



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

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

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I <sup>2</sup> C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f4802vs020ec">https://www.e-xfl.com/product-detail/zilog/z8f4802vs020ec</a>



External Pin Reset . . . . .	29
Stop Mode Recovery . . . . .	29
Stop Mode Recovery Using Watch-Dog Timer Time-Out . . . . .	29
Stop Mode Recovery Using a GPIO Port Pin Transition . . . . .	30
<b>Low-Power Modes . . . . .</b>	<b>31</b>
Overview . . . . .	31
Stop Mode . . . . .	31
Halt Mode . . . . .	31
<b>General-Purpose I/O . . . . .</b>	<b>33</b>
Overview . . . . .	33
GPIO Port Availability By Device . . . . .	33
Architecture . . . . .	34
GPIO Alternate Functions . . . . .	34
GPIO Interrupts . . . . .	36
GPIO Control Register Definitions . . . . .	36
Port A-H Address Registers . . . . .	37
Port A-H Control Registers . . . . .	38
Port A-H Input Data Registers . . . . .	42
Port A-H Output Data Register . . . . .	43
<b>Interrupt Controller . . . . .</b>	<b>44</b>
Overview . . . . .	44
Interrupt Vector Listing . . . . .	44
Architecture . . . . .	46
Operation . . . . .	46
Master Interrupt Enable . . . . .	46
Interrupt Vectors and Priority . . . . .	47
Interrupt Assertion Types . . . . .	47
Interrupt Control Register Definitions . . . . .	48
Interrupt Request 0 Register . . . . .	48
Interrupt Request 1 Register . . . . .	49
Interrupt Request 2 Register . . . . .	50
IRQ0 Enable High and Low Bit Registers . . . . .	51
IRQ1 Enable High and Low Bit Registers . . . . .	52
IRQ2 Enable High and Low Bit Registers . . . . .	53
Interrupt Edge Select Register . . . . .	54
Interrupt Port Select Register . . . . .	55
Interrupt Control Register . . . . .	56
<b>Timers . . . . .</b>	<b>57</b>
Overview . . . . .	57
Architecture . . . . .	57
Operation . . . . .	58



DMA_ADC Operation .....	123
Configuring DMA_ADC for Data Transfer .....	124
DMA Control Register Definitions .....	124
DMAx Control Register .....	124
DMAx I/O Address Register .....	125
DMAx Address High Nibble Register .....	126
DMAx Start/Current Address Low Byte Register .....	127
DMAx End Address Low Byte Register .....	128
DMA_ADC Address Register .....	128
DMA_ADC Control Register .....	130
DMA Status Register .....	131
<b>Analog-to-Digital Converter .....</b>	<b>132</b>
Overview .....	132
Architecture .....	132
Operation .....	133
Automatic Power-Down .....	133
Single-Shot Conversion .....	133
Continuous Conversion .....	134
DMA Control of the ADC .....	135
ADC Control Register Definitions .....	135
ADC Control Register .....	135
ADC Data High Byte Register .....	137
ADC Data Low Bits Register .....	137
<b>Flash Memory .....</b>	<b>138</b>
Overview .....	138
Operation .....	139
Flash Operation Timing Using the Flash Frequency Registers ..	141
Flash Code Protection Against External Access .....	141
Flash Code Protection Against Accidental Program and Erasure	141
Byte Programming .....	142
Page Erase .....	143
Mass Erase .....	143
Flash Controller Bypass .....	143
Flash Control Register Definitions .....	144
Flash Control Register .....	144
Flash Status Register .....	145
Flash Page Select Register .....	146
Flash Frequency High and Low Byte Registers .....	147



