Zilog - Z16F2810FI20EG 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f2810fi20eg

Signal and Pin Descriptions

The ZNEO® Z16F Series products are available in various package styles and pin configurations. This chapter describes the signals and available pin configurations for each package style. For more information about the physical package specifications, see the Packaging chapter on page 356.

Available Packages

Table 1 lists the package styles available for each device within the ZNEO Z16F Series product line.

Table 1. ZNEO Z16F Series Package Options

Part Number	64-pin LQFP	68-pin PLCC	80-Pin QFP	100-pin LQFP
Z16F2811			X	X
Z16F2810*	X	X	X	
Z16F6411			X	X
Z16F3211			X	X

Note: *The Z16F2810 MCU does not feature an external bus interface.

Pin Configurations

Figures 2 through 5 display the configurations of all of the packages available in the ZNEO Z16F Series. For description of each signal, see Table 2 on page 12.

Table 2. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
UART Controllers		
TXD0/TXD1	O	Transmit data: These signals transmit outputs from the UARTs.
RXD0/RXD1	I	Receive data: These signals receives inputs for the UARTs and IrDAs.
$\overline{\text{CTS0}}$/$\overline{\text{CTS1}}$	I	Clear to Send: These signals are control inputs for the UARTs.
DE0/DE1	O	Driver Enable (DE): This signal allows automatic control of external RS-485 drivers. This signal is approximately the inverse of the Transmit Empty (TXE) bit in the UART Status 0 Register. The DE signal is used to ensure an external RS-485 driver is enabled when data is transmitted by the UART.
General-Purpose Timers		
T0OUT/$\overline{\text{T0OUT}}$ T1OUT/$\overline{\text{T1OUT}}$ T2OUT/$\overline{\text{T2OUT}}$	O	General-purpose timer outputs: These signals are output pins from the timers.
T0IN/T0IN1/T0IN2 /T1IN/T2IN	I	General-purpose timer inputs: These signals are used as the capture, gating and counter inputs.
Pulse-Width Modulator for Motor Control		
PWMH0/PWMH1/ PWMH2	O	PWM High output.
PWML0/PWML1/ PWML2	O	PWM Low output.
FAULT0/FAULTY	I	PWM Fault condition input: FAULT0 and FAULTY are active Low.
Analog		
ANA[11:0]	I	Analog input: These signals are inputs to the ADC.
VREF	I	ADC reference voltage input or internal reference output: The VREF pin must be capacitively coupled to analog ground, if the internal voltage reference is selected as the ADC reference voltage. A 10 mF capacitor is recommended.
CINP	I	Comparator positive input
CINN	I	Comparator negative input
COMPOUT	O	Comparator output
OPINP	I	Operational amplifier positive input
OPINN	I	Operational amplifier negative input
OPOUT	O	Operational amplifier output

