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	60
Program Memory Size	64KB (64K 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/z16f6411fi20sg

Introduction

Zilog's ZNEO® Z16F family of products are optimized for demanding applications. The ZNEO line of Zilog® microcontroller products is based on the ZNEO CPU.

Features

ZNEO family of products include the following features:

- 20 MHz ZNEO CPU
- 128 KB internal Flash memory with 16-bit access and In-Circuit Programming (ICP)
- 4 KB internal RAM with 16-bit access
- External interface allows seamless connection to external data memory and peripheral with:
  - Six chip selects with programmable wait states
  - 24-bit address bus supports 16 MB
  - Selectable 8-bit or 16-bit data bus widths
  - Programmable chip select signal polarity
  - ISA-compatible mode
- 12-channel, 10-bit Analog-to-Digital Converter (ADC)
- Operational Amplifier
- Analog Comparator
- 4-channel Direct Memory Access (DMA) controller supports internal or external DMA requests
- Two full-duplex 9-bit Universal Asynchronous Receiver/Transmitter (UARTs) with support for Local Interconnect Network (LIN) and Infrared Data Association (IrDA)
- Internal Precision Oscillator (IPO)
- Inter-Integrated Circuit (I^2C) master/slave controller
- Enhanced Serial Peripheral Interface (ESPI)
- 12-bit Pulse Width Modulation (PWM) module with three complementary pairs or six independent PWM outputs with deadband generation and fault trip input
- Three standard 16-bit timers with Capture, Compare and PWM capability
- Watchdog Timer (WDT) with internal RC oscillator

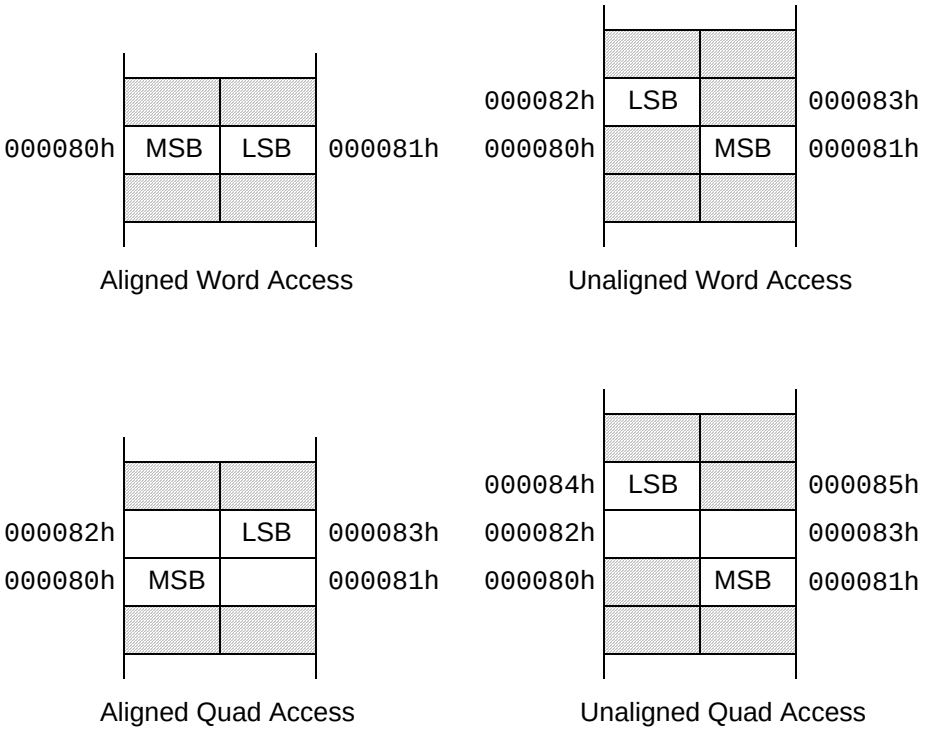

Figure 8. Alignment of Word and Quad Operations on 16-bit Memories

External Interface Timing

The following sections describe the ZNEO Z16F Series MCU's external interface timing.

External Interface Write Timing, Normal Mode

Figure 11 and Table 14 provide timing information for the external interface performing a write operation. In Figure 11, it is assumed that the wait state generator is configured to provide 1 wait state during write operations. The external $\overline{\text{WAIT}}$ input pin is generating an additional Wait period. It is assumed in Figure 11 that the chip select ($\overline{\text{CS}}$) signal has been configured for active Low operation. Though the internal system clock is not provided as an external signal, it provides a useful reference for control signal events.

