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	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	-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/z16f6411al20eg

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.

Address Space

The ZNEO CPU features a unique architecture with a single, unified 24-bit address space. It supports up to four memory areas:

- Internal nonvolatile memory (Flash, EEPROM, EPROM or ROM).
- Internal RAM.
- Internal I/O memory (internal peripherals).
- External memory (and/or memory-mapped peripherals).

The 24-bit address space supports up to 16MB (16,777,216 bytes) of memory. The ZNEO CPU accesses any two of the above memory areas in parallel. In addition, the ZNEO CPU supports three different data widths:

- Byte (8-bit)
- Word (16-bit)
- Quad (32-bit)

The ZNEO CPU accesses memories of different bus width:

- 8-bit wide memories
- 16-bit wide memories

Memory Map

The memory map of the ZNEO CPU displayed in Figure 6 shows the locations of internal nonvolatile memory, internal RAM and internal I/O memory. External memory, however, is placed at addresses which are not occupied by internal memory.

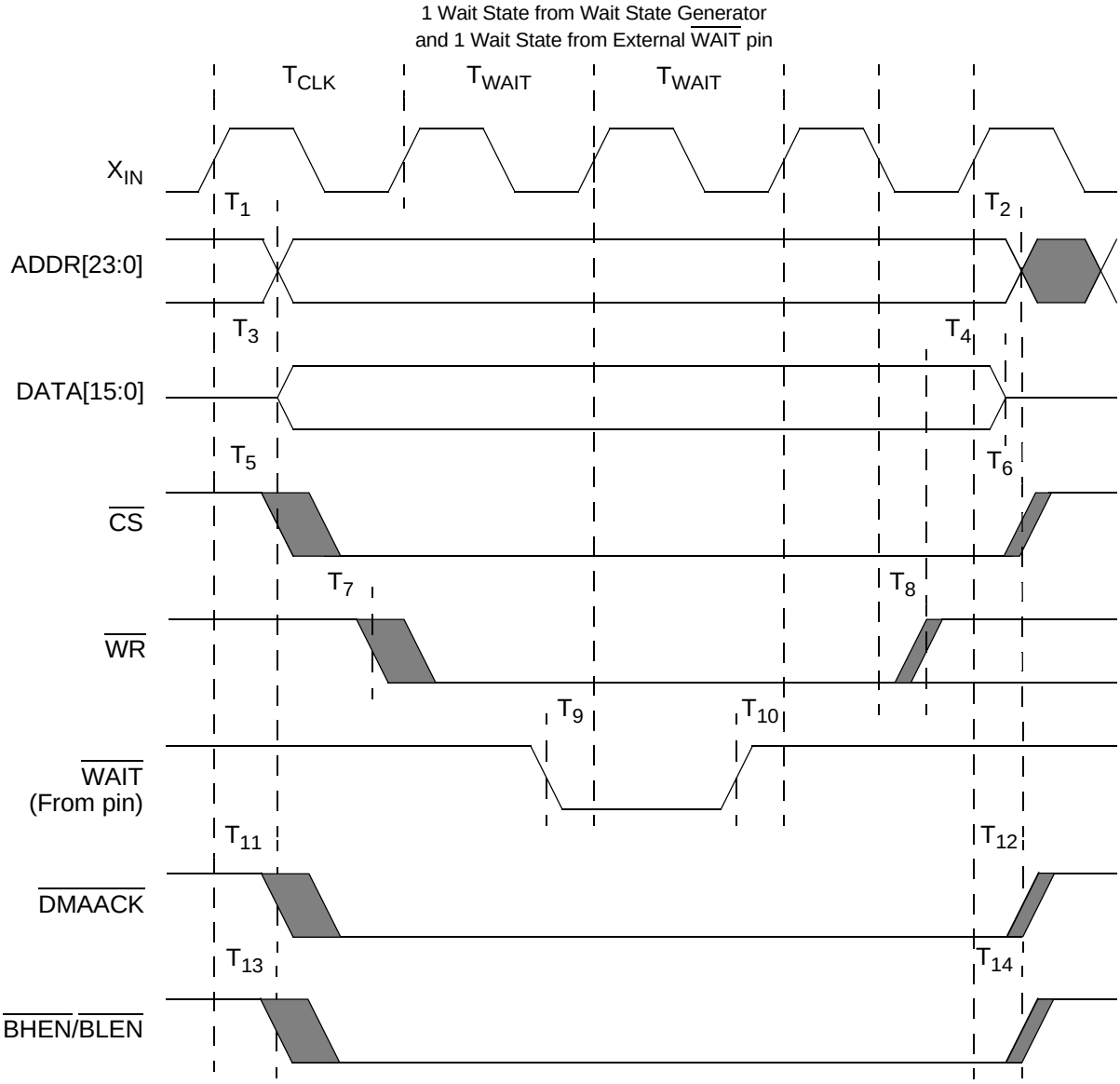


Figure 12. External Interface Timing for a Write Operation, ISA Mode

External Interface Read Timing, Normal Mode

Figure 13 and Table 16 provide timing information for the external interface performing a read operation in Normal Mode. In Figure 13, it is assumed the wait state generator has been configured to provide 2 wait states during read operations. For proper data hold time

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.

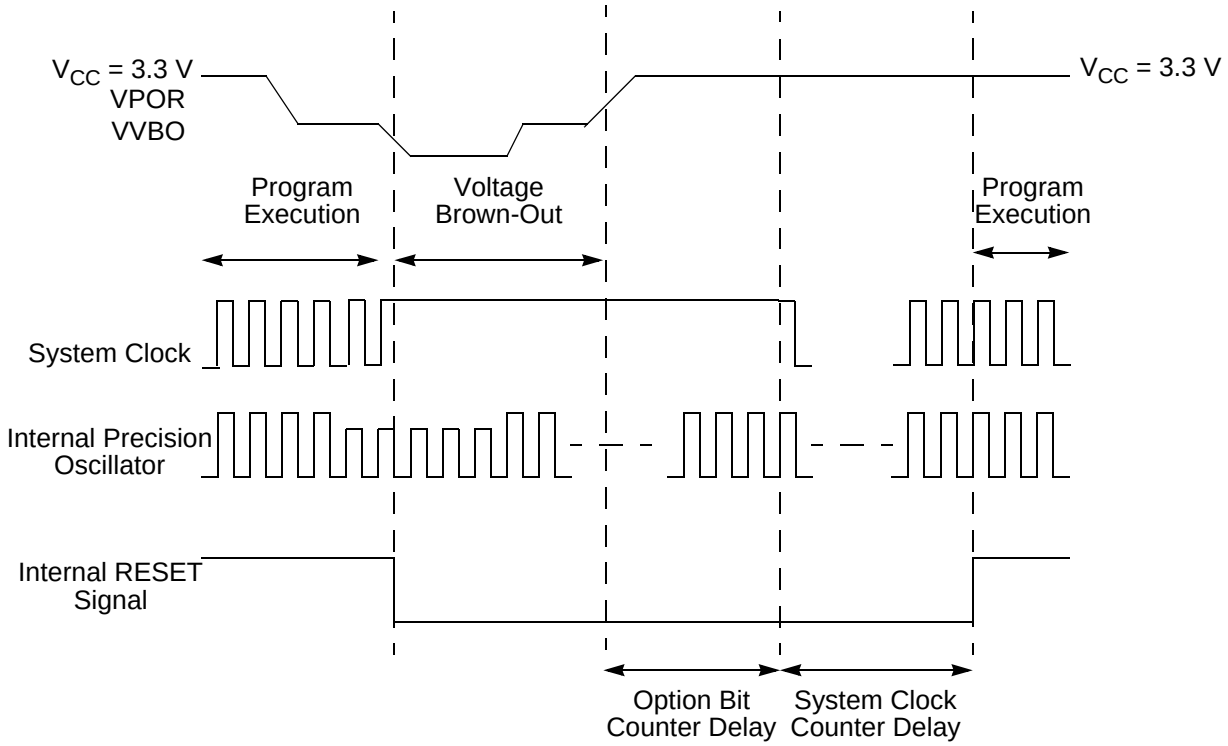


Figure 17. Voltage Brown-Out Reset Operation

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-

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.

