E·XFL



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>"

D	et	ai	ls
υ	eu	aı	15

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, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f3202vs020ec00tr

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



Signal Mnemonic	I/O	Description
Reset		
RESET	Ι	RESET. Generates a Reset when asserted (driven Low).
Power Supply		
VDD	Ι	Power Supply.
AVDD	Ι	Analog Power Supply.
VSS	Ι	Ground.
AVSS	Ι	Analog Ground.

Table 2. Signal Descriptions (Continued)

Pin Characteristics

Table 3 provides detailed information on the characteristics for each pin available on the Z8F640x family products. Data in Table 3 is sorted alphabetically by the pin symbol mnemonic.

Table 3. Pin Characteristics of the Z8F640x family

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tri-State Output	Internal Pull-up or Pull-down	Schmitt Trigger Input	Open Drain Output
AVSS	N/A	N/A	N/A	N/A	No	No	N/A
AVDD	N/A	N/A	N/A	N/A	No	No	N/A
DBG	I/O	Ι	N/A	Yes	No	Yes	Yes
VSS	N/A	N/A	N/A	N/A	No	No	N/A
PA[7:0]	I/O	Ι	N/A	Yes	No	Yes	Yes, Programmable
PB[7:0]	I/O	Ι	N/A	Yes	No	Yes	Yes, Programmable
PC[7:0]	I/O	Ι	N/A	Yes	No	Yes	Yes, Programmable
PD[7:0]	I/O	Ι	N/A	Yes	No	Yes	Yes, Programmable
PE7:0]	I/O	Ι	N/A	Yes	No	Yes	Yes, Programmable
x represents int	teger 0, 1, to	o indicate mul	tiple pins with s	ymbol mnen	nonics that dif	fer only by tl	ne integer



Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
FCE	Interrupt Port Select	IRQPS	00	55
FCF	Interrupt Control	IRQCTL	00	56
GPIO Port A				
FD0	Port A Address	PAADDR	00	37
FD1	Port A Control	PACTL	00	38
FD2	Port A Input Data	PAIN	XX	42
FD3	Port A Output Data	PAOUT	00	43
GPIO Port B				
FD4	Port B Address	PBADDR	00	37
FD5	Port B Control	PBCTL	00	38
FD6	Port B Input Data	PBIN	XX	42
FD7	Port B Output Data	PBOUT	00	43
GPIO Port C	*			
FD8	Port C Address	PCADDR	00	37
FD9	Port C Control	PCCTL	00	38
FDA	Port C Input Data	PCIN	XX	42
FDB	Port C Output Data	PCOUT	00	43
GPIO Port D	*			
FDC	Port D Address	PDADDR	00	37
FDD	Port D Control	PDCTL	00	38
FDE	Port D Input Data	PDIN	XX	42
FDF	Port D Output Data	PDOUT	00	43
GPIO Port E				
FE0	Port E Address	PEADDR	00	37
FE1	Port E Control	PECTL	00	38
FE2	Port E Input Data	PEIN	XX	42
FE3	Port E Output Data	PEOUT	00	43
GPIO Port F				
FE4	Port F Address	PFADDR	00	37
FE5	Port F Control	PFCTL	00	38
FE6	Port F Input Data	PFIN	XX	42
FE7	Port F Output Data	PFOUT	00	43
GPIO Port G				
FE8	Port G Address	PGADDR	00	37
FE9	Port G Control	PGCTL	00	38
FEA	Port G Input Data	PGIN	XX	42
FEB	Port G Output Data	PGOUT	00	43
GPIO Port H				
FEC	Port H Address	PHADDR	00	37
XX-Undefined				

Table 6. Register File Address Map (Continued)



Architecture

Figure 65 illustrates a block diagram of the interrupt controller.



