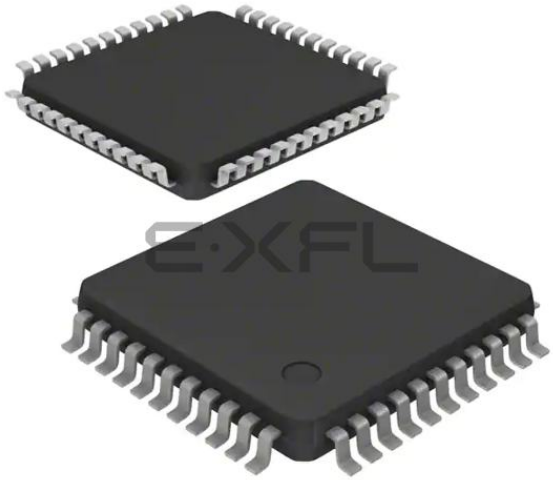

**Welcome to E-XFL.COM**

## 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 | I²C, IrDA, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, POR, PWM, WDT |
| Number of I/O | 31 |
| 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 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LQFP |
| Supplier Device Package | 44-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f3201an020ec |

# List of Figures

- Power-On Reset (POR)
- 3.0-3.6V operating voltage with 5V-tolerant inputs
- 0° to +70°C standard temperature and -40° to +105°C extended temperature operating ranges

## Part Selection Guide

Table 1 identifies the basic features and package styles available for each device within the Z8F640x family product line.

**Table 1. Z8F640x Family Part Selection Guide**

| Part Number | Flash (KB) | RAM (KB) | I/O | 16-bit Timers with PWM | ADC Inputs | UARTs with IrDA | $I^2C$ | SPI | 40/44-pin packages | 64/68-pin packages | 80-pin package |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Z8F1601 | 16 | 2 | 31 | 3 | 8 | 2 | 1 | 1 | X | | |
| Z8F1602 | 16 | 2 | 46 | 4 | 12 | 2 | 1 | 1 | | X | |
| Z8F2401 | 24 | 2 | 31 | 3 | 8 | 2 | 1 | 1 | X | | |
| Z8F2402 | 24 | 2 | 46 | 4 | 12 | 2 | 1 | 1 | | X | |
| Z8F3201 | 32 | 2 | 31 | 3 | 8 | 2 | 1 | 1 | X | | |
| Z8F3202 | 32 | 2 | 46 | 4 | 12 | 2 | 1 | 1 | | X | |
| Z8F4801 | 48 | 4 | 31 | 3 | 8 | 2 | 1 | 1 | X | | |
| Z8F4802 | 48 | 4 | 46 | 4 | 12 | 2 | 1 | 1 | | X | |
| Z8F4803 | 48 | 4 | 60 | 4 | 12 | 2 | 1 | 1 | | | X |
| Z8F6401 | 64 | 4 | 31 | 3 | 8 | 2 | 1 | 1 | X | | |
| Z8F6402 | 64 | 4 | 46 | 4 | 12 | 2 | 1 | 1 | | X | |
| Z8F6403 | 64 | 4 | 60 | 4 | 12 | 2 | 1 | 1 | | | X |

**Table 6. Register File Address Map (Continued)**

| 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 19. Port A-H STOP Mode Recovery Source Enable Sub-Registers**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | PSMRE7 | PSMRE6 | PSMRE5 | PSMRE4 | PSMRE3 | PSMRE2 | PSMRE1 | PSMRE0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| ADDR | If 05H in Port A-H Address Register, accessible via Port A-H Control Register | | | | | | | |

PSMRE[7:0]—Port STOP Mode Recovery Source Enabled
0 = The Port pin is not configured as a STOP Mode Recovery source. Transitions on this pin during Stop mode do not initiate STOP Mode Recovery.

1 = The Port pin is configured as a STOP Mode Recovery source. Any logic transition on this pin during Stop mode initiates STOP Mode Recovery.

## Port A-H Input Data Registers

Reading from the Port A-H Input Data registers (Table 20) returns the sampled values from the corresponding port pins. The Port A-H Input Data registers are Read-only.

**Table 20. Port A-H Input Data Registers (PxIN)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | PIN7 | PIN6 | PIN5 | PIN4 | PIN3 | PIN2 | PIN1 | PIN0 |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R | R | R | R | R | R | R | R |
| ADDR | FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH | | | | | | | |

PIN[7:0]—Port Input Data
Sampled data from the corresponding port pin input.
0 = Input data is logical 0 (Low).
1 = Input data is logical 1 (High).