## *List of Tables*

Table 1.	Z8F640x Family Part Selection Guide .....	2
Table 2.	Z8F640x Family Package Options .....	6
Table 3.	Signal Descriptions .....	13
Table 4.	Pin Characteristics of the Z8F640x family .....	15
Table 5.	Z8F640x Family Program Memory Maps .....	18
Table 6.	Z8F640x Family Data Memory Maps .....	19
Table 7.	Register File Address Map .....	20
Table 8.	Reset and STOP Mode Recovery Characteristics and Latency .....	25
Table 9.	Reset Sources and Resulting Reset Type .....	26
Table 10.	STOP Mode Recovery Sources and Resulting Action ...	29
Table 11.	Port Availability by Device and Package Type .....	33
Table 12.	Port Alternate Function Mapping .....	35
Table 13.	Port A-H GPIO Address Registers (PxADDR) .....	37
Table 14.	GPIO Port Registers and Sub-Registers .....	37
Table 15.	Port A-H Control Registers (PxCTL) .....	38
Table 16.	Port A-H Data Direction Sub-Registers .....	39
Table 17.	Port A-H Alternate Function Sub-Registers .....	39
Table 18.	Port A-H Output Control Sub-Registers .....	40
Table 19.	Port A-H High Drive Enable Sub-Registers .....	41
Table 20.	Port A-H Input Data Registers (PxIN) .....	42
Table 21.	Port A-H STOP Mode Recovery Source Enable Sub-Registers .....	42
Table 22.	Port A-H Output Data Register (PxOUT) .....	43
Table 23.	Interrupt Vectors in Order of Priority .....	45
Table 24.	Interrupt Request 0 Register (IRQ0) .....	48
Table 25.	Interrupt Request 1 Register (IRQ1) .....	49
Table 26.	Interrupt Request 2 Register (IRQ2) .....	50
Table 27.	IRQ0 Enable and Priority Encoding .....	51
Table 28.	IRQ0 Enable High Bit Register (IRQ0ENH) .....	51
Table 29.	IRQ0 Enable Low Bit Register (IRQ0ENL) .....	52
Table 30.	IRQ1 Enable and Priority Encoding .....	52
Table 31.	IRQ1 Enable Low Bit Register (IRQ1ENL) .....	53



## Program Memory

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

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

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

# *Reset and Stop Mode Recovery*

## Overview

The Reset Controller within the Z8F640x family devices controls Reset and STOP Mode Recovery operation. In typical operation, the following events cause a Reset to occur:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO)
- Watch-Dog Timer time-out (when configured via the WDT\_RES Option Bit to initiate a reset)
- External  $\overline{\text{RESET}}$  pin assertion
- On-Chip Debugger initiated Reset (OCDCTL[1] set to 1)

When the Z8F640x family device is in Stop mode, a Stop Mode Recovery is initiated by either of the following:

- Watch-Dog Timer time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source
- DBG pin driven Low

## Reset Types

The Z8F640x family provides several different types of Reset operation. Stop Mode Recovery is considered a form of Reset. The type of Reset is a function of both the current operating mode of the Z8F640x family device and the source of the Reset. Table 7 lists the types of Reset and their operating characteristics. The System Reset is longer than the Short Reset to allow additional time for external oscillator start-up.

**Table 7. Reset and Stop Mode Recovery Characteristics and Latency**

Reset Type	Reset Characteristics and Latency		
	Control Registers	eZ8 CPU	Reset Latency (Delay)
System Reset	Reset (as applicable)	Reset	514 WDT Oscillator cycles + 16 System Clock cycles
Short Reset	Reset (as applicable)	Reset	66 WDT Oscillator cycles + 16 System Clock cycles
Stop Mode Recovery	Unaffected, except WDT_CTL register	Reset	514 WDT Oscillator cycles + 16 System Clock cycles



**I<sup>2</sup>C—I<sup>2</sup>C Interrupt Request**

0 = No interrupt request is pending for the I<sup>2</sup>C.

1 = An interrupt request from the I<sup>2</sup>C is awaiting service.

**SPI—SPI Interrupt Request**

0 = No interrupt request is pending for the SPI.

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

**ADC—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.

**Table 24. Interrupt Request 1 Register (IRQ1)**

<b>BITS</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>FIELD</b>	PAD7I	PAD6I	PAD5I	PAD4I	PAD3I	PAD2I	PAD1I	PAD0I
<b>RESET</b>	0	0	0	0	0	0	0	0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>ADDR</b>	FC3H							

**PADxI—Port A or Port D Pin  $x$  Interrupt Request**

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

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

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.

set to 2-byte transfers, the temporary holding register for the Timer Reload High Byte is not bypassed.

