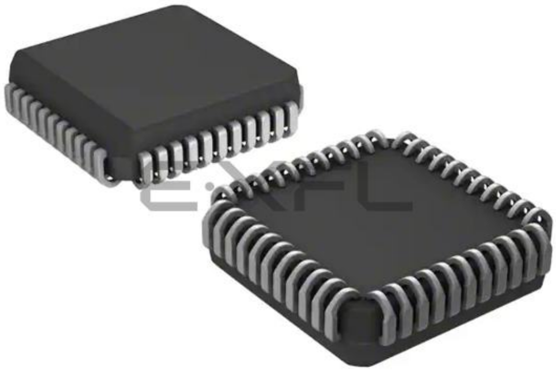
Zilog - Z8F6421VN020SC00TR Datasheet



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	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 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6421vn020sc00tr

Use of All Uppercase Letters

The use of all uppercase letters designates the names of states, modes, and commands.

- Example 1: The bus is considered BUSY after the Start condition.
- Example 2: A START command triggers the processing of the initialization sequence.
- Example 3: STOP mode.

Bit Numbering

Bits are numbered from *0* to *n–1* where *n* indicates the total number of bits. For example, the 8 bits of a register are numbered from 0 to 7.

Safeguards

It is important that you understand the following safety terms, which are defined here.

⚠ **Caution:** *Indicates a procedure or file may become corrupted if you do not follow directions.*

The eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required Program Memory
- Software stack allows much greater depth in subroutine calls and interrupts than hardware stacks
- Compatible with existing Z8 code
- Expanded internal Register File allows access of up to 4 KB
- New instructions improve execution efficiency for code developed using higher-level programming languages, including C
- Pipelined instruction fetch and execution
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT, and SRL
- New instructions support 12-bit linear addressing of the Register File
- Up to 10 MIPS operation
- C-Compiler friendly
- 2 to 9 clock cycles per instruction

For more information on the eZ8 CPU, refer to *eZ8™ CPU Core User Manual (UM0128)* available for download at www.zilog.com.

General-Purpose Input/Output

The 64K Series features seven 8-bit ports (Ports A-G) and one 4-bit port (Port H) for general-purpose input/output (GPIO). Each pin is individually programmable. All ports (except B and H) support 5 V-tolerant inputs.

Flash Controller

The Flash Controller programs and erases the Flash memory.

10-Bit Analog-to-Digital Converter

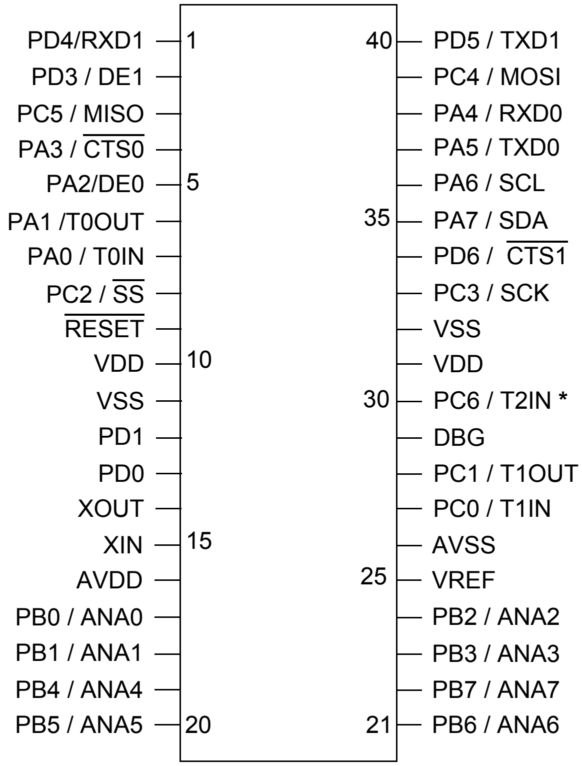
The Analog-to-Digital Converter converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from up to 12 different analog input sources.

UARTs

Each UART is full-duplex and capable of handling asynchronous data transfers. The UARTs support 8- and 9-bit data modes, selectable parity, and an efficient bus transceiver Driver Enable signal for controlling a multi-transceiver bus, such as RS-485.

Pin Configurations

Figure 2 through Figure 7 on page 13 display the pin configurations for all of the packages available in the Z8 Encore! XP 64K Series Flash Microcontrollers. For description of the signals, see Table 3 on page 14. Timer 3 is not available in the 40-pin and 44-pin packages.