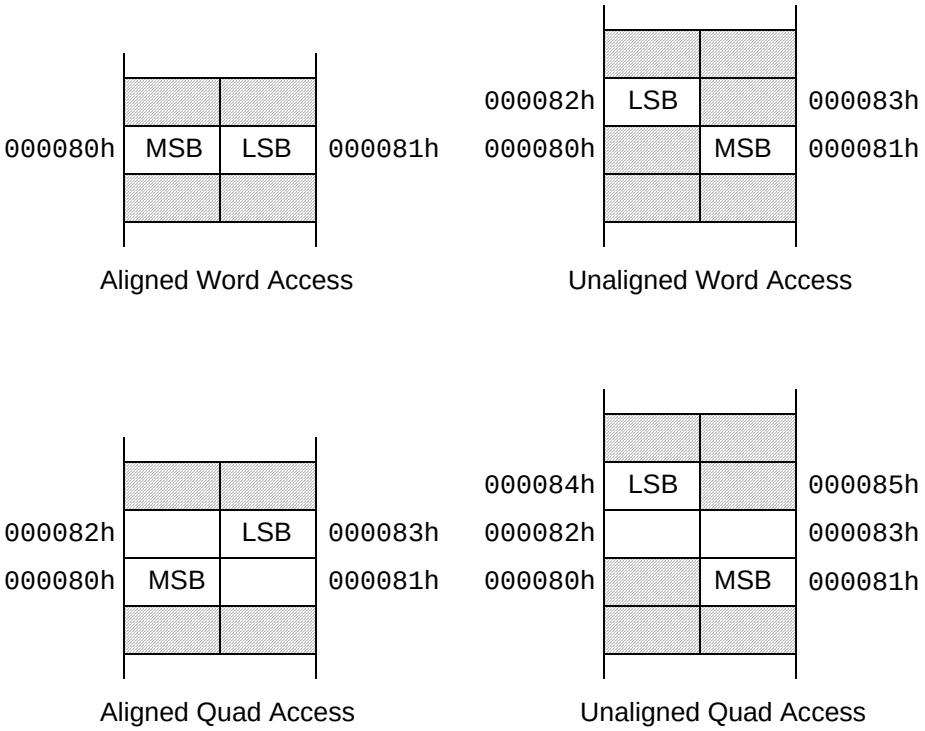

Figure 8. Alignment of Word and Quad Operations on 16-bit Memories

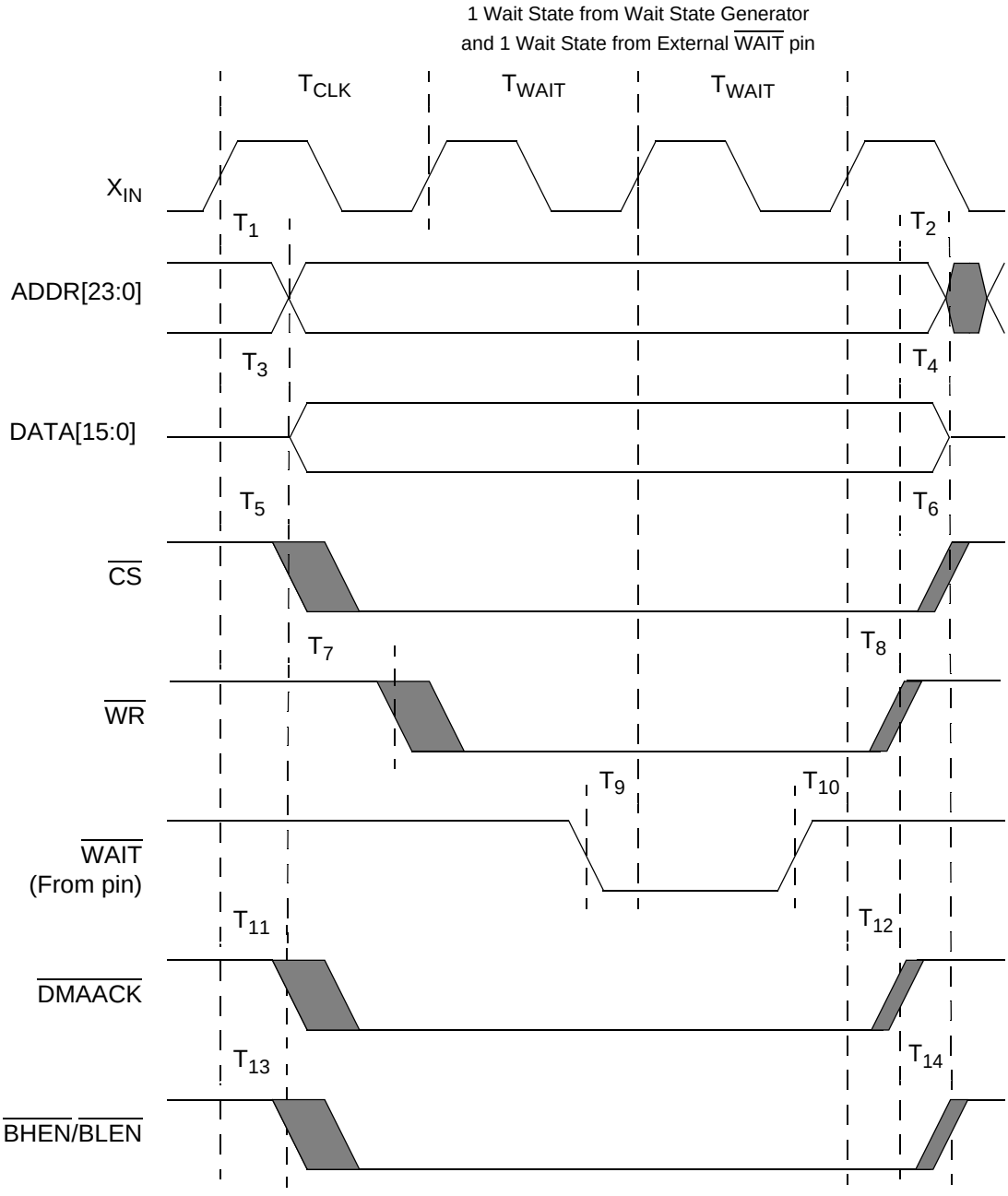

Figure 11. External Interface Timing for a Write Operation, Normal Mode

Reset on page 57. Following Power-On Reset, the POR status bit in the reset source register is set to 1. Figure 17 displays Voltage Brown-Out operation. For VBO and POR threshold voltages (V_{VBO} and V_{POR}), see Figure 75 on page 343.

The VBO circuit is either enabled or disabled during Stop Mode. Operation during Stop Mode