



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	19
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
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	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0412pj020sc

Email: info@E-XFL.COM

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

Braces

The curly braces { }, indicate a single register or bus created by concatenating some combination of smaller registers, buses, or individual bits.

• **Example:** The 12-bit register address {0H, RP[7:4], R1[3:0]} is composed of a 4-bit hexadecimal value (0H) and two 4-bit register values taken from the Register Pointer (RP) and Working Register R1. 0H is the most significant nibble (4-bit value) of the 12-bit register, and R1[3:0] is the least significant nibble of the 12-bit register.

Parentheses

The parentheses (), indicate an indirect register address lookup.

• **Example:** (R1) is the memory location referenced by the address contained in the Working Register R1.

Parentheses/Bracket Combinations

The parentheses (), indicate an indirect register address lookup and the square brackets, [], indicate a register or bus.

• **Example:** Assume PC[15:0] contains the value 1234h. (PC [15:0]) then refers to the contents of the memory location at address 1234h.

Use of the Words Set, Reset and Clear

The word *set* implies that a register bit or a condition contains a logical 1. The words re*set* or *clear* imply that a register bit or a condition contains a logical 0. When either of these terms is followed by a number, the word *logical* cannot be included; however, it is implied.

Notation for Bits and Similar Registers

A field of bits within a register is designated as: Register[*n*:*n*].

• **Example:** ADDR[15:0] refers to bits 15 through bit 0 of the Address.

Use of the Terms LSB, MSB, Isb, and msb

In this document, the terms *LSB* and *MSB*, when appearing in upper case, mean *least* significant byte and most significant byte, respectively. The lowercase forms, *lsb* and *msb*, mean *least significant bit* and most significant bit, respectively.

Use of Initial Uppercase Letters

Initial uppercase letters designate settings and conditions in general text.

- **Example 1**: The receiver forces the SCL line to Low.
- **Example 2:** The Master generates a STOP condition to abort the transfer.













Figure 4. Z8F0811 and Z8F0411 in 20-Pin SSOP and PDIP Packages

UARTO Status 0 U0STAT0 (F41H - Read Only) **UART0** Control 0 D7 D6 D5 D4 D3 D2 D1 D0 U0CTL0 (F42H - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 -CTS signal Returns the level of the CTS -Loop Back Enable 0 = Normal operation 1 = Transmit data is looped signal Transmitter Empty 0 = Data is currently back to the receiver transmitting 1 = Transmission is STOP Bit Select 0 = Transmitter sends 1 complete STOP bit -Transmitter Data Register 0 = Transmit Data Register is 1 = Transmitter sends 2STOP bits full 1 = Transmit Data register is Send Break 0 = No break is sent empty 1 = Output of the transmitter Break Detect 0 = No break occurred is zero Parity Select 0 = Even parity 1 = A break occurred 1 = Odd parity Framing Error 0 = No framing error Parity Enable 0 = Parity is disabled occurred 1 = A framing occurred 1 = Parity is enabled Overrun Error CTS Enable 0 = No overrrun error0 = CTS signal has no effect occurred on the 1 = An overrun error transmitter occurred 1 = UART recognizes \overline{CTS} signal as a Parity Error transmit enable control 0 = No parity error occurred 1 = A parity error occurred signal Receive Data Available Receive Enable 0 = Receive Data Register is 0 = Receiver disabled 1 = Receiver enabled empty 1 = A byte is available in the Transmit Enable 0 = Transmitter disabled Receive Data Register 1 = Transmitter enabled

21

Port	Pin	Mnemonic	Alternate Function Description
Port C	Port C PC0 T1		Timer 1 Input
	PC1	T1OUT	Timer 1 Output
	PC2	SS	SPI Slave Select
	PC3	SCK	SPI Serial Clock
	PC4	MOSI	SPI Master Out Slave In
	PC5	MISO	SPI Master In Slave Out

Table 12. Port Alternate Function Mapping (Continued)

GPIO Interrupts

Many of GPIO port pins are used as interrupt sources. Some port pins are 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). For more details on interrupts using the GPIO pins, see GPIO Port Pin Block Diagram on page 48.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data, and output data. Table 13 lists the GPIO Port Registers and Sub-Registers. Use the Port A–C Address and Control Registers together to provide access to sub-registers for Port configuration and control.

