Zilog - Z16F2810FI20SG Datasheet

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	ZNEO
Core Size	16-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	60
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f2810fi20sg

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Section	Description	Page No.
Nov 2013	12	Signal Descriptions	Corrected active status of RD, WR and $\overline{CS}$ signals.	12
Jul 2013	11	Analog Functions	Updated the Analog Functions Block Diagram.	242
Aug 2011	10	Multi-Channel PWM Timer	Per CR#13095, corrected PWMEN description in PWM Control 0 Register (PWMCTL0) table; corrected description in PWM Deadband Register (PWMDB) table and added footnote; added same footnote to PWM Minimum Pulse Width Filter (PWMMPF), PWM Fault Mask Register (PWMFM), and PWM Fault Control Register (PWMFCTL) tables.	125, 127–129, 131
Jun 2011	09	Electrical Characteristics	Corrected V_{COFF} input offset value in Comparator Electrical Characteristics table	347
Aug 2010	08	N/A	Removed ISO information.	ii
		All	Updated logos.	All
		Table 191	Changed the Minimum, Typical and Maximum values for V_{REF} (Externally supplied Voltage Reference only).	346
Jan 2009	07	Timer 0–2 Control 0 Register	Table 62: added "Only Counter Mode should be used with this feature" to Bit 4 description.	109
		Analog Functions	ADC Overview, updated fast conversion time to 2.5µs.	243
		Electrical Characteristics	Updated Table 185.	337
		Internal Precision Oscillator	Removed reference to 32kHz.	336

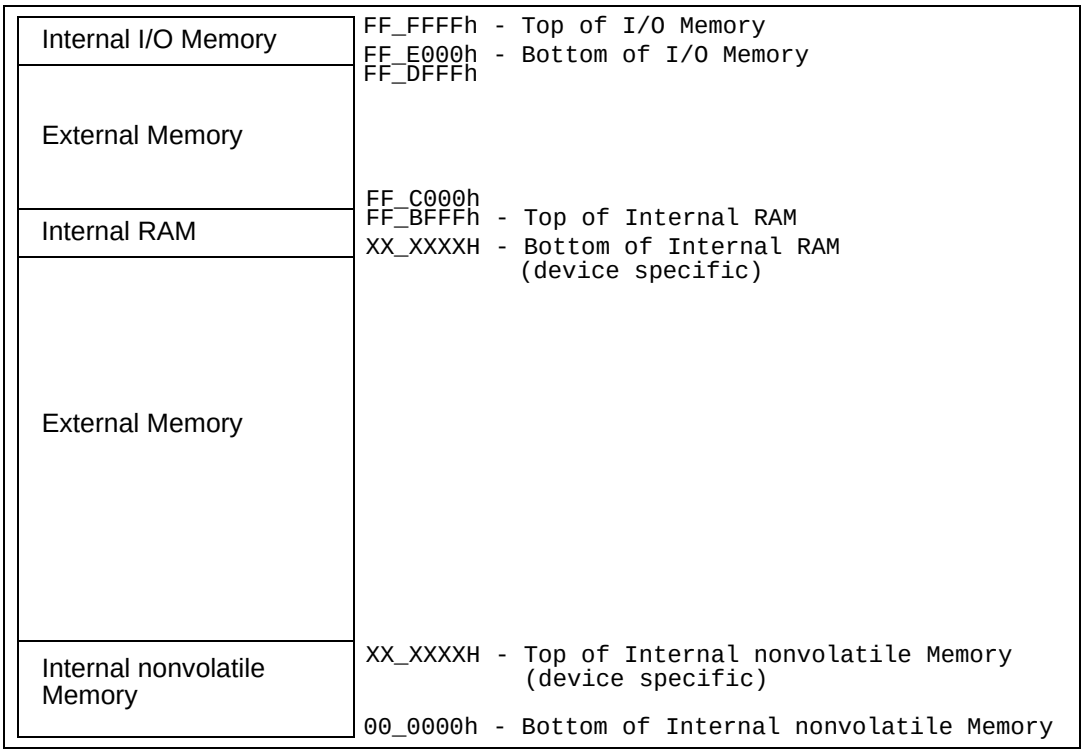

Figure 6. Physical Memory Map

To determine the amount of internal RAM and internal nonvolatile memory available for the specific device, see the Ordering Information section on page 356.

