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 | 46 |
| Program Memory Size | 64KB (64K 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 | https://www.e-xfl.com/product-detail/zilog/z8f6422vs020ec00tr |

## I²C

The I²C controller makes the Z8 Encore! XP compatible with the I²C protocol. The I²C controller consists of two bidirectional bus lines, a serial data (SDA) line and a serial clock (SCL) line.

## Serial Peripheral Interface

The serial peripheral interface allows the Z8 Encore! XP to exchange data between other peripheral devices such as EEPROMs, A/D converters and ISDN devices. The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface.

## Timers

Up to four 16-bit reloadable timers can be used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in One-Shot, Continuous, Gated, Capture, Compare, Capture and Compare, and PWM modes. Only 3 timers (Timers 0-2) are available in the 44-pin packages.

## Interrupt Controller

The 64K Series products support up to 24 interrupts. These interrupts consist of 12 internal and 12 GPIO pins. The interrupts have 3 levels of programmable interrupt priority.

## Reset Controller

The Z8 Encore! can be reset using the $\overline{\text{RESET}}$ pin, Power-On Reset, Watchdog Timer, STOP mode exit, or Voltage Brownout (VBO) warning signal.

## On-Chip Debugger

The Z8 Encore! XP features an integrated On-Chip Debugger. The OCD provides a rich set of debugging capabilities, such as reading and writing registers, programming the Flash, setting breakpoints and executing code. A single-pin interface provides communication to the OCD.

## DMA Controller

The 64K Series features three channels of DMA. Two of the channels are for register RAM to and from I/O operations. The third channel automatically controls the transfer of data from the ADC to the memory.

**Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)**

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No |
|---|---|---|---|---|
| **Timer 3 (unavailable in the 44-pin packages)** | | | | |
| F18 | Timer 3 High Byte | T3H | 00 | 90 |
| F19 | Timer 3 Low Byte | T3L | 01 | 90 |
| F1A | Timer 3 Reload High Byte | T3RH | FF | 91 |
| F1B | Timer 3 Reload Low Byte | T3RL | FF | 91 |
| F1C | Timer 3 PWM High Byte | T3PWMH | 00 | 92 |
| F1D | Timer 3 PWM Low Byte | T3PWML | 00 | 92 |
| F1E | Timer 3 Control 0 | T3CTL0 | 00 | 93 |
| F1F | Timer 3 Control 1 | T3CTL1 | 00 | 94 |
| 20-3F | Reserved | — | XX | |
| **UART 0** | | | | |
| F40 | UART0 Transmit Data | U0TXD | XX | 114 |
| | UART0 Receive Data | U0RXD | XX | 115 |
| F41 | UART0 Status 0 | U0STAT0 | 0000011Xb | 115 |
| F42 | UART0 Control 0 | U0CTL0 | 00 | 117 |
| F43 | UART0 Control 1 | U0CTL1 | 00 | 117 |
| F44 | UART0 Status 1 | U0STAT1 | 00 | 115 |
| F45 | UART0 Address Compare Register | U0ADDR | 00 | 120 |
| F46 | UART0 Baud Rate High Byte | U0BRH | FF | 120 |
| F47 | UART0 Baud Rate Low Byte | U0BRL | FF | 120 |
| **UART 1** | | | | |
| F48 | UART1 Transmit Data | U1TXD | XX | 114 |
| | UART1 Receive Data | U1RXD | XX | 115 |
| F49 | UART1 Status 0 | U1STAT0 | 0000011Xb | 115 |
| F4A | UART1 Control 0 | U1CTL0 | 00 | 117 |
| F4B | UART1 Control 1 | U1CTL1 | 00 | 117 |
| F4C | UART1 Status 1 | U1STAT1 | 00 | 115 |
| F4D | UART1 Address Compare Register | U1ADDR | 00 | 120 |
| F4E | UART1 Baud Rate High Byte | U1BRH | FF | 120 |
| F4F | UART1 Baud Rate Low Byte | U1BRL | FF | 120 |
| **$I^2C$** | | | | |
| F50 | $I^2C$ Data | I2CDATA | 00 | 156 |
| F51 | $I^2C$ Status | I2CSTAT | 80 | 157 |
| F52 | $I^2C$ Control | I2CCTL | 00 | 158 |
| F53 | $I^2C$ Baud Rate High Byte | I2CBRH | FF | 160 |
| F54 | $I^2C$ Baud Rate Low Byte | I2CBRL | FF | 160 |
| F55 | $I^2C$ Diagnostic State | I2CDST | C0 | 161 |
| F56 | $I^2C$ Diagnostic Control | I2CDIAG | 00 | 163 |
| F57-F5F | Reserved | — | XX | |
| **Serial Peripheral Interface (SPI)** | | | | |
| F60 | SPI Data | SPIDATA | XX | 137 |

