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	Active
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0811hh020eg

Pin Characteristics

Table 4 provides detailed information about the characteristics for each pin available on Z8 Encore! XP® F0822 Series products. The data in Table 4 is sorted alphabetically by pin symbol mnemonic.

Table 4. Pin Characteristics

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tri-State Output	Internal Pull-Up or Pull-Down	Schmitt-Trigger Input	Open Drain Output
AV_{DD}	N/A	N/A	N/A	N/A	No	No	N/A
AV_{SS}	N/A	N/A	N/A	N/A	No	No	N/A
DBG	I/O	I	N/A	Yes	No	Yes	Yes
PA[7:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable
PB[4:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable
PC[5:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable
RESET	I	I	Low	N/A	Pull-up	Yes	N/A
V_{DD}	N/A	N/A	N/A	N/A	No	No	N/A
V_{REF}	Analog	N/A	N/A	N/A	No	No	N/A
V_{SS}	N/A	N/A	N/A	N/A	No	No	N/A
X_{IN}	I	I	N/A	N/A	No	No	N/A
X_{OUT}	O	O	N/A	No	No	No	No

Stop Mode Recovery Using WDT Time-Out

If the WDT times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the WDT Control Register, the WDT and stop bits are set to 1. If the WDT is configured to generate an interrupt upon time-out and the Z8 Encore! XP® F0822 Series device is configured to respond to interrupts, the eZ8 CPU services the WDT interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a STOP Mode Recover source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. The GPIO Stop Mode Recovery signals are filtered to reject pulses less than 10 ns (typical) in duration. In the WDT Control Register, the stop bit is set to 1.

Caution: In STOP Mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the port pin through the end of the Stop Mode Recovery delay. Therefore, short pulses on the port pin initiates Stop Mode Recovery without being written to the Port Input Data Register or without initiating an interrupt (if enabled for that pin).

Table 24. Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt Source
Highest	0002H	Reset (not an interrupt)
	0004H	WDT (see the Watchdog Timer chapter on page 70)
	0006H	Illegal Instruction Trap (not an interrupt)
	0008H	Reserved
	000AH	Timer 1
	000CH	Timer 0
	000EH	UART 0 receiver
	0010H	UART 0 transmitter
	0012H	I^2C
	0014H	SPI
	0016H	ADC
	0018H	Port A7, rising or falling input edge
	001AH	Port A6, rising or falling input edge
	001CH	Port A5, rising or falling input edge
	001EH	Port A4, rising or falling input edge
	0020H	Port A3, rising or falling input edge
	0022H	Port A2, rising or falling input edge
	0024H	Port A1, rising or falling input edge
	0026H	Port A0, rising or falling input edge
	0028H	Reserved
	002AH	Reserved
	002CH	Reserved
	002EH	Reserved
	0030H	Port C3, both input edges
	0032H	Port C2, both input edges
	0034H	Port C1, both input edges
Lowest	0036H	Port C0, both input edges

Architecture

Figure 9 displays a block diagram of the interrupt controller.

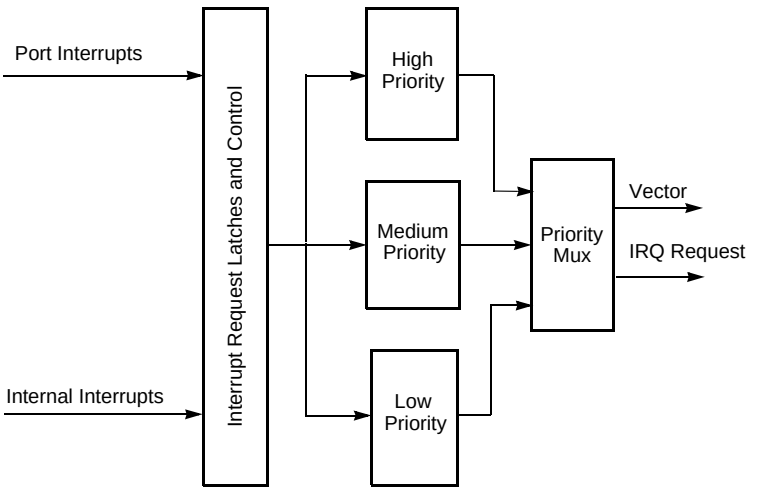

Figure 9. Interrupt Controller Block Diagram

Operation

This section describes the operational aspects of the following functions.