Internal Nonvolatile Memory

Internal nonvolatile memory contains executable program code, constants and data. For each product within the ZNEO CPU family, a memory block beginning at address 00_0000h is reserved for user option bits and system vectors (for example, RESET, Trap, Interrupts and System Exceptions, etc.). Table 4 provides an example of reserved memory map for a ZNEO CPU product with 24 interrupt vectors.

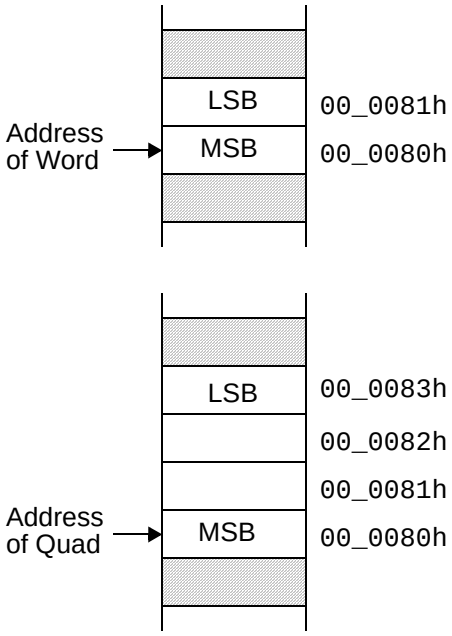

Figure 7. Endianness of Words and Quads

Bus Widths

The ZNEO CPU accesses 8-bit or 16-bit memories. The data buses of the internal nonvolatile memory and internal RAM are 16-bit wide. The internal peripherals are a mix of 8-bit and 16-bit peripherals. The external memory bus is configured as an 8-bit or 16-bit memory bus.

If a Word or Quad operation occurs on a 16-bit wide memory, the number of memory accesses depends on the alignment of the address. If the address is aligned on an even boundary, a Word operation takes one memory access and a Quad operation takes two memory accesses. If the address is on an odd boundary (unaligned), a Word operation takes two memory accesses and a Quad operation takes three memory accesses. Figure 8 displays the alignment Word and Quad operations on 16-bit memories.

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FF_E03B	IRQ2 Enable Low Bit	IRQ2ENL	00	92
FF_E03C-FF_E03F	Reserved	—	XX	—
Watchdog Timer Base Address = FF_E040				
FF_E040-FF_E041	Reserved	—	—	—
FF_E042	Watchdog Timer Reload High Byte	WDTH	04	241
FF_E043	Watchdog Timer Reload Low Byte	WDTL	00	241
FF_E044-FF_E04F	Reserved	—	—	—
Reset Base Address = FF_E050				
FF_E050	Reset Status and Control Register	RSTSCR	XX	62
FF_E051-FF_E06F	Reserved	—	XX	—
Flash Controller Base Address = FF_E060				
FF_E060	Flash Command Register	FCMD	XX	261
FF_E060	Flash Status Register	FSTAT	00	262
FF_E061	Flash Control Register	FCTL	00	263
FF_E062	Flash Sector Protect Register	FSECT	00	264
FF_E063	Reserved	—	XX	—
FF_E064-FF_E065	Flash Page Select Register	FPAGE	0000	265
FF_E066-FF_E067	Flash Frequency Register	FFREQ	0000	266
External Interface Base Address = FF_E070				
FF_E070	External Interface Control	EXTCT		42
FF_E071	Reserved	—	—	—
FF_E072	Chip Select 0 Control High	EXTCS0H		43
FF_E073	Chip Select 0 Control Low	EXTCS0L		44
FF_E074	Chip Select 1 Control High	EXTCS1H		43
FF_E075	Chip Select 1 Control Low	EXTCS1L		45
FF_E076	Chip Select 2 Control High	EXTCS2H		43
FF_E077	Chip Select 2 Control Low	EXTCS2L		46
FF_E078	Chip Select 3 Control High	EXTCS3H		43
FF_E079	Chip Select 3 Control Low	EXTCS3L		46
FF_E07A	Chip Select 4 Control High	EXTCS4H		43

XX = Undefined.