**Table 40. Timer 0-3 Reload High Byte Register (TxRH)**

BITS	7	6	5	4	3	2	1	0
FIELD	TRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F02H, F0AH, F12H, F1AH							

**Table 41. Timer 0-3 Reload Low Byte Register (TxRL)**

BITS	7	6	5	4	3	2	1	0
FIELD	TRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F03H, F0BH, F13H, F1BH							

TRH and TRL—Timer Reload Register High and Low

These two bytes form the 16-bit Reload value, {TRH[7:0], TRL[7:0]}. This value is used to set the maximum count value which initiates a timer reload to 0001H. In Compare mode, these two byte form the 16-bit Compare value.



### **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  $2^{\text{PRES}}$ , 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

## Watch-Dog Timer Control Register Definitions

### Watch-Dog Timer Control Register

The Watch-Dog Timer Control (WDTCTL) register, detailed in Table 46, is a Read-Only register that indicates the source of the most recent Reset event, indicates a Stop Mode Recovery event, and indicates a Watch-Dog Timer time-out. Reading this register resets the upper four bits to 0.

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

**Table 46. Watch-Dog Timer Control Register (WDTCTL)**

BITS	7	6	5	4	3	2	1	0
FIELD	POR	STOP	WDT	EXT	Reserved			
RESET	X	X	X	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
ADDR	FF0							

**POR**—Power-On Reset Indicator

If this bit is set to 1, a Power-On Reset 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.

**STOP**—STOP Mode Recovery Indicator

If this bit is set to 1, a STOP Mode Recovery occurred. If the STOP and WDT bits are both set to 1, the STOP Mode Recovery occurred due to a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the STOP Mode Recovery was not caused by a WDT time-out. This bit is reset by a Power-On Reset or a WDT time-out that occurred while not in STOP mode. Reading this register also resets this bit.

**WDT**—Watch-Dog Timer Time-Out Indicator

If this bit is set to 1, a WDT time-out occurred. A Power-On Reset resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit.

**EXT**—External Reset Indicator

If this bit is set to 1, a Reset initiated by the external RESET pin occurred. A Power-On Reset or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.



**CTSE—CTS Enable**

0 = The  $\overline{\text{CTS}}$  signal has no effect on the transmitter.

1 = The UART recognizes the  $\overline{\text{CTS}}$  signal as an enable control from the transmitter.

**PEN—Parity Enable**

This bit enables or disables parity. Even or odd is determined by the PSEL bit.

0 = Parity is disabled.

1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.

**PSEL—Parity Select**

0 = Even parity is transmitted and expected on all received data.

1 = Odd parity is transmitted and expected on all received data.

**SBRK—Send Break**

This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so insure that the transmitter has finished sending data before setting this bit.

0 = No break is sent.

1 = The output of the transmitter is zero.

**STOP—Stop Bit Select**

0 = The transmitter sends one stop bit.

1 = The transmitter sends two stop bits.

**LBEN—Loop Back Enable**

0 = Normal operation.

1 = All transmitted data is looped back to the receiver.

**Table 55. UARTx Control 1 Register (UxCTL1)**

BITS	7	6	5	4	3	2	1	0
FIELD	BIRQ	MPM	MPE	MPBT	Reserved		$\overline{\text{RDAIRQ}}$	IREN
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	F43H and F4BH							

**BIRQ—Baud Rate Generator Interrupt Request**

This bit sets an interrupt request when the Baud Rate Generator times out and is only set if a UART is not enabled. The is bit produces no effect when the UART is enabled.

0 = Interrupts behave as set by UART control.

1 = The Baud Rate Generator generates a receive interrupt when it counts down to zero.

**MPM—Multiprocessor (9-bit) mode Select**

This bit is used to enable Multiprocessor (9-bit) mode.

## SPI Baud Rate High and Low Byte Registers

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 reload value must be greater than or equal to 0002H for proper SPI operation (maximum baud rate is system clock frequency divided by 4). The SPI baud rate is calculated using the following equation:

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

**Table 64. SPI Baud Rate High Byte Register (SPIBRH)**

BITS	7	6	5	4	3	2	1	0
FIELD	BRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F66H							

BRH = SPI Baud Rate High Byte

Most significant byte, BRG[15:8], of the SPI Baud Rate Generator's reload value.