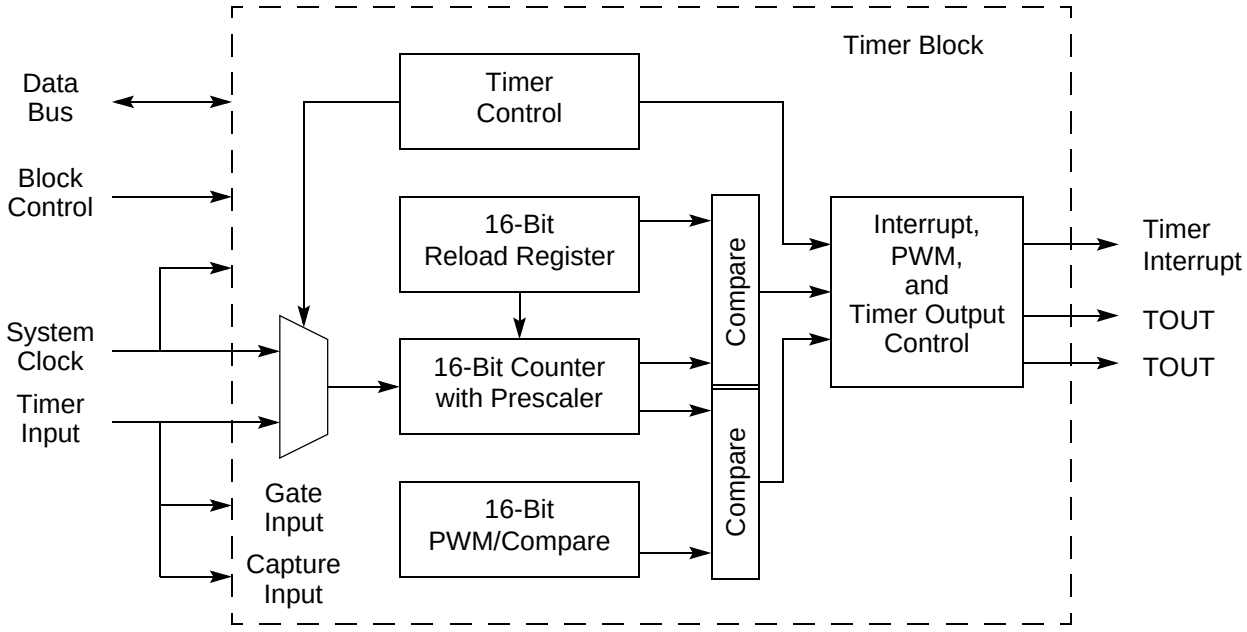


Figure 20. Timer Block Diagram

Operation

The general-purpose timer is a 16-bit up-counter. In normal operation, the timer is initialized to 0001h. When the timer is enabled, it counts up to the value contained in the Reload High and Low Byte registers, then resets to 0001h. The counter either halts or continues depending on the mode.

Minimum time-out delay (1 system clock) is set by loading the value 0001h into the Timer Reload High and Low byte registers and setting the prescale value to 1.

Maximum time-out delay ($2^{16} * 2^7$ system clocks) is set by loading the value 0000h into the Timer Reload High and Low byte registers and setting the prescale value to 128. When the timer reaches FFFFh, the timer rolls over to 0000h.

If the reload register is set to a value less than the current counter value, the counter continues counting until it reaches FFFFh and then resets to 0000h. Then the timer continues to count until it reaches the reload value and it resets to 0001h.

Note: When T0IN0, T0IN1 and T0IN2 functions are enabled on the PB0, PB1 and PB2 pins, each Timer 0 input will have the same effect as the single Timer 0 Input pin T0IN. For example, if the Timer 0 is in Capture Mode, any transitions on any of the PB0, PB1 and PB2 pins will cause a Capture.

PWM Fault Status Register (PWMFSTAT) is read to determine which fault source caused the interrupt.

When a fault is detected and the PWM outputs are disabled, modulator control of the PWM outputs are reenabled either by the software or by the fault input signal deasserting. Selection of the reenable method is made using the PWM Fault Control Register (PWMFCTL). Configuration of the fault modes and reenable methods allow pulse-by-pulse limiting and hard shutdown. When configured in Automatic Restart Mode, the PWM outputs are reengaged at beginning of the next PWM cycle (master timer value is equal to 0) if all fault signals are deasserted. In software controlled restart, all fault inputs must be deasserted and the fault flags must be cleared.

The fault input pin is Schmitt-triggered. The input signal from the pin as well as the comparators pass though an analog filter to reject high-frequency noise.

