bits, parity/multiprocessor bit, and Stop bit(s) are all 0s this bit is set to 1. Reading the UART Receive Data register clears this bit.
0 = No break occurred.
1 = A break occurred.

TDRE—Transmitter Data Register Empty
This bit indicates that the UART Transmit Data register is empty and ready for additional data. Writing to the UART Transmit Data register resets this bit.
0 = Do not write to the UART Transmit Data register.
1 = The UART Transmit Data register is ready to receive an additional byte to be transmitted.

TXE—Transmitter Empty
This bit indicates that the transmit shift register is empty and character transmission is finished.
0 = Data is currently transmitting.
1 = Transmission is complete.

CTS—$\overline{\text{CTS}}$ signal
When this bit is read it returns the level of the $\overline{\text{CTS}}$ signal. This signal is active Low.

UART Status 1 Register

This register contains multiprocessor control and status bits.

Table 64. UART Status 1 Register (U0STAT1)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved						NEWFRM	MPRX
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R
ADDR	F44H							

Reserved—Must be 0.

NEWFRM—Status bit denoting the start of a new frame. Reading the UART Receive Data register resets this bit to 0.
0 = The current byte is not the first data byte of a new frame.
1 = The current byte is the first data byte of a new frame.

Flash Operation Timing Using the Flash Frequency Registers

Before performing either a program or erase operation on Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasing of the Flash with system clock frequencies ranging from 32 kHz (32768 Hz) through 20 MHz.

The Flash Frequency High and Low Byte registers combine to form a 16-bit value, `FFREQ`, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz). This value is calculated using the following equation:

$$\text{FFREQ[15:0]} = \frac{\text{System Clock Frequency (Hz)}}{1000}$$

Caution: *Flash programming and erasure are not supported for system clock frequencies below 32 kHz (32768 Hz) or above 20 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure operation of the Z8 Encore! XP® F082A Series devices.*

Flash Code Protection Against External Access

The user code contained within the Flash memory can be protected against external access by the on-chip debugger. Programming the `FRP` Flash Option Bit prevents reading of the user code with the On-Chip Debugger. See Flash Option Bits on page 153 and On-Chip Debugger on page 173 for more information.

Flash Code Protection Against Accidental Program and Erasure

The Z8 Encore! XP F082A Series provides several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by a combination of the Flash Option bits, the register locking mechanism, the page select redundancy and the sector level protection control of the Flash Controller.

Flash Code Protection Using the Flash Option Bits

The `FRP` and `FWP` Flash Option Bits combine to provide three levels of Flash Program Memory protection as listed in Table 77. See Flash Option Bits on page 153 for more information.

SPROT7-SPROT0—Sector Protection
Each bit corresponds to a 512 byte Flash sector. For the Z8F08xx devices, the upper 3 bits must be zero. For the Z8F04xx devices all bits are used. For the Z8F02xx devices, the upper 4 bits are unused. For the Z8F01xx devices, the upper 6 bits are unused.

Flash Frequency High and Low Byte Registers

The Flash Frequency High (FFREQH) and Low Byte (FFREQL) registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:

$$\text{FFREQ[15:0]} = \{\text{FFREQH[7:0]},\text{FFREQL[7:0]}\} = \frac{\text{System Clock Frequency}}{1000}$$

Caution: *The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device. Also, Flash programming and erasure is not supported for system clock frequencies below 20 kHz or above 20 MHz.*

Table 82. Flash Frequency High Byte Register (FFREQH)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQH							
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	FFAH							

FFREQH—Flash Frequency High Byte
High byte of the 16-bit Flash Frequency value.

Table 83. Flash Frequency Low Byte Register (FFREQL)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQL							
RESET	0							
R/W	R/W							
ADDR	FFBH							

FFREQL—Flash Frequency Low Byte
Low byte of the 16-bit Flash Frequency value.

Trim Bit Data Register

The Trim Bid Data (TRMDR) register contains the read or write data for access to the trim option bits (Table 85).

Table 85. Trim Bit Data Register (TRMDR)

BITS	7	6	5	4	3	2	1	0
FIELD	TRMDR - Trim Bit Data							
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	FF7H							

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses `0000H` and `0001H` are reserved for the user-programmable Flash option bits.

Flash Program Memory Address 0000H

Table 86. Flash Option Bits at Program Memory Address 0000H

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	FRP	Reserved	FWP
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	Program Memory 0000H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

WDT_RES—Watchdog Timer Reset
0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request.
1 = Watchdog Timer time-out causes a system reset. This setting is the default for unprogrammed (erased) Flash.

WDT_AO—Watchdog Timer Always On
0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled.