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
GPIO Port G Base Address = FF_E160				
FF_E160	Port G Input Data	PGIN	XX	71
FF_E161	Port G Output Data	PGOUT	00	72
FF_E162	Port G Data Direction	PGDD	00	73
FF_E163	Port G High Drive Enable	PGHDE	00	74
FF_E164	Reserved	—	—	—
FF_E165	Port G Alternate Function Low	PGAFL	00	76
FF_E166	Port G Output Control	PGOC	00	76
FF_E167	Port G Pull-Up Enable	PGPUE	00	76
FF_E168	Port G Stop Mode Recovery Enable	PGSMRE	00	77
FF_E169-FF_E16F	Port G Reserved	—	—	—
GPIO Port H Base Address = FF_E170				
FF_E170	Port H Input Data	PHIN	XX	71
FF_E171	Port H Output Data	PHOUT	00	72
FF_E172	Port H Data Direction	PHDD	00	73
FF_E173	Port H High Drive Enable	PHHDE	00	74
FF_E174	Port H Alternate Function High	PHAFH	00	75
FF_E175	Port H Alternate Function Low	PHAFL	00	76
FF_E176	Port H Output Control	PHOC	00	76
FF_E177	Port H Pull-Up Enable	PHPUE	00	76
FF_E178	Port H Stop Mode Recovery Enable	PHSMRE	00	77
FF_E179-FF_E17F	Port H Reserved	—	—	—
GPIO Port J Base Address = FF_E180				
FF_E180	Port J Input Data	PJIN	XX	71
FF_E181	Port J Output Data	PJOUT	00	72
FF_E182	Port J Data Direction	PJDD	00	73
FF_E183	Port J High Drive Enable	PJHDE	00	74
FF_E184	Reserved	—	—	—
FF_E185	Reserved	—	—	—
FF_E186	Port J Output Control	PJOC	00	76
FF_E187	Port J Pull-Up Enable	PJPUE	00	76

XX = Undefined.