**Table 22. Interrupt Vectors in Order of Priority**

| Priority | Program Memory Vector Address | Interrupt Source | Interrupt Assertion Type |
|---|---|---|---|
| Highest | 0002h | Reset (not an interrupt) | Not applicable |
| | 0004h | Watch-Dog Timer | Continuous assertion |
| | 0006h | Illegal Instruction Trap (not an interrupt) | Not applicable |
| | 0008h | Timer 2 | Single assertion (pulse) |
| | 000Ah | Timer 1 | Single assertion (pulse) |
| | 000Ch | Timer 0 | Single assertion (pulse) |
| | 000Eh | UART 0 receiver | Continuous assertion |
| | 0010h | UART 0 transmitter | Continuous assertion |
| | 0012h | $I^2C$ | Continuous assertion |
| | 0014h | SPI | Continuous assertion |
| | 0016h | ADC | Single assertion (pulse) |
| | 0018h | Port A7 or Port D7, rising or falling input edge | Single assertion (pulse) |
| | 001Ah | Port A6 or Port D6, rising or falling input edge | Single assertion (pulse) |
| | 001Ch | Port A5 or Port D5, rising or falling input edge | Single assertion (pulse) |
| | 001Eh | Port A4 or Port D4, rising or falling input edge | Single assertion (pulse) |
| | 0020h | Port A3 or Port D3, rising or falling input edge | Single assertion (pulse) |
| | 0022h | Port A2 or Port D2, rising or falling input edge | Single assertion (pulse) |
| | 0024h | Port A1 or Port D1, rising or falling input edge | Single assertion (pulse) |
| | 0026h | Port A0 or Port D0, rising or falling input edge | Single assertion (pulse) |
| | 0028h | Timer 3 *(not available in 40/44-pin packages)* | Single assertion (pulse) |
| | 002Ah | UART 1 receiver | Continuous assertion |
| | 002Ch | UART 1 transmitter | Continuous assertion |
| | 002Eh | DMA | Single assertion (pulse) |
| | 0030h | Port C3, both input edges | Single assertion (pulse) |
| | 0032h | Port C2, both input edges | Single assertion (pulse) |
| | 0034h | Port C1, both input edges | Single assertion (pulse) |
| Lowest | 0036h | Port C0, both input edges | Single assertion (pulse) |

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

**Table 49. Watch-Dog Timer Reload Low Byte Register (WDTL)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | WDTL | | | | | | | |
| 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 | FF3H | | | | | | | |
| 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.

0 = No parity error has occurred.
1 = A parity error has occurred.

OE—Overrun Error
This bit indicates that an overrun error has occurred. An overrun occurs when new data is received and the UART Receive Data register has not been read. If the RDA bit is reset to 0, then reading the UART Receive Data register clears this bit.
0 = No overrun error occurred.
1 = An overrun error occurred.

FE—Framing Error
This bit indicates that a framing error (no Stop bit following data reception) was detected. Reading the UART Receive Data register clears this bit.
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 zeros then 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.

**Table 53. UART*x* Status 1 Register (U*x*STAT1)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | Reserved | | | | | | | MPRX |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |
| ADDR | F44H and F4CH | | | | | | | |

Reserved
These bits are reserved and must be 0.

MPRX—Multiprocessor Receive
This status bit is for the receiver and reflects the actual status of the last multiprocessor bit received. Reading from the UART Data register resets this bit to 0.

## UART*x* Control 0 and Control 1 Registers

The UART*x* Control 0 and Control 1 registers (Tables 54 and 55) configure the properties of the UART's transmit and receive operations. The UART Control registers must ben be written while the UART is enabled.

**Table 54. UART*x* Control 0 Register (U*x*CTL0)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | TEN | REN | CTSE | PEN | PSEL | SBRK | STOP | LBEN |
| 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 | F42H and F4AH | | | | | | | |

TEN—Transmit Enable
This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled.
0 = Transmitter disabled.
1 = Transmitter enabled.

REN—Receive Enable
This bit enables or disables the receiver.
0 = Receiver disabled.
1 = Receiver enabled.

