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	EBI/EMI, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	76
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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f2811al20eg

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

Brackets

The square brackets, [ ], indicate a register or bus.

Example. For the register R1[7:0], R1 is an 8-bit register, R1[7] is the most significant bit, and R1[0] is the least significant bit.

Braces

The curly braces { }, indicate a single register or bus created by concatenating some combination of smaller registers, buses, or individual bits.

Example. The 12-bit register address {0h, RP[7:4], R1[3:0]} is composed of a 4-bit hexadecimal value (0h) and two 4-bit register values taken from the register pointer (RP) and working register R1. 0h is the most significant nibble (4-bit value) of the 12-bit register, and R1[3:0] is the least significant nibble of the 12-bit register.

Parentheses

The parentheses ( ), indicate an indirect register address lookup.

Example. (R1) is the memory location referenced by the address contained in the working register R1.

Parentheses/Bracket Combinations

The parentheses ( ), indicate an indirect register address lookup and the square brackets [ ], indicate a register or bus.

Example. Assume PC[15:0] contains the value 1234h. (PC[15:0]) refers to the contents of the memory location at the address 1234h.

Use of the Words Set, Reset, and Clear

The word set implies that a register bit or a condition contains a logical 1. The words reset or clear imply that a register bit or a condition contains a logical 0. When either of these terms is followed by a number, the word logical may not be included; however, it is implied.

Notation for Bits and Similar Registers

A field of bits within a register is designated as: Register[*n*:*n*].

Example. ADDR[15:0] refers to bit 15 through bit 0 of the address.

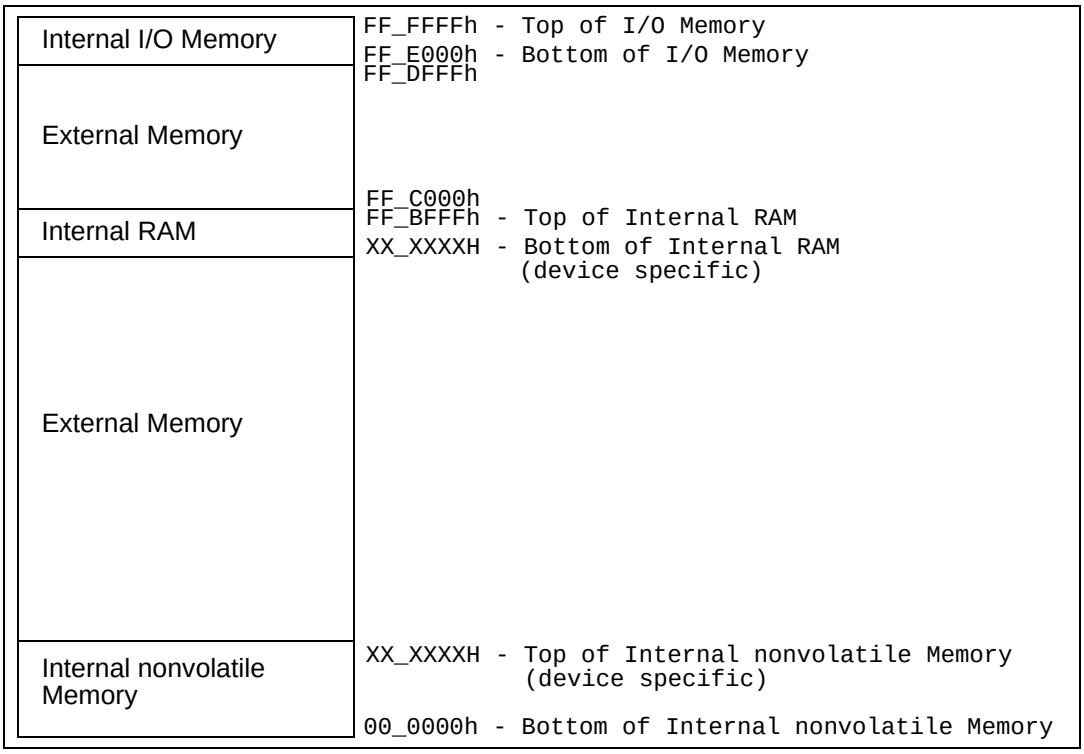

Figure 6. Physical Memory Map

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.