Note: At the completion of a Write cycle, the deassertion of the $\overline{\text{WR}}$ signal is fed back from the pin and used on chip to control the deassertion of the data, $\overline{\text{CS}}$, address and byte enable signals to assure proper timing of the data hold.

Table 14. External Interface Timing for a Write Operation, Normal Mode

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_1	SYS CLK Rise to Address Valid Delay		10
T_2	$\overline{\text{WR}}$ Rise to Address Output Hold Time	3	
T_3	SYS CLK Rise to Data Valid Delay		10
T_4	$\overline{\text{WR}}$ Rise to Data Output Hold Time	3	
T_5	SYS CLK Rise to $\overline{\text{CS}}$ Assertion Delay		10
T_6	$\overline{\text{WR}}$ Rise to $\overline{\text{CS}}$ Deassertion Hold Time	3	
T_7	SYS CLK Rise to $\overline{\text{WR}}$ Assertion Delay		$1/2T_{CLK}$ +10
T_8	SYS CLK Rise to $\overline{\text{WR}}$ Deassertion Hold Time	3	
T_9	$\overline{\text{WAIT}}$ Input Pin Assertion to X_{IN} Rise Setup Time	1	
T_{10}	$\overline{\text{WAIT}}$ Input Pin Deassertion to X_{IN} Rise Setup Time	1	
T_{11}	SYS CLK Rise to $\overline{\text{DMAACK}}$ Assertion Delay		10
T_{12}	SYS CLK Rise to $\overline{\text{DMAACK}}$ Deassertion Hold Time	3	
T_{13}	SYS CLK Rise to $\overline{\text{BHEN}}$ or $\overline{\text{BLEN}}$ Assertion Delay		10
T_{14}	$\overline{\text{WR}}$ Rise to $\overline{\text{BHEN}}$ or $\overline{\text{BLEN}}$ Deassertion Hold Time	3	

Low-Power Modes

The ZNEO® Z16F Series products contain advanced integrated power-saving features. Power management functions are divided into three categories to include CPU operating modes, peripheral power control and programmable option bits. The highest level of power reduction is provided through a combination of all functions.

Stop Mode

Execution of the ZNEO CPU's Stop instruction places the device into Stop Mode. In Stop Mode, the operating characteristics are:

- IPO is stopped; X_{IN} and X_{OUT} pins are driven to V_{SS}.
- System clock is stopped.
- ZNEO CPU is stopped.
- Program counter (PC) stops incrementing.
- If enabled for operation during Stop Mode, the WDT and its internal RC oscillator continue to operate.
- If enabled for operation in Stop Mode through the associated option bit, the VBO protection circuit continues to operate.
- All other on-chip peripherals are nonactive.

To minimize current in Stop Mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails (V_{DD} or V_{SS}), the VBO protection must be disabled and WDT must be disabled. The device is brought out of Stop Mode using Stop Mode Recovery. For detailed information about Stop Mode Recovery, see the Reset and Stop Mode Recovery section on page 56.

! **Caution:** To prevent excess current consumption when using an external clock source in Stop Mode, the external clock must be disabled.

Port A-K Data Direction Registers

The Port A-K Data Direction registers, shown in Table 27, configure the specified port pins as either inputs or outputs.

Table 27. Port A-K Data Direction Registers (PxDD)

Bits	7	6	5	4	3	2	1	0
Field	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF_E102, FF_E112, FF_E122, FF_E132, FF_E142, FF_E152, FF_E162, FF_E172, FF_E182, FF_E192							

Bit	Description
[7:0] DD[7:0]	**Data Direction** These bits control the direction of the associated port pin. Port alternate function operation overrides the data direction register setting. 0 = Output Data in the Port A-K Output Data Register is driven onto the port pin. 1 = Input The port pin is sampled and the value written into the Port A-K Input Data Register. The output driver is high impedance.

Table 46. Interrupt Request 2 Register (IRQ2) and Interrupt Request 2 Set Register (IRQ2SET)

Bits	7	6	5	4	3	2	1	0
Field	PWMTI	U1RXI	U1TXI	PWMFI	PC3I/ DMA3I	PC2I/ DMA2I	PC1I/ DMA1I	PC0I/ DMA0I
RESET	0	0	0	0	0	0	0	0
R/W	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C	R/W1C
Addr	FF_E038h							
Field	PWMTI	U1RXI	U1TXI	PWMFI	PC3I/ DMA3I	PC2I/ DMA2I	PC1I/ DMA1I	PC0I/ DMA0I
RESET	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Addr	FF_E039h							

Note: IRQ2SET at address FF_E039h is write only and used to set the interrupts identified.