is controlled by the VBO_AO option bit. For information about configuring VBO_AO, see the Option Bits chapter on page 292.

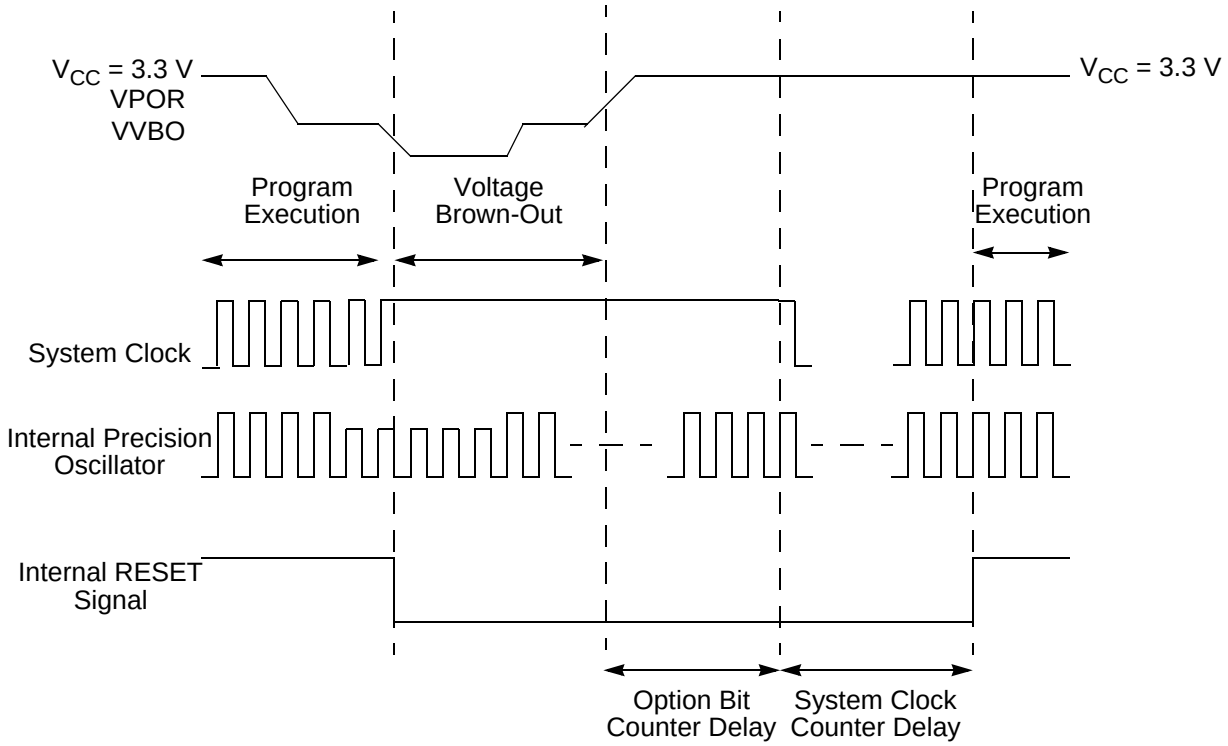

Figure 17. Voltage Brown-Out Reset Operation

Watchdog Timer Reset

If the device is in NORMAL or Halt Mode, the WDT initiates a System Reset at time-out if the WDT_RES option bit is set to 1. This setting is the default (unprogrammed) setting of the WDT_RES option bit. The WDT status bit in the Reset Status and Control Register is set to signify that the reset was initiated by the WDT.