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FF_E132	Port D Data Direction	PDDD	00	73
FF_E133	Port D High Drive Enable	PDHDE	00	74
FF_E134	Port D Alternate Function High	PDAFH	00	75
FF_E135	Port D Alternate Function Low	PDAFL	00	76
FF_E136	Port D Output Control	PDOC	00	76
FF_E137	Port D Pull-Up Enable	PDPUE	00	76
FF_E138	Port D Stop Mode Recovery Enable	PDSMRE	00	77
FF_E139-FF_E13F	Port D Reserved	—	—	—
GPIO Port E Base Address = FF_E140				
FF_E140	Port E Input Data	PEIN	XX	71
FF_E141	Port E Output Data	PEOUT	00	72
FF_E142	Port E Data Direction	PEDD	00	73
FF_E143	Port E High Drive Enable	PEHDE	00	74
FF_E144	Reserved	—	—	—
FF_E145	Reserved	—	—	—
FF_E146	Port E Output Control	PEOC	00	76
FF_E147	Port E Pull-Up Enable	PEPUE	00	76
FF_E148	Port E Stop Mode Recovery Enable	PESMRE	00	77
FF_E149-FF_E14F	Port E Reserved	—	—	—
GPIO Port F Base Address = FF_E150				
FF_E150	Port F Input Data	PFIN	XX	71
FF_E151	Port F Output Data	PFOUT	00	72
FF_E152	Port F Data Direction	PFDD	00	73
FF_E153	Port F High Drive Enable	PFHDE	00	74
FF_E154	Reserved	—	—	—
FF_E155	Port F Alternate Function Low	PFAFL	00	76
FF_E156	Port F Output Control	PFOC	00	76
FF_E157	Port F Pull-Up Enable	PFPUE	00	76
FF_E158	Port F Stop Mode Recovery Enable	PFSMRE	00	77
FF_E159-FF_E15F	Port F Reserved	—	—	—

XX = Undefined.

Bit	Description
[4] PWMFENL	PWM Fault Interrupt Request Enable Low Bit.
[3:0] CxENL/ DMAxENL	Port C*x* or *DMAx* Interrupt Request Enable Low Bit.