Bit	Description
[7] PWMTI	**PWM Timer Interrupt Request** 0 = No interrupt request is pending for the PWM timer. 1 = An interrupt request from the PWM timer is awaiting service. Writing 1 to this bit resets it to 0.
[6] U1RXI	**UART 1 Receiver Interrupt Request** 0 = No interrupt request is pending for the UART 1 receiver. 1 = An interrupt request from the UART 1 receiver is awaiting service. Writing 1 to this bit resets it to 0.
[5] U1TXI	**UART 1 Transmitter Interrupt Request** 0 = No interrupt request is pending for the UART 1 transmitter. 1 = An interrupt request from the UART 1 transmitter is awaiting service. Writing 1 to this bit resets it to 0.
[4] PWMFI	**PWM Fault Interrupt Request** 0 = No interrupt request is pending for the PWM fault. 1 = An interrupt request from the PWM fault is awaiting service. Writing 1 to this bit resets it to 0.
[3:0] PCxI/ DMAxI	**Port C Pin *x* or DMA x Interrupt Request** 0 = No interrupt request is pending for GPIO port C pin x or DMA x. 1 = An interrupt request from GPIO port C pin *x* or DMAx is awaiting service. Writing 1 to this bit resets it to 0. Where *x* indicates the specific GPIO port C pin or DMA number (0 through 3).

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.

byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits in the LIN-UART Status 0 Register.

In LIN Mode, an overrun error is signaled for receive data overruns as described above and in the LIN Slave, if the BRG counter overflows during the autobaud sequence (the ATB bit will also be set in this case). There is no data associated with the autobaud overflow interrupt, however the Receive Data Register must be read to clear the OE bit. In this case software must write `10b` to the LinState field, forcing the LIN slave back to Wait for Break state.

LIN-UART Data and Error Handling Procedure

Figure 29 displays the recommended procedure for use in LIN-UART receiver interrupt service routines.

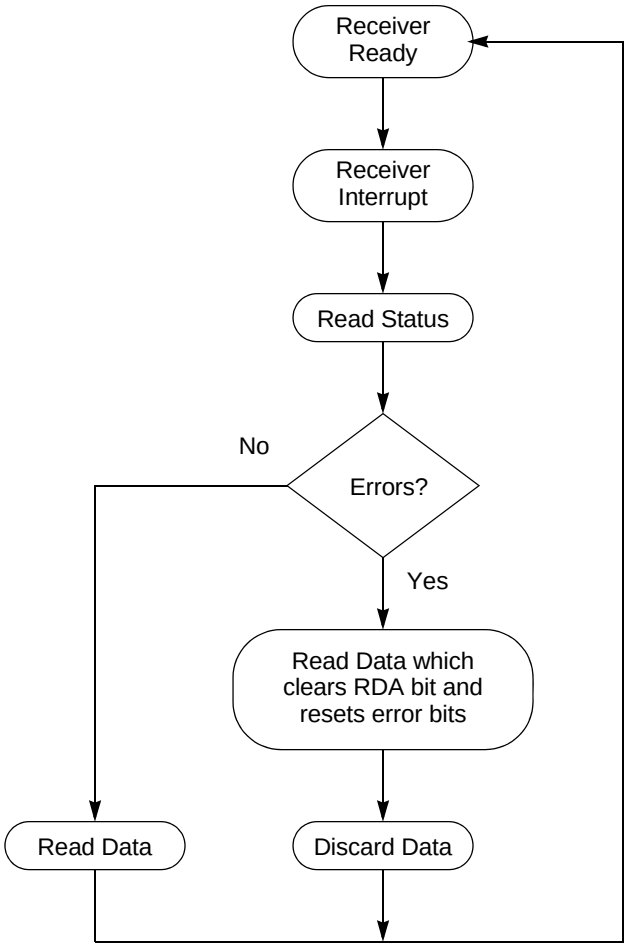

Figure 29. LIN-UART Receiver Interrupt Service Routine Flow

Table 80. LIN-UART Transmit Data Register (UxTXD)

Bits	7	6	5	4	3	2	1	0
Field	TXD							
RESET	X							
R/W	W							
Addr	FF–E200h, FF–E210h							

Bit	Description
[7:0] TXD	**Transmit Data** LIN-UART transmitter data byte to be shifted out through the TXD pin.