Master Interrupt Enable: see page 42

Interrupt Vectors and Priority: see page 43

Interrupt Assertion: see page 43

Software Interrupt Assertion: see page 44

Master Interrupt Enable

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

Interrupts are globally enabled by any of the following actions:

- Execution of an Enable Interrupt (EI) instruction
- Execution of an Return from Interrupt (IRET) instruction

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT Mode equation is used to determine the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is calculated using the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is calculated using the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

CAPTURE Mode

In CAPTURE Mode, the current timer count value is recorded when the appropriate external Timer Input transition occurs. The capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the capture event occurs, an interrupt is generated and the timer continues counting.

The timer continues counting up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt and continues counting.

Observe the following procedure for configuring a timer for CAPTURE Mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE Mode
   - Set the prescale value
   - Set the Capture edge (rising or falling) for the Timer Input
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. This allows user software to determine if interrupts were generated by either a capture event or a reload. If

the PWM High and Low Byte registers still contains 0000H after the interrupt, then the interrupt was generated by a reload.

5. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to capture event is calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

COMPARE Mode

In COMPARE Mode, the timer counts up to the 16-bit maximum Compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Compare value, the timer generates an interrupt and counting continues (the timer value is not reset to 0001H). 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 Compare.

If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting.

Observe the following procedure for configuring a timer for COMPARE Mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARE Mode
   - Set the prescale value
   - Set the initial logic level (High or Low) for the Timer Output alternate function, if required
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 Compare value.
4. If required, 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 Register to enable the timer and initiate counting.

UART Status 0 Register

The UART Status 0 and Status 1 registers, shown in Tables 55 and 56, identify the current UART operating configuration and status.

Table 55. UART Status 0 Register (U0STAT0)

Bit	7	6	5	4	3	2	1	0
Field	RDA	PE	OE	FE	BRKD	TDRE	TXE	CTS
RESET	0					1		X
R/W	R							
Address	F41H							

Bit	Description
[7] RDA	**Receive Data Available** This bit indicates that the UART Receive Data Register has received data. Reading the UART Receive Data Register clears this bit. 0 = The UART Receive Data Register is empty. 1 = There is a byte in the UART Receive Data Register.
[6] PE	**Parity Error** This bit indicates that a parity error has occurred. Reading the UART Receive Data Register clears this bit. 0 = No parity error has occurred. 1 = A parity error has occurred.
[5] 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.
[4] 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.
[3] 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.
[2] TDRE	

clock periods since the previous pulse was detected). This period allows the endec a sampling window of –4 to +8 baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window, this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to its initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset to resynchronize the endec to the incoming signal. This routine allows the endec to tolerate jitter and baud rate errors in the incoming data stream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a start bit is received.

Infrared Endec Control Register Definitions

All infrared endec configuration and status information is set by the UART control registers as defined in the UART Control Register Definitions section on page 88.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the infrared endec before enabling the GPIO port alternate function for the corresponding pin.

Bit	Description (Continued)
[1] FLUSH	**Flush Data** Setting this bit to 1 clears the I2C Data Register and sets the TDRE bit to 1. This bit allows flushing of the I2C Data Register when a Not Acknowledge interrupt is received after the data has been sent to the I2C Data Register. Reading this bit always returns 0.
[0] FILTEN	**I2C Signal Filter Enable** This bit enables low-pass digital filters on the SDA and SCL input signals. These filters reject any input pulse with periods less than a full system clock cycle. The filters introduce a 3-system clock cycle latency on the inputs. 1 = Low-pass filters are enabled. 0 = Low-pass filters are disabled.

I2C Baud Rate High and Low Byte Registers

The I2C Baud Rate High and Low Byte registers, shown in Tables 74 and 75, combine to form a 16-bit reload value, BRG[15:0], for the I2C Baud Rate Generator. When configured as a general purpose timer, the interrupt interval is calculated using the following equation:

Interrupt Interval (s) = System Clock Period (s) ×BRG[15:0]

Table 74. I2C Baud Rate High Byte Register (I2CBRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	FFH							
R/W	R/W							
Address	F53H							

Bit	Description
[7:0] BRH	**I2C Baud Rate High Byte** Most significant byte, BRG[15:8], of the I2C Baud Rate Generator's reload value. **Note:** If the DIAG bit in the I2C Diagnostic Control Register is set to 1, a read of the I2CBRH Register returns the current value of the I2C Baud Rate Counter[15:8].