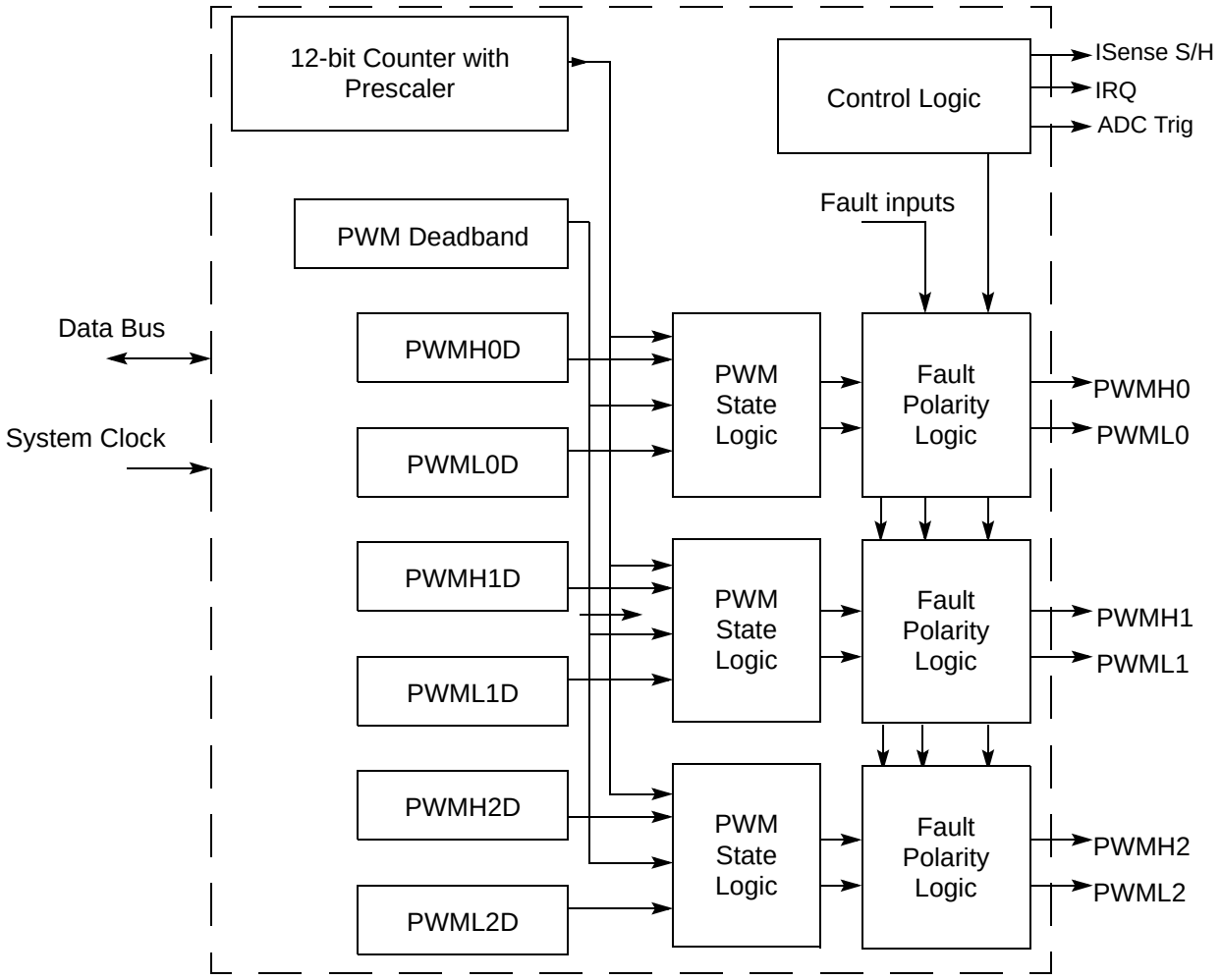

Figure 21. PWM Block Diagram

Operation

PWM Option Bits

To protect the configuration of critical PWM parameters, settings to enable output channels and the default off-state are maintained as user option bits. These values are set when the user program code is written to the part; the software cannot change these values. For more, see the Option Bits chapter on page 292.

PWM 0–2 Duty Cycle High and Low Byte Registers

The PWM 0–2 H/L (High Side/Low Side) Duty Cycle High and Low Byte (PWM*x*DH and PWM*x*DL) registers, shown in Tables 68 and 69, set the duty cycle of the PWM signal. This 14-bit signed value is compared to the PWM count value to determine the PWM output. Reads from these registers always return the values from the temporary holding registers. The PWM generator does not use the PWM duty cycle value until the next PWM reload event occurs.

$$\text{PWM Duty Cycle} = 100 \times \frac{\text{PWM Duty Cycle Value}}{\text{PWM Reload Value}}$$

Writing a negative value (DUTYH[7] = 1) forces the PWM to be OFF for the full PWM period. Writing a positive value greater than the 12-bit PWM reload value forces the PWM to be ON for the full PWM period.

Table 68. PWM 0–2 H/L Duty Cycle High Byte Register (PWMH*x*DH, PWML*x*DH)

Bits	7	6	5	4	3	2	1	0
Field	SIGN	Reserved		DUTYH				
RESET	X	XX		X_XXXX				
R/W	R/W	R/W		R/W				
Addr	FF_E390h, FF_E392h, FF_E394h, FF_E396h, FF_E398h, FF_E39Ah							

Bit	Description
[7] SIGN	**Duty Cycle Sign** 0 = Duty cycle is a positive two's complement number. 1 = Duty cycle is a negative two's complement number. Output is forced to the off-state.
[6:5]	**Reserved** These bits are reserved and must be programmed to 00.
[4:0] DUTYH	**PWM Duty Cycle High and Low Bytes** The upper byte of two bytes {DUTYH[7:0], DUTYL[7:0]} that form a 14-bit signed value; bits 5 and 6 of the High byte are always 0. The value is compared to the current 12-bit PWM count.

Bit	Description (Continued)
[1] Fault0INT	**Fault 0 Interrupt** 0 = Interrupt on fault 0 pin assertion disabled. 1 = Interrupt on Fault0 pin assertion enabled.
[0] Fault0RST	**Fault 0 Restart** 0 = Automatic recovery. PWM resumes control of outputs when all fault sources have deasstered. **Software Controlled Recovery** 1 = PWM resumes control of outputs only after all fault sources have deasserted and all fault flags are cleared and a PWM reload occurs.

Note: This register can only be written when PWMEN is cleared.

PWM Input Sample Register

The PWM pin value is sampled by reading the PWM Input Sample Register, shown in Table 77.