External Pin Reset

The input-only $\overline{\text{RESET}}$ pin has a schmitt-triggered input, an internal pull-up, an analog filter and a digital filter to reject noise. After the $\overline{\text{RESET}}$ pin is asserted for at least four sys-

changes the state of the IRQE bit. For the IRET instruction, the bit is set based on what has been pushed on the stack.

Interrupts are globally enabled by any of the following actions:

- Execution of an Enable Interrupt (EI) instruction
- Writing 1 to the IRQE bit in the flag register

Interrupts are globally disabled by any of the following actions:

- Execution of a Disable Interrupt (DI) instruction
- ZNEO CPU acknowledgement of an interrupt service request from the interrupt controller
- Writing 0 to the IRQE bit in the flag register
- Reset
- Execution of a TRAP instruction
- All System Exceptions

Interrupt Vectors and Priority

The interrupt controller supports three levels of interrupt priority. Level 3 is the highest priority, Level 2 is the second highest priority and Level 1 is the lowest priority. If all of the interrupts are enabled with identical interrupt priority (for example, all interrupts enabled as Level 2 interrupts), the interrupt priority is assigned from highest to lowest as specified in Table 39 on page 81. Level 3 interrupts always have higher priority than Level 2 interrupts, which in turn, always have higher priority than Level 1 interrupts. Within each interrupt priority levels (Level 1, Level 2 or Level 3), priority is assigned as specified in Table 39. Reset and System Exceptions have the highest priority.

System Exceptions

System Exceptions are generated for stack overflow, illegal instructions, divide-by-zero and divide overflow, etc. The System Exceptions are not affected by the IRQE and share a single vector.

Each exception has a bit in the system exception status register. When a system exception occurs it pushes the program counter and the flags on the stack, fetches the system exception vector from 000008h (similar to a IRQ) and the bit associated with that exception is set in the status register. Additional exceptions from the same source are blocked until the status bit of the particular exception is cleared by writing 1 tothat status bit. Other types of exceptions occur while servicing an exception. When this happens the processor again

mine the active time of a PWM output. The following sections describe the PWM TIMER modes and the registers controlling the duty-cycle and deadband time.