LIN-UART Receive Data Register

Data bytes received through the RXD pin are stored in the LIN-UART Receive Data Register, shown in Table 81. The Read-only LIN-UART Receive Data Register shares a register file address with the Write-only LIN-UART Transmit Data Register.

Table 81. LIN-UART Receive Data Register (UxRXD)

Bits	7	6	5	4	3	2	1	0
Field	RXD							
RESET	X							
R/W	R							
Addr	FF–E200h, FF–E210h							

Bit	Description
[7:0] RXD	**Receive Data** LIN-UART receiver data byte from the RXD pin

Table 95. LIN-UART Baud Rate Low Byte Register (UxBRL)

Bits	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Addr	FF–E207h, FF–E217h							

The LIN-UART data rate is calculated using the following equation for standard UART modes. For LIN protocol, the baud rate registers must be programmed with the baud period rather than 1/16 baud period.

Note: The UART must be disabled when updating the baud rate registers because High and Low registers must be written independently.

The LIN-UART data rate is calculated using the following equation for standard UART operation:

$$\text{UART Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

The LIN-UART data rate is calculated using the following equation for LIN Mode UART operation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{\text{UART Baud Rate Divisor Value}}$$

For a given LIN-UART data rate, the integer baud rate divisor value is calculated using the following equation for standard UART operation:

$$\text{UART Baud Rate Divisor Value (BRG)} = \text{Round}\left(\frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Data Rate (bits/s)}}\right)$$

For a given LIN-UART data rate, the integer baud rate divisor value is calculated using the following equation for LIN Mode UART operation:

$$\text{UART Baud Rate Divisor Value (BRG)} = \text{Round}\left(\frac{\text{System Clock Frequency (Hz)}}{\text{UART Data Rate (bits/s)}}\right)$$