Table 100. Serialization Data Locations

Info Page Address	Memory Address	Usage
1C	FE1C	Serial Number Byte 3 (most significant)
1D	FE1D	Serial Number Byte 2
1E	FE1E	Serial Number Byte 1
1F	FE1F	Serial Number Byte 0 (least significant)

Randomized Lot Identifier

Table 101. Lot Identification Number (RAND_LOT)

BITS	7	6	5	4	3	2	1	0
FIELD	RAND_LOT							
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	Interspersed throughout Information Page Memory							
Note: U = Unchanged by Reset. R/W = Read/Write.								

RAND_LOT—Randomized Lot ID
The randomized lot ID is a 32 byte binary value that changes for each production lot.

Table 102. Randomized Lot ID Locations

Info Page Address	Memory Address	Usage
3C	FE3C	Randomized Lot ID Byte 31 (most significant)
3D	FE3D	Randomized Lot ID Byte 30
3E	FE3E	Randomized Lot ID Byte 29
3F	FE3F	Randomized Lot ID Byte 28
58	FE58	Randomized Lot ID Byte 27
59	FE59	Randomized Lot ID Byte 26
5A	FE5A	Randomized Lot ID Byte 25
5B	FE5B	Randomized Lot ID Byte 24

On-Chip Debugger

The Z8 Encore! XP® F082A Series devices contain an integrated On-Chip Debugger (OCD) that provides advanced debugging features including:

- Single pin interface.
- Reading and writing of the register file.
- Reading and writing of program and data memory.
- Setting of breakpoints and watchpoints.
- Executing eZ8 CPU instructions.
- Debug pin sharing with general-purpose input-output function to maximize pins available to the user (8-pin product only).

Architecture

The on-chip debugger consists of four primary functional blocks: transmitter, receiver, auto-baud detector/generator, and debug controller. Figure 23 displays the architecture of the on-chip debugger.

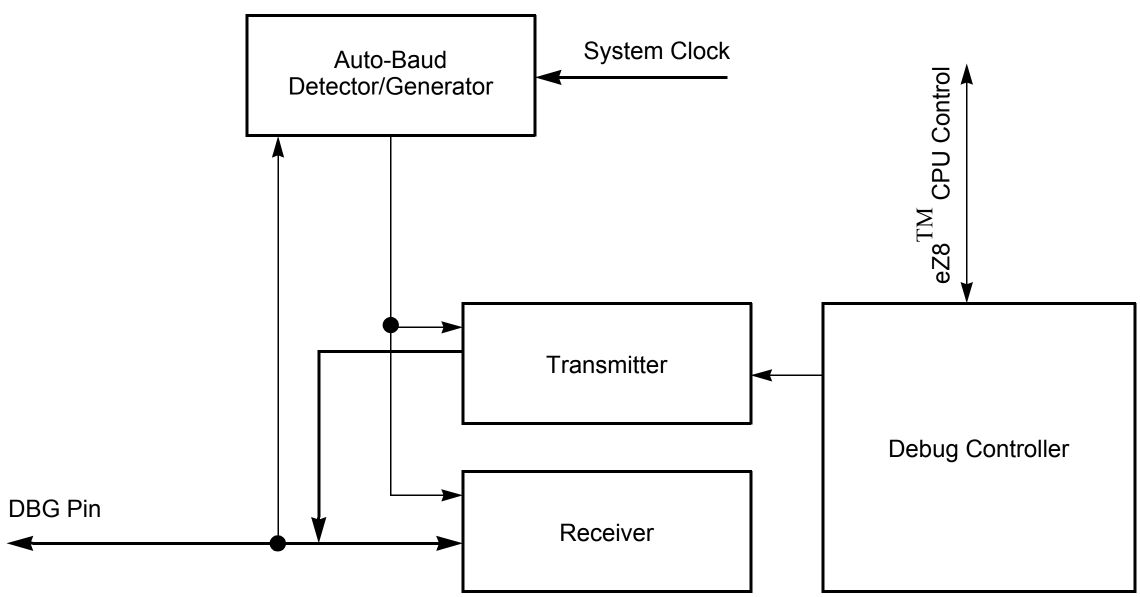


Figure 23. On-Chip Debugger Block Diagram

The Auto-Baud Detector/Generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. For optimal operation with asynchronous datastreams, the maximum recommended baud rate is the system clock frequency divided by 8. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by 4, but this theoretical maximum is possible only for low noise designs with clean signals. Table 105 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 105. OCD Baud-Rate Limits

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (Kbps)	Recommended Standard PC Baud Rate (bps)	Minimum Baud Rate (Kbps)
20.0	2500.0	1,843,200	39
1.0	125.0	115,200	1.95
0.032768 (32 kHz)	4.096	2,400	0.064

If the OCD receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. Reconfigure the Auto-Baud Detector/Generator by sending `80H`.

OCD Serial Errors

The On-Chip Debugger can detect any of the following error conditions on the DBG pin:

- Serial Break (a minimum of nine continuous bits Low)
- Framing Error (received `Stop` bit is Low)
- Transmit Collision (OCD and host simultaneous transmission detected by the OCD)

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a four character long Serial Break back to the host, and resets the Auto-Baud Detector/Generator. A Framing Error or Transmit Collision may be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the interface, returning a Serial Break break back to the host only extends the length of the Serial Break if the host releases the Serial Break early.