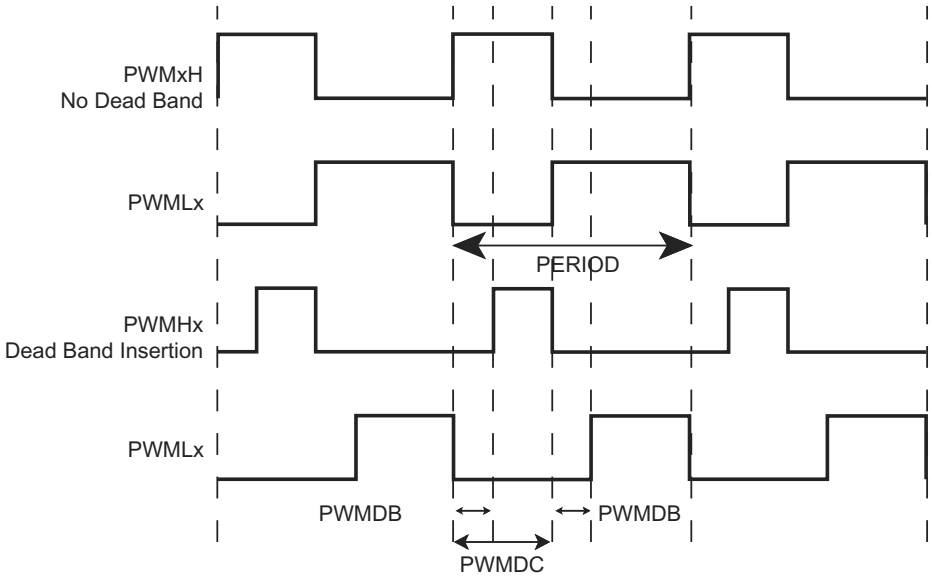

Figure 22. Edge-Aligned PWM Output

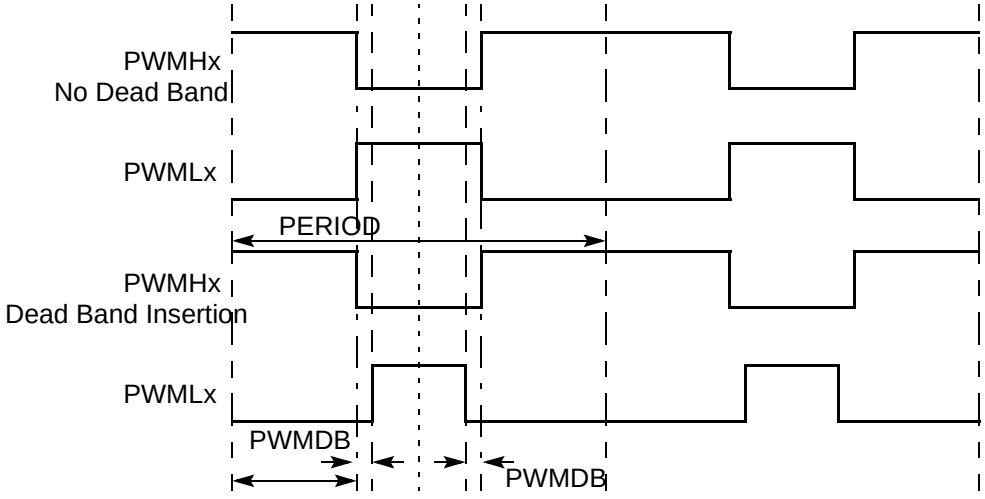

Figure 23. Center-Aligned PWM Output

Manual Off-State Control of PWM Output Channels

Each PWM output is controlled directly by the modulator logic or set to the off-state. To manually set the PWM output to the off-state, set the OUTCTL bit and the associated OUTx bits in the PWM Output Control Register (PWMOUT). Off-state control operates individually by channel. For example, suppressing a single output of pair allows the complementary channel to continue operating. Similarly, if the outputs are operating independently disabling one output channel has no effect on the other PWM outputs.

Deadband Insertion

When the PWM outputs are configured to operate as complementary pairs, an 8-bit deadband value is defined in the PWM Deadband Register (PWMDB). Inserting deadband time causes the modulator to separate the deassertion of one PWM signal from the assertion of its complement. This separation is essential for many motor control applications to prevent simultaneous turn-on of the high-side and low-side drive transistors. The deadband counter directly counts system clock cycles and is unaffected by PWM prescaler settings. The width of this deadband is the number of system clock cycles specified in the PWM Deadband Register (PWMDB).

The minimum deadband duration is zero system clocks; the maximum time is 255 system clocks. Both PWM outputs of a complementary pair is deasserted during the deadband period. Generation of deadband time does not alter the PWM period but the deadband time is subtracted from the active time of the PWM outputs. Figure 22 on page 117 displays the effect of deadband insertion on the PWM output.

Minimum PWM Pulse Width Filter

The PWM modulator is capable of producing pulses as narrow as a single system clock cycle in width. The response time of external drive circuit is slower than the period of a system clock. Therefore, a filter is implemented to enforce a minimum width pulse on the PWM output pins. All output pulses, either High or Low, must be at least the minimum number of PWM clock cycles in width, as specified in the PWM Minimum Pulse Width Filter (PWMMPF) Register (for more details, see the PWM Prescaler section on page 116).

If the expected pulse width is less than the threshold, the associated PWM output does not change state until the duty cycle value has changed sufficiently to allow pulse generation of an acceptable width. The minimum pulse width filter also accounts for the duty cycle variation caused by the deadband insertion. The PWM output pulse is filtered even if the programmed duty cycle is greater than the threshold but the decrease in pulse width because of deadband insertion causes the pulse to be too narrow. The pulse width filter value is calculated as:

PWM Fault Control Register

The PWM Fault Control (PWMFCTL) Register, shown in Table 76, determines how the PWM recovers from a fault condition. Settings in this register select either an automatic or a software-controlled PWM restart.

Table 76. PWM Fault Control Register (PWMFCTL)

Bits	7	6	5	4	3	2	1	0
Field	Reserved	DBGRST	CMP1INT	CMP1RST	CMPINT	CMPRST	Fault0INT	Fault0RST
RESET	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	FF_E388h							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] DBGRST	**Debug Restart** 0 = Automatic recovery. PWM resumes control of outputs when all fault sources have deasstered and a new PWM period begins. 1 = Software controlled recovery. PWM resumes control of outputs only after all fault sources have deasserted and all fault flags are cleared and a PWM reload occurs.
[5] CMP1INT	**Comparator 1 Interrupt** 0 = Interrupt on comparator assertion disabled. 1 = Interrupt on comparator assertion enabled.
[4] CMP1RST	**Comparator 1 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.
[3 CMP0INT	**Comparator 0 Interrupt** 0 = Interrupt on comparator 0 assertion disabled. 1 = Interrupt on comparator 0 assertion enabled.
[2] CMP0RST	**Comparator 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.