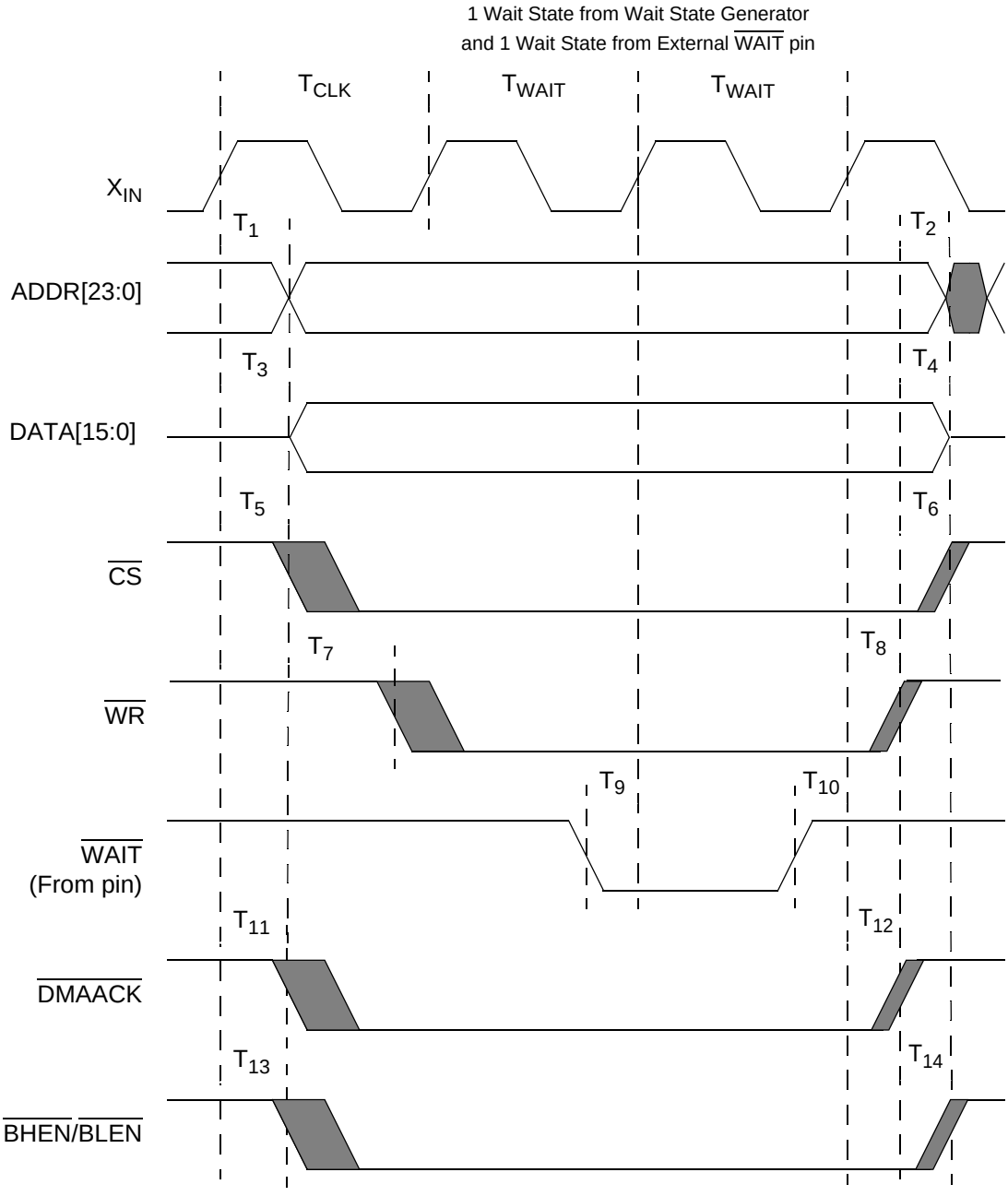

Figure 11. External Interface Timing for a Write Operation, Normal Mode

Reset Status and Control Register

The Reset Status and Control Register (RSTSCR), shown in Table 21, records the cause of the most recent RESET or Stop Mode Recovery. All status bits are updated on each RESET or Stop Mode Recovery event. Table 22 indicates the possible states of the Reset status bits following a RESET or Stop Mode Recovery event.

Table 21. Reset Status and Control Register (RSTSCR)

Bits	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	FLT	USR	Reserved	USER_RST
RESET	See Table 22 for a description of these bits.							
R/W	R	R	R	R	R	R	R	W
Addr	FF–E050h							

Bit	Description
[7] POR	For a description of bits [7:2], see Table 22.
[6] STOP	
[5] WDT	
[4] EXT	
[3] FLT	
[2] USR	
[1]	**Reserved** These bits are reserved and must be programmed to 0.
[0] USER_RST	The USER_RST bit in this register allows software controlled RESET of the part pin. This bit is a Write Only bit that causes a System Reset with the result identified by the USR bit after being executed. 0 = No action. 1 = Causes System Reset.

