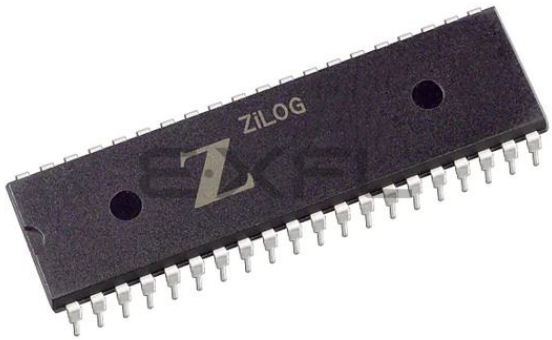

Welcome to E-XFL.COM

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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	29
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	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.620", 15.75mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6421pm020eg

Revision History

Each instance in the Revision History table reflects a change to this document from its previous revision. For more details, refer to the corresponding pages or appropriate links listed in the table below.

Date	Revision Level	Description	Page
Jan 2013	24	Restored 40-pin PDIP package to Signal and Pin Descriptions and Packaging chapters.	7, 286
Feb 2012	23	Corrected formatting of I_{DDS} section, Table 107; corrected language in the General Purpose RAM section of Appendix A;	202, 248
Sep 2011	22	Revised Flash Sector Protect Register description; revised Packaging chapter.	178, 286
Mar 2008	21	Changed title to Z8 Encore! XP F64xx Series.	All
Feb 2008	20	Changed Z8 Encore! XP 64K Series Flash Microcontrollers to Z8 Encore! XP F64xx Series Flash Microcontrollers. Deleted three sentences that mentioned Z8R642. Removed the 40 PDIP package. Added ZENETSC0100ZACG to the end of the Ordering Information table. Changed the flag status to unaffected for BIT, BSET, and BCLR in the eZ8 CPU Instruction Summary table.	287, 234
Dec 2007	19	Updated Zilog logo, Disclaimer section, and implemented style guide. Updated Table 113. Changed Z8 Encore! 64K Series to Z8 Encore! XP 64K Series Flash Microcontrollers throughout the document.	All
Dec 2006	18	Updated Flash Memory Electrical Characteristics and Timing table and Ordering Information chapter.	213, 287
Nov 2006	17	Updated Part Number Suffix Designations section.	292

List of Figures

- 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 4KB
- 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 about the eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com.

General-Purpose Input/Output

The Z8 Encore! XP F64xx 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 5V-tolerant inputs.

Flash Controller

The Flash Controller programs and erases the contents of 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 multitransceiver bus, such as RS-485.