Table 77. PWM Input Sample Register (PWMIN)

Bits	7	6	5	4	3	2	1	0
Field	Reserved	FAULT	IN2L	IN2H	IN1L	IN1H	IN0L	IN0H
RESET	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF_E386h							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] FAULT	**Sample Fault0 pin** 0 = A Low-level signal was read on the fault pin. 1 = A High-level signal was read on the fault pin.
[5:0] IN2L/IN2H/ IN1L/IN1H/ IN0L/IN0H	**Sample PWM Pins** 0 = A Low-level signal was read on the pins. 1 = A High-level signal was read on the pins.

passed to the UART. Communication is half-duplex, which means that simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART's baud rate generator and supports IrDA standard baud rates from 9600 baud to 115.2 Kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the infrared endec. The infrared endec data rate is calculated using the following equation:

$$\text{Infrared Data Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/infrared data bit is 16-clocks wide. If the data to be transmitted is 1, the IR_TXD signal remains Low for the full 16-clock period. If the data to be transmitted is 0, a 3-clock high pulse is output following a 7-clock low period. After the 3-clock high pulse, a 6-clock low pulse is output to complete the full 16-clock data period. Figure 33 displays IrDA data transmission. When the infrared endec is enabled, the UART's TXD signal is internal to the ZNEO Z16F Series products while the IR_TXD signal is output through the TXD pin.