Note: Timer 3 is not supported.

* T2OUT is not supported.

Figure 2. Z8 Encore! XP 64K Series Flash Microcontrollers in 40-Pin Dual Inline Package (PDIP)

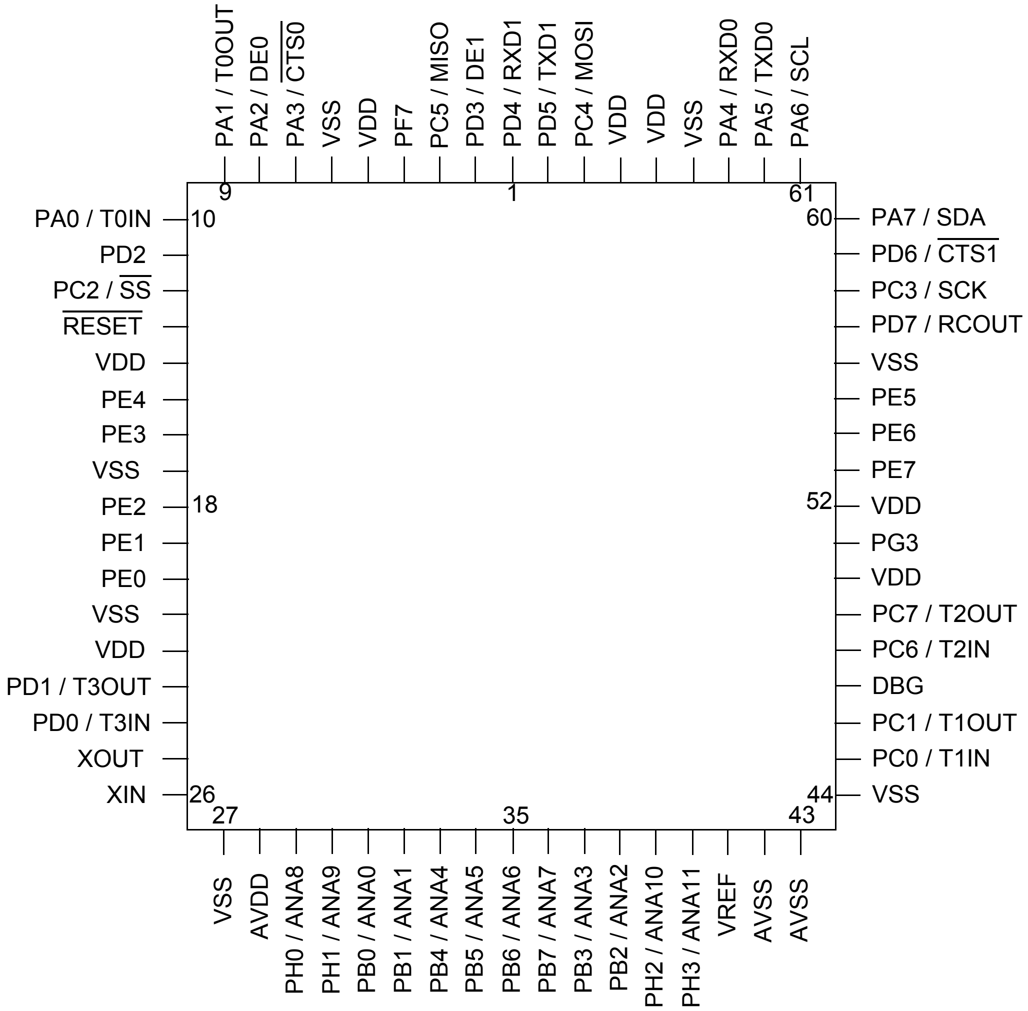


Figure 6. Z8 Encore! XP 64K Series Flash Microcontrollers in 68-Pin Plastic Leaded Chip Carrier (PLCC)