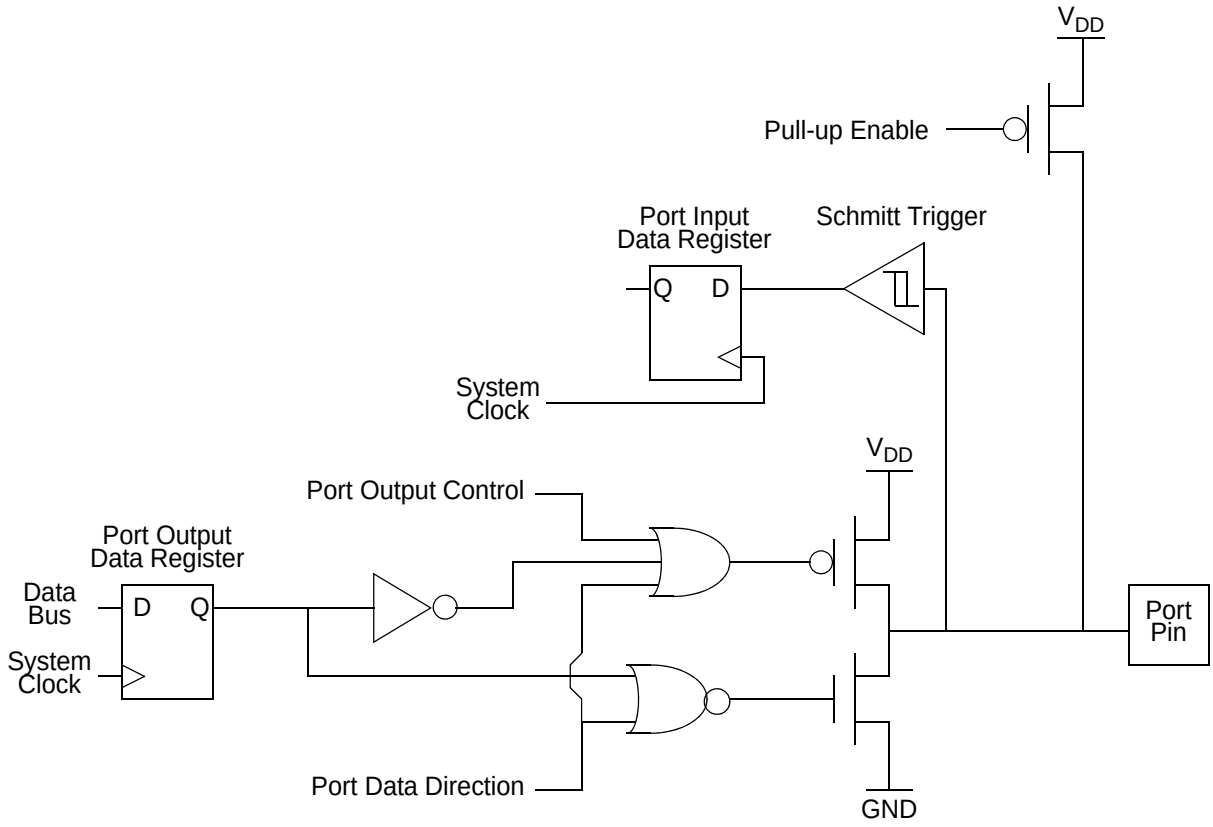

Figure 18. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many GPIO port pins are used for GPIO and to provide access to the on-chip peripheral functions such as timers, serial communication devices and external data and address bus. The Port A–K alternate function registers configure these pins for either GPIO or alternate function operation. When a pin is configured for alternate function, control of the port pin direction (I/O) is passed from the Port A–K data direction registers to the alternate function assigned to this pin. Table 24 on page 68 lists the alternate functions associated with each port pin.

For detailed information about enabling the external interface data signals, see the External Interface chapter on page 37. When the external interface data signals are enabled for an 8-bit port, the other GPIO functionality including alternate functions cannot be used.

Table 24. Port Alternate Function Mapping

Port	Pin	Alternate Function 1	Alternate Function 2	Alternate Function 3	External Interface
Port A	PA0	T0IN/$\overline{\text{T0OUT}}$	DMA0REQ	T0INPB	
	PA1	T0OUT	DMA0ACK		
	PA2	DE0	FAULTY		
	PA3	CTS0	FAULT0		
	PA4	RXD0	$\overline{\text{CS1}}$		
	PA5	TXD0	$\overline{\text{CS2}}$		
	PA6	SCL	$\overline{\text{CS3}}$		
	PA7	SDA	$\overline{\text{CS4}}$		
Port B	PB0/T0IN0	ANA0			
	PB1/T0IN1	ANA1			
	PB2/T0IN2	ANA2			
	PB3	ANA3/OPOUT			
	PB4	ANA4			
	PB5	ANA5			
	PB6	ANA6/OPINP/CINN			
	PB7	ANA7/OPINN			
Port C	PC0	T1IN/$\overline{\text{T1OUT}}$	DMA1REQ	CINN	
	PC1	T1OUT	DMA1ACK	COMPOUT	
	PC2	SS	$\overline{\text{CS4}}$		
	PC3	SCK	DMA2REQ		
	PC4	MOSI	DMA2ACK		
	PC5	MISO	$\overline{\text{CS5}}$		
	PC6	T2IN/$\overline{\text{T2OUT}}$	PWMH0		
	PC7	T2OUT	PWML0		
Port D	PD0	PWMH1	ADDR[20]		
	PD1	PWML1	ADDR[21]		
	PD2	PWMH2	ADDR[22]		
	PD3	DE1	ADDR[16]		
	PD4	RXD1	ADDR[18]		
	PD5	TXD1	ADDR[19]		
	PD6	CTS1	ADDR[17]		
	PD7	PWML2	ADDR[23]		

5. Enable the timer interrupt, if required and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin(s) for the timer output alternate function.
7. Write to the Timer Control 1 Register to enable the timer and initiate counting.

The PWM period is determined by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001h is loaded into the Timer High and Low Byte registers, use the One-Shot Mode equation 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 determined by:

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

If TPOL is set to 1, the ratio of the PWM output High time to the total period is determined by:

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

Capture Modes

There are three Capture modes which provide slightly different methods for recording the time or time interval between timer input events. These modes are Capture Mode, Capture Restart Mode and Capture Compare Mode. In all of the three modes, when the appropriate timer input transition (capture event) occurs, the timer counter value is captured and stored in the PWM High and Low Byte registers. The TPOL bit in the Timer Control 1 Register determines if the Capture occurs on a rising edge or a falling edge of the timer input signal. The TICONFIG bit determines whether interrupts are generated on capture events, reload events or both. The INCAP bit in Timer Control 0 Register clears to indicate an interrupt caused by a reload event and sets to indicate the timer interrupt is caused by an input capture event.

If the timer output alternate function is enabled, the timer output pin changes state (from Low to High or High to Low) at timer Reload. The initial value is determined by the TPOL bit.