The host transmits a Serial Break on the `DBG` pin when first connecting to the Z8 Encore! XP F082A Series devices or when recovering from an error. A Serial Break from the host resets the Auto-Baud Generator/Detector but does not reset the OCD Control register. A Serial Break leaves the device in DEBUG mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns

Caution: *It is possible to disable the clock failure detection circuitry as well as all functioning clock sources. In this case, the Z8 Encore! XP F082A Series device ceases functioning and can only be recovered by Power-On-Reset.*

Oscillator Control Register Definitions

Oscillator Control Register

The Oscillator Control Register (OSCCTL) enables/disables the various oscillator circuits, enables/disables the failure detection/recovery circuitry and selects the primary oscillator, which becomes the system clock.

The Oscillator Control Register must be unlocked before writing. Writing the two step sequence `E7H` followed by `18H` to the Oscillator Control Register unlocks it. The register is locked at successful completion of a register write to the OSCCTL.

Table 109. Oscillator Control Register (OSCCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	INTEN	XTLEN	WDTEN	SOFEN	WDFEN	SCKSEL		
RESET	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F86H							

INTEN—Internal Precision Oscillator Enable
1 = Internal precision oscillator is enabled
0 = Internal precision oscillator is disabled

XTLEN—Crystal Oscillator Enable; this setting overrides the GPIO register control for PA0 and PA1
1 = Crystal oscillator is enabled
0 = Crystal oscillator is disabled

WDTEN—Watchdog Timer Oscillator Enable
1 = Watchdog Timer oscillator is enabled
0 = Watchdog Timer oscillator is disabled

SOFEN—System Clock Oscillator Failure Detection Enable
1 = Failure detection and recovery of system clock oscillator is enabled
0 = Failure detection and recovery of system clock oscillator is disabled

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 if manual program coding is preferred or if you intend to implement your 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 112. 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 113. 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 114.

Table 114. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B).
cc	Condition Code	—	Refer to Condition Codes section in the *eZ8 CPU Core User Manual (UM0128)*.
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
Ir	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
p	Polarity	p	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	X	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 115 lists additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 116 through Table 123 lists 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 116. 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
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using Extended Addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using Extended Addressing

Table 123. Rotate and Shift Instructions (Continued)

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

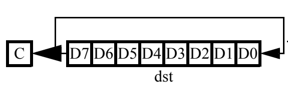
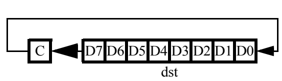
eZ8 CPU Instruction Summary

Table 124 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 124. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
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
ADD dst, src	dst ← dst + 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	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							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 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
OR dst, src	dst ← dst OR src	r	r	42	–	*	*	0	–	–	2	3
		r	Ir	43							2	4
		R	R	44							3	3
		R	IR	45							3	4
		R	IM	46							3	3
		IR	IM	47							3	4
ORX dst, src	dst ← dst OR src	ER	ER	48	–	*	*	0	–	–	4	3
		ER	IM	49							4	3
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
		IM		IF70							3	2
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

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

AC Characteristics

The section provides information about the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs.

Table 129. AC Characteristics

		V_{DD} = 2.7 V to 3.6 V T_A = -40 °C to +105 °C (unless otherwise stated)			
Symbol	**Parameter**	**Minimum**	**Maximum**	**Units**	**Conditions**
F_{SYSCLK}	System Clock Frequency	–	20.0	MHz	Read-only from Flash memory
		0.032768	20.0	MHz	Program or erasure of the Flash memory
F_{XTAL}	Crystal Oscillator Frequency	–	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver
T_{XIN}	System Clock Period	50	–	ns	$T_{CLK} = 1/F_{sysclk}$
T_{XINH}	System Clock High Time	20	30	ns	T_{CLK} = 50 ns
T_{XINL}	System Clock Low Time	20	30	ns	T_{CLK} = 50 ns
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50 ns
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50 ns

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP® F082A Series with 1 KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0 °C to 70 °C											
Z8F012APB020SC	1 KB	256 B	16 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F012AQB020SC	1 KB	256 B	16 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F012ASB020SC	1 KB	256 B	16 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F012ASH020SC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F012AHH020SC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F012APH020SC	1 KB	256 B	16 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F012ASJ020SC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F012AHJ020SC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F012APJ020SC	1 KB	256 B	16 B	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C											
Z8F012APB020EC	1 KB	256 B	16 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F012AQB020EC	1 KB	256 B	16 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F012ASB020EC	1 KB	256 B	16 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F012ASH020EC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F012AHH020EC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F012APH020EC	1 KB	256 B	16 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F012ASJ020EC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F012AHJ020EC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F012APJ020EC	1 KB	256 B	16 B	23	20	2	8	1	1	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging											

X

Z