Figure 65. Interrupt Controller Block Diagram

Operation

Master Interrupt Enable

The master interrupt enable bit (IRQE) in the Interrupt Control register globally enables and disables interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an EI (Enable Interrupt) instruction
- Execution of an IRET (Return from Interrupt) instruction
- Writing a 1 to the IRQE bit in the Interrupt Control register

Interrupts are globally disabled by any of the following actions:

- Execution of a DI (Disable Interrupt) instruction
- eZ8 CPU acknowledgement of an interrupt service request from the interrupt controller
- Writing a 0 to the IRQE bit in the Interrupt Control register
- Reset



I²CI—I²C Interrupt Request

0 = No interrupt request is pending for the I²C.

1 = An interrupt request from the I²C is awaiting service.

SPII-SPI Interrupt Request

- 0 = No interrupt request is pending for the SPI.
- 1 = An interrupt request from the SPI is awaiting service.

ADCI-ADC Interrupt Request

0 = No interrupt request is pending for the Analog-to-Digital Converter.

1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 24) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 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 1 register to determine if any interrupt requests are pending.

able 24. li	nterrupt Req	luest I Regis	ter (IRQI)				
BITS	7	6	5	4	3	2	
FIELD	PAD7I	PAD6I	PAD5I	PAD4I	PAD3I	PAD2I	
RESET	0	0	0	0	0	0	

Тź

R/W

R/W

R/W

ADDR

PADxI—Port A or Port D Pin x Interrupt Request

R/W

0 = No interrupt request is pending for GPIO Port A or Port D pin x.

R/W

1 = An interrupt request from GPIO Port A or Port D pin x is awaiting service.

FC3H

where x indicates the specific GPIO Port pin number (0 through 7). For each pin, only 1 of either Port A or Port D can be enabled for interrupts at any one time. Port selection (A or D) is determined by the values in the Interrupt Port Select Register.

R/W

R/W

1

PAD1I

0

R/W

0

PAD01

0

R/W



Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register (Table 25) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 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 1 register to determine if any interrupt requests are pending.

Table 25. Interrupt Request 2 Register (IRQ2)

BITS	7	6	5	4	3	2	1	0
FIELD	T3I	U1RXI	UITXI	DMAI	PC3I	PC2I	PC1I	PC0I
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				FC	6H			

T3I—Timer 3 Interrupt Request

- 0 = No interrupt request is pending for Timer 3.
- 1 = An interrupt request from Timer 3 is awaiting service.

U1RXI—UART 1 Receive Interrupt Request

- 0 = No interrupt request is pending for the UART1 receiver.
- 1 = An interrupt request from UART1 receiver is awaiting service.
- U1TXI-UART 1 Transmit Interrupt Request
- 0 = No interrupt request is pending for the UART 1 transmitter.
- 1 = An interrupt request from the UART 1 transmitter is awaiting service.

DMAI—DMA Interrupt Request

- 0 = No interrupt request is pending for the DMA.
- 1 = An interrupt request from the DMA is awaiting service.

PCxI—Port C Pin x Interrupt Request

- 0 = No interrupt request is pending for GPIO Port C pin *x*.
- 1 = An interrupt request from GPIO Port C pin x is awaiting service.

where *x* indicates the specific GPIO Port C pin number (0 through 3).



Capture mode

0 = Count is captured on the rising edge of the Timer Input signal.

1 = Count is captured on the falling edge of the Timer Input signal.

Compare mode

When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

Gated mode

0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input.

1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.

Capture/Compare mode

0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal.

1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.

PRES—Prescale value.

The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This insures proper clock division each time the Timer is restarted.

- 000 = Divide by 1
- 001 = Divide by 2
- 010 = Divide by 4
- 011 = Divide by 8
- 100 = Divide by 16
- 101 = Divide by 32
- 110 = Divide by 64
- 111 = Divide by 128

TMODE—Timer mode

- 000 =One-Shot mode
- 001 =Continuous mode
- 010 =Counter mode
- 011 = PWM mode
- 100 = Capture mode
- 101 = Compare mode
- 110 = Gated mode
- 111 = Capture/Compare mode



- 5. Check the TDRE bit in the UART Status 0 register to determine if the Transmit Data register is empty (indicated by a 1). If empty, continue to Step 6. If the Transmit Data register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data register becomes available to receive new data.
- 6. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmit the data.
- 7. To transmit additional bits, return to Step 5.