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

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FEF	Port H Output Data	PHOUT	00	66
Watchdog Timer				
FF0	Watchdog Timer Control	WDTCTL	XXX00000b	100
FF1	Watchdog Timer Reload Upper Byte	WDTU	FF	101
FF2	Watchdog Timer Reload High Byte	WDTH	FF	101
FF3	Watchdog Timer Reload Low Byte	WDTL	FF	101
FF4-FF7	Reserved	—	XX	
Flash Memory Controller				
FF8	Flash Control	FCTL	00	190
FF8	Flash Status	FSTAT	00	190
FF9	Page Select	FPS	00	191
FF9 (if enabled)	Flash Sector Protect	FPROT	00	192
FFA	Flash Programming Frequency High Byte	FFREQH	00	192
FFB	Flash Programming Frequency Low Byte	FFREQL	00	192
FF4-FF8	Reserved	—	XX	
Read-Only Memory Controller				
FF9	Page Select	RPS	00	
FFA-FFB	Reserved	—	XX	
eZ8 CPU				
FFC	Flags	—	XX	Refer to *eZ8™ CPU Core User Manual (UM0128)*
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	
Note: XX=Undefined				

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

General-Purpose I/O

Overview

The 64K Series products support a maximum of seven 8-bit ports (Ports A–G) and one 4-bit port (Port H) for general-purpose input/output (GPIO) operations. Each port consists of control and data registers. The GPIO control registers are used to determine data direction, open-drain, output drive current and alternate pin functions. Each port pin is individually programmable. All ports (except B and H) support 5 V-tolerant inputs.

GPIO Port Availability By Device

Table 11 lists the port pins available with each device and package type.

Table 11. Port Availability by Device and Package Type

Device	Packages	Port A	Port B	Port C	Port D	Port E	Port F	Port G	Port H
Z8X1621	40-pin	[7:0]	[7:0]	[6:0]	[6:3, 1:0]	-	-	-	-
Z8X1621	44-pin	[7:0]	[7:0]	[7:0]	[6:0]	-	-	-	-
Z8X1622	64- and 68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]
Z8X2421	40-pin	[7:0]	[7:0]	[6:0]	[6:3, 1:0]	-	-	-	-
Z8X2421	44-pin	[7:0]	[7:0]	[7:0]	[6:0]	-	-	-	-
Z8X2422	64- and 68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]
Z8X3221	40-pin	[7:0]	[7:0]	[6:0]	[6:3, 1:0]	-	-	-	-
Z8X3221	44-pin	[7:0]	[7:0]	[7:0]	[6:0]	-	-	-	-
Z8X3222	64- and 68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]
Z8X4821	40-pin	[7:0]	[7:0]	[6:0]	[6:3, 1:0]	-	-	-	-
Z8X4821	44-pin	[7:0]	[7:0]	[7:0]	[6:0]	-	-	-	-
Z8X4822	64- and 68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]

Port A–H Data Direction Sub-Registers

The Port A–H Data Direction sub-register is accessed through the Port A–H Control register by writing 01H to the Port A–H Address register (Table 16).

Table 16. Port A–H Data Direction Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1							
R/W	R/W							
ADDR	If 01H in Port A–H Address Register, accessible through Port A–H Control Register							

DD[7:0]—Data Direction
These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.
0 = Output. Data in the Port A–H Output Data register is driven onto the port pin.
1 = Input. The port pin is sampled and the value written into the Port A–H Input Data Register. The output driver is tri-stated.

Port A–H Alternate Function Sub-Registers

The Port A–H Alternate Function sub-register (Table 17) is accessed through the Port A–H Control register by writing 02H to the Port A–H Address register. The Port A–H Alternate Function sub-registers select the alternate functions for the selected pins. To determine the alternate function associated with each port pin, see GPIO Alternate Functions on page 59.

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

Table 17. Port A–H Alternate Function Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	0							
R/W	R/W							
ADDR	If 02H in Port A–H Address Register, accessible through Port A–H Control Register							

T1I—Timer 1 Interrupt Request
0 = No interrupt request is pending for Timer 1.
1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request
0 = No interrupt request is pending for Timer 0.
1 = An interrupt request from Timer 0 is awaiting service.

U0RXI—UART 0 Receiver Interrupt Request
0 = No interrupt request is pending for the UART 0 receiver.
1 = An interrupt request from the UART 0 receiver is awaiting service.

U0TXI—UART 0 Transmitter Interrupt Request
0 = No interrupt request is pending for the UART 0 transmitter.
1 = An interrupt request from the UART 0 transmitter is awaiting service.

I^2CI— I^2C Interrupt Request
0 = No interrupt request is pending for the I^2C.
1 = An interrupt request from the I^2C is awaiting service.

SPII—SPI Interrupt Request
0 = No interrupt request is pending for the SPI.
1 = An interrupt request from the SPI is awaiting service.