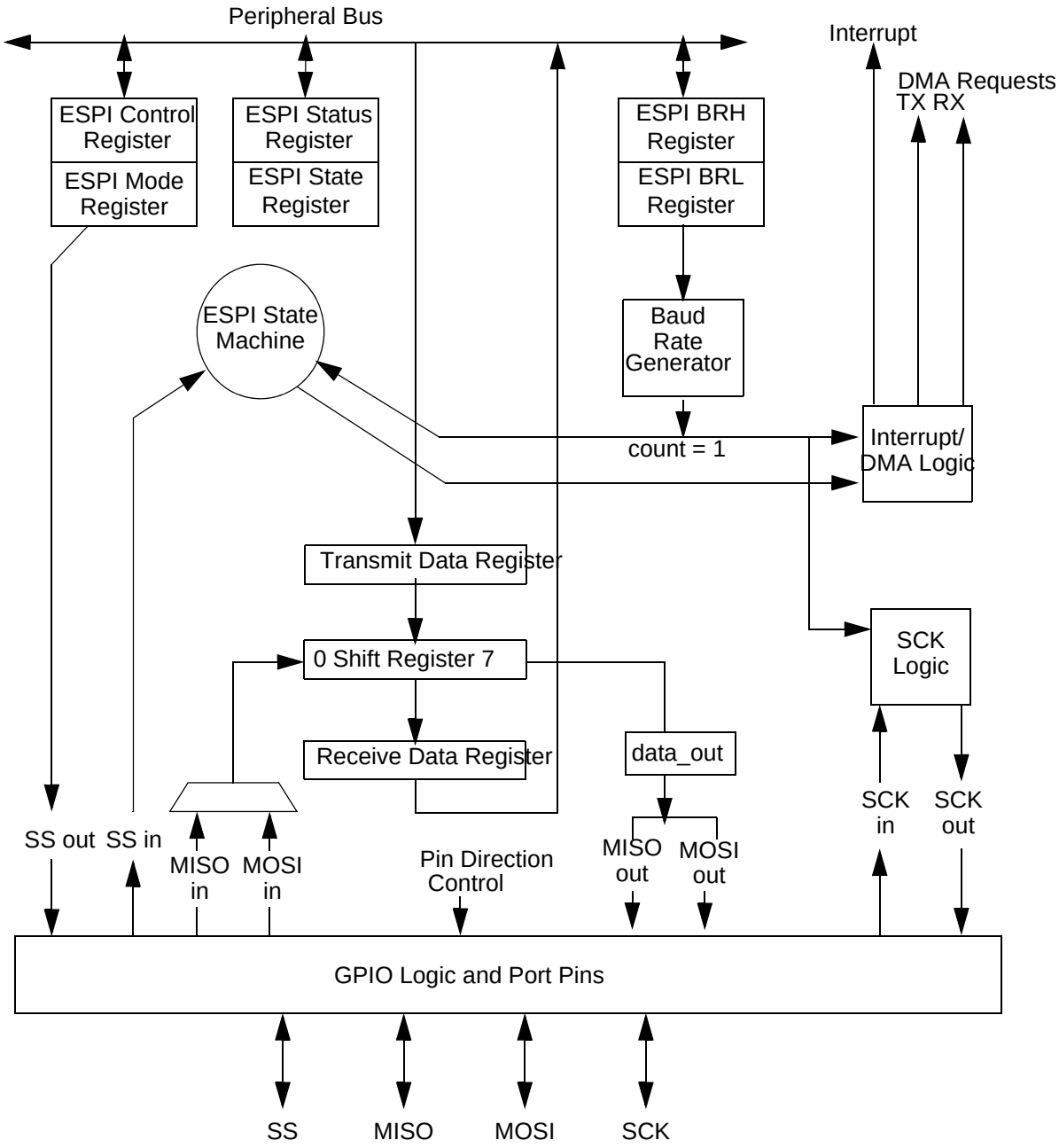

Figure 35. ESPI Block Diagram

number of bits per frame is a value other than an integral number of 8-bits by setting NUMBITS to a value other than 0.

Example. To send 20 bits/frame, set NUMBITS = 5 and read/write 4 bytes per frame. The transmit data must be left justified and the receive data must be right justified.

The transaction is terminated when the master has no more data to transmit. After the last bit is transferred, SCLK stops and $\overline{SS}$ and SSV returns to their default states. If TEOF is not set on the last byte, a transmit underrun error occurs at this point.

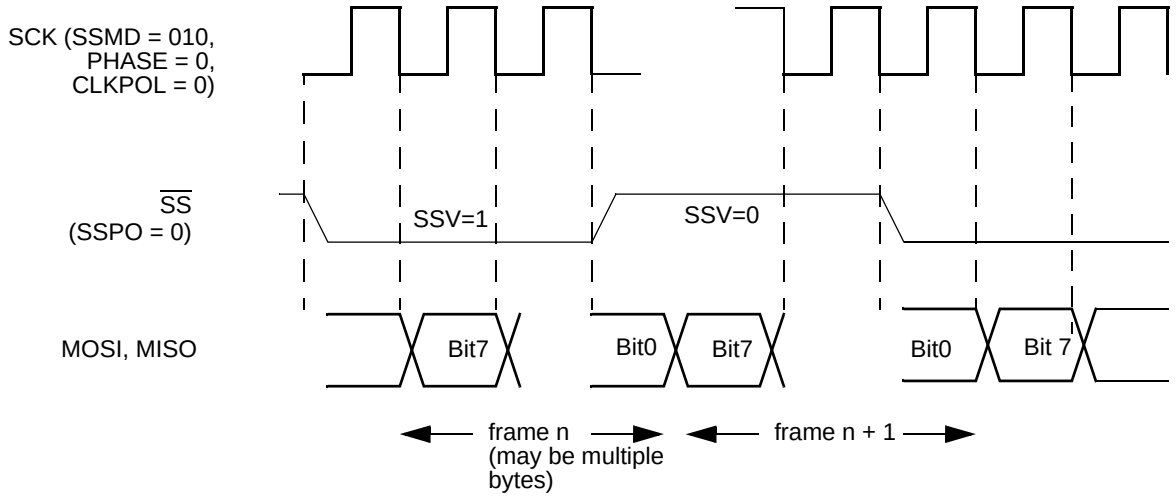

Figure 39. I²S Mode (SSMD = 010)

SPI Protocol Configuration

This section describes in detail how to configure the ESPI block for the SPI protocol. In the SPI protocol the master sources the SCK and asserts slave select signals to one or more slaves. The slave select signals are typically active Low.

SPI Master Operation

The ESPI block is configured for Master Mode operation by setting the MMEN bit = 1 in the ESPICTL Register. The SSMD field of the ESPI Mode register is set to 000 for SPI protocol mode. The PHASE, CLKPOL and WOR bits in the ESPICTL Register and the NUMBITS field in the ESPI Mode Register mustbe consistent with the Slave SPI devices. Typically for an SPI master SSIO = 1 and SSPO = 0. The appropriate GPIO pins are configured for the ESPI alternate function on the MOSI, MISO and SCK pins. The GPIO for the ESPI $\overline{SS}$ pin is configured in alternate function mode as well though software uses any GPIO pin(s) to drive one or more slave select lines. If the ESPI $\overline{SS}$ signal is not used to

first byte of the transaction is compared against {11110,SLA[9:8],R/$\overline{W}$} and the second byte is compared against SLA[7:0].