Bit	Description (Continued)
[2] SBRK	**Send Break** This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has finished sending data before setting this bit. In standard UART Mode, the duration of the break is determined by how long software leaves this bit asserted. Also the duration of any required Stop bits following the break must be timed by software before writing a new byte to be transmitted to the Transmit Data Register. In LIN Mode, the master sends a Break character by asserting SBRK. The duration of the break is timed by hardware and the SBRK bit is deasserted by hardware when the Break is completed. The duration of the Break is determined by the TxBreakLength field of the LIN Control Register. One or two Stop bits are automatically provided by the hardware in LIN Mode as defined by the Stop bit. 0 = No break is sent. 1 = The output of the transmitter is 0.
[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 within the IrDA module.

Enhanced Serial Peripheral Interface

The Enhanced Serial Peripheral Interface (ESPI) supports SPI (Serial Peripheral Interface) and Inter IC Sound (I^2S) modes of operation.

The features of the ESPI include:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface ($\overline{SS}$, SCK, MOSI, MISO)
- Transmit and receive buffer registers to enable high throughput
- Transfer rates up to maximum of one-fourth the system clock frequency in Slave Mode
- Error detection
- Dedicated programmable baud rate generator (BRG)
- Data transfer control through polling, interrupt, or DMA

Architecture

The ESPI is a full-duplex, synchronous, character-oriented channel that supporting a four-wire interface (serial clock, transmit and receive data and Slave select). The ESPI block consists of a shift register, transmit and receive data buffer registers, a baud rate (clock) generator, control/status registers and a control state machine. Transmit and receive transfers are in sync as there is a single shift register for both transmit and receive data. Figure 35 displays a block diagram of the ESPI.

Multi-Master SPI Operation

In a multi-master SPI system, all SCK pins are tied together, all MOSI pins are tied together and all MISO pins are tied together. All SPI pins must be configured in open-drain mode to prevent bus contention. At any time, only one SPI device is configured as the master and all other devices on the bus are configured as slaves. The master asserts the $\overline{SS}$ pin on the selected slave. Then, the active master drives the clock and transmit data on the SCK and MOSI pins to the SCK and MOSI pins on the slave (including those slaves which are not enabled). The enabled slave drives data out its MISO pin to the MISO master pin.

When the ESPI is configured as a master in a multi-master SPI system, the $\overline{SS}$ pin must be configured as an input. The $\overline{SS}$ input signal on a device configured as a master must remain High. If the $\overline{SS}$ signal on the active master goes Low (indicating another master is accessing this device as a slave), a collision error flag is set in the ESPI Status Register. The slave select outputs on a master in a multi-master system must come from GPIO pins.

SPI Slave Operation

The ESPI block is configured for Slave Mode operation by setting the MMEN bit = 0 in the ESPICTL Register and setting the SSIO bit = 0 in the ESPIMODE Register. The SSMD field of the ESPI Mode Register is set to 00 for SPI protocol mode. The PHASE, CLKPOL and WOR bits in the ESPICTL Register and the NUMBITS field in the ESPIMODE Register must be set to be consistent with the other SPI devices. Typically for an SPI slave SSPO = 0.

If the slave has data to send to the master, the data must be written to the data register before the transaction starts (first edge of SCK when $\overline{SS}$ is asserted). If the data register is not written prior to the slave transaction, the MISO pin outputs all 1s.

Due to the delay resulting from synchronization of the $\overline{SS}$ and SCK input signals to the internal system clock, the maximum SCK baud rate which is supported in Slave Mode is the system clock frequency divided by 8. This rate is controlled by the SPI master.

Figure 42 displays the ESPI configuration in SPI Slave Mode.

ESPI error interrupts occur if any of the TUND, COL, ABT and ROVR bits in the ESPI Status register are set. These bits are cleared by writing a 1 to the corresponding bit.