The logic path from the fault sources to the PWM output is asynchronous ensuring that the fault inputs forces the PWM outputs to their off-state even if the system clock is stopped.

PWM Operation in CPU Halt Mode

When the ZNEO CPU is operating in Halt Mode, the PWM continues to operate if it is enabled. To minimize current in Halt Mode, the PWM must be disabled by clearing the PWMEN bit to 0.

PWM Operation in CPU Stop Mode

When the ZNEO CPU is operating in Stop Mode, the PWM is disabled as the system clock ceases to operate in Stop Mode. The PWM output remains in the same state as they were prior to entering the Stop Mode. In normal operation, the PWM outputs must be disabled by software prior to the CPU entering the Stop Mode. A fault condition detected in Stop Mode forces the PWM outputs to the predefined off-state.

Observing the State of PWM Output Channels

The logic value of the PWM outputs is sampled by reading the PWMIN Register. If a PWM channel pair is disabled (option bit is not set), the associated PWM outputs are forced to high impedance and are used as general purpose inputs.

PWM Control Register Definitions

The following sections describe the various PWM control registers.

ESPI Signals

The four ESPI signals are:

- Master-In/Slave-Out (MISO)
- Master-Out/Slave-In (MOSI)
- Serial clock (SCK)
- Slave select ($\overline{SS}$)

The following paragraphs describe these signals in both Master and Slave modes. The appropriate GPIO pins must be configured using the GPIO alternate function registers.

Master-In/Slave-Out

The MISO pin is configured as an input in a master device and as an output in a slave device. Data is transferred to most significant bit first. The MISO pin of a slave device is placed in a high-impedance state if the slave is not selected. When the ESPI is not enabled, this signal is in a high-impedance state. The direction of this pin is controlled by the MMEN bit of the ESPI Control Register.

Master-Out/Slave-In

The MOSI pin is configured as an output in a master device and as an input in a slave device. Data is transferred to most significant bit first. When the ESPI is not enabled, this signal is in a high-impedance state. The direction of this pin is controlled by the MMEN bit of the ESPI Control Register.

Serial Clock

The SCK synchronizes data movement both in and out of the shift register via the MOSI and MISO pins. In Master Mode (MMEN = 1), the ESPI's baud rate generator creates the serial clock and drives it out via its SCK pin to the slave devices. In Slave Mode, the SCK pin is an input. Slave devices ignore the SCK signal unless their $\overline{SS}$ pin is asserted.

The master and slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles; see the NUMBITS field in the ESPI Mode Register (ESPIMODE) on page 197. In both master and slave ESPI devices, data is shifted on one edge of the SCK and is sampled on the opposite edge where data is stable. SCK phase and polarity is determined by the PHASE and CLKPOL bits in the ESPI Control Register (ESPICTL) on page 194.