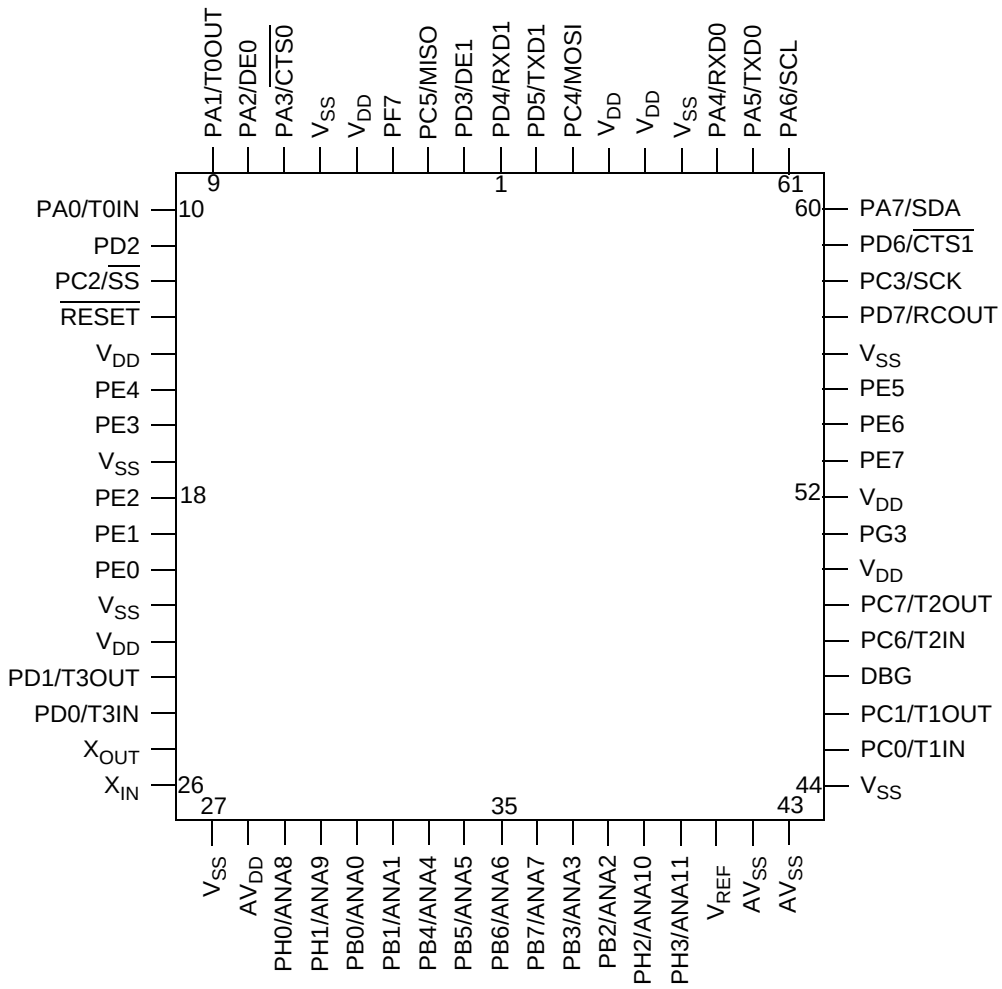

Figure 6. Z8 Encore! XP F64xx Series in 68-Pin Plastic Leaded Chip Carrier (PLCC)

Table 7. Z8 Encore! XP F64xx Series Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
DMA 0 (continued)				
FB2	DMA0 End/Start Address High Nibble	DMA0H	XX	155
FB3	DMA0 Start Address Low Byte	DMA0START	XX	156
FB4	DMA0 End Address Low Byte	DMA0END	XX	156
DMA 1				
FB8	DMA1 Control	DMA1CTL	00	153
FB9	DMA1 I/O Address	DMA1IO	XX	154
FBA	DMA1 End/Start Address High Nibble	DMA1H	XX	155
FBB	DMA1 Start Address Low Byte	DMA1START	XX	156
FBC	DMA1 End Address Low Byte	DMA1END	XX	156
DMA ADC				
FBD	DMA_ADC Address	DMAA_ADDR	XX	157
FBE	DMA_ADC Control	DMAACTL	00	158
FBF	DMA_ADC Status	DMAASTAT	00	159
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	51
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	55
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	55
FC3	Interrupt Request 1	IRQ1	00	53
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	56
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	56
FC6	Interrupt Request 2	IRQ2	00	54
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	58
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	58
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	60
FCE	Interrupt Port Select	IRQPS	00	60
FCF	Interrupt Control	IRQCTL	00	61
GPIO Port A				
FD0	Port A Address	PAADDR	00	40
FD1	Port A Control	PACTL	00	41
FD2	Port A Input Data	PAIN	XX	46

Note: XX = Undefined.

Table 7. Z8 Encore! XP F64xx Series Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
GPIO Port A (continued)				
FD3	Port A Output Data	PAOUT	00	46
GPIO Port B				
FD4	Port B Address	PBADDR	00	40
FD5	Port B Control	PBCTL	00	41
FD6	Port B Input Data	PBIN	XX	46
FD7	Port B Output Data	PBOUT	00	46
GPIO Port C				
FD8	Port C Address	PCADDR	00	40
FD9	Port C Control	PCCTL	00	41
FDA	Port C Input Data	PCIN	XX	46
FDB	Port C Output Data	PCOUT	00	46
GPIO Port D				
FDC	Port D Address	PDADDR	00	40
FDD	Port D Control	PDCTL	00	41
FDE	Port D Input Data	PDIN	XX	46
FDF	Port D Output Data	PDOUT	00	46
GPIO Port E				
FE0	Port E Address	PEADDR	00	40
FE1	Port E Control	PECTL	00	41
FE2	Port E Input Data	PEIN	XX	46
FE3	Port E Output Data	PEOUT	00	46
GPIO Port F				
FE4	Port F Address	PFADDR	00	40
FE5	Port F Control	PFCTL	00	41
FE6	Port F Input Data	PFIN	XX	46
FE7	Port F Output Data	PFOUT	00	46
GPIO Port G				
FE8	Port G Address	PGADDR	00	40
FE9	Port G Control	PGCTL	00	41
FEA	Port G Input Data	PGIN	XX	46
FEB	Port G Output Data	PGOUT	00	46

Note: XX = Undefined.

Architecture

Figure 10 displays a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength are not illustrated.