Reading Timer Count Values

The current count value in the timer is read while counting (enabled). This has no effect on timer operation. Normally, the count must be read with one 16-bit operation. However, 8-bit reads are done with the following method. 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 register. This operation allows accurate reads of the full 16-bit timer count value when enabled. When the timer is not enabled, a read from the timer low byte register returns the actual value in the counter.

The Timers can be cascaded by using the Cascade bit in the Timer control registers. When this bit is set for a Timer, the input source is redefined. When the Cascade bit is set for Timer 0, the input for Timer 0 is the output of the Analog Comparator. When the Cascade bit is set for Timer 1 and Timer 2, the output of Timer 0 and Timer 1 become the input for Timer 1 and Timer 2, respectively. Any Timer Mode can be used. Timer 0 can be cascaded to Timer 1 only by setting the Cascade bit for Timer 1. Timer 1 cascaded to Timer 2 only by setting the Cascade bit for Timer 2. Or all three cascaded, Timer 0 to Timer 1 or Timer 2 for really long counts by setting the Cascade bit for Timer 1 and Timer 2.

Timer Control Register Definitions

Timer 0–2 High and Low Byte Registers

The Timer 0–2 High and Low Byte (TxH and TxL) registers, shown in Tables 56 and 57, contain the current 16-bit timer count value. When the timer is enabled, a read from TxH stores the value in TxL to a temporary holding register. A read from TxL always returns this temporary register when the timer is enabled. When the timer is disabled, reads from the TxL reads the register directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations, so simultaneous 16-bit writes are not possible. When either of the timer high or low byte registers are written during counting, the 8-bit written value is placed in the counter (High or Low Byte) at the next clock edge. The counter continues counting from the new value.

ESPI Control Register Definitions

ESPI Data Register

The ESPI Data Register, shown in Table 102, addresses both the outgoing Transmit Data register and the incoming Receive Data register. Reads from the ESPI Data register return the contents of the Receive Data register. The Receive Data register is updated with the contents of the shift register at the end of each transfer. Writes to the ESPI Data register load the Transmit Data register unless TDRE = 0. Data is shifted out starting with bit 7. The last bit received resides in bit position 0.

With the ESPI configured as a Master, writing a data byte to this register initiates the data transmission. With the ESPI configured as a Slave, writing a data byte to this register loads the shift register in preparation for the next data transfer with the external Master. In either the Master or Slave modes, if TDRE = 0, writes to this register are ignored.

When the character length is less than 8 bits (as set by the NUMBITS field in the ESPI Mode register), the transmit character must be left justified in the ESPI Data register. A received character of less than 8 bits is right justified (last bit received is in bit position 0). For example, if the ESPI is configured for 4-bit characters, the transmit characters must be written to ESPIDATA[7:4] and the received characters are read from ESPIDATA[3:0].

Table 102. ESPI Data Register (ESPIDATA)

Bits	7	6	5	4	3	2	1	0
Field	DATA							
RESET	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF_E260h							

Bit	Description
[7:0] DATA	**Data** Transmit and/or receive data. Writes to the ESPIDATA Register load the shift register. Reads from the ESPIDATA Register return the value of the Receive Data Register.

ESPI Transmit Data Command Register

The ESPI Transmit Data Command Register, shown in Table 103, provides control of the $\overline{SS}$ pin when it is configured as an output (Master Mode). The TEOF and SSV bits are controlled by the DMA interface as well as by a bus write to this register.

ters accessing different Slaves) or during the data phase when the Masters are attempting to write different data to the same Slave.

When a Master loses arbitration, software is informed by means of the Arbitration Lost interrupt. Software repeats the same transaction again at a later time.

A special case occurs when a slave transaction starts just before software attempts to start a new master transaction by setting the Start bit. In this case the state machine enters the slave states before the Start bit is set and the I^2C Controller does not arbitrate. If a slave address match occurs and the I^2C Controller receives or transmits data, the Start bit is cleared and an Arbitration Lost interrupt is asserted. Software minimizes the chance of this occurring by checking the BUSY bit in the I2CSTATE Register before initiating a master transaction. If a slave address match does not occur, the Arbitration Lost interrupt does not occur and the Start bit is not cleared. The I^2C Controller initiates the master transaction after the I^2C bus is no longer busy.

Master Address Only Transactions

It is sometimes appropriate to perform an address-only transaction to determine if a particular Slave device is able to respond. This transaction is performed by monitoring the ACKV bit in the I2CSTATE Register after the address has been written to the I2CDATA Register and the Start bit has been set. After ACKV is set, the ACK bit in the I2CSTATE Register determines if the Slave is able to communicate. The Stop bit must be set in the I2CCTL Register to terminate the transaction without transferring data. For a 10-bit slave address, if the first address byte is acknowledged, the second address byte must also be sent to determine if the appropriate slave is responding.