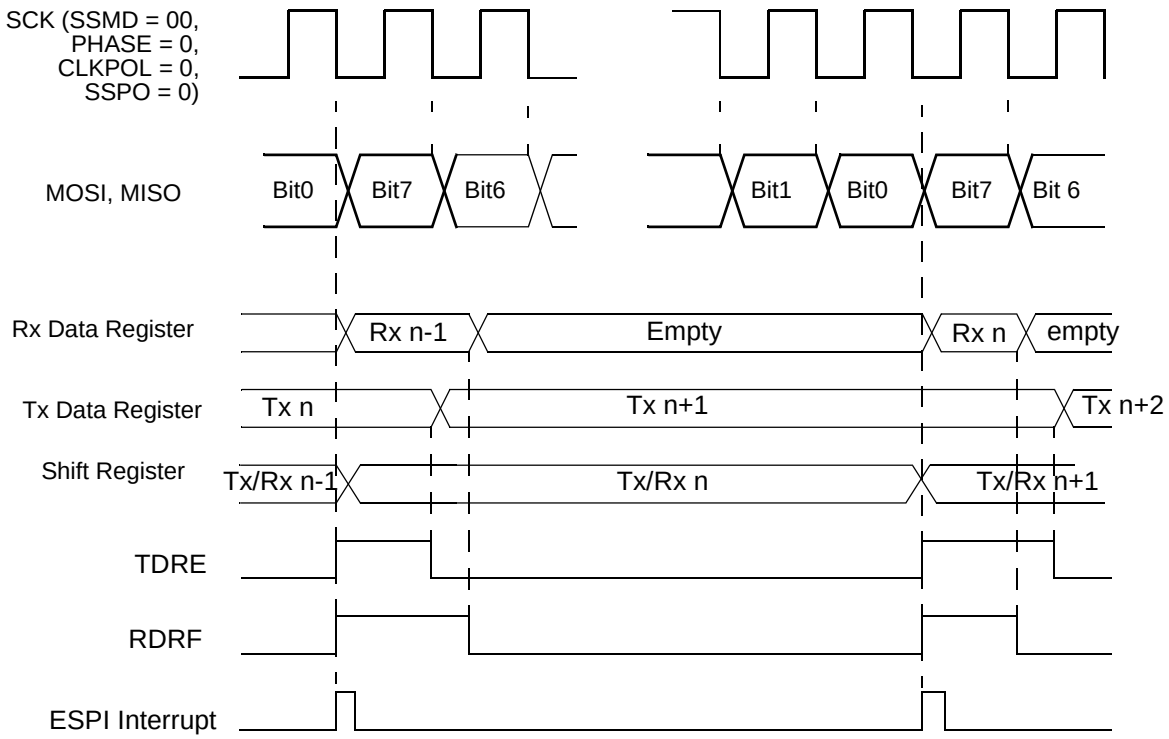


Figure 38. SPI Mode (SSMD = 000)

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

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

Software Control of I2C Transactions

The I2C Controller is configured using the I2C Control and I2C Mode registers. The MODE[1:0] field of the I2C Mode Register allows configuring the I2C Controller for Master/Slave or Slave Only Mode and configures the slave for 7-bit or 10-Bit Address recognition. 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.

Master/Slave Mode is used for:

- Master Only operation in a single master, one or more slave I2C system
- Master/Slave in a multi-master, multi-slave I2C system
- Slave Only operation in an I2C system

In Slave Only Mode, the Start bit of the I2C Control Register is ignored (software cannot initiate a master transaction by accident). This restricts the operation to Slave Only Mode and prevents accidental operation in Master Mode.

Software controls I2C transactions by enabling the I2C Controller interrupt in the interrupt controller or by polling the I2C Status Register.

To use interrupts, the I2C interrupt must be enabled in the Interrupt Controller and followed by executing an EI instruction. The TXI bit in the I2C Control Register must be set to enable transmit interrupts. An I2C interrupt service routine then verifies the I2C Status Register to determine the cause of the interrupt.

To control transactions by polling, the interrupt bits (TDRE, RDRF, SAM, ARBLST, SPRS and NCKI) in the I2C Status Register must be polled. The TDRE bit asserts regardless of the state of the TXI bit.

Master Transactions

The following sections describe the Master read and write transactions to both 7- and 10-bit slaves.

Master Arbitration

If a Master loses arbitration during the address byte, it releases the SDA line, switches to Slave Mode and monitors the address to determine if it is selected as a Slave. If a Master loses arbitration during a transmit data byte, it releases the SDA line and waits for the next Stop or Start condition.

The Master detects a loss of arbitration when a 1 is transmitted but a 0 is received from the bus in the same bit time. This loss occurs if more than one Master is simultaneously accessing the bus. Loss of arbitration occurs during the address phase (two or more Mas-

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

Reference Buffer, RBUF

The reference buffer, RBUF, supplies the reference voltage for the ADC. When enabled, the internal voltage reference generator supplies the ADC and the voltage is available on the VREF pin. When RBUF is disabled, the reference voltage must be supplied externally through the VREF pin. RBUF is controlled by the REFEN bit in the ADC0 Control Register.

Internal Voltage Reference Generator

The internal voltage reference generator provides the voltage to RBUF. The internal reference voltage is 2 V.

ADC Control Register Definitions

The following sections describe the control registers for the ADC.

ADC0 Control Register 0

The ADC0 Control Register initiates the A/D conversion and provides ADC0 status information.