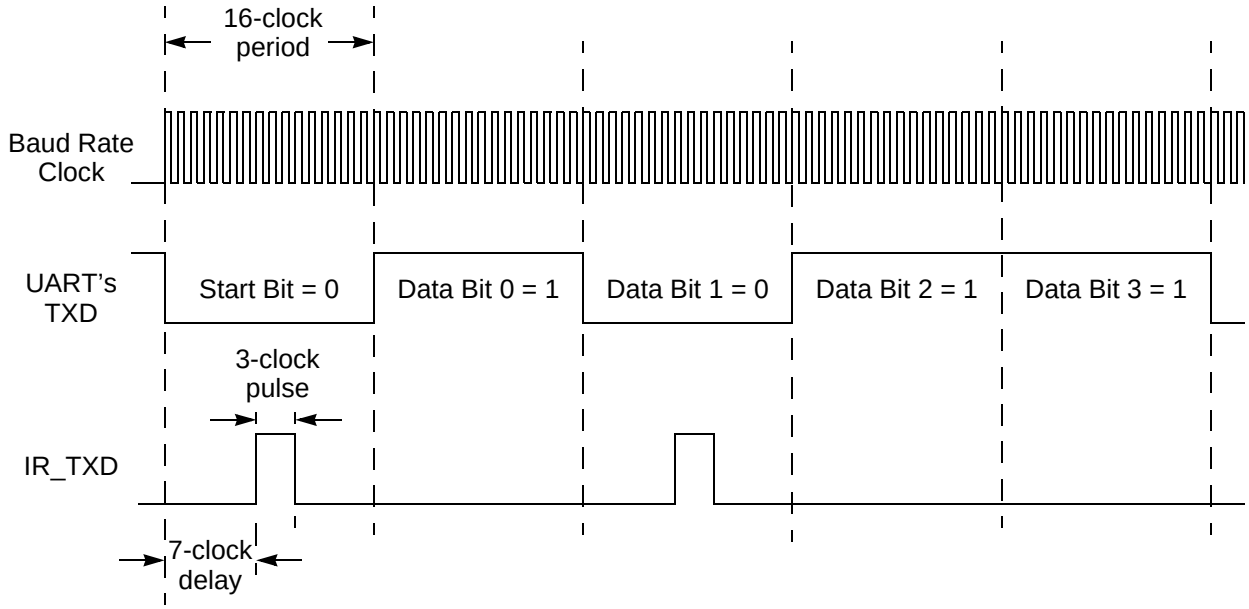

Figure 33. Infrared Data Transmission

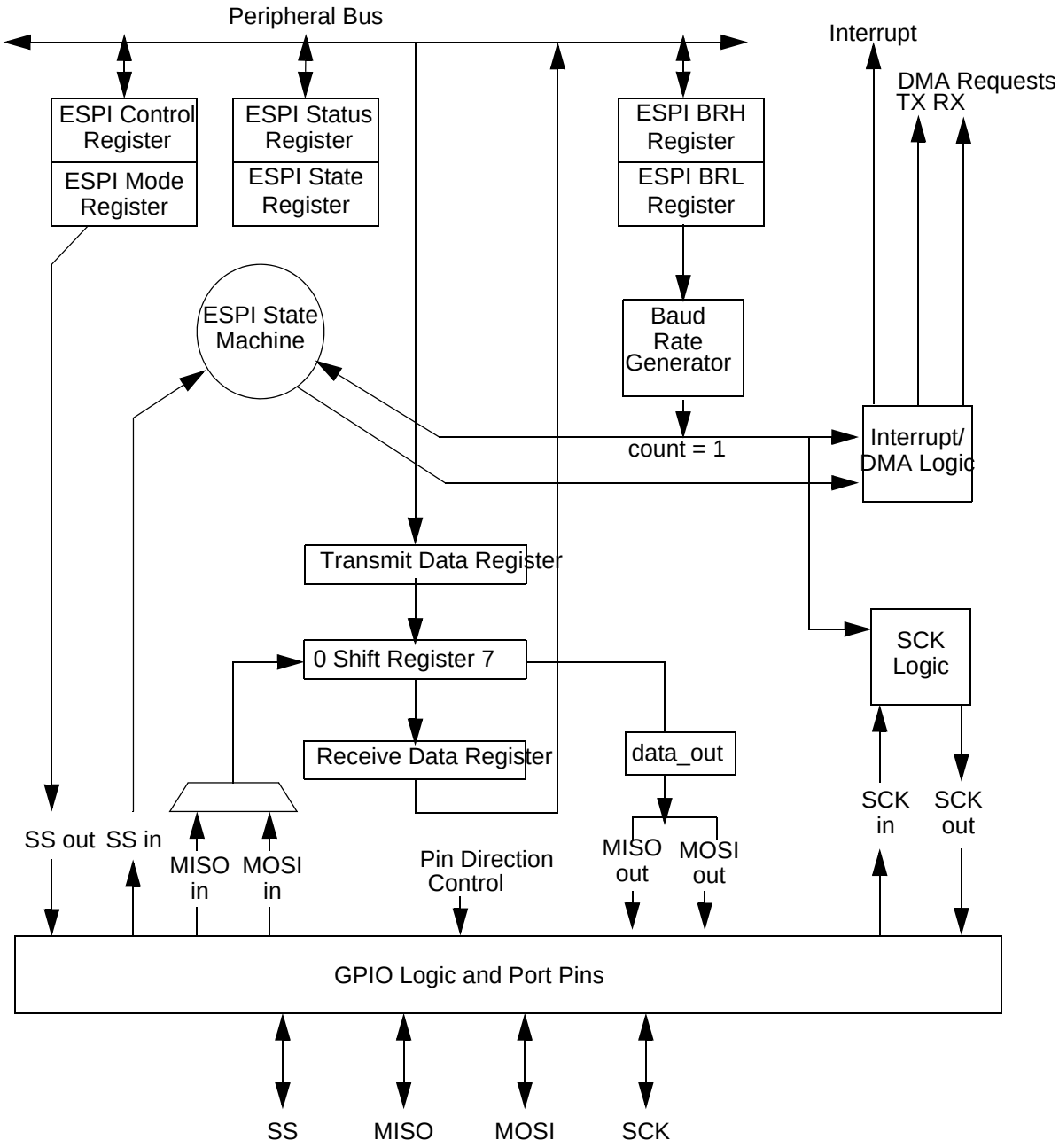

Figure 35. ESPI Block Diagram

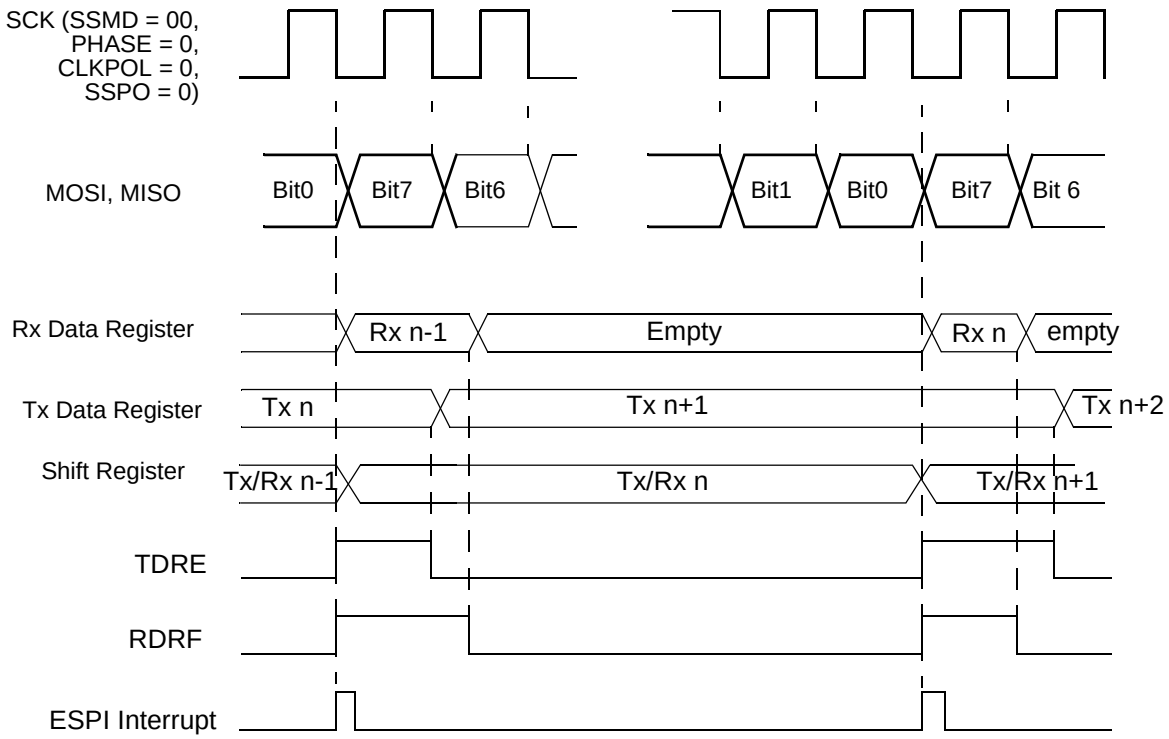

Figure 38. SPI Mode (SSMD = 000)

I2S (Inter-IC Sound) Mode

This mode is selected by setting the SSMD field of the mode register to 010. The PHASE and CLKPOL bits of the control register must be set to 0. This mode is illustrated in Figure 39 with $\overline{SS}$ alternating between consecutive frames. A frame consists of a fixed number of data bytes as defined in the DMA buffer descriptor or by software. I^2S (Inter-IC Sound) mode is typically used to transfer left or right channel audio data.

The SSV indicates whether the corresponding bytes are left or right channel data. The SSV value must be updated when servicing the TDRE interrupt/request for the first byte in a left or write channel frame. This update is accomplished by performing a word write when writing the first byte of the audio word, which updates both the ESPI data and transmit data command words or by doing a byte write to update SSV followed by a byte write to the data register. The $\overline{SS}$ signal leads the data by one SCK period.

If a DMA Channel is controlling data transfer, each sequence of left (or right) channel byte is considered a frame with a buffer descriptor. The SSV bit is defined in the buffer descriptor command field and is automatically written to the transmit data command register just prior to or in synchronous with the first data byte of the frame being written. Note that the

- Support for multi-master environments. If arbitration is lost when operating as a Master, the ARBLST bit in the I2CISTAT Register is set and the mode automatically switches to Slave Mode.

Operation

The I^2C Master/Slave Controller operates in either Slave Only Mode or Master/Slave Mode with Master arbitration. In Master/Slave Mode, it is used as the only Master on the bus or as one of several Masters on the bus with arbitration. In a multi-Master environment, the controller switches from Master to Slave Mode on losing arbitration.