If the ESPI is disabled (ESPIEN1,0 = 00), an ESPI interrupt is generated by a BRG time-out. This timer function must be enabled by setting the BRGCTL bit in the ESPICTL Register. This timer interrupt does not set any of the bits of the ESPI Status register.

DMA Interface

The assertion of the TDRE and RDRF signals generate transmit and receive DMA requests (SPITxReq, SPIRxReq), allowing data movement to be handled by a DMA Controller rather than directly by software. The DMA acknowledges these requests through the SPITxAck and SPIRxAck signals). Inputs allow the SSV and TEOF bits of the Transmit Data Command register to be controlled by the DMA. The SPITxReqEOF and SPIRxReqEOF outputs to the DMA provides an indication that $\overline{SS}$ has deasserted (transaction complete).

If the software application is moving data in only one direction, the ESPIEN1,0 bits are set to 10 or 01, allowing a single DMA Channel to control the ESPI data transfer. For a master, the valid options are transmit only or transmit-receive. For a slave, all options are valid. When a slave is operating in Receive Only Mode, it will transmit characters of all 1s.

DMA Descriptors

For ESPI Transmit DMA descriptors, the 4-bit CMDSTAT field of the descriptor is in the format shown in Table 99. The SSV bit in the Master's transmit buffer descriptor CMDSTAT field controls the ESPI $\overline{SS}$ output. The SSV bit in the descriptor is transferred to the SSV bit in the ESPI Data Command register with the first byte of the buffer. If the EOF bit is set in the DMA descriptor control word, the end of frame signal from the DMA (EOFSync) will assert coincident with writing the last byte in the buffer to the ESPI Data register, setting the TEOF bit of the ESPI Data Command register. After this last byte has been transferred, the Master's $\overline{SS}$ output will deassert and the SSV and TEOF bits in the Data Command register will be cleared. The CMDSTAT field in ESPI Receive DMA Descriptors has no function.

Table 99. ESPI Tx DMA Descriptor Command Field

Reserved	Reserved	Reserved	SSV

For ESPI DMA descriptors, the 4-bit frame status field of the descriptor has the format shown in Tables 100 and 101.

Table 100. ESPI Tx DMA Descriptor Status Field

0	0	COL	TUND

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.

Flash Sector Protect Register

The Flash Sector Protect Register, shown in Table 142, protects Flash memory sectors from being programmed or erased from user code. User code can only write bits in this register to 1 (bits cannot be cleared to 0 by user code).

Table 142. Flash Sector Protect Register (FSECT)

Bits	7	6	5	4	3	2	1	0
Field	SECT7	SECT6	SECT5	SECT4	SECT3	SECT2	SECT1	SECT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Addr	FF_E062h							

Note: R/W1 = Register is accessible for read operations. Register is written to 1 only (via user code).

Bit	Description
[7:0] SECT*n*	**Sector Protect** 0 = Sector *n* is programmed or erased from user code. 1 = Sector *n* is protected and cannot be programmed or erased from user code.

Note: User code write bits from 0 to 1 only.

Option Bits

Option bits allow user configuration of certain aspects of the ZNEO® Z16F Series operation. The feature configuration data is stored in the Program memory and read during Reset. The features available for control using the option bits are:

- WDT time-out response selection–interrupt or Reset
- WDT enabled at Reset
- The ability to prevent unwanted read access to user code in Program memory
- The ability to prevent accidental programming and erasure of the user code in Program memory
- Voltage Brown-Out (VBO) configuration—always enabled or disabled during Stop Mode to reduce Stop Mode power consumption
- Oscillator mode selection for high, medium and low power crystal oscillators, or external RC oscillator
- PWM pin setup for motor control application

Operation

Each time the option bits are programmed or erased, the device must be Reset for the change to take place. During any reset operation (System Reset, Short Reset, or Stop Mode Recovery), the option bits are automatically read from the Program memory and written to Option Configuration registers. The Option Configuration registers control operation of the device. Option Bit Control Register are loaded before the device exits Reset and the ZNEO CPU begins code execution. The Option Configuration registers are not part of the Register file and are not accessible for read or write access.

Option Bit Address Space

The first four bytes of Program Memory at addresses 0000h through 0003h, shown in Tables 161 and 162, respectively, are reserved for the user option bits. These bytes are used to configure user specific options. You can change the option bits to meet application requirements.