Port Register Mnemonic	Port Register Name
P <i>x</i> ADDR	Port A–C Address Register (selects sub-registers)
P <i>x</i> CTL	Port A–C Control Register (provides access to sub-registers)
PxIN	Port A–C Input Data Register
P <i>x</i> OUT	Port A–C Output Data Register
Port Sub-Register Mnemonic	Port Register Name
PxDD	Data Direction
P <i>x</i> AF	Alternate Function

Table 13. GPIO Port Registers and Sub-Registers

pins.To determine the alternate function associated with each port pin, see GPIO Port Pin Block Diagram on page 48.

Caution: Do not enable alternate function for GPIO port pins which do not have an associated alternate function. Failure to follow this guideline can result in unpredictable operation.

Table 17. Port A–CA–C Alternate Function Sub-Registers

BITS	7	6	5	4	3	2	1	0		
FIELD	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0		
RESET		0								
R/W		R/W								
ADDR	lf 02H i	n Port A–C	Address Reo	gister, acces	sible throug	n the Port A-	-C Control F	Register		

AF[7:0]—Port Alternate Function enabled

- 0 = The port pin is in NORMAL mode and the DDx bit in the Port A–C 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–C Output Control Sub-Registers

The Port A–C Output Control sub-register (Table 18) is accessed through the Port A–C Control Register by writing 03H to the Port A–C Address Register. Setting the bits in the Port A–C 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 18. Port	A-C Output	Control	Sub-Registers
----------------	------------	---------	---------------

BITS	7	6	5	4	3	2	1	0		
FIELD	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0		
RESET		0								
R/W		R/W								
ADDR	lf 03H i	n Port A–C	Address Reo	gister, acces	sible throug	h the Port A-	-C Control F	Register		

POC[7:0]—Port Output Control

These bits function independently of the alternate function bit and always disable the drains if set to 1.

0 = The drains are enabled for any output mode (unless overridden by the

Table 29. IRQ0 Enable High Bit Register (IRQ0ENH)

BITS	7	6	5	4	3	2	1	0			
FIELD	Reserved	T1ENH	T0ENH	UORENH	U0TENH	I2CENH	SPIENH	ADCENH			
RESET		0									
R/W		R/W									
ADDR				FC	1H						

Reserved—Must be 0 T1ENH—Timer 1 Interrupt Request Enable High Bit

TOENH—Timer 0 Interrupt Request Enable High Bit **UORENH**—UART 0 Receive Interrupt Request Enable High Bit **UOTENH**—UART 0 Transmit Interrupt Request Enable High Bit **I2CENH**—I²C Interrupt Request Enable High Bit **SPIENH**—SPI Interrupt Request Enable High Bit **ADCENH**—ADC Interrupt Request Enable High Bit

Table 30. IRQ0 Enable Low Bit Register (IRQ0ENL)

BITS	7	6	5	4	3	2	1	0		
FIELD	Reserved	T1ENL	T0ENL	UORENL	U0TENL	I2CENL	SPIENL	ADCENL		
RESET		0								
R/W		R/W								
ADDR				FC	2H					

Reserved—Must be 0

T1ENL—Timer 1 Interrupt Request Enable Low Bit **T0ENL**—Timer 0 Interrupt Request Enable Low Bit **U0RENL**—UART 0 Receive Interrupt Request Enable Low Bit **U0TENL**—UART 0 Transmit Interrupt Request Enable Low Bit **I2CENL**—I²C Interrupt Request Enable Low Bit **SPIENL**—SPI Interrupt Request Enable Low Bit **ADCENL**—ADC Interrupt Request Enable Low Bit

IRQ1 Enable High and Low Bit Registers

Table 31 describes the priority control for IRQ1. The IRQ1 Enable High and Low Bit Registers (Table 32 and Table 33) form a priority encoded enabling for interrupts in the Interrupt Request 1 Register. Priority is generated by setting bits in each register.

All three Watchdog Timer Reload Registers must be written in this order. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes occur unless the sequence is restarted. The value in the Watchdog Timer Reload Registers is loaded into the counter when the WDT is first enabled and every time a WDT instruction is executed.

Watchdog Timer Control Register Definitions

Watchdog Timer Control Register

The Watchdog Timer Control Register (WDTCTL), detailed in Table 48, is a Read-Only Register that indicates the source of the most recent Reset event, a Stop Mode Recovery event, and a WDT time-out. Reading this register resets the upper four bits to 0.