Arbitration Lost Interrupts

Arbitration Lost interrupts (ARBLST bit = 1 in I2CISTAT) occur when the I^2C Controller is in Master Mode and loses arbitration (outputs a 1 on SDA and receives a 0 on SDA). The I^2C Controller switches to Slave Mode when this occurs. This bit clears automatically when the I2CISTAT Register is read.

Stop/Restart Interrupts

A Stop/Restart event interrupt (SPRS bit = 1 in I2CISTAT) occurs when the I^2C Controller is in Slave Mode and a Stop or Restart condition is received, indicating the end of the transaction. The RSTR bit in the I^2C State Register indicates whether the bit was set due to a Stop or Restart condition. When a Restart occurs, a new transaction by the same Master is expected to follow. This bit is cleared automatically when the I2CISTAT Register is read. The Stop/Restart interrupt only occurs on a selected (address match) slave.

Not Acknowledge interrupts

Not Acknowledge interrupts (NCKI bit = 1 in I2CISTAT) occur in Master Mode when a Not Acknowledge is received or sent by the I^2C Controller and the Start or Stop bit is not set in the I^2C Control Register. In Master Mode the Not Acknowledge interrupt clears by setting the Start or Stop bit. When this interrupt occurs in Master Mode, the I^2C Controller waits until it is cleared before performing any action. In Slave Mode, the Not Acknowledge interrupt occurs when a Not Acknowledge is received in response to the data sent. The NCKI bit clears in Slave Mode when software reads the I2CISTAT Register.

General Purpose Timer Interrupt from Baud Rate Generator

If the I^2C Controller is disabled (IEN bit in the I2CCTL Register = 0) and the BIRQ bit in the I2CCTL Register = 1, an interrupt is generated when the BRG counts down to 1. The BRG reloads and continues counting, providing a periodic interrupt. None of the bits in the I2CISTAT Register are set, allowing the BRG in the I^2C Controller to be used as a general purpose timer when the I^2C Controller is disabled.

Start and Stop Conditions

The Master generates the Start and Stop conditions to start or end a transaction. To start a transaction, the I^2C Controller generates a Start condition by pulling the SDA signal Low while SCL is High. To complete a transaction, the I^2C Controller generates a Stop condition by creating a Low-to-High transition of the SDA signal while the SCL signal is High. The Start and Stop events occur when the Start and Stop bits in the I^2C Control Register are written by software to begin or end a transaction. Any byte transfer currently under way finishes, including the acknowledge phase before the Start or Stop condition occurs.

9. The Master sends the Stop or Restart signal on the bus. Either of these signals cause the I2C Controller to assert the Stop interrupt (Stop bit = 1 in the I2CISTAT Register). When the Slave receive data from the Master, software takes no action in response to the Stop interrupt other than reading the I2CISTAT Register, clearing the Stop bit.

Slave Transmit Transaction with 7-Bit Address

The data transfer format for a Master reading data from a Slave in 7-bit address Mode is shown in Figure 49. The following procedure describes the I2C Master/Slave Controller operating as a Slave in 7-Bit Addressing Mode, transmitting data to the bus Master.

S	Slave Address	R=1	A	Data	A	Data	A	P/S

Figure 50. Data Transfer Format, Slave Transmit Transaction with 7-Bit Addressing

1. Software configures the controller for operation as a Slave in 7-Bit Addressing Mode as follows.
   a. Initialize the MODE field in the I2C Mode Register for either Slave Only Mode or Master/Slave Mode with 7-Bit Addressing.
   b. Optionally set the GCE bit.
   c. Initialize the SLA[6:0] bits in the I2C Slave Address Register.
   d. Set IEN = 1 in the I2C Control Register. Set NAK = 0 in the I2C Control Register.
   e. Program the Baud Rate High and Low Byte registers for the I2C baud rate.