Transmitting Data using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data register to accept new data for transmission. Follow these steps to configure the UART for interrupt-driven data transmission:

- 1. Write to the UART Baud Rate High and Low Byte registers to set the desired baud rate.
- 2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
- 3. Execute a DI instruction to disable interrupts.
- 4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the desired priority.
- 5. Write to the UART Control 1 register to enable Multiprocessor (9-bit) mode functions, if desired.
- 6. Write to the UART Control 0 register to:
 - Set the transmit enable bit (TEN) to enable the UART for data transmission
 - Enable parity, if desired, and select either even or odd parity.
 - Set or clear the CTSE bit to enable or disable control from the receiver via the $\overline{\text{CTS}}$ pin.
- 7. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data transmission. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) should perform the following:

- 8. Write the data byte to the UART Transmit Data register. The transmitter will automatically transfer the data to the Transmit Shift register and transmit the data.
- 9. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
- 10. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.



- 7. Write to the UART Control 0 register to:
 - Set the receive enable bit (REN) to enable the UART for data reception
 - Enable parity, if desired, and select either even or odd parity.
- 8. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) should perform the following:

- 9. Check the UART Status 0 register to determine the source of the interrupt error, break, or received data.
- 10. If the interrupt was due to data available, read the data from the UART Receive Data register. If operating in Multiprocessor (9-bit) mode, first read the Multiprocessor Receive flag (MPRX) to determine if the data was directed to this UART before reading the data.
- 11. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
- 12. Execute the IRET instruction to return from the interrupt-service routine and await more data.

Receiving Data using the Direct Memory Access Controller (DMA)

The DMA and UART can coordinate automatic data transfer from the UART Receive Data register to general-purpose Register File RAM. This reduces the eZ8 CPU processing overhead required to support UART data reception. The UART Receiver interrupt must then only notify the eZ8 CPU of error conditions. Follow these steps to configure the UART and DMA for automatic data handling:

- 1. Write to the DMA control registers to configure the DMA to transfer data from the UART Receive Data register to general-purpose Register File RAM.
- 2. Write to the UART Baud Rate High and Low Byte registers to set the desired baud rate.
- 3. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
- 4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the desired priority.
- 5. Write to the UART Control 1 register to:
 - Enable Multiprocessor (9-bit) mode functions, if desired.
 - Disable the UART interrupt for received data by clearing $\overline{\text{RDAIRQ}}$ to 0.



- 6. Write to the UART Control 0 register to:
 - Set the receive enable bit (REN) to enable the UART for data reception
 - Enable parity, if desired, and select either even or odd parity.

The UART and DMA are now configured for data reception and automatic data transfer to the Register File. When a valid data byte is received by the UART the following occurs:

- 7. The UART notifies the DMA Controller that a data byte is available in the UART Receive Data register.
- 8. The DMA Controller requests control of the system bus from the eZ8 CPU.
- 9. The eZ8 CPU acknowledges the bus request.
- 10. The DMA Controller transfers the data from the UART Receive Data register to another location in RAM and then return bus control back to the eZ8 CPU.

The UART and DMA can continue to transfer incoming data bytes without eZ8 CPU intervention. When a UART error is detected, the UART Receiver interrupt is generated. The associated interrupt service routine (ISR) should perform the following:

11. Check the UART Status 0 register to determine the source of the UART error or break condition and then respond appropriately.

Multiprocessor (9-bit) mode

The UART has a Multiprocessor mode that uses an extra (9th) bit for selective communication when a number of processors share a common UART bus. In Multiprocessor (9-bit) mode (also referred to as 9-Bit mode), the multiprocessor bit (MP) is transmitted immediately following the 8-bits of data and immediately preceding the STOP bit(s) as illustrated in Figure 70. The character format is:



Figure 70. UART Asynchronous Multiprocessor (9-bit) Mode Data Format

In Multiprocessor (9-bit) mode, parity is not an option as the Parity bit location (9th bit) becomes the Multiprocessor control bit. The UART Control 1 and Status 1 registers provide multiprocessor (9-bit) mode control and status information.