7. The I2C Controller loads the I2C Shift register with the contents of the I2C Data register.
8. After one bit of address is shifted out by the SDA signal, the Transmit interrupt is asserted.
9. Software responds by writing the second byte of address into the contents of the I2C Data register.
10. The I2C Controller shifts the rest of the first byte of address and write bit out by the SDA signal.
11. The I2C slave sends an acknowledge by pulling the SDA signal low during the next high period of SCL. The I2C Controller sets the ACK bit in the I2C Status register.
12. The I2C Controller loads the contents of the I2C Shift register with the contents of the I2C Data register.
13. The I2C Controller shifts the data out by the SDA signal. After the first bit has been sent, the Transmit interrupt is asserted.
14. Software responds by writing the data to be written out to the I2C Control register.
15. The I2C Controller shifts out the rest of the second byte of slave address by the SDA signal.
16. The I2C slave sends an acknowledge by pulling the SDA signal low during the next high period of SCL. The I2C Controller sets the ACK bit in the I2C Status register.
17. The I2C Controller shifts the data out by the SDA signal. After the first bit is sent, the Transmit interrupt is asserted.
18. Software responds by asserting the STOP bit of the I2C Control register.
19. The I2C Controller completes transmission of the data on the SDA signal.
20. The I2C Controller sends the STOP condition to the I2C bus.

## Reading a Transaction with a 7-Bit Address

Figure 81 illustrates the data transfer format for a receive operation on a 7-bit addressed slave. The shaded regions indicate data transferred from the I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

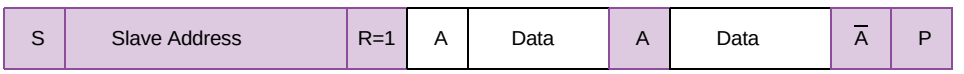


**Figure 81. Receive Data Transfer Format for a 7-Bit Addressed Slave**

The data transfer format for a receive operation on a 7-bit addressed slave is as follows:

1. Software writes the I2C Data register with a 7-bit slave address followed by a 1 (read).
2. Software asserts the START bit of the I2C Control register.
3. Software asserts the NAK bit of the I2C Control register so that after the first byte of data has been read by the I2C Controller, a Not Acknowledge is sent to the I2C slave.
4. The I2C Controller sends the START condition.
5. The I2C Controller sends the address and read bit by the SDA signal.
6. The I2C slave sends an Acknowledge by pulling the SDA signal Low during the next high period of SCL.
7. The I2C Controller reads the first byte of data from the I2C slave.
8. The I2C Controller asserts the Receive interrupt.
9. Software responds by reading the I2C Data register.
10. The I2C Controller sends a NAK to the I2C slave.
11. A NAK interrupt is generated by the I2C Controller.
12. Software responds by setting the STOP bit of the I2C Control register.
13. A STOP condition is sent to the I2C slave.

## Reading a Transaction with a 10-Bit Address

Figure 82 illustrates the receive format for a 10-bit addressed slave. The shaded regions indicate data transferred from the I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

| S | Slave Address 1st 7 bits | W=0 | A | Slave address 2nd Byte | A | S | Slave Address 1st 7 bits | R=1 | A | Data | A | Data | $\overline{A}$ | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Figure 82. Receive Data Format for a 10-Bit Addressed Slave**

The first seven bits transmitted in the first byte are 11110XX. The two bits XX are the two most-significant bits of the 10-bit address. The lowest bit of the first byte transferred is the write signal.

The data transfer format for a receive operation on a 10-bit addressed slave is as follows:

1. Software writes an address 11110B followed by the two address bits and a 0 (write).
2. Software asserts the START bit of the I2C Control register.
3. The I2C Controller sends the Start condition.

**Table 74. DMA*x* Start/Current Address Low Byte Register (DMA*x*START)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | DMA_START | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| ADDR | FB3H, FHBH | | | | | | | |

DMA_START—DMA*x* Start/Current Address Low
These bits, with the four lower bits of the DMA*x*_H register, form the 12-bit Start/Current address. The full 12-bit address is given by {DMA_START_H[3:0], DMA_START[7:0]}.

## DMA*x* End Address Low Byte Register

The DMA*x* End Address Low Byte register, in conjunction with the DMA*x*_H register, forms a 12-bit End Address.

**Table 75. DMA*x* End Address Low Byte Register (DMA*x*END)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | DMA_END | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| ADDR | FB4H, FBCH | | | | | | | |

DMA_END—DMA*x* End Address Low
These bits, with the four upper bits of the DMA*x*_H register, form a 12-bit address. This address is the ending location of the DMA*x* transfer. The full 12-bit address is given by {DMA_END_H[3:0], DMA_END[7:0]}.