2. The Master initiates a transfer, sending the address byte. The Slave Mode I2C Controller finds an address match and detects the R/$\overline{W}$ bit = 1 (read by Master from Slave). The I2C Controller acknowledges, indicating that it is ready to accept the transaction.The SAM bit in the I2CISTAT Register is set = 1, causing an interrupt. The RD bit is set = 1, indicating a read from the Slave.
3. Software responds to the interrupt by reading the I2CISTAT Register, clearing the SAM bit. When RD = 1, software responds by loading the first data byte into the I2CDATA Register. Software sets the TXI bit in the I2CCTL Register to enable transmit interrupts. When the Master initiates the data transfer, the I2C Controller holds SCL Low until software has written the first data byte to the I2CDATA Register.
4. SCL is released and the first data byte is shifted out.
5. When the first bit of the first data byte is transferred, the I2C controller sets the TDRE bit, which asserts the transmit data interrupt.
6. Software responds to the transmit data interrupt (TDRE = 1) by loading the next data byte into the I2CDATA Register, which clears TDRE.

Bit	Description (Continued)
[1] SCLOUT	**Serial Clock Output** Current value of Serial Clock being output onto the bus. The actual values of the SCL and SDA signals on the I2C bus is observed via the GPIO Input Register.
[0] BUSY	**I2C Bus Busy** 0 = No activity on the I2C Bus. 1 = A transaction is underway on the I2C bus.

Table 118. I2C State Register (I2CSTATE), Description when DIAG = 1

Bits	7	6	5	4	3	2	1	0
Field	I2CSTATE_H				I2CSTATE_L			
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Addr	FF–E245h							

Bit	Description
[7:4] I2CSTATE_H	**I2C State High** This field defines the current state of the I2C Controller. It is the most significant nibble of the internal state machine. Table 119 defines the states for this field.
[3:0] I2CSTATE_L	**I2C State Low** Least significant nibble of the I2C state machine. This field defines the substates for the states defined by I2CSTATE_H. Table 120 defines the values for this field.

Table 119. I2CSTATE_H

State Encoding	State Name	State Description
0000	Idle	I2C bus is idle or I2C Controller is disabled.
0001	Slave Start	I2C Controller has received a start condition.
0010	Slave Bystander	Address did not match–ignore remainder of transaction.
0011	Slave Wait	Waiting for Stop or Restart condition after sending a Not Acknowledge instruction.
0100	Master Stop2	Master completing Stop condition (SCL = 1, SDA = 1).
0101	Master Start/Restart	Master Mode sending Start condition (SCL = 1, SDA = 0).
0110	Master Stop1	Master initiating Stop condition (SCL = 1, SDA = 0).

The output of the operational amplifier is also connected to an analog input (ANA3) of the ADC multiplexer.

The operational amplifier does not automatically power-down. To reduce operating current when not in use, the operational amplifier is disabled by clearing the OPEN bit in the comparator and op-amp register to 0.

When the operational amplifier is disabled, the output is high impedance.

Interrupts

The comparator generates an interrupt on any change in the logic output value (from 0 to 1 and from 1 to 0). For information about enabling and prioritization of the comparator interrupt, see the Interrupt Controller chapter on page 80.

Comparator Control Register Definitions

The following sections describe the comparator control registers.

Comparator and Operational Amplifier Control Register

The Comparator and Operational Amplifier Control Register (CMPOPC), shown in Table 135, enables the comparator and operational amplifier and provides access to the comparator output.

Table 135. Comparator and Op Amp Control Register (CMPOPC)

Bits	7	6	5	4	3	2	1	0
Field	OPEN	Reserved		CPISEL	CMPIRQ	CMPIV	CMPOUT	CMPEN
RESET	0	00		0	0	0	X	0
R/W	R/W	R		R/W	R/W	R/W	R	R/W
Addr	FF_E510h							

Bit	Description
[7] OPEN	**Operational Amplifier Disable** 0 = Operational amplifier is disabled. 1 = Operational amplifier is enabled.
[6:5]	**Reserved** These bits are reserved and must be programmed to 00.