Table 58. UART Baud Rates

20.0 MHz System Clock

Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)
1250.0	1	1250.0	0.00
625.0	2	625.0	0.00
250.0	5	250.0	0.00
115.2	11	113.6	-1.36
57.6	22	56.8	-1.36
38.4	33	37.9	-1.36
19.2	65	19.2	0.16
9.60	130	9.62	0.16
4.80	260	4.81	0.16
2.40	521	2.40	-0.03
1.20	1042	1.20	-0.03
0.60	2083	0.60	0.02
0.30	4167	0.30	-0.01

18.432 MHz Syst	em Clock		
Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)
1250.0	1	1152.0	-7.84%
625.0	2	576.0	-7.84%
250.0	5	230.4	-7.84%
115.2	10	115.2	0.00
57.6	20	57.6	0.00
38.4	30	38.4	0.00
19.2	60	19.2	0.00
9.60	120	9.60	0.00
4.80	240	4.80	0.00
2.40	480	2.40	0.00
1.20	960	1.20	0.00
0.60	1920	0.60	0.00
0.30	3840	0.30	0.00

16.667 MHz System Clock

Desired Rate	BRG Divisor	Actual Rate	Error	Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)	(kHz)	(Decimal)	(kHz)	(%)
1250.0	1	1041.69	-16.67	1250.0	N/A	N/A	N/A
625.0	2	520.8	-16.67	625.0	1	691.2	10.59
250.0	4	260.4	4.17	250.0	3	230.4	-7.84
115.2	9	115.7	0.47	115.2	6	115.2	0.00
57.6	18	57.87	0.47	57.6	12	57.6	0.00
38.4	27	38.6	0.47	38.4	18	38.4	0.00
19.2	54	19.3	0.47	19.2	36	19.2	0.00
9.60	109	9.56	-0.45	9.60	72	9.60	0.00
4.80	217	4.80	-0.83	4.80	144	4.80	0.00
2.40	434	2.40	0.01	2.40	288	2.40	0.00
1.20	868	1.20	0.01	1.20	576	1.20	0.00
0.60	1736	0.60	0.01	0.60	1152	0.60	0.00
0.30	3472	0.30	0.01	0.30	2304	0.30	0.00

11.0592 MHz System Clock



10.0 MHz System	n Clock			5.5296 MHz Syst	em Clock		
Desired Rate	BRG Divisor	Actual Rate	Error	Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)	(kHz)	(Decimal)	(kHz)	(%)
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	1	625.0	0.00	625.0	N/A	N/A	N/A
250.0	3	208.33	-16.67	250.0	1	345.6	38.24
115.2	5	125.0	8.51	115.2	3	115.2	0.00
57.6	11	56.8	-1.36	57.6	6	57.6	0.00
38.4	16	39.1	1.73	38.4	9	38.4	0.00
19.2	33	18.9	0.16	19.2	18	19.2	0.00
9.60	65	9.62	0.16	9.60	36	9.60	0.00
4.80	130	4.81	0.16	4.80	72	4.80	0.00
2.40	260	2.40	-0.03	2.40	144	2.40	0.00
1.20	521	1.20	-0.03	1.20	288	1.20	0.00
0.60	1042	0.60	-0.03	0.60	576	0.60	0.00
0.30	2083	0.30	0.02	0.30	1152	0.30	0.00

Table 58. UART Baud Rates (Continued)

3.579545 MHz System Clock

Desired Rate	BRG Divisor	Actual Rate	Error	Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)	(kHz)	(Decimal)	(kHz)	(%)
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	N/A	N/A	N/A	625.0	N/A	N/A	N/A
250.0	1	223.72	-10.51	250.0	N/A	N/A	N/A
115.2	2	111.9	-2.90	115.2	1	115.2	0.00
57.6	4	55.9	-2.90	57.6	2	57.6	0.00
38.4	6	37.3	-2.90	38.4	3	38.4	0.00
19.2	12	18.6	-2.90	19.2	6	19.2	0.00
9.60	23	9.73	1.32	9.60	12	9.60	0.00
4.80	47	4.76	-0.83	4.80	24	4.80	0.00
2.40	93	2.41	0.23	2.40	48	2.40	0.00
1.20	186	1.20	0.23	1.20	96	1.20	0.00
0.60	373	0.60	-0.04	0.60	192	0.60	0.00
0.30	746	0.30	-0.04	0.30	384	0.30	0.00