ADCI—ADC Interrupt Request
0 = No interrupt request is pending for the Analog-to-Digital Converter.
1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 25) 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 25. Interrupt Request 1 Register (IRQ1)

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

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)

- Master receives from a 7-bit slave
- Master receives from a 10-bit slave

SDA and SCL Signals

I^{2}C sends all addresses, data and acknowledge signals over the SDA line, most-significant bit first. SCL is the common clock for the I^{2}C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I^{2}C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a high level. When the slave releases the clock, the I^{2}C Controller continues the transaction. All data is transferred in bytes and there is no 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^{2}C Interrupts

The I^{2}C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The Transmit interrupt is enabled by the IEN and TXI bits of the Control register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I^{2}C Controller and neither the `START` or `STOP` bit is set. The Not Acknowledge event sets the NCKI bit of the I^{2}C Status register and can only be cleared by setting the `START` or `STOP` bit in the I^{2}C Control register. When this interrupt occurs, the I^{2}C Controller waits until either the STOP or START bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I^{2}C Controller (master reading data from slave). This procedure sets the RDRF bit of the I^{2}C Status register. The RDRF bit is cleared by reading the I^{2}C Data register. The RDRF bit is set during the acknowledge phase. The I^{2}C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

In order for a receive (read) DMA transaction to send a Not Acknowledge on the last byte, the receive DMA must be set up to receive n-1 bytes, then software must set the NAK bit and receive the last (nth) byte directly.

Start and Stop Conditions

The master (I^2C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I^2C Controller generates a START condition by pulling the SDA signal Low while SCL is High. To complete a transaction, the I^2C Controller generates a Stop condition by creating a low-to-high transition of the SDA signal while the SCL signal is high. The START and STOP bits in the I^2C Control register control the sending of the Start and Stop conditions. A master is also allowed to end one transaction and begin a new one by issuing a Restart. This is accomplished by setting the START bit at the end of a transaction, rather than the STOP bit. Note that the Start condition not sent until the START bit is set and data has been written to the I^2C Data register.

Master Write and Read Transactions

The following sections provide a recommended procedure for performing I^2C write and read transactions from the I^2C Controller (master) to slave I^2C devices. In general software should rely on the TDRE, RDRF and NCKI bits of the status register (these bits generate interrupts) to initiate software actions. When using interrupts or DMA, the TXI bit is set to start each transaction and cleared at the end of each transaction to eliminate a 'trailing' Transmit interrupt.

Caution should be used in using the ACK status bit within a transaction because it is difficult for software to tell when it is updated by hardware.

When writing data to a slave, the I^2C pauses at the beginning of the Acknowledge cycle if the data register has not been written with the next value to be sent (TDRE bit in the I^2C Status register = 1). In this scenario where software is not keeping up with the I^2C bus (TDRE asserted longer than one byte time), the Acknowledge clock cycle for byte n is delayed until the Data register is written with byte n + 1, and appears to be grouped with the data clock cycles for byte n+1. If either the START or STOP bit is set, the I^2C does not pause prior to the Acknowledge cycle because no additional data is sent.

When a Not Acknowledge condition is received during a write (either during the address or data phases), the I^2C Controller generates the Not Acknowledge interrupt (NCKI = 1) and pause until either the STOP or START bit is set. Unless the Not Acknowledge was received on the last byte, the Data register will already have been written with the next address or data byte to send. In this case the FLUSH bit of the Control register should be set at the same time the STOP or START bit is set to remove the stale transmit data and enable subsequent Transmit interrupts.

When reading data from the slave, the I^2C pauses after the data Acknowledge cycle until the receive interrupt is serviced and the RDRF bit of the status register is cleared by

Table 70. I^{2}C Data Register (I2CDATA)

BITS	7	6	5	4	3	2	1	0
FIELD	DATA							
RESET	0							
R/W	R/W							
ADDR	F50H							

I^{2}C Status Register

The Read-only I^{2}C Status register (Table 71) indicates the status of the I^{2}C Controller.

Table 71. I^{2}C Status Register (I2CSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	TDRE	RDRF	ACK	10B	RD	TAS	DSS	NCKI
RESET	1	0						
R/W	R							
ADDR	F51H							

TDRE—Transmit Data Register Empty
When the I^{2}C Controller is enabled, this bit is 1 when the I^{2}C Data register is empty. When this bit is set, an interrupt is generated if the TXI bit is set, except when the I^{2}C Controller is shifting in data during the reception of a byte or when shifting an address and the RD bit is set. This bit is cleared by writing to the I2CDATA register.