**Timer 1 PWM High Byte**
T1PWMH (F0CH - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 1 PWM value [15:8]

**Timer 1 PWM Low Byte**
T1PWML (F0DH - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 1 PWM value [7:0]

**Timer 1 Control 0**
T1CTL0 (F0EH - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Reserved

Cascade Timer
0 = Timer 1 Input signal is GPIO pin
1 = Timer 1 Input signal is Timer 0 out

Reserved

**Timer 1 Control 1**
T1CTL1 (F0FH - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

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

Prescale Value
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

Timer Input/Output Polarity
Operation of this bit is a function of the current operating mode of the timer

Timer Enable
0 = Timer is disabled
1 = Timer is enabled

**Timer 2 High Byte**
T2H (F10H - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 2 current count value [15:8]

**Timer 2 Low Byte**
T2L (F11H - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 2 current count value [7:0]

**Timer 2 Reload High Byte**
T2RH (F12H - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 2 reload value [15:8]

**Timer 2 Reload Low Byte**
T2RL (F13H- Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 2 reload value [7:0]

**Timer 2 PWM High Byte**
T2PWMH (F14H - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 2 PWM value [15:8]

**Timer 2 PWM Low Byte**
T2PWML (F15H - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Timer 2 PWM value [7:0]

**Timer 2 Control 0**
T2CTL0 (F16H - Read/Write)

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|

Reserved

Cascade Timer
0 = Timer 2 Input signal is GPIO pin
1 = Timer 2 Input signal is Timer 1 out

Reserved

One-Shot time-out, first set the TPOL bit in the Timer Control 1 Register to the start value before beginning ONE-SHOT mode. Then, after starting the timer, set TPOL to the opposite bit value.

Follow the steps below for configuring a timer for ONE-SHOT mode and initiating the count:

1. Write to the Timer Control 1 register to:
   - Disable the timer
   - Configure the timer for ONE-SHOT mode
   - Set the prescale value
   - If using the Timer Output alternate function, set the initial output level (High or Low)
2. Write to the Timer High and Low Byte registers to set the starting count value
3. Write to the Timer Reload High and Low Byte registers to set the Reload value
4. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function
6. Write to the Timer Control 1 register to enable the timer and initiate counting

In ONE-SHOT mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{ONE-SHOT Mode Time-Out Period (s)} = \frac{(\text{Reload Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

### CONTINUOUS Mode

In CONTINUOUS mode, the timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon timer Reload.

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

1. Write to the Timer Control 1 register to:
   - Disable the timer
   - Configure the timer for CONTINUOUS mode
   - Set the prescale value
   - If using the Timer Output alternate function, set the initial output level (High or Low)

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

8. Execute an EI instruction to enable interrupts.

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

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

## Receiving Data using the Polled Method

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

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

### Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte has been received and is available in the UART Receive Data register. This interrupt can be disabled independent of the other receiver interrupt sources. The received data interrupt occurs once the receive character has been received and placed in the Receive Data register. Software must respond to this received data available condition before the next character is completely received to avoid an overrun error.

**Note:** *In MULTIPROCESSOR mode (`MPEN` = 1), the receive data interrupts are dependent on the multiprocessor configuration and the most recent address byte.*

- A break is received
- An overrun is detected
- A data framing error is detected

### UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the UART Status 0 register is updated to indicate the overrun condition (and Break Detect, if applicable). The `RDA` bit is set to 1 to indicate that the Receive Data register contains a data byte. However, because the overrun error occurred, this byte may not contain valid data and should be ignored. The `BRKD` bit indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data register must be read again to clear the error bits is the UART Status 0 register. Updates to the Receive Data register occur only when the next data word is received.

### UART Data and Error Handling Procedure

Figure 18 on page 113 displays the recommended procedure for use in UART receiver interrupt service routines.

**Table 55. UART Status 1 Register (UxSTAT1)**

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

Reserved—Must be 0.

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

MPRX—Multiprocessor Receive
Returns the value of the last multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

## UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (see Table 56 and Table 57 on page 118) configure the properties of the UART's transmit and receive operations. The UART Control registers must not been written while the UART is enabled.