**Table 65. SPI Baud Rate Low Byte Register (SPIBRL)**

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/w
ADDR	F67H							

BRL = SPI Baud Rate Low Byte

Least significant byte, BRG[7:0], of the SPI Baud Rate Generator's reload value.

limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the high period of SCL.

## I<sup>2</sup>C Interrupts

the I<sup>2</sup>C Controller contains three sources of interrupts—Transmit, Receive and Not Acknowledge (NAK) interrupts. NAK interrupts occur when a Not Acknowledge is received from the slave or sent by the I<sup>2</sup>C Controller and the Start or Stop bit is set. This source sets bit 0 and can only be cleared by setting the Start or Stop bit. When this interrupt occurs, the I<sup>2</sup>C Controller waits until it is cleared before performing any action. In an interrupt service routine, this interrupt must be the first thing polled. Receive interrupts occur when a byte of data has been received by the I<sup>2</sup>C master. This interrupt is cleared by reading from the I<sup>2</sup>C Data register. If no action is taken, the I<sup>2</sup>C Controller waits until this interrupt is cleared before performing any other action.

For Transmit interrupts to occur, the TXI bit must be 1 in the I<sup>2</sup>C Control register. Transmit interrupts occur under the following conditions when the transmit data register is empty:

- The I<sup>2</sup>C Controller is idle (not performing an operation).
- The START bit is set and there is no valid data in the I<sup>2</sup>C Shift or I<sup>2</sup>C Data register to shift out.
- The first bit of the byte of an address is shifting out and the RD bit of the I<sup>2</sup>C Status register is deasserted.
- The first bit of a 10-bit address shifts out.
- The first bit of write data shifted out.

► **Note:** Writing to the I<sup>2</sup>C Data register always clears a Transmit interrupt.

## Start and Stop Conditions

The master (I<sup>2</sup>C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I<sup>2</sup>C Controller generates a START condition by pulling the SDA signal low while SCL is high. Then a high-to-low transition occurs on the SDA signal while the clock is High. To complete a transaction, the I<sup>2</sup>C Controller generates a Stop condition by creating a low-to-high transition of the SDA signal in the middle of the high period of the SCL signal. When the SCL signal is High, the master generates a Start bit by pulling a High SDA signal Low and generates a Stop bit by releasing the SDA signal. The Start and Stop signals are found in the I<sup>2</sup>C Control register and must be written by software when the Z8F640x family device must begin or end a transaction.

## Writing a Transaction with a 7-Bit Address

1. The I<sup>2</sup>C Controller shifts the I<sup>2</sup>C Shift register out onto SDA signal.



# *Direct Memory Access Controller*

## Overview

The Z8F640x family device's Direct Memory Access (DMA) Controller provides three independent Direct Memory Access channels. Two of the channels (DMA0 and DMA1) transfer data between the on-chip peripherals and the Register File. The third channel (DMA\_ADC) controls the Analog-to-Digital Converter (ADC) operation and transfers the Single-Shot mode ADC output data to the Register File.

## Operation

### DMA0 and DMA1 Operation

DMA0 and DMA1, referred to collectively as DMA<sub>x</sub>, transfer data either from the on-chip peripheral control registers to the Register File, or from the Register File to the on-chip peripheral control registers. The sequence of operations in a DMA<sub>x</sub> data transfer is:

1. DMA<sub>x</sub> trigger source requests a DMA data transfer.
2. DMA<sub>x</sub> requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA<sub>x</sub> transfers either a single byte or a two-byte word (depending upon configuration) and then returns system bus control back to the eZ8 CPU.
4. If Current Address equals End Address:
  - DMA<sub>x</sub> reloads the original Start Address
  - If configured to generate an interrupt, DMA<sub>x</sub> sends an interrupt request to the Interrupt Controller
  - If configured for single-pass operation, DMA<sub>x</sub> resets the DEN bit in the DMA<sub>x</sub> Control register to 0 and the DMA is disabled.

If Current Address does not equal End Address, the Current Address increments by 1 (single-byte transfer) or 2 (two-byte word transfer).