Writing the 55H, AAH unlock sequence to the Watchdog Timer Control Register (WDTCTL) address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL address produce no effect on the bits in the WDTCTL. The locking mechanism prevents spurious writes to the Reload registers.

BITS	7	6	5	4	3	2	1	0	
FIELD	POR	STOP	WDT	EXT	Reserved				
RESET	See d	lescriptions l	below	0					
R/W		R							
ADDR				FF	0H				

Table 48. Watchdog Timer Control Register (WDTCTL)

Reset or Stop Mode Recovery Event	POR	STOP	WDT	EXT
Power-On Reset	1	0	0	0
Reset through RESET pin assertion	0	0	0	1
Reset through WDT time-out	0	0	1	0
Reset through the OCD (OCTCTL[1] set to 1)	1	0	0	0
Reset from STOP Mode through the DBG Pin driven Low	1	0	0	0
Stop Mode Recovery through GPIO pin transition	0	1	0	0
Stop Mode Recovery through WDT time-out	0	1	1	0

POR—Power-On Reset Indicator

If this bit is set to 1, a POR event occurred. This bit is reset to 0, if a WDT time-out or Stop Mode Recovery occurs. This bit is also reset to 0, when the register is read.

Table 50. Watchdog Timer Reload High Byte Register (WDTH)

BITS	7	6	5	4	3	2	1	0			
FIELD		WDTH									
RESET		1									
R/W		R/W*									
ADDR		FF2H									
R/W*–Re	R/W*–Read returns the current WDT count value. Write sets the desired Reload Value.										

WDTH—WDT Reload High Byte

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

Table 51. Watchdog Timer Reload Low Byte Register (WDTL)

BITS	7	6	5	4	3	2	1	0			
FIELD	WDTL										
RESET		1									
R/W				R/	W*						
ADDR		FF3H									
R/W*-Re	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.

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data Register is empty, an interrupt is generated immediately. When the UART Transmit Interrupt is detected, the associated ISR performs the following:

- 1. Write the UART Control 1 Register to select the outgoing address bit:
 - Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
- 2. Write the data byte to the UART Transmit Data Register. The transmitter automatically transfers data to the Transmit Shift Register and then transmits the data.
- 3. Clear the UART Transmit Interrupt bit in the applicable Interrupt Request Register.
- 4. Execute the IRET instruction to return from the ISR and waits for the Transmit Data Register to again become empty.

Receiving Data using the Polled Method

Follow the steps below to configure the UART for polled data reception:

- 1. Write to the UART Baud Rate High and Low Byte Registers to set the required baud rate.
- 2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
- 3. Write to the UART Control 1 Register to enable Multiprocessor mode functions, if desired.
- 4. Write to the UART Control 0 Register to:
 - Set the receive enable bit (REN) to enable the UART for data reception
 - Enable parity, if required, and if MULTIPROCESSOR mode is not enabled, and select either even or odd parity.
- 5. Check the RDA bit in the UART Status 0 Register to determine if the Receive Data Register contains a valid data byte (indicated by 1). If RDA is set to 1 to indicate available data, continue to step 6. If the Receive Data Register is empty (indicated by a 0), continue to monitor the RDA bit awaiting reception of the valid data.
- 6. Read data from the UART Receive Data Register. If operating in Multiprocessor (9-bit) mode, further actions may be required depending on the Multiprocessor Mode bits MPMD[1:0].
- 7. Return to step 5 to receive additional data.

MPBT—Multiprocessor Bit Transmit

This bit is applicable only when Multiprocessor (9-bit) mode is enabled. 0 =Send a 0 in the multiprocessor bit location of the data stream (9th bit). 1 =Send a 1 in the multiprocessor bit location of the data stream (9th bit).

DEPOL—Driver Enable Polarity

0 = DE signal is Active High.

1 = DE signal is Active Low.

BRGCTL—Baud Rate Control

This bit causes different UART behavior depending on whether the UART receiver is enabled (REN = 1 in the UART Control 0 Register).

When the UART receiver is <u>not</u> enabled, this bit determines whether the BRG will issue interrupts.

- 0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value
- 1 = The BRG generates a receive interrupt when it counts down to zero. Reads from the Baud Rate High and Low Byte registers return the current BRG count value.

When the UART receiver is enabled, this bit allows reads from the Baud Rate Registers to return the BRG count value instead of the Reload Value.

- 0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value.
- 1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the Timers, there is no mechanism to latch the High Byte when the Low Byte is read.

RDAIRQ—Receive Data Interrupt Enable

- 0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller.
- 1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.