Table 126. ADC0 Control Register 0 (ADC0CTL)

Bits	7	6	5	4	3	2	1	0
Field	START0	CVTRD0	REFEN	ADC0EN	ANAIN0[3:0]			
RESET	0	0	0	0	0	0	0	0
R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FF–E500h							

Bit	Description
[7] START0	**ADC0 Start/Busy** 0 = Writing to 0 has no effect. Reading a 0 indicates the ADC0 is available to begin a conversion. 1 = Writing to 1 starts a conversion on ADC0. Reading a 1 indicates a conversion is currently in progress.
[6] CVTRD0	**Convert On Read** 0 = The ADC0 operates normally. 1 = If this bit is set to 1, whenever the ADC0D Register is read it increments the ANAIN field by one and start a new conversion. The ANAIN field increments until it reaches the value set in the ADC0MAX Register. After doing the conversion on the channel specified by the ADC0MAX Register, the next read resets the ANAIN field to 0. This function is used with the DMA to perform continuous conversions.

tion is complete, the Flash Controller returns to its locked state. Only pages located in unprotected sectors are erased.

The four steps to performing a Page Erase operation are:

1. Write the page to be erased to the Flash Page Select Register.
2. Write the first unlock command 73h to the Flash Command Register.
3. Write the second unlock command 8Ch to the Flash Command Register.
4. Write the Page Erase command 95h to the Flash Command Register.

Mass Erase

The Flash memory cannot be Mass Erased by user code.

Flash Controller Bypass

The Flash Controller is bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Programming algorithms by controlling the Flash programming signals directly.

Flash Controller Bypass is recommended for large volume gang programming applications, which do not require in-circuit programming of the Flash memory.

Flash Controller Behavior using the On-Chip Debugger

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Controller does not have to be unlocked for program and erase operations.
- The Flash Write Protect option bit is ignored.
- The Flash Sector Protect register is ignored for programming and erase operations.
- Programming operations are not limited to the page selected in the Flash Page Select Register.
- Bits in the Flash Sector Protect register is written to 1 or 0.
- The Flash Page Select Register is written when the Flash Controller is unlocked.
- The Mass Erase command is enabled.

bit 7. For the auto-baud character 0Dh, the auto-baud detector measures the period from the rising edge at the end of the start bit to the rising edge at the beginning of the stop bit. This measured value is automatically written to the BRG reload register after the auto-baud character is received. Once configured, the BRG will generate a bit clock based on this measured character time.

Line Control

When operating at high speeds, it is appropriate to speed up the rise and fall times of the single wire bus. Three control bits are used to control the bus rise and fall times, the high drive strength enable bit, the drive high enable bit and the output enable control bit.

The high drive strength enable bit puts the pin into High Driv eMode. For information about high drive strength, see the Electrical Characteristics chapter on page 337.

If the output enable control bit is set, the line is driven High and Low during transmission. If the drive high control bit is set, it drives the line high for short periods when transmitting a logic one. This rapidly charges the inherent capacitance of the single wire bus.

If both the output enable and drive high control bits are set, the line is driven high for one clock cycle when transmitting a one. If the output enable bit is clear and the drive high bit is set, the line is driven high until the input is detected High or the center of the bit time occurs, whichever is first.

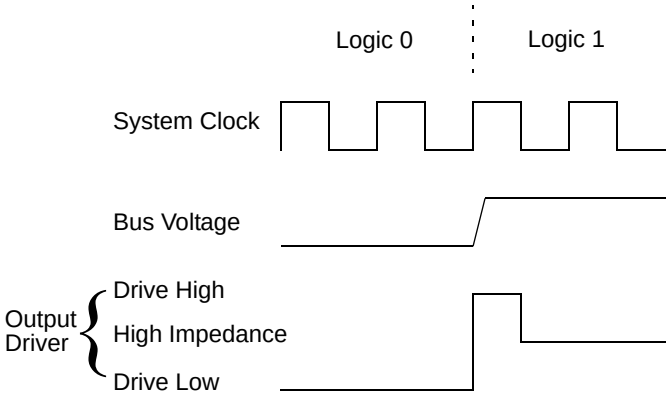


Figure 66. Output Driver when Drive High and Open Drain Enabled

9-Bit Mode

The serial interface is configured to transmit and receive a ninth data bit. This ninth bit is used to transmit or receive a software generated parity bit. It is used as an address/data bit in a multi-node system such as RS-485.