Though slave operation is fully supported in Master/Slave Mode, if a device is intended to operate only as a slave, the Slave Only Mode is selected. In Slave Only Mode, the device does not initiate a transaction even if software inadvertently sets the Start bit.

SDA and SCL Signals

I^2C sends all addresses, data and acknowledge signals over the SDA line, the most-significant bit first. SCL is the clock for the I^2C bus. 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 is responsible for driving the SCL clock signal. During the Low period of the clock, a Slave holds the SCL signal Low to suspend the transaction if it is not ready to proceed. The Master releases the clock at the end of the Low period and notices that the clock remains Low instead of returning to a High level. When the Slave releases the clock, the I^2C Master 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 address, data or acknowledge, the SDA signal changes in the middle of the Low period of SCL. When receiving address, data, or acknowledge, the SDA signal is sampled in the middle of the High period of SCL.

A low-pass digital filter is applied to the SDA and SCL receive signals by setting the filter enable (FILTEN) bit in the I^2C Control Register. When the filter is enabled, any glitch, which is less than a system clock period in width is rejected. This filter must be enabled when running in I^2C Fast Mode (400 kbps) and is also used at lower data rates.

I^2C Interrupts

The I^2C Controller contains multiple interrupt sources that are combined into one interrupt request signal to the interrupt controller. If the I^2C Controller is enabled, the source of the interrupt is determined by bits, which are set in the I2CISTAT Register. If the I^2C Controller is disabled, the BRG Controller is used to generate general-purpose timer interrupts.