ADC Data High Byte Register

The ADC Data High Byte Register, shown in Table 79, contains the upper eight bits of the 10-bit ADC output. During a SINGLE-SHOT conversion, this value is invalid. Access to the ADC Data High Byte Register is read-only. The full 10-bit ADC result is furnished by {ADCD_H[7:0], ADCD_L[7:6]}. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

Table 79. ADC Data High Byte Register (ADCD_H)

Bit	7	6	5	4	3	2	1	0
Field	ADCD_H							
RESET	X							
R/W	R							
Address	F72H							

Bit	Description
[7:0] ADCD_H	**ADC Data High Byte** This byte contains the upper eight bits of the 10-bit ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the last conversion output is held in this register. These bits are undefined after a Reset.

Flash Memory

The products in the Z8 Encore! XP® F0822 Series feature either 8KB (8192) or 4KB (4096) bytes of Flash memory with Read/Write/Erase capability. Flash memory is programmed and erased in-circuit by either user code or through the OCD.

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. Flash memory is divided into eight sectors which is protected from programming and erase operations on a per sector basis.

Table 81 describes the Flash memory configuration for each device in the Z8F082xfamily. Table 82 lists the sector address ranges. Figure 33 displays the Flash memory arrangement.

Table 81. Flash Memory Configurations

Part Number	Flash Size	Number of Pages	Flash Memory Addresses	Sector Size	Number of Sectors	Pages per Sector
Z8F08xx	8KB (8192)	16	0000H–1FFFH	1KB (1024)	8	2
Z8F04xx	4KB (4096)	8	0000H–0FFFH	0.5KB (512)	8	1

Table 82. Flash Memory Sector Addresses

	Flash Sector Address Ranges	
Sector Number	Z8F04xx	Z8F08xx
0	0000H–01FFH	0000H–03FFH
1	0200H–03FFH	0400H–07FFH
2	0400H–05FFH	0800H–0BFFH
3	0600H–07FFH	0C00H–0FFFH
4	0800H–09FFH	1000H–13FFH
5	0A00H–0BFFH	1400H–17FFH
6	0C00H–0DFFH	1800H–1BFFH
7	0E00H–0FFFH	1C00H–1FFFH

For more information about bypassing the Flash Controller, refer to the Third Party Flash Programming Support for Z8 Encore! MCU Application Note (AN0117), 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 OCD:

- 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 1 or 0
- The second write of the Page Select Register to unlock the Flash Controller is not necessary
- The Page Select Register is written when the Flash Controller is unlocked
- The Mass Erase command is enabled

Flash Control Register Definitions

This section defines the features of the following Flash Control registers.

Flash Control Register: see page 150

Flash Status Register: see page 151

Page Select Register: see page 152

Flash Sector Protect Register: see page 152

Flash Frequency High and Low Byte Registers: see page 153

On-Chip Peripheral AC and DC Electrical Characteristics

Table 100 provides information about the Power-On Reset and Voltage Brown-Out electrical characteristics.

Table 100. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = –40°C to 105°C				
Symbol	**Parameter**	**Minimum**	**Typical***	**Maximum**	**Units**	**Conditions**
V_{POR}	Power-On Reset Voltage Threshold	2.15	2.40	2.60	V	$V_{DD} = V_{POR}$
V_{VBO}	Voltage Brown-Out Reset Voltage Threshold	2.05	2.30	2.55	V	$V_{DD} = V_{VBO}$
	V_{POR} to V_{VBO} hysteresis	50	100	–	mV	
	Starting V_{DD} voltage to ensure valid POR	–	V_{SS}	–	V	
T_{ANA}	POR Analog Delay	–	50	–	µs	$V_{DD} > V_{POR}$; T_{POR} Digital Reset delay follows T_{ANA}
T_{POR}	POR Digital Delay	–	5.0	–	ms	50 WDT Oscillator cycles (10kHz) + 16 System Clock cycles (20MHz)
T_{VBO}	Voltage Brown-Out Pulse Rejection Period	–	10	–	µs	$V_{DD} < V_{VBO}$ to generate a Reset.
T_{RAMP}	Time for VDD to transition from V_{SS} to V_{POR} to ensure valid Reset	0.10	–	100	ms	

Note: *Data in the typical column is from characterization at 3.3 V and 25°C. These values are provided for design guidance only and are not tested in production.