Read Memory. The Read memory command reads data from memory. The memory address is sign extended.

```
DBG <-- {1000,Size[3:0]}
DBG <-- addr[15:8]
DBG <-- addr[7:0]
DBG ->> 1 to 16 bytes of data
DBG --> CRC[0:7]
```

Write Memory. The Write memory command writes data to memory. The memory address is sign extended.

```
DBG <-- {1001,size[3:0}
DBG <-- addr[15:8]
DBG <-- addr[7:0]
DBG <<- 1 to 16 bytes of data
DBG --> CRC[0:7]
```

Read Memory. The Read memory command reads data from memory**.**

```
DBG <-- {1010,size[3:0}
DBG <-- size[11:4]
DBG <-- 00h
DBG <-- addr[23:16]
DBG <-- addr[15:8]
DBG <-- addr[7:0]
DBG ->> 1 to 4096 bytes of data
DBG --> CRC[0:7]
```

Write Memory. The Write memory command writes data to memory**.**

```
DBG <-- {1011,size[3:0}
DBG <-- size[11:4]
DBG <-- 00h
DBG <-- addr[23:16]
DBG <-- addr[15:8]
DBG <-- addr[7:0]
DBG <<- 1 to 4096 bytes of data
DBG --> CRC[0:7]
```

Read Memory CRC. The Read memory CRC command computes and return the CRC of a block of memory.

```
DBG <-- {1110,BlockCount[3:0]}
DBG <-- BlockCount[11:4]
DBG <-- 00h
DBG <-- addr[23:16]
DBG <-- {addr[15:12],xxxx}
DBG --> MemoryCRC[0:7]
DBG --> MemoryCRC[8:15]
DBG --> CRC[0:7]
```

MemoryCRC is computed on memory in increments of 4K blocks. The BlockCount field determines how many blocks of memory to compute the MemoryCRC on.

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.

On-Chip Oscillator

The products in the ZNEO® Z16F Series feature an on-chip oscillator for use with external crystals with frequencies from 32 kHz to 20 MHz. In addition, the oscillator supports external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with oscillation frequencies up to 20 MHz. This oscillator generates the primary system clock for the internal ZNEO CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin also accept a CMOS-level clock input signal (32 kHz to 20 MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected.

When configured for use with crystal oscillators or external clock drivers, the frequency of the signal on the X_{IN} input pin determines the frequency of the system clock (that is, no internal clock divider). In RC operation, the system clock is driven by a clock divider (divide by 2) to ensure 50% duty cycle.

Operating Modes

The ZNEO Z16F Series products support four different oscillator modes:

- On-chip oscillator configured for use with external RC networks (<4MHz).
- Minimum power for use with very low frequency crystals (32kHz to 1.0MHz).
- Medium power for use with medium frequency crystals or ceramic resonators (0.5MHz to 10.0MHz)
- Maximum power for use with high frequency crystals or ceramic resonators (8.0MHz to 20.0MHz)

The oscillator mode is selected through user-programmable option bits. For more information, see the Option Bits chapter on page 292.