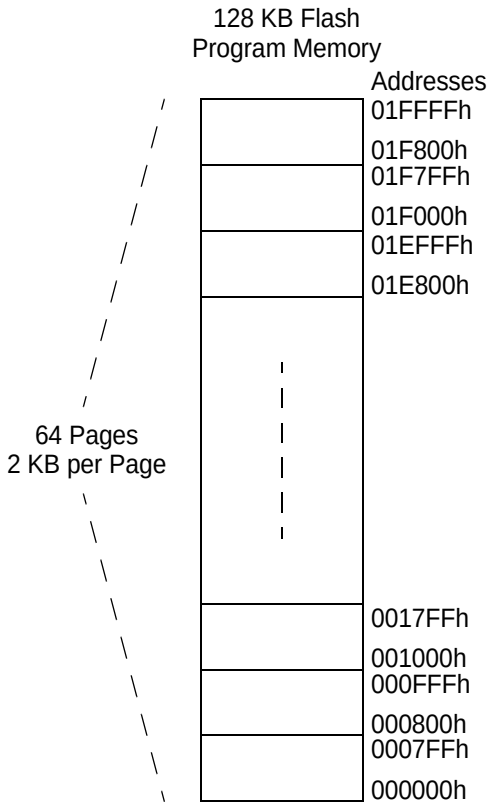

Figure 55. Flash Memory Arrangement

Information Area

Table 138 describes the ZNEO Z16F Series Information Area. This 128-byte Information Area is accessed by setting bit 7 of the Flash Control register to 1. When access is enabled, the Information Area is mapped into program memory and overlays the 128 bytes at addresses 000000h to 00007Fh. When the Information Area access is enabled, instructions access data from the Information Area. The CPU instruction fetches always come from Main Memory regardless of the Information Area access bit. Access to the Information Area is read-only.

Flash Status Register

The Flash Status Register, shown in Table 140, indicates the current state of the Flash Controller. This register is read at any time. The Read-only Flash Status Register shares its address with the Write-only Flash Command Register.

Table 140. Flash Status Register (FSTAT)

Bits	7	6	5	4	3	2	1	0
Field	UNLOCK	Reserved	FSTAT					
RESET	0	0	00h					
R/W	R	R	R					
Addr	FF_E060h							

Bit	Description
[7] UNLOCK	**Unlocked** This status bit is set when the Flash Controller is unlocked. 0 = Flash Controller locked. 1 = Flash Controller unlocked.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5:0] FSTAT	**Flash Controller Status** 00_0000 = Flash Controller idle. 00_1xxx = Program operation in progress. 01_0xxx = Page erase operation in progress. 10_0xxx = Mass erase operation in progress.

DMA List Address Register

The DMA List Address Register is written when the list mode for the DMA is used. This register contains the address of the current list the DMA is operating on. Writing the DMAxLARL Register (shown in Table 160) enables the DMA for list operation.

Table 158. DMA X List Address Register Upper DMAxLARU

Bits	7	6	5	4	3	2	1	0
Field	DMAxLARU							
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	FFE41Dh, FFE42Dh, FFE43Dh, FFE44Dh							

In DIRECT Mode, this register is used to set a watermark interrupt. This interrupt occurs when the DMATXLN[15:8] equals 0 and DMAxTXLN[7:0] equals DMAxLARU. Note when using the watermark the DMAxLARL must not be written.

Table 159. DMA X List Address Register High (DMAxLARH)

Bits	7	6	5	4	3	2	1	0
Field	DMAxLARH							
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	FFE41Eh, FFE42Eh, FFE43Eh, FFE44Eh							

Table 160. DMA X List Address Register Low (DMAxLARL)

Bits	7	6	5	4	3	2	1	0
Field	DMAxLARL							
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	FFE41Fh, FFE42Fh, FFE43Fh, FFE44Fh							

Writing to the DMAxLARL Register causes the DMA to enter LINKED LIST Mode.

Bit	Description (Continued)
[5:4]	**Reserved** These bits are reserved and must be programmed to 00.
[3] CRCEN	**CRC Enable** If this bit is set, a CRC is appended to the end of each debug command. Clearing this bit will disable transmission of the CRC. 0 = CRC disabled 1 = CRC enabled
[2] UARTEN	**UART Enable** This bit is used to enable or disable the UART. This bit is ignored when OCDEN is set. 0 = UART Disabled. 1 = UART Enabled.
[1] ABCHAR	**Auto-Baud Character** This bit selects the character used during auto-baud detection. This bit cannot be written by the CPU if OCDEN is set. 0 = Auto-baud character to be measured is 80h. 1 = Auto-baud character to be measured is 0Dh.
[0] ABSRCH	**Auto-Baud Search Mode** This bit enables auto-baud search mode. When this bit is set, the next character received is measured to set the Baud Rate Reload register. This bit clears itself to 0 after the reload register has been written. This bit is automatically set when OCDEN is set if a serial communication error occurs. This bit cannot be written by the CPU if the OCDEN bit is set. 0 = Auto-baud search disabled. 1 = Auto-baud search enabled.