General Purpose I/O Port Output Timing

Figure 49 and Table 107 provide timing information for GPIO port pins.

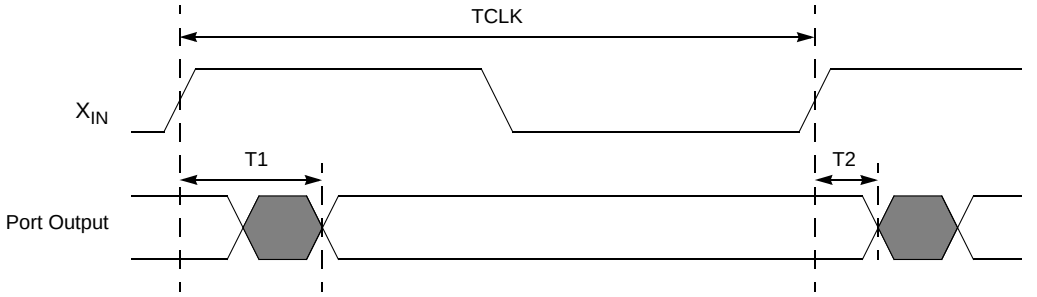

Figure 49. GPIO Port Output Timing

Table 107. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
GPIO Port Pins			
T_1	X_{IN} Rise to Port Output Valid Delay	–	15
T_2	X_{IN} Rise to Port Output Hold Time	2	–

Assembly Language Source Program Example

```
JP START      ; Everything after the semicolon is a comment.
START:        ; A label called “START”. The first instruction (JP
              ; START) in this example causes program execution to
              ; jump to the point within the program where the
              ; START label occurs.
LD R4, R7     ; A Load (LD) instruction with two operands. The
              ; first operand, Working Register R4, is the
              ; destination. The second operand, Working Register
              ; R7, is the source. The contents of R7 is written
              ; into R4.
LD 234H, 01H  ; Another Load (LD) instruction with two operands.
              ; The first operand, Extended Mode Register Address
              ; 234H, is the destination. The second operand,
              ; Immediate Data value 01H, is the source. The value
              ; 01H is written into the Register at address 234H.
```

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as ‘destination, source’. After assembly, the object code usually has the operands in the order ’source, destination’, but ordering is op code-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed by users that prefer manual program coding or intend to implement their own assembler.

Example 1. If the contents of registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code result is shown in Table 114.

Table 114. Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2. In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0–R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code result is shown in Table 115.

Hex Address: F01

Table 131. Timer 0–1 Low Byte Register (TxL)

Bit	7	6	5	4	3	2	1	0
Field	TL							
RESET	0							1
R/W	R/W							
Address	F01H, F09H							

Hex Address: F02

Table 132. Timer 0–1 Reload High Byte Register (TxRH)

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1							
R/W	R/W							
Address	F02H, F0AH							

Hex Address: F03

Table 133. Timer 0–1 Reload Low Byte Register (TxRL)

Bit	7	6	5	4	3	2	1	0
Field	TRL							
RESET	1							
R/W	R/W							
Address	F03H, F0BH							

Hex Address: F45

Table 152. UART Address Compare Register (U0ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
RESET	0							
R/W	R/W							
Address	F45H							

Hex Address: F46

Table 153. UART Baud Rate High Byte Register (U0BRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1							
R/W	R/W							
Address	F46H							

Hex Address: F47

Table 154. UART Baud Rate Low Byte Register (U0BRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Address	F47H							

Hex Addresses: F48–F4F

This address range is reserved.

Hex Address: FC1

Table 173. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	I2CENH	SPIENH	ADCENH
RESET	0							
R/W	R/W							
Address	FC1H							

Hex Address: FC2

Table 174. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENL	T0ENL	U0RENL	U0TENL	I2CENL	SPIENL	ADCENL
RESET	0							
R/W	R/W							
Address	FC2H							

Hex Address: FC3

Table 175. Interrupt Request 1 Register (IRQ1)

Bit	7	6	5	4	3	2	1	0
Field	PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
RESET	0							
R/W	R/W							
Address	FC3H							

Hex Address: FC4

Table 176. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENH	PA6ENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
RESET	0							
R/W	R/W							
Address	FC4H							

U

V

W

Customer Support

To share comments, get your technical questions answered or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.