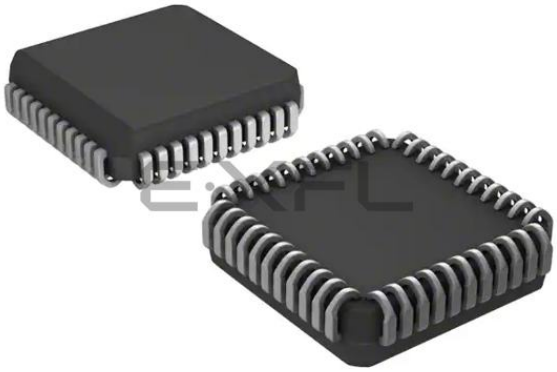

Welcome to E-XFL.COM

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	31
Program Memory Size	48KB (48K 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	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4821vn020eg

Watchdog Timer Reset

If the device is in normal or HALT Mode, the Watchdog Timer can initiate a system reset at time-out if the WDT_RES option bit is set to 1. This capability is the default (unprogrammed) setting of the WDT_RES option bit. The WDT status bit in the WDT Control Register is set to signify that the reset was initiated by the Watchdog Timer.

External Pin Reset

The $\overline{\text{RESET}}$ pin has a Schmitt-triggered input, an internal pull-up, an analog filter and a digital filter to reject noise. Once the $\overline{\text{RESET}}$ pin is asserted for at least 4 system clock cycles, the devices progress through the system reset sequence. While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8 Encore! XP F64xx Series devices continue to be held in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the system reset time-out, the devices exit the Reset state immediately following $\overline{\text{RESET}}$ pin deassertion. Following a system reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the Watchdog Timer Control (WDTCTL) Register is set to 1.

On-Chip Debugger Initiated Reset

A Power-On Reset can be initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The On-Chip Debugger block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the system reset. Following the system reset the POR bit in the WDT Control Register is set.

Stop Mode Recovery

STOP Mode is entered by the eZ8 executing a stop instruction. For detailed STOP Mode information, see the Low-Power Modes chapter on page 34. During Stop Mode Recovery, the devices are held in reset for 66 cycles of the Watchdog Timer oscillator followed by 16 cycles of the system clock. Stop Mode Recovery only affects the contents of the Watchdog Timer Control Register. Stop Mode Recovery does not affect any other values in the Register File, including the Stack Pointer, Register Pointer, Flags, peripheral control registers, and general-purpose RAM.

The eZ8 CPU fetches the Reset vector at program memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the stop bit in the Watchdog Timer Control Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions.

Port A–H Output Control Subregisters

The Port A–H Output Control Subregister, shown in Table 18, is accessed through the Port A–H Control Register by writing 03H to the Port A–H Address Register. Setting the bits in the Port A–H Output Control subregisters to 1 configures the specified port pins for open-drain operation. These subregisters affect the pins directly and, as a result, alternate functions are also affected.

Table 18. Port A–H Output Control Subregisters

Bit	7	6	5	4	3	2	1	0
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	0							
R/W	R/W							
Address	See note.							

Note: If a 03H exists in the Port A–H Address Register, it is accessible through the Port A–H Control Register.

Bit	Description
[7:0] POCx	**Port Output Control** These bits function independently of the alternate function bit and disables the drains if set to 1. 0 = The drains are enabled for any output mode. 1 = The drain of the associated pin is disabled (open-drain mode).

Note: x indicates register bits in the range [7:0].

Port A–H High Drive Enable Subregisters

The Port A–H High Drive Enable Subregister, shown in Table 19, is accessed through the Port A–H Control Register by writing 04H to the Port A–H Address Register. Setting the bits in the Port A–H High Drive Enable subregisters to 1 configures the specified port pins for high-current output drive operation. The Port A–H High Drive Enable Subregister affects the pins directly and, as a result, alternate functions are also affected.

Table 19. Port A–H High Drive Enable Subregisters

Bit	7	6	5	4	3	2	1	0
Field	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
RESET	0							
R/W	R/W							
Address	See note.							

Note: If a 04H exists in the Port A–H Address Register, it is accessible through the Port A–H Control Register.

Bit	Description
[7:0] PHDEx	**Port High Drive Enabled** 0 = The Port pin is configured for standard output current drive. 1 = The Port pin is configured for high output current drive.

Note: x indicates register bits in the range [7:0].