## Configuring DMA0 and DMA1 for Data Transfer

Follow these steps to configure and enable DMA0 or DMA1:

1. Write to the DMAx I/O Address register to set the Register File address identifying the on-chip peripheral control register. The upper nibble of the 12-bit address for on-chip peripheral control registers is always FH. The full address is {FH, DMAx\_IO[7:0]}
2. Determine the 12-bit Start and End Register File addresses. The 12-bit Start Address is given by {DMAx\_H[3:0], DMA\_START[7:0]}. The 12-bit End Address is given by {DMAx\_H[7:4], DMA\_END[7:0]}.
3. Write the Start and End Register File address high nibbles to the DMAx End/Start Address High Nibble register.
4. Write the lower byte of the Start Address to the DMAx Start/Current Address register.
5. Write the lower byte of the End Address to the DMAx End Address register.
6. Write to the DMAx Control register to complete the following:
  - Select loop or single-pass mode operation
  - Select the data transfer direction (either from the Register File RAM to the on-chip peripheral control register; or from the on-chip peripheral control register to the Register File RAM)
  - Enable the DMAx interrupt request, if desired
  - Select Word or Byte mode
  - Select the DMAx request trigger
  - Enable the DMAx channel

## DMA\_ADC Operation

DMA\_ADC transfers data from the ADC to the Register File. The sequence of operations in a DMA\_ADC data transfer is:

1. ADC completes conversion on the current ADC input channel and signals the DMA controller that two-bytes of ADC data are ready for transfer.
2. DMA\_ADC requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA\_ADC transfers the two-byte ADC output value to the Register File and then returns system bus control back to the eZ8 CPU.
4. If the current ADC Analog Input is the highest numbered input to be converted:
  - DMA\_ADC resets the ADC Analog Input number to 0 and initiates data conversion on ADC Analog Input 0.
  - If configured to generate an interrupt, DMA\_ADC sends an interrupt request to the Interrupt Controller



# *Option Bits*

## Overview

Option Bits allow user configuration of certain aspects of Z8F640x family device operation. The feature configuration data is stored in the Program Memory and read during Reset. The features available for control via the Option Bits are:

- Watch-Dog Timer time-out response selection—interrupt or Short Reset.
- Watch-Dog Timer enabled at Reset.
- The ability to prevent unwanted read access to user code in Program Memory.
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory.

## Operation

### Option Bit Configuration By Reset

Each time the Option Bits are programmed or erased, the Z8F640x family device must be Reset for the change to take place. During any reset operation (System Reset, Short Reset, or Stop Mode Recovery), the Option Bits are automatically read from the Program Memory and written to Option Configuration registers. The Option Configuration registers control operation of the Z8F640x family device. Option Bit control of the Z8F640x family device is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access.

### Option Bit Address Space

The first two bytes of Program Memory at addresses 0000H and 0001H are reserved for the user Option Bits. The byte at Program Memory address 0000H is used to configure user options. The byte at Program Memory address 0001H is reserved for future use and must be left in its unprogrammed state.





- **Read Data Memory (0DH)**—The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the Z8F640x family device is not in Debug mode, this command returns FFH for the data.

```
DBG <-- 0DH
DBG <-- Data Memory Address [15:8]
DBG <-- Data Memory Address [7:0]
DBG <-- Size [15:8]
DBG <-- Size [7:0]
DBG --> 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 Z8F640x family 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 <-- 0EH
DBG --> CRC [15:8]
DBG --> CRC [7:0]
```

- **Step Instruction (10H)**—The Step Instruction command steps one assembly instruction at the current Program Counter (PC) location. If the Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, the OCD ignores this command.

```
DBG <-- 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 Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, the OCD ignores this command.

```
DBG <-- 11H
DBG <-- 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 Z8F640x family device is not in Debug mode or the Read Protect Option Bit is enabled, this command reads and discards one byte.

```
DBG <-- 12H
DBG <-- 1-5 byte opcode
```

**Table 123. Logical Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using Extended Addressing
COM	dst	Complement
OR	dst, src	Logical OR
ORX	dst, src	Logical OR using Extended Addressing
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using Extended Addressing

**Table 124. Program Control Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap



For valuable information about hardware and software development tools, visit the ZiLOG web site at [www.zilog.com](http://www.zilog.com). The latest released version of ZDS can be downloaded from this site.