Another approach is to set both the Stop and Start bits (for sending a 7-bit address). After both bits are cleared (7-bit address has been sent and transaction is complete), the ACK bit is read to determine if the slave is acknowledged. For a 10-bit slave, set the Stop bit after the second TDRE interrupt (second address byte is being sent).

Master Transaction Diagrams

In the following transaction diagrams, shaded regions indicate data transferred from the Master to the Slave and unshaded regions indicate data transferred from the Slave to the Master. The transaction field labels are defined as follows:

S: Start

W: Write

A: Acknowledge

$\overline{A}$: Not Acknowledge

P: Stop

Bit	Description (Continued)
[5] STOP	**Send Stop Condition** When set, this bit causes the I2C Controller (when configured as the Master) to send the Stop condition after the byte in the I2C Shift Register has completed transmission or after a byte has been received in a receive operation. When set, this bit is reset by the I2C Controller after a Stop condition has been sent or by deasserting the IEN bit. If this bit is 1, it cannot be cleared to 0 by writing to the register. If Stop is set while a Slave Mode transaction is underway, the Stop bit will be cleared by hardware.
[4] BIRQ	**Baud Rate Generator Interrupt Request** This bit is ignored when the I2C Controller is enabled. If this bit is set = 1 when the I2C Controller is disabled (IEN = 0) the baud rate generator is used as an additional timer causing an interrupt to occur every time the baud rate generator counts down to 1. The baud rate generator runs continuously in this Mode, generating periodic interrupts.
[3] TXI	**Enable TDRE Interrupts** This bit enables interrupts when the I2C Data Register is empty.
[2] NAK	**Send NAK** Setting this bit sends a Not Acknowledge condition after the next byte of data has been received. It is automatically deasserted after the Not Acknowledge is sent or the IEN bit is cleared. If this bit is 1, it cannot be cleared to 0 by writing to the register.
[1] FLUSH	**Flush Data** Setting this bit clears the I2C Data Register and sets the TDRE bit to 1. This bit allows flushing of the I2C Data Register when an NAK condition is received after the next data byte has been written to the I2C Data Register. Reading this bit always returns 0.
[0] FILTEN	**I2C Signal Filter Enable** Setting this bit enables low-pass digital filters on the SDA and SCL input signals. This function provides the spike suppression filter required in I2C Fast Mode. 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.

I2C Baud Rate High and Low Byte Registers

The I2C Baud Rate High and Low Byte registers, shown in Tables 115 and 116, combine to form a 16-bit reload value, BRG[15:0], for the I2C Baud Rate Generator. The baud rate High and Low Byte Registers must be programmed for the I2C baud rate in Slave Mode as well as in Master Mode. In Slave Mode, the baud rate value programmed must match the master's baud rate within ±25% for proper operation.

The I2C baud rate is calculated using the following equation.

$$\text{I2C Baud Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{4 \times \text{BRG[15:0]}}$$

ADC0 Data Low Bits Register

The ADC0 Data Low Bits Register contains the lower bits of the ADC0 output. Access to the ADC0 Data Low Bits Register is Read-Only.

Table 128. ADC0 Data Low Bits Register (ADC0D_L)

Bits	7	6	5	4	3	2	1	0
Field	ADC0D_L		Reserved					
RESET	X		X					
R/W	R		R					
Addr	FF–E503h							

Bit	Description
[7:6] ADC0D_L	**ADC0 Low Bits** 00–11b = These bits are the 2 least significant bits of the 10-bit ADC0 output. These bits are undefined after a Reset.
[5:0]	**Reserved** These bits are reserved and must be programmed to 0.

Sample Settling Time Register

The Sample Settling Time Register is used to program the length of time from the SAMPLE/$\overline{\text{HOLD}}$ signal to the Start signal, when the conversion begins. The number of clock cycles required for settling varies from system to system depending on the system clock period used. This register must be programmed to contain the number of clocks required to meet a 0.5µs minimum settling time.

Table 129. Sample and Settling Time (ADCSST)

Bits	7	6	5	4	3	2	1	0
Field	Reserved			SST				
RESET	0	0	0	1	1	1	1	1
R/W	R			R/W				
Addr	FF–E504h							

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 0.
[4:0] SST	**Sample Settling Time** 00h–1Fh = Sample settling time in number of system clock periods to meet 0.5µs minimum.