Crystal Oscillator Operation

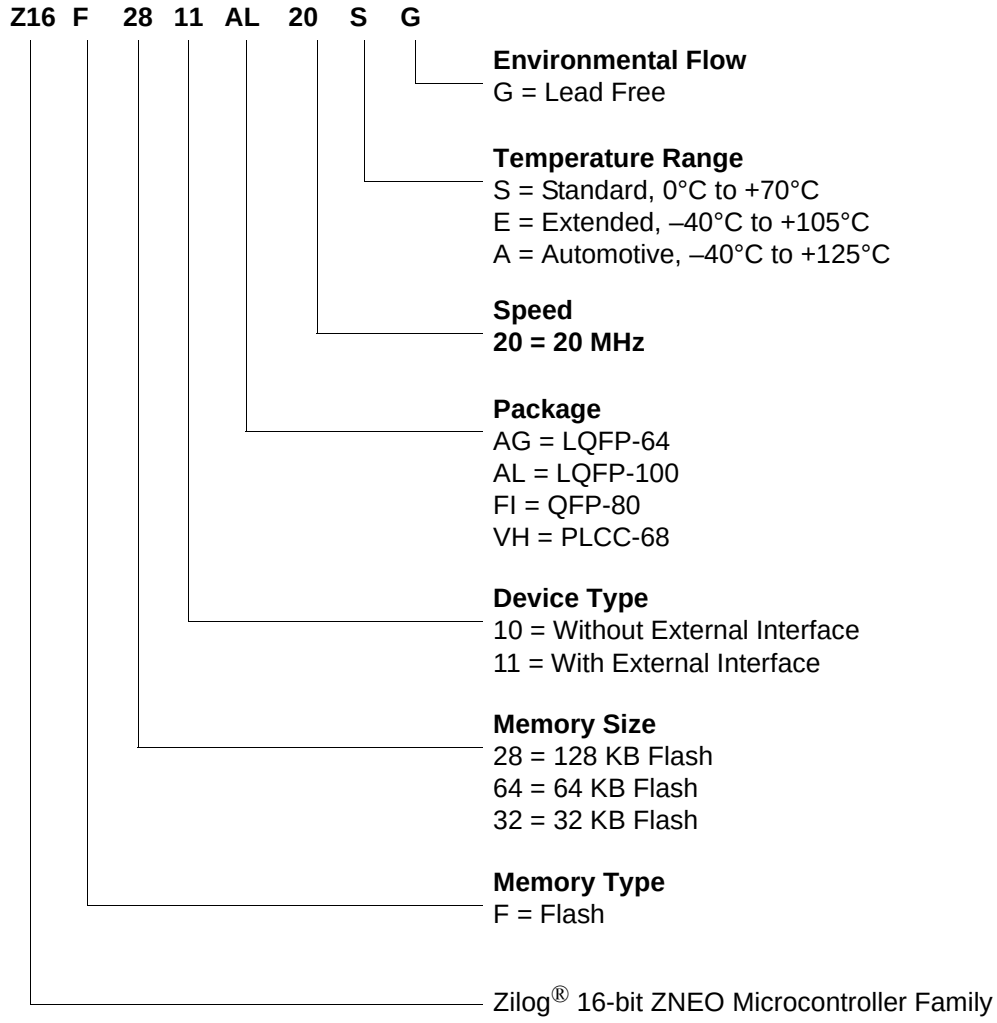
Figure 70 displays a recommended configuration for connection with an external fundamental mode, parallel-resonant crystal operating at 20MHz. Recommended 20MHz crystal specifications are provided in Table 181. Resistor R1 is optional and limits total power dissipation by the crystal. The printed circuit board layout must add no more than 4 pF of stray capacitance to either the X_{IN} or X_{OUT} pins. If oscillation does not occur, it reduce the values of capacitors C1 and C2 to decrease loading.

Table 186. DC Characteristics (Continued)

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
I_{TL}	Tri-State Leakage Current	–5	—	+5	µA	V_{DD} = 3.6 V
C_{PAD}	GPIO Port Pad Capacitance	—	8.0[2]	—	pF	
C_{XIN}	X_{IN} Pad Capacitance	—	8.0[2]	—	pF	
C_{XOUT}	X_{OUT} Pad Capacitance	—	9.5[2]	—	pF	
I_{PU}	Weak Pull-up Current	30	100	350	µA	V_{DD} = 2.7 V to 3.6 V
I_{CCS1}	Supply Current in Stop Mode with VBO enabled		600		µA	V_{DD} = 3.0 V; 25°C
I_{CCS2}	Supply Current in Stop Mode with VBO disabled		2		µA	V_{DD} = 3.0 V; 25°C
I_{CCS3}	Supply Current in Stop Mode with VBO disabled and WDT disabled		1		µA	V_{DD} = 3.0 V; 25°C
I_{CCA}	Active I_{DD} at 20MHz	—	18	35	mA	Typ: V_{DD}=3.0 V/30°C Max: V_{DD}=3.6 V/125°C Peripherals enabled, no loads
I_{CCH}	I_{DD} in Halt Mode at 20MHz	—	4	6	mA	Typ: V_{DD}=3.0 V/30°C Max: V_{DD}=3.6 V/125°C Peripherals off, no loads

Note:
1. This condition excludes all pins that have on-chip pull-ups enabled, when driven Low.

Part Number Suffix Designations



Note: Packages are not available for all memory sizes. See the Ordering Information section on page 356 for available packages.

Precharacterization Product

The product represented by this document is newly introduced and Zilog® has not completed the full characterization of the product. The document states what Zilog knows about this product at this time, but additional features or nonconformance with some aspects of the document might be found, either by Zilog or its customers in the course of further application and characterization work. In addition, Zilog cautions that delivery might be uncertain at times, due to start-up yield issues. For more information, please visit www.zilog.com.