## Part Number Description

ZiLOG part numbers consist of a number of components, as indicated in the following examples:

ZiLOG Base Products	
Z8	ZiLOG 8-bit microcontroller product
F6	Flash Memory
64	Program Memory Size
01	Device Number
A	Package
N	Pin Count
020	Speed
S	Temperature Range
C	Environmental Flow

<b>Packages</b>	A = LQFP
	S = SOIC
	H = SSOP
	P = PDIP
	V = PLCC
	F = QFP
<b>Pin Count</b>	H = 20 pins
	J = 28 pins
	M = 40 pins
	N = 44 pins
	R = 64 pins
	S = 68 pins
	T = 80 pins
<b>Speed</b>	020 = 20MHz
<b>Temperature</b>	S = 0°C to +70°C
	E = -40°C to +105°C
<b>Environmental Flow</b>	C = Plastic-Standard

Example: Part number Z8F06401AN020SC is an 8-bit microcontroller product in an LQFP package, using 44 pins, operating with a maximum 20MHz external clock frequency over a 0°C to +70°C temperature range and built using the Plastic-Standard environmental flow.



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

## D

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

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

## E

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



- register 109, 126, 184
  - ADC control (ADCCTL) 135
  - ADC data high byte (ADCDH) 137
  - ADC data low bits (ADC DL) 137
  - baud low and high byte (I2C) 121
  - baud rate high and low byte (SPI) 110
  - control (SPI) 107
  - control, I2C 119
  - data, SPI 106
  - DMA status (DMAA\_STAT) 131
  - DMA\_ADC address 128
  - DMA\_ADC control DMAACTL) 130
  - DMAx address high nibble (DMAxH) 126
  - DMAx control (DMAxCTL) 124
  - DMAx end/address low byte (DMAxEND) 128
  - DMAx start/current address low byte register (DMAxSTART) 128
  - flash control (FCTL) 144
  - flash high and low byte (FFREQH and FREEQL) 147
  - flash page select (FPS) 146
  - flash status (FSTAT) 145
  - GPIO port A-H address (PxADDR) 37
  - GPIO port A-H alternate function
    - sub-registers 39
  - GPIO port A-H control address (PxCTL) 38
  - GPIO port A-H data direction sub-registers 39
  - I2C baud rate high (I2CBRH) 121
  - I2C control (I2CCTL) 119
  - I2C data (I2CDATA) 118
  - I2C status 118
  - I2C status (I2CSTAT) 118
  - I2Cbaud rate low (I2CBRL) 121
  - mode, SPI 109
  - OCD control 161
  - OCD status 162
  - OCD watchpoint address 164
  - OCD watchpoint control 163
  - OCD watchpoint data 164
  - SPI baud rate high byte (SPIBRH) 110
  - SPI baud rate low byte (SPIBRL) 110
  - SPI control (SPICTL) 107
  - SPI data (SPIDATA) 106
  - SPI status (SPISTAT) 108
  - status, I2C 118
  - status, SPI 108
  - UARTx baud rate high byte (UxBRH) 91
  - UARTx baud rate low byte (UxBRL) 92
  - UARTx Control 0 (UxCTL0) 89
  - UARTx control 1 (UxCTL1) 90
  - UARTx receive data (UxRXD) 87
  - UARTx status 0 (UxSTAT0) 87
  - UARTx status 1 (UxSTAT1) 89
  - UARTx transmit data (UxTXD) 86
  - watch-dog timer control (WDTCTL) 75
  - watch-dog timer reload high byte (WDTH) 76
  - watch-dog timer reload low byte (WDTL) 77
  - watch-dog timer reload upper byte (WDTU) 76
- register file 17
- register file address map 20
- register pair 184
- register pointer 185
- reset
  - and stop mode characteristics 25
  - and stop mode recovery 25
  - carry flag 188
  - controller 5
  - sources 26
- RET 190
- return 190
- return information 216
- RL 191
- RLC 191
- rotate and shift instructions 191
- rotate left 191
- rotate left through carry 191
- rotate right 191
- rotate right through carry 191
- RP 185
- RR 184, 191
- rr 184
- RRC 191
  
- S**
  - SBC 188
  - SCF 188, 189
  - SKC 101