1.8432 MHz System Clock



Transfer Format PHASE Equals Zero

Figure 77 illustrates the timing diagram for an SPI transfer in which PHASE is cleared to 0. The two SCK waveforms show polarity with CLKPOL reset to 0 and with CLKPOL set to one. The diagram may be interpreted as either a Master or Slave timing diagram since the SCK Master-In/Slave-Out (MISO) and Master-Out/Slave-In (MOSI) pins are directly connected between the Master and the Slave.



Figure 77. SPI Timing When PHASE is 0

Transfer Format PHASE Equals One

Figure 78 illustrates the timing diagram for an SPI transfer in which PHASE is one. Two waveforms are depicted for SCK, one for CLKPOL reset to 0 and another for CLKPOL set to 1.



Table 76 provides an example of the Register File addresses if the DMA_ADC Address register contains the value 72H.

ADC Analog Input	Register File Address (Hex) ¹
0	720H-721H
1	722H-723H
2	724H-725H
3	726H-727H
4	728H-729H
5	72AH-72BH
6	72CH-72DH
7	72EH-72FH
8	730H-731H
9	732H-733H
10	734H-735H
11	736H-737H

Table 76. DMA_ADC Register File Address Example

¹ DMAA_ADDR set to 72H.

Table 77. DMA_ADC Address Register (DMAA_ADDR)

BITS	7	6	5	4	3	2	0		
FIELD	DMAA_ADDR								
RESET	Х	Х	Х	Х	Х	Х	Х	Х	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
ADDR	FBDH								

DMAA_ADDR—DMA_ADC Address

These bits specify the seven most-significant bits of the 12-bit Register File addresses used for storing the ADC output data. The ADC Analog Input Number defines the five least-significant bits of the Register File address. Full 12-bit address is {DMAA_ADDR[7:1], 4-bit ADC Analog Input Number, 0}.

Reserved This bit is reserved and must be 0.



- 1. Enable the desired analog inputs by configuring the general-purpose I/O pins for alternate function. This configuration disables the digital input and output drivers.
- 2. Write to the ADC Control register to configure the ADC and begin the conversion. The bit fields in the ADC Control register can be written simultaneously:
 - Write to ANAIN [3:0] to select one of the 12 analog input sources.
 - Clear CONT to 0 to select a single-shot conversion.
 - Write to VREF to enable or disable the internal voltage reference generator.
 - Set CEN to 1 to start the conversion.
- 3. CEN remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power-up before beginning the 5129 cycle conversion.
- 4. When the conversion is complete, the ADC control logic performs the following operations:
 - 10-bit data result written to {ADCD_H[7:0], ADCD_L[7:6]}.
 - CEN resets to 0 to indicate the conversion is complete.
 - An interrupt request is sent to the Interrupt Controller.
- 5. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

Continuous Conversion

When configured for continuous conversion, the ADC continuously performs an analogto-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated only at the end of the first conversion after enabling.

Caution:

In Continuous mode, users must be aware that ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not seen at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.

The steps for setting up the ADC and initiating continuous conversion are as follows:

- 1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This disables the digital input and output driver.
- 2. Write to the ADC Control register to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
 - Write to ANAIN [3:0] to select one of the 12 analog input sources.



```
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.

```
DEG <-- 07H
DEG --> ProgramCounter[15:8]
DEG --> 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



AC Characteristics

The section provides information on the AC characteristics and timing of the Z8 Encore!TM. All AC timing information assumes a standard load of 50pF on all outputs.