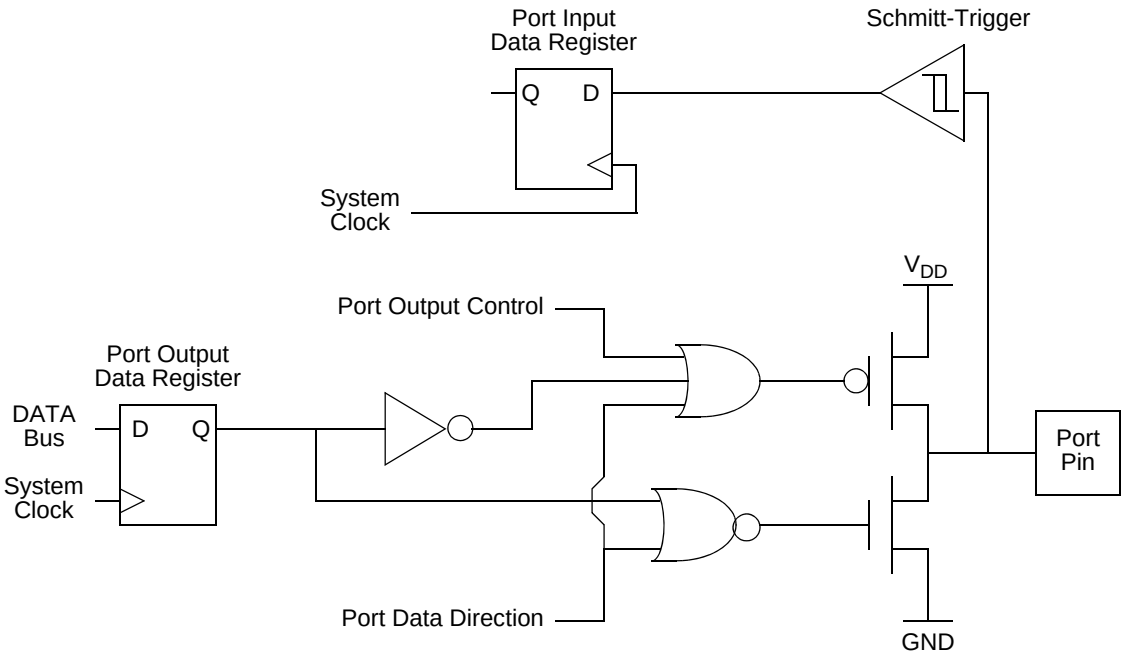

Figure 10. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins can be used as both general-purpose I/O and to provide access to on-chip peripheral functions such as the timers and serial communication devices. The Port A–H Alternate Function subregisters configure these pins for either general-purpose I/O or alternate function operation. When a pin is configured for alternate function, control of the port pin direction (input/output) is passed from the Port A–H Data Direction registers to the alternate function assigned to this pin. Table 12 lists the alternate functions associated with each port pin.

Every subsequent appropriate transition (after the first) of the timer input signal captures the current count value. The Capture value is written to the Timer PWM High and Low Byte Registers. When the capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H, and counting resumes.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

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

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for CAPTURE/COMPARE 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 compare value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin for the timer input alternate function.
6. Write to the Timer Control 1 Register to enable the timer.
7. Counting begins on the first appropriate transition of the timer input signal. No interrupt is generated by this first edge.

In COMPARE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

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

Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte Register is read, the contents of the Timer Low Byte Register are placed in a holding register. A subsequent read from the Timer Low Byte Register returns the value in the holding reg-

Table 64. SPI Data Register (SPIDATA)

Bit	7	6	5	4	3	2	1	0
Field	DATA							
RESET	X							
R/W	R/W							
Address	F60H							

Bit	Description
[7:0] DATA	**Data** Transmit and/or receive data.

SPI Control Register

The SPI Control Register, shown in Table 65, configures the SPI for transmit and receive operations.

Table 65. SPI Control Register (SPICTL)

Bit	7	6	5	4	3	2	1	0
Field	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN
RESET	0							
R/W	R/W							
Address	F61H							

Bit	Description
[7] 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.
[6] 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.
[5] 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.

Table 71. I²C Data Register (I2CDATA)

Bit	7	6	5	4	3	2	1	0
Field	DATA							
RESET	0							
R/W	R/W							
Address	F50H							

I²C Status Register

The read-only I²C Status Register, shown in Table 72, indicates the status of the I²C Controller.