**Table 56. UART Control 0 Register (UxCTL0)**

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

**Table 64. SPI Control Register (SPICTL)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | IRQE | STR | BIRQ | PHASE | CLKPOL | WOR | MMEN | SPIEN |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | F61H | | | | | | | |

IRQE—Interrupt Request Enable
0 = SPI interrupts are disabled. No interrupt requests are sent to the Interrupt Controller.
1 = SPI interrupts are enabled. Interrupt requests are sent to the Interrupt Controller.

STR—Start an SPI Interrupt Request
0 = No effect.
1 = Setting this bit to 1 also sets the `IRQ` bit in the SPI Status register to 1. Setting this bit forces the SPI to send an interrupt request to the Interrupt Control. This bit can be used by software for a function similar to transmit buffer empty in a UART. Writing a 1 to the `IRQ` bit in the SPI Status register clears this bit to 0.

BIRQ—BRG Timer Interrupt Request
If the SPI is enabled, this bit has no effect. If the SPI is disabled:
0 = The Baud Rate Generator timer function is disabled.
1 = The Baud Rate Generator timer function and time-out interrupt are enabled.

PHASE—Phase Select
Sets the phase relationship of the data to the clock. For more information on operation of the PHASE bit, see SPI Clock Phase and Polarity Control on page 132.

CLKPOL—Clock Polarity
0 = SCK idles Low (0).
1 = SCK idle High (1).

WOR—Wire-OR (OPEN-DRAIN) Mode Enabled
0 = SPI signal pins not configured for open-drain.
1 = All four SPI signal pins (SCK, $\overline{SS}$, MISO, MOSI) configured for open-drain function. This setting is typically used for multi-master and/or multi-slave configurations.

MMEN—SPI Master Mode Enable
0 = SPI configured in Slave mode.
1 = SPI configured in Master mode.

SPIEN—SPI Enable
0 = SPI disabled.
1 = SPI enabled.

# Direct Memory Access Controller

## Overview

The 64K Series 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 ADC operation and transfers SINGLE-SHOT mode ADC output data to the Register File.

## Operation

### DMA0 and DMA1 Operation

DMA0 and DMA1, referred to collectively as DMA*x*, 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*x* data transfer is:

1. DMA*x* trigger source requests a DMA data transfer.
2. DMA*x* requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA*x* 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*x* reloads the original Start Address
   - If configured to generate an interrupt, DMA*x* sends an interrupt request to the Interrupt Controller
   - If configured for single-pass operation, DMA*x* resets the DEN bit in the DMA*x* 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).

0101 = ADC Analog Inputs 0-5 updated.
0110 = ADC Analog Inputs 0-6 updated.
0111 = ADC Analog Inputs 0-7 updated.
1000 = ADC Analog Inputs 0-8 updated.
1001 = ADC Analog Inputs 0-9 updated.
1010 = ADC Analog Inputs 0-10 updated.
1011 = ADC Analog Inputs 0-11 updated.
1100-1111 = Reserved.

## DMA Status Register

The DMA Status register (Table 85 on page 173) indicates the DMA channel that generated the interrupt and the ADC Analog Input that is currently undergoing conversion. Reads from this register reset the Interrupt Request Indicator bits (IRQA, IRQ1, and IRQ0) to 0. Therefore, software interrupt service routines that read this register must process all three interrupt sources from the DMA.

**Table 85. DMA_ADC Status Register (DMAA_STAT)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | CADC[3:0] | | | | Reserved | IRQA | IRQ1 | IRQ0 |
| RESET | 0 | | | | | | | |
| R/W | R | | | | | | | |
| ADDR | FBFH | | | | | | | |

CADC[3:0]—Current ADC Analog Input
This field identifies the Analog Input that the ADC is currently converting.

Reserved
This bit is reserved and must be 0.

IRQA—DMA_ADC Interrupt Request Indicator
This bit is automatically reset to 0 each time a read from this register occurs.
0 = DMA_ADC is not the source of the interrupt from the DMA Controller.
1 = DMA_ADC completed transfer of data from the last ADC Analog Input and generated an interrupt.

IRQ1—DMA1 Interrupt Request Indicator
This bit is automatically reset to 0 each time a read from this register occurs.
0 = DMA1 is not the source of the interrupt from the DMA Controller.
1 = DMA1 completed transfer of data to/from the End Address and generated an interrupt.

IRQ0—DMA0 Interrupt Request Indicator
This bit is automatically reset to 0 each time a read from this register occurs.

5. Re-write the page written in step 2 to the Page Select register.
6. Write the Page Erase command 95H to the Flash Control register.

### Mass Erase

The Flash memory cannot be Mass Erased by user code.

### Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Programming algorithms by controlling the Flash programming signals directly.