RDRF—Receive Data Register Full
This bit is set = 1 when the I^{2}C Controller is enabled and the I^{2}C Controller has received a byte of data. When asserted, this bit causes the I^{2}C Controller to generate an interrupt. This bit is cleared by reading the I^{2}C Data register (unless the read is performed using execution of the On-Chip Debugger's Read Register command).

ACK—Acknowledge
This bit indicates the status of the Acknowledge for the last byte transmitted or received. When set, this bit indicates that an Acknowledge occurred for the last byte transmitted or received. This bit is cleared when IEN = 0 or when a Not Acknowledge occurred for the last byte transmitted or received. It is not reset at the beginning of each transaction and is not reset when this register is read.

Flash Memory

Overview

The products in the Z8 Encore! XP 64K Series Flash Microcontrollers feature up to 64 KB (65,536 bytes) of non-volatile Flash memory with read/write/erase capability. The Flash memory can be programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash memory array is arranged in 512-byte per page. The 512-byte page is the minimum Flash block size that can be erased. The Flash memory is also divided into 8 sectors which can be protected from programming and erase operations on a per sector basis.

Table 89 describes the Flash memory configuration for each device in the 64K Series. Table 90 on page 184 lists the sector address ranges. Figure 35 on page 184 displays the Flash memory arrangement.

Table 89. Flash Memory Configurations

Part Number	Flash Size	Number of Pages	Flash Memory Addresses	Sector Size	Number of Sectors	Pages per Sector
Z8F162x	16K (16,384)	32	0000H - 3FFFH	2K (2048)	8	4
Z8F242x	24K (24,576)	48	0000H - 5FFFH	4K (4096)	6	8
Z8F322x	32K (32,768)	64	0000H - 7FFFH	4K (4096)	8	8
Z8F482x	48K (49,152)	96	0000H - BFFFH	8K (8192)	6	16
Z8F642x	64K (65,536)	128	0000H - FFFFH	8K (8192)	8	16

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.

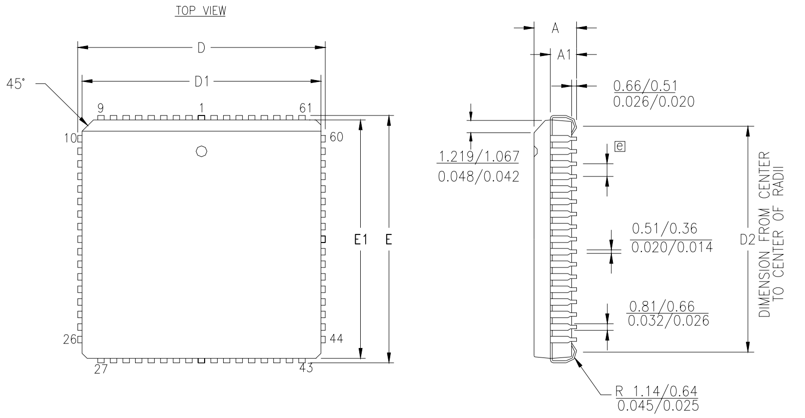
Caution: *For security reasons, Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the Flash controller must go through the unlock sequence again to select another page.*

Table 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
XOR dst, src	dst ← dst XOR src	r	r	B2	-	*	*	0	-	-	2	3
		r	Ir	B3							2	4
		R	R	B4							3	3
		R	IR	B5							3	4
		R	IM	B6							3	3
		IR	IM	B7							3	4
XORX dst, src	dst ← dst XOR src	ER	ER	B8	-	*	*	0	-	-	4	3
		ER	IM	B9							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

Figure 66 displays the 68-pin Plastic Lead Chip Carrier (PLCC) package available for the Z8X1622, Z8X2422, Z8X3222, Z8X4822, and Z8X6422 devices.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.32	4.57	.170	.180
A1	2.43	2.92	.095	.115
D/E	25.02	25.40	.985	1.000
D1/E1	24.13	24.33	.950	.958
D2	22.86	23.62	.900	.930
e	1.27 BSC		.050 BSC	

NOTE:
1. CONTROLLING DIMENSIONS : INCH.
2. LEADS ARE COPLANAR WITHIN 0.004 IN. RANGE.
3. DIMENSION : $\frac{\text{MM}}{\text{INCH.}}$

Figure 66. 68-Lead Plastic Lead Chip Carrier Package (PLCC)

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