Table 72. I²C Status Register (I2CSTAT)

Bit	7	6	5	4	3	2	1	0
Field	TDRE	RDRF	ACK	10B	RD	TAS	DSS	NCKI
RESET	1	0						
R/W	R							
Address	F51H							

Bit	Description
[7] TDRE	**Transmit Data Register Empty** When the I²C Controller is enabled, this bit is 1 when the I²C Data Register is empty. When this bit is set, an interrupt is generated if the TXI bit is set, except when the I²C 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.
[6] RDRF	**Receive Data Register Full** This bit is set = 1 when the I²C Controller is enabled and the I²C Controller has received a byte of data. When asserted, this bit causes the I²C Controller to generate an interrupt. This bit is cleared by reading the I²C Data Register (unless the read is performed using execution of the On-Chip Debugger's Read Register command).

Flash Read Protection

The user code contained within Flash memory can be protected from external access. Programming the Flash Read Protect option bit prevents reading of user code by the On-Chip Debugger or by using the Flash Controller Bypass mode. For more information, see the Option Bits chapter on page 180 and the On-Chip Debugger chapter on page 183.

Flash Write/Erase Protection

The Z8 Encore! XP F64xx Series provides several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by the Flash Controller unlock mechanism, the Flash Sector Protect Register, and the Flash Write Protect option bit.

Flash Controller Unlock Mechanism

At Reset, the Flash Controller locks to prevent accidental program or erasure of Flash memory. To program or erase Flash memory, the Flash Controller must be unlocked. After unlocking the Flash Controller, the Flash can be programmed or erased. Any value written by user code to the Flash Control Register or Page Select Register out of sequence will lock the Flash Controller.

Observe the following procedure to unlock the Flash Controller from user code:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write the page to be programmed or erased to the Page Select Register.
3. Write the first unlock command 73H to the Flash Control Register.
4. Write the second unlock command 8CH to the Flash Control Register.
5. Rewrite the page written in Step 2 to the Page Select Register.

Flash Sector Protection

The Flash Sector Protect Register can be configured to prevent sectors from being programmed or erased. After a sector is protected, it cannot be unprotected by user code. The Flash Sector Protect Register is cleared after reset and any previously written protection values is lost. User code must write this register in their initialization routine if they want to enable sector protection.

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Register with 5EH. After the Flash Sector Protect Register is selected, it can be accessed at the Page Select Register address. When user code writes the Flash Sector Protect Register, bits can only be set to 1. Thus, sectors can be protected, but not unprotected, via register write operations. Writing a value other than 5EH to the Flash Control Register deselects the Flash Sector Protect Register and reenables access to the Page Select Register.

Bit	Description (Continued)
[5] RPEN	**Read Protect Option Bit Enabled** 0 = The Read Protect option bit is disabled (1). 1 = The Read Protect option bit is enabled (0), disabling many OCD commands.
[4:0]	**Reserved** These bits are reserved and must be programmed to 00000.

Table 136. 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	4
		R	IM	E6							3	2
		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 r ← r + 1 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 r ← r + 1 rr ← rr + 1	Ir	Irr	83	–	–	–	–	–	–	2	9
		Irr	Ir	93							2	9
LDWX dst, src	dst ← src	ER	ER	1F E8	–	–	–	–	–	–	5	4

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

Hex Address: F09

Table 147. Timer 0–3 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, F11H, F19H							

Hex Address: F0A

Table 148. Timer 0–3 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, F12H, F1AH							

Hex Address: F0B

Table 149. Timer 0–3 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, F13H, F1BH							

Hex Address: F0C

Table 150. Timer 0–3 PWM High Byte Register (TxPWMH)

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	0							
R/W	R/W							
Address	F04H, F0CH, F14H, F1CH							

Hex Address: FE7

Table 252. Port A–H Output Data Register (P*x*OUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
Address	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

Hex Address: FE8

Table 253. Port A–H GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

Hex Address: FE9

Table 254. Port A–H Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W							
Address	FD1H, FD5H, FD9H, FDDH, FE1H, FE5H, FE9H, FEDH							

Hex Address: FEA

Table 255. Port A–H Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X							
R/W	R							
Address	FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH							

Hex Address: FEF

Table 260. Port A–H Output Data Register (P*x*OUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
Address	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

Watchdog Timer

For more information about these Watchdog Timer Control registers, see the Watchdog Timer Control Register Definitions section on page 83.

Hex Address: FF0

Table 261. Watchdog Timer Control Register (WDTCTL)

Bit	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	Reserved			SM
RESET	See Table 48 on page 84.			0				
R/W	R							
Address	FF0H							

Hex Address: FF1

Table 262. Watchdog Timer Reload Upper Byte Register (WDTU)

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	1							
R/W	R/W*							
Address	FF1H							

Note: *R/W = Read returns the current WDT count value; write sets the appropriate reload value.

D

E

F

V

W

X

Z