Table 24. Interrupt Request 0 Register (IRQ0)

Bit	7	6	5	4	3	2	1	0
Field	T2I	T1I	T0I	U0RXI	U0TXI	I2CI	SPII	ADCI
RESET	0							
R/W	R/W							
Address	FC0H							

Bit	Description
[7] T2I	**Timer 2 Interrupt Request** 0 = No interrupt request is pending for Timer 2. 1 = An interrupt request from Timer 2 is awaiting service.
[6] T1I	**Timer 1 Interrupt Request** 0 = No interrupt request is pending for Timer 1. 1 = An interrupt request from Timer 1 is awaiting service.
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for Timer 0. 1 = An interrupt request from Timer 0 is awaiting service.
[4] 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.
[3] 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.
[2] I2CI	**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.
[1] SPII	**SPI Interrupt Request** 0 = No interrupt request is pending for the SPI. 1 = An interrupt request from the SPI is awaiting service.
[0] 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.

IRQ2 Enable High and Low Bit Registers

Table 33 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers, shown in Tables 34 and 35, form a priority-encoded enabling for interrupts in the Interrupt Request 2 Register. Priority is generated by setting bits in each register.

Table 33. IRQ2 Enable and Priority Encoding

IRQ2ENH[*x*]	IRQ2ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: *x* indicates register bits in the range [7:0].

Table 34. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	7	6	5	4	3	2	1	0
Field	T3ENH	U1RENH	U1TENH	DMAENH	C3ENH	C2ENH	C1ENH	C0ENH
RESET	0							
R/W	R/W							
Address	FC7H							

Bit	Description
[7] T3ENH	**Timer 3 Interrupt Request Enable High Bit**
[6] U1RENH	**UART 1 Receive Interrupt Request Enable High Bit**
[5] U1TENH	**UART 1 Transmit Interrupt Request Enable High Bit**
[4] DMAENH	**DMA Interrupt Request Enable High Bit**
[3] C3ENH	**Port C3 Interrupt Request Enable High Bit**
[2] C2ENH	**Port C2 Interrupt Request Enable High Bit**
[1] C1ENH	**Port C1 Interrupt Request Enable High Bit**
[0] C0ENH	**Port C0 Interrupt Request Enable High Bit**

enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

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

1. Write to the Timer Control 1 Register to:
   - Disable the timer.
   - Configure the timer for COUNTER Mode.
   - Select either the rising edge or falling edge of the timer input signal for the count. This also sets the initial logic level (High or Low) for the timer output alternate function. However, the timer output function does not have to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in COUNTER Mode. After the first timer reload in COUNTER Mode, counting always begins at the reset value of 0001H. Generally, in COUNTER Mode the Timer High and Low Byte registers must be written with the value 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload 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. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
7. Write to the Timer Control 1 Register to enable the timer.

In COUNTER Mode, the number of timer input transitions since the timer start is calculated using the following equation:

COUNTER Mode Timer Input Transitions = Current Count Value – Start Value

PWM Mode

In PWM Mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the timer output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. 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.

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

GATED Mode

In GATED Mode, the timer counts only when the timer input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control 1 Register. When the timer input signal is asserted, counting begins. A timer interrupt is generated when the timer input signal is deasserted or a timer reload occurs. To determine if a timer input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

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. When 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 (assuming the timer input signal is still asserted). Also, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reset.

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

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for GATED Mode
   - Set the prescale value
2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload 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. Assert the timer input signal to initiate the counting.

CAPTURE/COMPARE Mode

In CAPTURE/COMPARE Mode, the timer begins counting on the first external timer input transition. The appropriate transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control 1 Register. The timer input is the system clock.

Bit	Description (Continued)
[1] STOP	**Stop Bit Select** 0 = The transmitter sends one stop bit. 1 = The transmitter sends two stop bits.
[0] LBEN	**Loop Back Enable** 0 = Normal operation. 1 = All transmitted data is looped back to the receiver.