IREN—Infrared Encoder/Decoder Enable

- 0 =Infrared Encoder/Decoder is disabled. UART operates normally operation.
- 1 = Infrared Encoder/Decoder is enabled. The UART transmits and receives data through the Infrared Encoder/Decoder.

UART Address Compare Register

The UART Address Compare register stores the multi-node network address of the UART. When the MPMD[1] bit of UART Control Register 0 is set, all incoming address bytes will be compared to the value stored in the Address Compare register. Receive interrupts and RDA assertions will only occur in the event of a match. of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal. This procedure allows the Endec to tolerate jitter and baud rate errors in the incoming data stream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Endec Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined in UART Control Register Definitions on page 100.

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

necessary for \overline{SS} to deassert between characters to generate the interrupt. The SPI in SLAVE mode also generates an interrupt if the \overline{SS} signal deasserts prior to transfer of all the bits in a character (see description of Slave Abort Error). Writing a 1 to the IRQ bit in the SPI Status Register clears the pending SPI interrupt request. The IRQ bit must be cleared to 0 by the ISR to generate future interrupts. To start the transfer process, an SPI interrupt can be forced by software writing a 1 to the STR bit in the SPICTL Register.

If the SPI is disabled, an SPI interrupt can be generated by a BRG time-out. This timer function must be enabled by setting the BIRQ bit in the SPICTL Register. This BRG time-out does not set the IRQ bit in the SPISTAT Register, just the SPI interrupt bit in the interrupt controller.

SPI Baud Rate Generator

In SPI MASTER mode, the BRG creates a lower frequency serial clock (SCK) for data transmission synchronization between the Master and the external Slave. The input to the BRG is the system clock. The SPI Baud Rate High and Low Byte Registers combine to form a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator. The SPI baud rate is calculated using the following equation:

SPI Baud Rate (bits/s) = $\frac{\text{System Clock Frequency (Hz)}}{2xBRG[15:0]}$

Minimum baud rate is obtained by setting BRG[15:0] to 0000H for a clock divisor value of $(2 \times 65536 = 131072)$.

When the SPI is disabled, BRG functions as a basic 16-bit timer with interrupt on time-out. Follow the steps below to configure BRG as a timer with interrupt on time-out:

- 1. Disable the SPI by clearing the SPIEN bit in the SPI Control Register to 0.
- 2. Load the desired 16-bit count value into the SPI Baud Rate High and Low Byte registers.
- 3. Enable BRG timer function and associated interrupt by setting the BIRQ bit in the SPI Control Register to 1.

When configured as a general-purpose timer, the interrupt interval is calculated using the following equation:

Interrupt Interval (s) = System Clock Period (s) ×BRG[15:0]]

frequency range for the device. The Flash Frequency High and Low Byte Registers must be loaded with the correct value to insure proper program and erase times.

Table 87. Flash Frequency High Byte Register (FFREQH)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQH							
RESET	0							
R/W	R/W							
ADDR	FFAH							

Table 88. 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							

FFREQH and FFREQL—Flash Frequency High and Low Bytes

These 2 bytes, {FFREQH[7:0], FFREQL[7:0]}, contain the 16-bit Flash Frequency value.

Flash Memory Address 0000H

Table 89. Option Bits at Flash Memory Address 0000H for 8K Series Flash Devices

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	OSC_SEL[1:0] V		VBO_AO	RP	Reserved	FWP
RESET	U							
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 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.
- 1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

OSC_SEL[1:0]—OSCILLATOR Mode Selection

- 00 = On-chip oscillator configured for use with external RC networks (<4 MHz).
- 01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).
- 10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).
- 11 = Maximum power for use with high frequency crystals (8.0 MHz to 20.0 MHz). This setting is the default for unprogrammed (erased) Flash.

VBO_AO—Voltage Brownout Protection Always On

- 0 = Voltage Brownout Protection is disabled in STOP mode to reduce total power consumption.
- 1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

RP—Read Protect

- 0 = User program code is inaccessible. Limited control features are available through the OCD.
- 1 = User program code is accessible. All OCD commands are enabled. This setting is the default for unprogrammed (erased) Flash.