Program Memory Address 0000h

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

OCD Control Register

The OCD Control Register (OCDCTL), shown in Table 176, controls the state of the CPU. This register puts the CPU in DEBUG Halt Mode, enables breakpoints, or single-steps an instruction.

Table 176. OCD Control Register (OCDCTL)

Bits	7	6	5	4	3	2	1	0
Field	DBGHALT	BRKHALT	BRKEN	DBGSTOP	Reserved			STEP
RESET	0	0	0	0	000			0
R/W	R/W	R/W	R/W	R/W	R			R/W

Bit	Description
[7] DBGHALT	**Debug Halt** Setting this bit to 1 causes the device to enter DEBUG HALT Mode. When in DEBUG HALT Mode, the CPU stops fetching instructions. Clearing this bit causes the CPU to start running again. This bit is automatically set to 1 when a breakpoint occurs if the BRKHALT bit is set. 0 = The device is running. 1 = The device is in DEBUG HALT Mode.
[6] BRKHALT	**Breakpoint Halt** This bit determines what action the OCD takes when a Breakpoint occurs. If this bit is set to 1, then the DBGHALT bit is automatically set to 1 when a breakpoint occurs. If BRKHALT is zero, then the CPU will loop on the breakpoint. 0 = CPU loops on current instruction when breakpoint occurs. 1 = A Breakpoint sets DBGHALT to 1.
[5] BRKEN	**Enable Breakpoints** This bit controls the behavior of the BRK instruction and the hardware breakpoint. By default, these generate an illegal instruction system trap. If this bit is set to 1, these events generate a Breakpoint instead of a system trap. The resulting action depends upon the BRKHALT bit. 0 = BRK instruction and hardware breakpoint generates system trap. 1 = BRK instruction and hardware breakpoint generates a breakpoint.
[4] DBGSTOP	**Debug Stop Mode** This bit controls the system clock behavior in Stop Mode. When set to 1, the system clock will continue to operate in Stop Mode. 0 = Stop Mode debug disabled. system clock stops in Stop Mode. 1 = Stop Mode debug enabled. system clock runs in Stop Mode.

General Purpose I/O Port Input Data Sample Timing

Figure 76 displays timing of the GPIO port input sampling. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The port value is then available to the ZNEO CPU on the second rising clock edge following the change of the port value. Table 195 lists the GPIO port input timing.

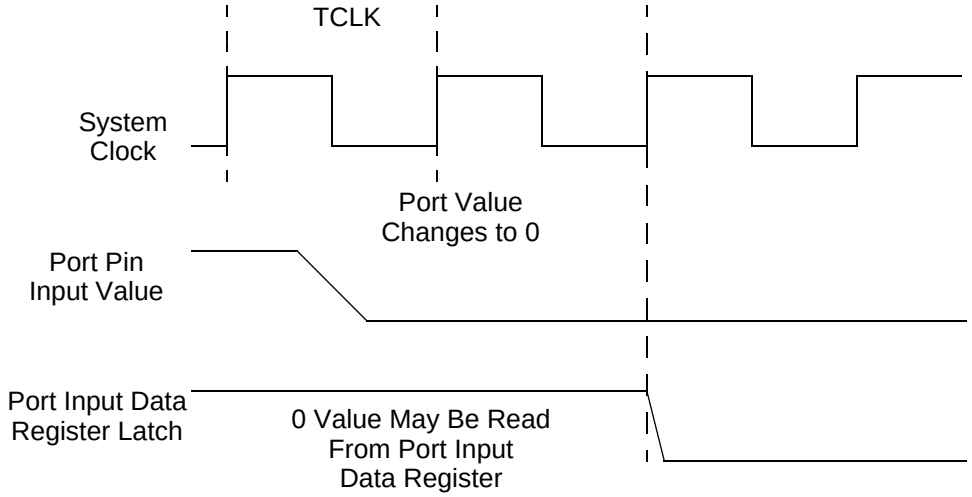

Figure 76. Port Input Sample Timing

Table 195. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Description**	**Min**	**Max**
T_{SMR}	GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources)	1µs	

On-Chip Debugger Timing

Table 196 provides timing information for the DBG pin. The DBG pin timing specifications assume a 4µs maximum rise and fall time.

Table 196. On-Chip Debugger Timing

		Delay (ns)	
Parameter	**Description**	**Min**	**Max**
DBG	Debug frequency.	—	System Clock/4