Table 58. UART Control 1 Register (U*x*CTL1)

Bit	7	6	5	4	3	2	1	0
Field	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
RESET	0							
R/W	R/W							
Address	F43H and F4BH							

Bit	Description
[7,5] MPMD[1,0]	**MULTIPROCESSOR Mode** If MULTIPROCESSOR (9-Bit) Mode is enabled, 00 = The UART generates an interrupt request on all received bytes (data and address). 01 = The UART generates an interrupt request only on received address bytes. 10 = The UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs. 11 = The UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register.
[6] MPEN	**MULTIPROCESSOR (9-bit) Enable** This bit is used to enable MULTIPROCESSOR (9-Bit) Mode. 0 = Disable MULTIPROCESSOR (9-Bit) Mode. 1 = Enable MULTIPROCESSOR (9-Bit) Mode.
[4] MPBT	**MULTIPROCESSOR Bit Transmit** This bit is applicable only when MULTIPROCESSOR (9-Bit) Mode is enabled. 0 = Send a 0 in the multiprocessor bit location of the data stream (9th bit). 1 = Send a 1 in the multiprocessor bit location of the data stream (9th bit).
[3] DEPOL	**Driver Enable Polarity** 0 = DE signal is Active High. 1 = DE signal is Active Low.

since the previous pulse was detected). This gives the endec a sampling window of minus four baud rate clocks to plus eight 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 the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal. This action 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 Encoder/Decoder 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 98.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UARTx Control 1 Register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.

Table 69. SPI Baud Rate High Byte Register (SPIBRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1							
R/W	R/W							
Address	F66H							

Bit	Description
[7:0] BRH	**SPI Baud Rate High Byte** Most significant byte, BRG[15:8], of the SPI Baud Rate Generator's reload value.

Table 70. SPI Baud Rate Low Byte Register (SPIBRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Address	F67H							

Bit	Description
[7:0] BRL	**SPI Baud Rate Low Byte** Least significant byte, BRG[7:0], of the SPI Baud Rate Generator's reload value.

SDA and SCL Signals

I²C sends all addresses, data and acknowledge signals over the SDA line, most significant bit first. SCL is the common clock for the I²C 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 (I²C) 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 I²C 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.

I²C Interrupts

The I²C 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 I²C Controller and neither the start or stop bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status Register and can only be cleared by setting the start or stop bit in the I²C Control Register. When this interrupt occurs, the I²C 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 I²C Controller (master reading data from slave). This procedure sets the RDRF bit of the I²C Status Register. The RDRF bit is cleared by reading the I²C Data Register. The RDRF bit is set during the acknowledge phase. The I²C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

Transmit interrupts occur when the TDRE bit of the I²C Status Register sets and the TXI bit in the I²C Control Register is set. transmit interrupts occur under the following conditions when the transmit data register is empty:

- The I²C Controller is enabled

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 the I²C is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

1. Disable the I²C by clearing the IEN bit in the I²C Control Register to 0.
2. Load the appropriate 16-bit count value into the I²C Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BIRQ bit in the I²C Control Register to 1.

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]

Configuring DMA_ADC for Data Transfer

Observe the following procedure to configure and enable the DMA_ADC:

1. Write the DMA_ADC Address Register with the 7 most significant bits of the Register File address for data transfers.
2. Write to the DMA_ADC Control Register to complete the following operations:
   - Enable the DMA_ADC interrupt request, if appropriate
   - Select the number of ADC analog inputs to convert
   - Enable the DMA_ADC channel

Caution: When using the DMA_ADC to perform conversions on multiple ADC inputs, the Analog-to-Digital Converter must be configured for SINGLE-SHOT Mode. If the ADC_IN field in the DMA_ADC Control Register is greater than 000b, the ADC must be in SINGLE-SHOT Mode.

CONTINUOUS Mode operation of the ADC can only be used in conjunction with the DMA_ADC if the ADC_IN field in the DMA_ADC Control Register is reset to 000b to enable conversion on ADC analog input 0 only.