Table 93. On-Chip Debugger Commands (Continued)

Debug Command	Command Byte	Enabled when NOT in DEBUG mode?	Disabled by Read Protect Option Bit
Execute Instruction	12H	-	Disabled
Reserved	13H - FFH	-	-

In the following bulleted list of OCD Commands, data and commands sent from the host to the OCD are identified by 'DBG \leftarrow Command/Data'. Data sent from the OCD back to the host is identified by 'DBG \rightarrow Data'

• **Read OCD Revision (00H)**—The Read OCD Revision command determines the version of the OCD. If OCD commands are added, removed, or changed, this revision number changes.

```
DBG \leftarrow 00H
DBG \rightarrow OCDREV[15:8] (Major revision number)
DBG \rightarrow OCDREV[7:0] (Minor revision number)
```

 Write OCD Counter Register (01H)—The Write OCD Counter Register command writes the data that follows to the OCDCNTR register. If the device is not in DEBUG mode, the data is discarded.

```
DBG \leftarrow 01H
DBG \leftarrow OCDCNTR [15:8]
DBG \leftarrow OCDCNTR [7:0]
```

• **Read OCD Status Register (02H)**—The Read OCD Status Register command reads the OCDSTAT register.

```
DBG \leftarrow 02H
DBG \rightarrow OCDSTAT[7:0]
```

• **Read OCD Counter Register (03H)**—The OCD Counter Register can be used to count system clock cycles in between Breakpoints, generate a BRK when it counts down to zero, or generate a BRK when its value matches the Program Counter. Since this register is really a down counter, the returned value is inverted when this register is read so the returned result appears to be an up counter. If the device is not in DEBUG mode, this command returns FFFFH.

```
DBG \leftarrow 03H
DBG \rightarrow ~OCDCNTR[15:8]
DBG \rightarrow ~OCDCNTR[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 device back into normal operating mode is to reset the device.

```
181
```

```
DBG \leftarrow Size[7:0]
DBG \rightarrow 1-65536 data bytes
```

• Read Program Memory CRC (0EH)—The Read Program Memory CRC command computes and returns the CRC (cyclic redundancy check) of Program Memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG mode, this command returns FFFFH for the CRC value. Unlike most other OCD Read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads the Program Memory, calculates the CRC value, and returns the result. The delay is a function of the Program Memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Program Memory.

```
DBG \leftarrow 0EH
DBG \rightarrow CRC[15:8]
DBG \rightarrow CRC[7:0]
```

• **Step Instruction (10H)**—The Step Instruction command steps one assembly instruction at the current Program Counter location. If the device is not in DEBUG mode or the Read Protect Option Bit is enabled, the OCD ignores this command.

DBG \leftarrow 10H

• **Stuff Instruction (11H)**—The Stuff Instruction command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0-4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the device is not in DEBUG mode or the Read Protect Option Bit is enabled, the OCD ignores this command.

DBG \leftarrow 11H DBG \leftarrow opcode[7:0]

• **Execute Instruction (12H)**—The Execute Instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over Breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not in DEBUG mode or the Read Protect Option Bit is enabled, the OCD ignores this command

DBG \leftarrow 12H DBG \leftarrow 1-5 byte opcode

On-Chip Debugger Control Register Definitions

OCD Control Register

The OCD Control Register controls the state of the OCD. This register enters or exits DEBUG mode and enables the BRK instruction. It can also reset the Z8 Encore! XP[®] F0822 Series device.



Figure 47 displays the input frequency response of the ADC.

Figure 47. Analog-to-Digital Converter Frequency Response

198

Table 116 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

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

Table 116. Additional Symbols

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

 $dst \leftarrow dst + src$

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

Abbreviation	Description	Abbreviation	Description
b	Bit position	IRR	Indirect Register Pair
СС	Condition code	р	Polarity (0 or 1)
Х	8-bit signed index or displacement	r	4-bit Working Register
DA	Destination address	R	8-bit register
ER	Extended Addressing register	r1, R1, Ir1, Irr1, IR1, rr1, RR1, IRR1, ER1	Destination address
IM	Immediate data value	r2, R2, Ir2, Irr2, IR2, rr2, RR2, IRR2, ER2	Source address
lr	Indirect Working Register	RA	Relative
IR	Indirect register	rr	Working Register Pair
Irr	Indirect Working Register Pair	RR	Register Pair

Table 127. Opcode Map Abbreviations

Z8 Encore! XP[®] F0822 Series Product Specification

250

working register pair 211 WTDU register 87

Х

X 211 XOR 217 XORX 217

Z

Z8 Encore! block diagram 3 features 1 introduction 1 part selection guide 2