## DMA_ADC Address Register

The DMA_ADC Address register points to a block of the Register File to store ADC conversion values as illustrated in Table 76. This register contains the seven most-significant bits of the 12-bit Register File addresses. The five least-significant bits are calculated from the ADC Analog Input number (5-bit base address is equal to twice the ADC Analog Input number). The 10-bit ADC conversion data is stored as two bytes with the most significant byte of the ADC data stored at the even numbered Register File address.

## Flash Frequency High and Low Byte Registers

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) and is calculated using the following equation:.

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

**Caution:** Flash programming and erasure is not supported for system clock frequencies below 32KHz (32768Hz) or above 20MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to insure proper operation of the Z8F640x family device.

**Table 88. 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 89. Flash Frequency Low Byte Register (FFREQL)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | FFREQL | | | | | | | |
| 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 | FFBH | | | | | | | |

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

If the OCD receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. The Auto-Baud Detector/Generator can then be reconfigured 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 should transmit a Serial Break on the DBG pin when first connecting to the Z8F640x family device 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 Z8F640x family 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 High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

## Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If Breakpoints are enabled, the OCD enters Debug mode and idles the eZ8 CPU. If Breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP.

### Breakpoints in Flash Memory

The BRK instruction is opcode 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a Breakpoint, write 00H to the desired address, overwriting the current instruction. To remove a Breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

## Watchpoints

The On-Chip Debugger can set one Watchpoint to cause a Debug Break. The Watchpoint identifies a single Register File address. The Watchpoint can be set to break on reads and/or writes of the selected Register File address. Additionally, the Watchpoint can be configured to break only when a specific data value is read and/or written from the specified reg-

ister. When the Watchpoint event occurs, the Z8F640x family device enters Debug mode and the DBGMODE bit in the OCDCTL register becomes 1.

### Runtime Counter

The On-Chip Debugger contains a 16-bit Runtime Counter. It counts system clock cycles between Breakpoints. The counter starts counting when the On-Chip Debugger leaves Debug mode and stops counting when it enters Debug mode again or when it reaches the maximum count of FFFFH.

## On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation of the Z8F640x family device, only a subset of the OCD commands are available. In Debug mode, all OCD commands become available unless the user code and control registers are protected by programming the Read Protect Option Bit (RP). The Read Protect Option Bit prevents the code in memory from being read out of the Z8F640x family device. When this option is enabled, several of the OCD commands are disabled. Table 93 contains a summary of the On-Chip Debugger commands. Each OCD command is described in further detail in the bulleted list following Table 93. Table 93 indicates those commands that operate when the Z8F640x family device is not in Debug mode (normal operation) and those commands that are disabled by programming the Read Protect Option Bit.

**Table 93. On-Chip Debugger Commands**

| Debug Command | Command Byte | Enabled when NOT in Debug mode? | Disabled by Read Protect Option Bit |
|---|---|---|---|
| Read OCD Revision | 00H | Yes | - |
| Reserved | 01H | - | - |
| Read OCD Status Register | 02H | Yes | - |
| Read Runtime Counter | 03H | - | - |
| Write OCD Control Register | 04H | Yes | Cannot clear DBGMODE bit |
| Read OCD Control Register | 05H | Yes | - |
| Write Program Counter | 06H | - | Disabled |
| Read Program Counter | 07H | - | Disabled |
| Write Register | 08H | - | Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control register. |
| Read Register | 09H | - | Disabled |

# Electrical Characteristics

## Absolute Maximum Ratings

Stresses greater than those listed in Table 100 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, unused inputs must be tied to one of the supply voltages ($V_{DD}$ or $V_{SS}$).

**Table 100. Absolute Maximum Ratings**

| Parameter | Minimum | Maximum | Units | Notes |
|---|---|---|---|---|
| Ambient temperature under bias | -40 | +105 | C | |
| Storage temperature | –65 | +150 | C | |
| Voltage on any pin with respect to $V_{SS}$ | –0.3 | +5.5 | V | 1 |
| Voltage on $V_{DD}$ pin with respect to $V_{SS}$ | –0.3 | +3.6 | V | |
| Maximum current on input and/or inactive output pin | –5 | +5 | µA | |
| Maximum output current from active output pin | -25 | +25 | mA | |
| **80-Pin QFP Maximum Ratings at -40°C to 70°C** | | | | |
| Total power dissipation | | 550 | mW | |
| Maximum current into $V_{DD}$ or out of $V_{SS}$ | | 150 | mA | |
| **80-Pin QFP Maximum Ratings at 70°C to 105°C** | | | | |
| Total power dissipation | | 200 | mW | |
| Maximum current into $V_{DD}$ or out of $V_{SS}$ | | 56 | mA | |
| **68-Pin PLCC Maximum Ratings at -40°C to 70°C** | | | | |
| Total power dissipation | | 1000 | mW | |
| Maximum current into $V_{DD}$ or out of $V_{SS}$ | | 275 | mA | |