Table 102. AC Characteristics

		$V_{DD} = 3.0 - 3.6V$ $T_A = -40^{\circ}C \text{ to } 105^{\circ}C$			
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	1.0	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} = 50ns
T _{XINR}	System Clock Rise Time	_	3	ns	$T_{CLK} = 50$ ns
T _{XINF}	System Clock Fall Time	_	3	ns	$T_{CLK} = 50$ ns



Condition Codes

The C, Z, S and V flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the flag settings are encoded in a 4-bit field called the condition code (cc), which forms Bits 7:4 of the conditional jump instructions. The condition codes are summarized in Table 117. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the flag test operation is used to decide if the conditional jump is executed.

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	-
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1
0011	3	ULE	Unsigned Less Than or Equal	(C OR Z) = 1
0100	4	OV	Overflow	V = 1
0101	5	Ml	Minus	S = 1
0110	6	Z	Zero	Z = 1
0110	6	EQ	Equal	Z = 1
0111	7	С	Carry	C = 1
0111	7	ULT	Unsigned Less Than	C = 1
1000	8	T (or blank)	Always True	-
1001	9	GE	Greater Than or Equal	(S XOR V) = 0
1010	А	GT	Greater Than	(Z OR (S XOR V)) = 0
1011	В	UGT	Unsigned Greater Than	(C = 0 AND Z = 0) = 1
1100	С	NOV	No Overflow	V = 0
1101	D	PL	Plus	S = 0
1110	Е	NZ	Non-Zero	Z = 0
1110	Е	NE	Not Equal	Z = 0
1111	F	NC	No Carry	C = 0
1111	F	UGE	Unsigned Greater Than or Equal	C = 0

Table 117. Condition Codes



Agaomhly		Address Mode				Fl	Fotob	Inctu				
Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Z	S	V	D	H	Cycles	Cycles
LDX dst, src	$dst \leftarrow src$	r	ER	84	-	-	-	-	-	-	3	2
		Ir	ER	85	•						3	3
		R	IRR	86	•						3	4
		IR	IRR	87	•						3	5
		r	X(rr)	88	•						3	4
		X(rr)	r	89	•						3	4
		ER	r	94	•						3	2
		ER	Ir	95	•						3	3
		IRR	R	96	•						3	4
		IRR	IR	97	•						3	5
		ER	ER	E8	•						4	2
		ER	IM	E9	•						4	2
LEA dst, X(src)	$dst \leftarrow src + X$	r	X(r)	98	-	-	-	-	-	-	3	3
		rr	X(rr)	99	•						3	5
MULT dst	dst[15:0] ← dst[15:8] * dst[7:0]	RR		F4	-	-	-	-	-	-	2	8
NOP	No operation			0F	-	-	-	-	-	-	1	2
OR dst, src	$dst \leftarrow dst \text{ OR } src$	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	$dst \leftarrow dst \text{ OR } src$	ER	ER	48	-	*	*	0	-	-	4	3
		ER	IM	49							4	3
Flags Notation:	* = Value is a function o - = Unaffected X = Undefined	of the resul	lt of the o	operation.		0 = 1 =	Res Set	et to to 1	0			

Table 126. eZ8 CPU Instruction Summary (Continued)



Assembly		Addres	ss Mode	Oncode(s)			Fla	ags			- Fetch	Instr
Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Z	S	V	D	Н	Cycles	Cycles
XOR dst, src	$dst \leftarrow dst \text{ XOR } src$	r	r	B2	-	*	*	0	-	-	2	3
		r	Ir	B3	-						2	4
		R	R	B4	-						3	3
		R	IR	B5	-						3	4
		R	IM	B6	-						3	3
		IR	IM	B7	-						3	4
XORX dst, src	$dst \leftarrow dst \text{ XOR } src$	ER	ER	B8	-	*	*	0	-	-	4	3
		ER	IM	B9	-						4	3
Flags Notation:	* = Value is a function of - = Unaffected X = Undefined	of the resu	lt of the o	operation.		0 = 1 =	Res Set	et to to 1	0			

Table 126. eZ8 CPU Instruction Summary (Continued)



Abbreviation	Description	Abbreviation	Description
b	Bit position	IRR	Indirect Register Pair
сс	Condition code	р	Polarity (0 or 1)
X	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
Ir	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