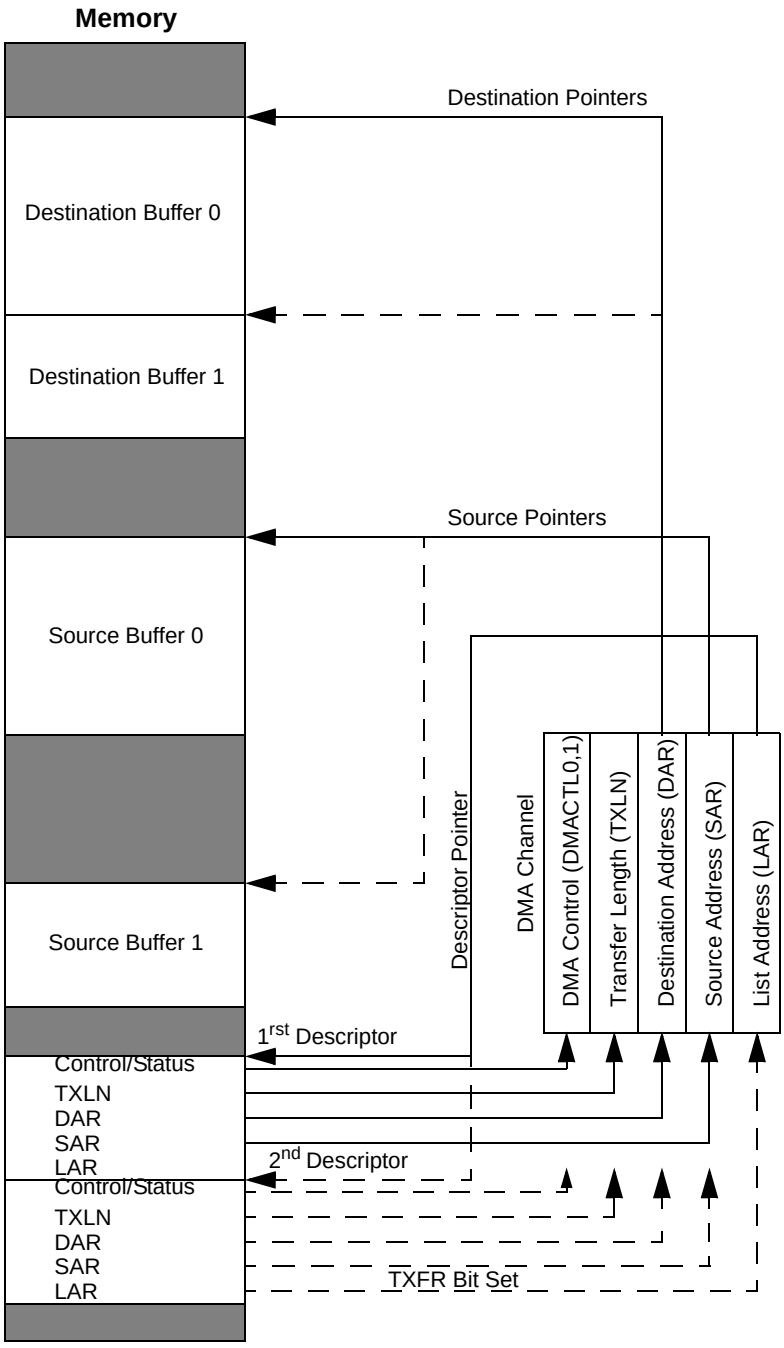

Figure 59. Linked List Diagram

Table 151. DMA X Transfer Length Low Register (DMAxTXLNL)

Bits	7	6	5	4	3	2	1	0
Field	DMAxTXLNL							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE413h, FFE423h, FFE433h, FFE443h							

DMA Destination Address

The DMA X Destination Address Register Upper, High and Low registers form the destination address. This address points to the location in which the data from the transfer will be stored.

Table 152. DMA X Destination Address Register Upper (DMAxDARU)

Bits	7	6	5	4	3	2	1	0
Field	DMAxDARU							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE415h, FFE425h,FFE435h,FFE445							

Table 153. DMA X Destination Address Register High (DMAxDARH)

Bits	7	6	5	4	3	2	1	0
Field	DMAxDARH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE416h, FFE426h, FFE436h, FFE446h							

Table 154. DMA X Destination Address Register Low (DMAxDARL)

Bits	7	6	5	4	3	2	1	0
Field	DMAxDARL							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFE417h, FFE427h, FFE437h, FFE447h							