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	Obsolete
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	46
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	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f2810vh20eg

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

the address space between internal and external nonvolatile memory and between nonvolatile and volatile memory.

- Any external volatile (random access) memory must be located at or above 80_0000h in the 24-bit address space (in the $\overline{CS1}$ range). This location is a requirement of the ZDSII GUI. Volatile memory on $\overline{CS0}$ is located in a lower address range if it is configured by adding an edited linker RANGE command to the Additional Linker Commands field of the ZDSII project settings.
- External volatile memory falling below FF_8000h must be addressed as a contiguous block. The ZDSII C-Compiler large model does not support holes in 32-bit addressed volatile memory. There is a hole between this memory and volatile memory at or above FF_8000h, however.
- External volatile memory at or above FF_8000h must be addressed as a block contiguous with the microcontroller's internal RAM. The ZDSII C-Compiler small model does not support holes in 16-bit addressable volatile memory.
- External volatile memory must not be located above internal RAM, which ends at FF_BFFFh. This location is a requirement of the ZDSII GUI. Volatile memory is located at FF_C000h and extend up to FF_DFFFh if the space is not used for I/O, but the range must be configured by adding an edited linker RANGE command to the Additional Linker Commands field of the project settings. The debugger memory window always displays this range as part of the I/O Data space, however.
- The ZDSII GUI assumes external I/O is located in the range FF_C000h to FF_DFFFh. Any external I/O that is located elsewhere is accessed using absolute addressing. The debugger memory window displays all addresses below FF_C000h as part of the Memory space.

For details about how ZDSII development tools use memory, refer to the Zilog Developer Studio II – ZNEO User Manual (UM0171).

External $\overline{WAIT}$ Pin Operation

Setup of the external $\overline{WAIT}$ pin is selected by the GPIO alternate function. When using the external $\overline{WAIT}$ pin, at least one internal wait state must be added to allow sufficient address valid to wait input setup time.

Table 13 lists the External Chip Select Control Registers Low for $\overline{CS2}$ to $\overline{CS5}$ (EXTCSxL). This register sets the number of wait states for chip selects 2 through 5. Waits are only added if the chip select is enabled.

Table 13. External Chip Select Control Registers Low for $\overline{CS2}$ to $\overline{CS5}$ (EXTCSxL)

Bits	7	6	5	4	3	2	1	0
Field	RESERVED		PRxWAIT		CSxWAIT			
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	FF_(E077, E079, E07B, E07D)H							

Bit	Description
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:4] PRxWAIT[2:0]	**Post Read Wait Selection** 00 = 0 wait state. 01 = 1 wait state. 10 = 2 wait states. 11 = 3 wait states.
[3:0] CSxWAIT	**Chip Select x Wait Selection** 0000 = 0 wait state. 0001 = 2 wait state. 0010 = 4 wait states. 0011 = 6 wait states. 0100 = 8 wait states. 0101 = 10 wait states. 0110 = 12 wait states. 0111 = 14 wait states. 1000 = 16 wait states. 1001 = 18 wait states. 1010 = 20 wait states. 1011 = 22 wait states. 1100 = 24 wait states. 1101 = 26 wait states. 1110 = 28 wait states. 1111 = 30 wait states.