7. When the Master receives the data byte, the Master transmits an Acknowledge instruction (or Not Acknowledge instruction for the last data byte).
8. The bus cycles through steps 5–7 until the last byte has been transferred. If software has not yet loaded the next data byte when the Master brings SCL Low to transfer the most significant data bit, the Slave I^2C Controller holds SCL Low until the data register is written. When the Slave receives a Not Acknowledge instruction, the I^2C Controller sets the NCKI bit in the I2CISTAT Register and generates the Not Acknowledge interrupt.
9. Software responds to the Not Acknowledge interrupt by clearing the TXI bit in the I2CCTL Register and by asserting the FLUSH bit of the I2CCTL Register to empty the data register.
10. When the Master completes the last acknowledge cycle, it asserts the Stop or Restart condition on the bus.
11. The Slave I^2C Controller asserts the Stop/Restart interrupt (set the SPRS bit in I2CISTAT Register).
12. Software responds to the Stop/Restart interrupt by reading the I2CISTAT Register which clears the SPRS bit.

Slave Transmit (Master Read) Transaction with 10-Bit Address

Figure 51 displays the data transfer format for a Master reading data from a Slave with 10-Bit Addressing.

S	Slave Address 1st Byte	W=0	A	Slave Address 2nd Byte	A	S	Slave Address 1st Byte	R=1	A	Data	A	Data	A	P

Figure 51. Data Transfer Format, Slave Transmit Transaction with 10-Bit Addressing

The following procedure describes the I^2C Master/Slave Controller operating as a Slave in 10-Bit Addressing Mode, transmitting data to the bus Master:

1. Software configures the controller for operation as a Slave in 10-Bit Addressing Mode.
   - Initialize the MODE field in the I^2C Mode Register for either Slave Only Mode or Master/Slave Mode with 10-Bit Addressing.
   - Optionally set the GCE bit.
   - Initialize the SLA[7:0] bits in the I2CSLVAD Register and SLA[9:8] in the I2CMODE Register.
   - Set IEN = 1, NAK = 0 in the I^2C Control Register.
   - Program the Baud Rate High and Low Byte registers for the I^2C baud rate.

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.

Buffer Closure

A DMA buffer closure is requested in two ways. The first is when the transfer length reaches zero. The second is when the DMA receives a request end of frame from the peripheral. When either of these cases occur, the DMA begins closure of the buffer.

Loop Mode Closure

If the LOOP bit is set then the current buffer descriptor is not modified. The DMAxLAR increments or a new LAR value is fetched from the descriptor.

EOF Closure

The DMAxEN bit is reset to 0. If the EOF bit is set, the CMDSTAT field is set with the status data from the peripheral. If the channel is in LINKED LIST Mode then the DMAxCTL word is written back to the CONTROL word of the descriptor. The DMAxLAR increments or is loaded with new LAR data from the descriptor if the TXFR bit is set.

Normal Closure

The DMAxEN bit is reset to 0. If the channel is in LINKED LIST Mode then the DMAxCTL word is written back to the CONTROL word of the descriptor. The DMAxLAR increments or is loaded with new LAR data from the descriptor if the TXFR bit is set.

DMA Modes

Each DMA channel operates in two modes, direct and linked list. Both modes use the DMA Channel registers. The only difference is in how they are loaded. In DIRECT Mode, the DMA Channel registers are directly loaded by software and when the transfer is complete, the DMA stops. In LINKED LIST Mode, the DMA will load its own registers from a descriptor list which is pointed to by the DMAxLAR Register. It then loads the next descriptor in the list and continues executing.

The descriptor Control/Status field and address bytes maintain the same format as the control and address registers in the DMA.

Direct Mode

DIRECT Mode only uses the registers in the DMA for operation. The software writes these registers directly to set up and enable the DMA. DIRECT Mode is entered by directly setting the DMAxEN bit in the DMAxCTL0 Register.

Figure 58 displays the DMA registers and how they point to the buffers allocated in memory.

Program Memory Address 0002h

Option Bits in this space are altered to change the chip configuration at reset.

Table 163. Options Bits at Program Memory Address 0002h

Bits	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	Program Memory 0002h							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7:0]	**Reserved** These option bits are reserved for future use and must always be 1. This setting is the default for unprogrammed (erased) Flash.

Program Memory Address 0003h

Option bits in this space are altered to change the chip configuration at reset.