Flash Controller Bypass is recommended for gang programming applications and large volume customers who do not require in-circuit programming of the Flash memory.

For more information on bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore!* available for download at www.zilog.com.

### Flash Controller Behavior in Debug Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored.
- The Flash Sector Protect register is ignored for programming and erase operations.
- Programming operations are not limited to the page selected in the Page Select register.
- Bits in the Flash Sector Protect register can be written to one or zero.
- The second write of the Page Select register to unlock the Flash Controller is not necessary.
- The Page Select register can be written when the Flash Controller is unlocked.
- The Mass Erase command is enabled through the Flash Control register.

**Caution:** *For security reasons, Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the Flash controller must go through the unlock sequence again to select another page.*

## I²C Timing

Figure 55 and Table 119 provide timing information for I²C pins.

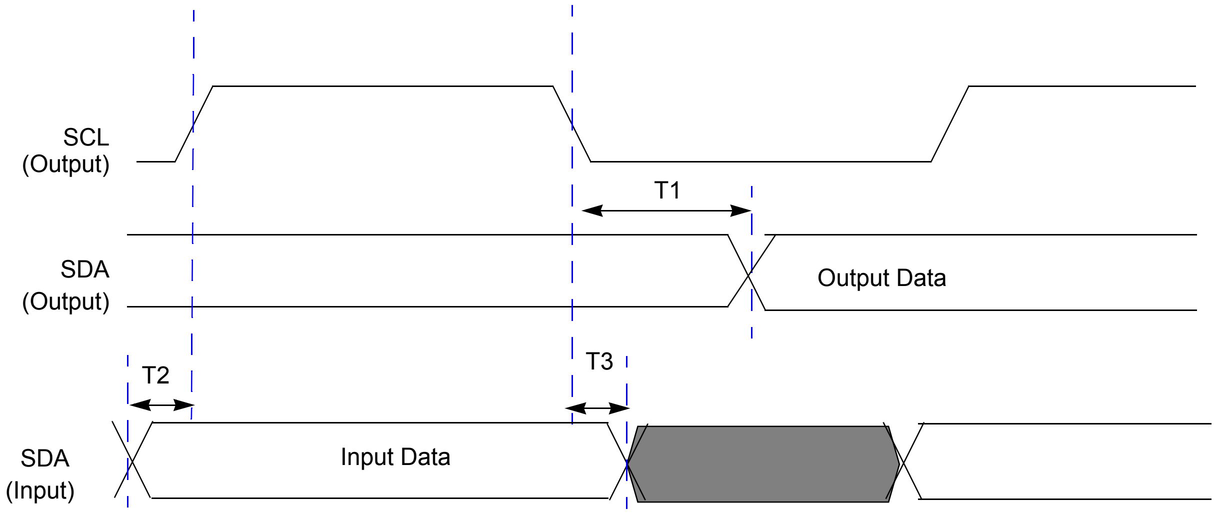


**Figure 55. I²C Timing**

**Table 119. I²C Timing**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| **I²C** | | | |
| $T_1$ | SCL Fall to SDA output delay | SCL period/4 | |
| $T_2$ | SDA Input to SCL rising edge Setup Time | 0 | |
| $T_3$ | SDA Input to SCL falling edge Hold Time | 0 | |

**Table 123. Additional Symbols**

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

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

dst ← dst + src

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

## 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 124. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the Flag test operation decides if the conditional jump is executed.

**Table 124. Condition Codes**

| 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 |

### Table 130. Logical Instructions (Continued)

| Mnemonic | Operands | Instruction |
|---|---|---|
| XOR | dst, src | Logical Exclusive OR |
| XORX | dst, src | Logical Exclusive OR using Extended Addressing |

### Table 131. Program Control Instructions

| Mnemonic | Operands | Instruction |
|---|---|---|
| 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 |

### Table 132. Rotate and Shift Instructions

| Mnemonic | Operands | Instruction |
|---|---|---|
| BSWAP | dst | Bit Swap |
| RL | dst | Rotate Left |
| RLC | dst | Rotate Left through Carry |
| RR | dst | Rotate Right |
| RRC | dst | Rotate Right through Carry |
| SRA | dst | Shift Right Arithmetic |