Notes:
1. This voltage applies to all pins except the following: $V_{DD}$, $AV_{DD}$, pins supporting analog input (Port B and Port H), $\overline{\text{RESET}}$, and where noted otherwise.

**Table 115. Notational Shorthand**

| Notation | Description | Operand | Range |
|---|---|---|---|
| b | Bit | b | b represents a value from 0 to 7 (000B to 111B). |
| cc | Condition Code | — | See Condition Codes overview in the eZ8 CPU User Manual. |
| 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 116 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

**Table 121. CPU Control Instructions**

| Mnemonic | Operands | Instruction |
|---|---|---|
| CCF | — | Complement Carry Flag |
| DI | — | Disable Interrupts |
| EI | — | Enable Interrupts |
| HALT | — | Halt Mode |
| NOP | — | No Operation |
| RCF | — | Reset Carry Flag |
| SCF | — | Set Carry Flag |
| SRP | src | Set Register Pointer |
| STOP | — | Stop Mode |
| WDT | — | Watch-Dog Timer Refresh |

**Table 122. Load Instructions**

| Mnemonic | Operands | Instruction |
|---|---|---|
| CLR | dst | Clear |
| LD | dst, src | Load |
| LDC | dst, src | Load Constant to/from Program Memory |
| LDCI | dst, src | Load Constant to/from Program Memory and Auto-Increment Addresses |
| LDE | dst, src | Load External Data to/from Data Memory |
| LDEI | dst, src | Load External Data to/from Data Memory and Auto-Increment Addresses |
| LDX | dst, src | Load using Extended Addressing |
| LEA | dst, X(src) | Load Effective Address |
| POP | dst | Pop |
| POPX | dst | Pop using Extended Addressing |
| PUSH | src | Push |
| PUSHX | src | Push using Extended Addressing |

**Table 126. 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LD dst, rc | dst ← src | r | IM | 0C-FC | - | - | - | - | - | - | 2 | 2 |
| | | r | X(r) | C7 | | | | | | | 3 | 3 |
| | | X(r) | r | D7 | | | | | | | 3 | 4 |
| | | r | Ir | E3 | | | | | | | 2 | 3 |
| | | R | R | E4 | | | | | | | 3 | 2 |
| | | R | IR | E5 | | | | | | | 3 | 3 |
| | | R | IM | E6 | | | | | | | 3 | 3 |
| | | IR | IM | E7 | | | | | | | 3 | 3 |
| | | Ir | r | F3 | | | | | | | 2 | 3 |
| | | IR | R | F5 | | | | | | | 3 | 3 |
| LDC dst, src | dst ← src | r | Irr | C2 | - | - | - | - | - | - | 2 | 5 |
| | | Ir | Irr | C5 | | | | | | | 2 | 9 |
| | | Irr | r | D2 | | | | | | | 2 | 5 |
| LDCI dst, src | dst ← src<br>r ← r + 1<br>rr ← rr + 1 | Ir | Irr | C3 | - | - | - | - | - | - | 2 | 9 |
| | | Irr | Ir | D3 | | | | | | | 2 | 9 |
| LDE dst, src | dst ← src | r | Irr | 82 | - | - | - | - | - | - | 2 | 5 |
| | | Irr | r | 92 | | | | | | | 2 | 5 |
| LDEI dst, src | dst ← src<br>r ← r + 1<br>rr ← rr + 1 | Ir | Irr | 83 | - | - | - | - | - | - | 2 | 9 |
| | | Irr | Ir | 93 | | | | | | | 2 | 9 |
| Flags Notation: | * = Value is a function of the result of the operation.<br>- = Unaffected<br>X = Undefined | | | | 0 = Reset to 0<br>1 = Set to 1 | | | | | | | |

**Table 126. 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LDX dst, src | dst ← 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 ← 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 ← 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 |

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