DMA Control Register Definitions

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

DMAx Control Register: see page 153

DMAx I/O Address Register: see page 154

DMAx Address High Nibble Register: see page 155

DMAx Start/Current Address Low Byte Register: see page 156

DMAx End Address Low Byte Register: see page 156

DMA_ADC Address Register: see page 157

DMA_ADC Control Register: see page 158

DMA_ADC Status Register: see page 159

Bit	Description (Continued)
[3] WSEL	**Word Select** 0 = DMA*x* transfers a single byte per request. 1 = DMA*x* transfers a two-byte word per request. The address for the on-chip peripheral control register must be an even address.
[2:0] RSS	**Request Trigger Source Select** The Request Trigger Source Select field determines the peripheral that can initiate a DMA transfer. The corresponding interrupts do not need to be enabled within the Interrupt Controller to initiate a DMA transfer. However, if the Request Trigger Source can enable or disable the interrupt request sent to the Interrupt Controller, the interrupt request must be enabled within the Request Trigger Source block. 000 = Timer 0. 001 = Timer 1. 010 = Timer 2. 011 = Timer 3. 100 = DMA0 Control Register: UART0 Received Data Register contains valid data. DMA1 Control Register: UART0 Transmit Data Register empty. 101 = DMA0 Control Register: UART1 Received Data Register contains valid data. DMA1 Control Register: UART1 Transmit Data Register empty. 110 = DMA0 Control Register: I^2C Receiver Interrupt. DMA1 Control Register: I^2C Transmitter Interrupt Register empty. 111 = Reserved.

DMA*x* I/O Address Register

The DMA*x* I/O Address Register, shown in Table 79, contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is provided by {FH, DMA*x*_IO[7:0]}. When the DMA is configured for two-byte word transfers, the DMA*x* I/O Address Register must contain an even-numbered address.

Table 79. DMA*x* I/O Address Register (DMA*x*IO)

Bit	7	6	5	4	3	2	1	0
Field	DMA_IO							
RESET	X							
R/W	R/W							
Address	FB1H, FB9H							

Bit	Description
[7:0] DMA_IO	**DMA On-Chip Peripheral Control Register Address** This byte sets the low byte of the on-chip peripheral control register address on Register File Page FH (addresses F00H to FFFH).

DMA Control of the ADC

The Direct Memory Access (DMA) Controller can control operation of the ADC including analog input selection and conversion enable. For more information about the DMA and configuring for ADC operations, see the Direct Memory Access Controller chapter on page 150.

ADC Control Register Definitions

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

ADC Control Register: see page 165

ADC Data High Byte Register: see page 167

ADC Data Low Bits Register: see page 168

ADC Control Register

The ADC Control Register selects the analog input channel and initiates the analog-to-digital conversion.

Table 87. ADC Control Register (ADCCTL)

Bit	7	6	5	4	3	2	1	0
Field	CEN	Reserved	VREF	CONT	ANAIN[3:0]			
RESET	0		1	0				
R/W	R/W							
Address	F70H							

Bit	Description
[7] CEN	**Conversion Enable** 0 = Conversion is complete. Writing a 0 produces no effect. The ADC automatically clears this bit to 0 when a conversion has been completed. 1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] VREF	**Voltage Reference** 0 = Internal voltage reference generator enabled. The V_{REF} pin should be left unconnected (or capacitively coupled to analog ground) if the internal voltage reference is selected as the ADC reference voltage. 1 = Internal voltage reference generator disabled. An external voltage reference must be provided through the V_{REF} pin.

Hex Address: F1D

Table 167. Timer 0–3 PWM Low Byte Register (TxPWML)

Bit	7	6	5	4	3	2	1	0
Field	PWML							
RESET	0							
R/W	R/W							
Address	F05H, F0DH, F15H, F1DH							

Hex Address: F1E

Table 168. Timer 0–3 Control 0 Register (TxCTL0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			CSC	Reserved			
RESET	0							
R/W	R/W							
Address	F06H, F0EH, F16H, F1EH							

Hex Address: F1F

Table 169. Timer 0–3 Control 1 Register (TxCTL1)

Bit	7	6	5	4	3	2	1	0
Field	TEN	TPOL	PRES			TMODE		
RESET	0							
R/W	R/W							
Address	F07H, F0FH, F17H, F1FH							

Hex Addresses: F20–F39

This address range is reserved.

Universal Asynchronous Receiver/Transmitter (UART)

For more information about these UART Control registers, see the UART Control Register Definitions section on page 98.

Hex Address: F4E

Table 186. UART Baud Rate High Byte Register (U*x*BRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1							
R/W	R/W							
Address	F46H and F4EH							

Hex Address: F4F

Table 187. UART Baud Rate Low Byte Register (U*x*BRL)

Bit7	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Address	F47H and F4FH							

Inter-Integrated Circuit (I^2C)

For more information about these I^2C Control registers, see the I2C Control Register Definitions section on page 141.

Hex Address: F50

Table 188. I^2C Data Register (I2CDATA)

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

D

E

F

G

H

I