# Ordering Information

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | $I^2C$ | SPI | UARTs with IrDA | Description |
|---|---|---|---|---|---|---|---|---|---|---|
| **Z8F642x with 64 KB Flash, 10-Bit Analog-to-Digital Converter** | | | | | | | | | | |
| Standard Temperature: 0 °C to 70 °C | | | | | | | | | | |
| Z8F6421PM020SC | 64 KB | 4 KB | 29 | 23 | 3 | 8 | 1 | 1 | 2 | PDIP 40-pin package |
| Z8F6421AN020SC | 64 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | LQFP 44-pin package |
| Z8F6421VN020SC | 64 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | PLCC 44-pin package |
| Z8F6422AR020SC | 64 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | LQFP 64-pin package |
| Z8F6422VS020SC | 64 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | PLCC 68-pin package |
| Z8F6423FT020SC | 64 KB | 4 KB | 60 | 24 | 4 | 12 | 1 | 1 | 2 | QFP 80-pin package |
| Extended Temperature: –40 °C to +105 °C | | | | | | | | | | |
| Z8F6421PM020EC | 64 KB | 4 KB | 29 | 23 | 3 | 8 | 1 | 1 | 2 | PDIP 40-pin package |
| Z8F6421AN020EC | 64 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | LQFP 44-pin package |
| Z8F6421VN020EC | 64 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | PLCC 44-pin package |
| Z8F6422AR020EC | 64 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | LQFP 64-pin package |
| Z8F6422VS020EC | 64 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | PLCC 68-pin package |
| Z8F6423FT020EC | 64 KB | 4 KB | 60 | 24 | 4 | 12 | 1 | 1 | 2 | QFP 80-pin package |
| Automotive/Industrial Temperature: –40 °C to +125 °C | | | | | | | | | | |
| Z8F6421PM020AC | 64 KB | 4 KB | 29 | 23 | 3 | 8 | 1 | 1 | 2 | PDIP 40-pin package |
| Z8F6421AN020AC | 64 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | LQFP 44-pin package |
| Z8F6421VN020AC | 64 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | PLCC 44-pin package |
| Z8F6422AR020AC | 64 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | LQFP 64-pin package |
| Z8F6422VS020AC | 64 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | PLCC 68-pin package |
| Z8F6423FT020AC | 64 KB | 4 KB | 60 | 24 | 4 | 12 | 1 | 1 | 2 | QFP 80-pin package |

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | $I^2C$ | SPI | UARTs with IrDA | Description |
|---|---|---|---|---|---|---|---|---|---|---|
| **Z8F482x with 48 KB Flash, 10-Bit Analog-to-Digital Converter** | | | | | | | | | | |
| Standard Temperature: 0 °C to 70 °C | | | | | | | | | | |
| Z8F4821PM020SC | 48 KB | 4 KB | 29 | 23 | 3 | 8 | 1 | 1 | 2 | PDIP 40-pin package |
| Z8F4821AN020SC | 48 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | LQFP 44-pin package |
| Z8F4821VN020SC | 48 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | PLCC 44-pin package |
| Z8F4822AR020SC | 48 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | LQFP 64-pin package |
| Z8F4822VS020SC | 48 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | PLCC 68-pin package |
| Z8F4823FT020SC | 48 KB | 4 KB | 60 | 24 | 4 | 12 | 1 | 1 | 2 | QFP 80-pin package |
| Extended Temperature: –40 °C to +105 °C | | | | | | | | | | |
| Z8F4821PM020EC | 48 KB | 4 KB | 29 | 23 | 3 | 8 | 1 | 1 | 2 | PDIP 40-pin package |
| Z8F4821AN020EC | 48 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | LQFP 44-pin package |
| Z8F4821VN020EC | 48 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | PLCC 44-pin package |
| Z8F4822AR020EC | 48 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | LQFP 64-pin package |
| Z8F4822VS020EC | 48 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | PLCC 68-pin package |
| Z8F4823FT020EC | 48 KB | 4 KB | 60 | 24 | 4 | 12 | 1 | 1 | 2 | QFP 80-pin package |
| Automotive/Industrial Temperature: –40 °C to +125 °C | | | | | | | | | | |
| Z8F4821PM020AC | 48 KB | 4 KB | 29 | 23 | 3 | 8 | 1 | 1 | 2 | PDIP 40-pin package |
| Z8F4821AN020AC | 48 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | LQFP 44-pin package |
| Z8F4821VN020AC | 48 KB | 4 KB | 31 | 23 | 3 | 8 | 1 | 1 | 2 | PLCC 44-pin package |
| Z8F4822AR020AC | 48 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | LQFP 64-pin package |
| Z8F4822VS020AC | 48 KB | 4 KB | 46 | 24 | 4 | 12 | 1 | 1 | 2 | PLCC 68-pin package |
| Z8F4823FT020AC | 48 KB | 4 KB | 60 | 24 | 4 | 12 | 1 | 1 | 2 | QFP 80-pin package |