Table 164. Options Bits at Program Memory Address 0003h

Bits	7	6	5	4	3	2	1	0
Field	ROMLESS 16	LPOPT	Reserved					
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	Program Memory 0003h							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7] ROMLESS 16	**ROMLESS 16 Select** 0 = If the device is ROMLESS, the data bus is 8 bits wide and is on Port E[7:0]. 1 = If the device is ROMLESS, the data bus is 16 bits wide and is on {Port J[7:0], Port E[7:0]}

The UARTEN control bit must be set to 1 to use the serial interface as a UART. Clearing the UARTEN control bit to 0 will prevent data received on the DBG pin from being written to the Receive Data register. Clearing the UARTEN control bit to 0 also prevents data written to the Transmit Data register from being transmitted on the single pin interface.

If the UART is disabled, data is still written to the Receive Data register and read from the Transmit Data register. These actions still generates UART interrupts. The UARTEN control bit only prevents data from being transmitted to or received from the DBG pin.

Serial Errors

The serial interface detects the following error conditions:

- Receive framing error (received Stop bit is Low)
- Transmit collision (OCD releases the bus high to send a logic 1 and detects it is Low)
- Receive overrun (received data before previously received data read)
- Receive break detect (10 or more bits Low)

Transmission of data is prevented if the transmit collision, receive framing error, receive break detect, receive overrun, or Receive Data Register full status bits are set.

Interrupts

The Debug UART generates interrupts during the following conditions:

- Receive Data register is Full (includes Rx Framing Error and Rx Overrun Error)
- Transmit Data register is empty
- Auto-Baud Detector loads the BRG (auto-baud character received)
- Receive Break detected

DBG Pin as a GPIO Pin

The DBG pin is used as a GPIO pin. The serial interface cannot be used for debugging when the DBG pin is configured as a GPIO pin. To set up the DBG pin as a GPIO pin, software must clear the DBGUART option bit and OCDEN control bit.

Software uses the pin as an input by clearing the output enable control bit. The PIN status bit in Line Control Register (DBGLCR) reflects the state of the DBG pin.

The DBG pin is configured as an output pin by setting the output enable control bit. The logic state of the IDLE bit in Line Control Register is driven onto the DBG pin.

Table 182. Oscillator Configuration and Selection

Clock Source	Characteristics	Required Setup
Internal Precision Oscillator	• 5.5 MHz • High precision possible when trimmed • No external components required	• The reset default.
External Crystal/ Resonator/ External Clock Drive	• 0 to 20 MHz • Very high accuracy (dependent on crystal/resonator or external source) • Requires external components	• Configure Option Bits for correct external oscillator mode • Unlock and write Oscillator Control Register (OSCCTL) to enable external oscillator • Wait for required stabilization time • Unlock and write Oscillator Control Register (OSCCTL) to select external oscillator
Internal Watchdog Timer Oscillator	• 10 kHz nominal • Low accuracy • No external components required • Low power consumption	• Unlock and write Oscillator Control Register (OSCCTL) to enable and select Internal WDT oscillator

Unintentional access to the Oscillator Control Register (OSCCTL) stops the chip by switching to a nonfunctioning oscillator. Accidental alteration of the OSCCTL Register is prevented by a locking/unlocking scheme. To write the register, unlock it by making two writes to the OSCCTL Register with the values E7h followed by 18h. A third write to the OSCCTL Register then changes the value of the register and returns the register to a locked state. Any other sequence of oscillator control register writes has no effect. The values written to unlock the register must be ordered correctly, but is not required to be consecutive. It is possible to access other registers within the locking/unlocking operation.

Clock Selection Following System Reset

The internal precision oscillator is selected following a System Reset. Startup code after the System Reset changes the system clock source by unlocking and configuring the OSCCTL Register. If the LPOPT bit in Program Memory Address 0003h is zero, Flash Low Power Mode is enabled during reset. When Flash Low Power Mode is enabled during reset, the FLPEN bit in the Oscillator Control Register (OSCCTL) will be set and the DIV field of the OSCDIV Register will be set to 08h.

Oscillator Divide Register

The Oscillator Divide Register (OSCDIV), shown in Table 184, provides the value to divide the system clock by. The Oscillator Divide Register must be unlocked before writing. Writing the two-step sequence E7h followed by 18h to the Oscillator Control Register address unlocks it. The register locks after completion of a register write to the OSCDIV.

Table 184. Oscillator Divide Register (OSCDIV)

Bits	7	6	5	4	3	2	1	0
Field	DIV							
RESET	00h*							
R/W	R/W							
Addr	FF_E0A1h							

Note: *The reset value is 08h if the option bit LPOPT is 0.

Bit	Value (H)	Description
[7:0] DIV	00h–FFh	**Oscillator Divide** 00h–divider is disabled; all other entries are the divide value for scaling the system clock.