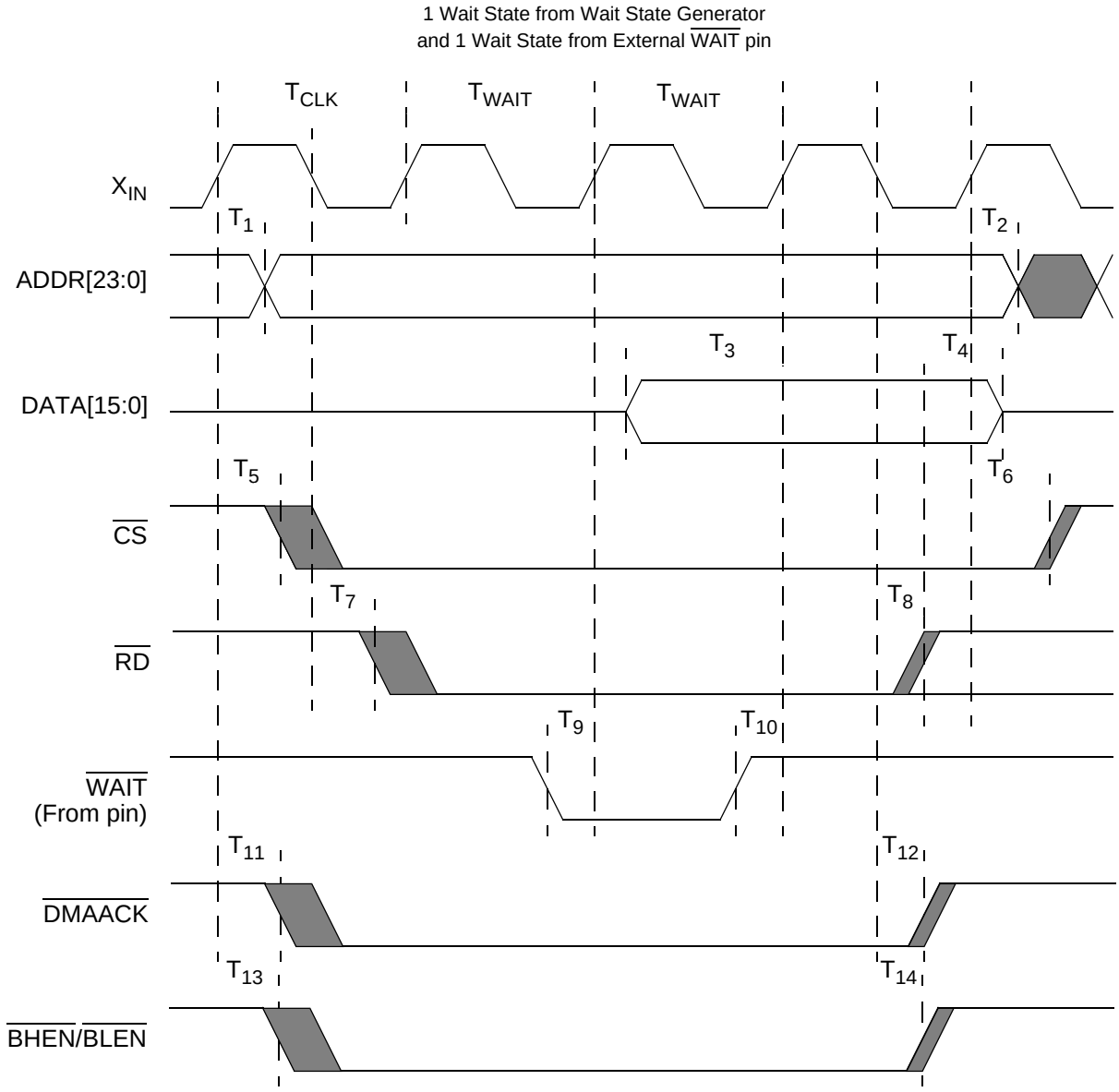

Figure 15. External Interface Timing for a Read Operation, ISA Mode

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

The PWM period is determined by the following equation:

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

If an initial starting value other than `0001h` is loaded into the Timer High and Low Byte registers, use the One-Shot Mode equation to determine the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is determined by:

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

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

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

Capture Modes

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

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

PWM Reload High and Low Byte Registers

The PWM Reload High and Low Byte (PWMRH and PWMRL) registers, shown in Tables 66 and 67, store a 12-bit reload value, {PWMRH[3:0], PWMRL[7:0]}. The PWM reload value is held in buffer registers. The PWM reload value written to the buffer registers are not used by the PWM generator until the next PWM reload event occurs. Reads from these registers always return the values from the buffer registers.

$$\text{Edge-Aligned PWM Mode Period} = \frac{\text{Prescaler} \times \text{Reload Value}}{f_{PWMclk}}$$

$$\text{Center-Aligned PWM Mode Period} = \frac{2 \times \text{Prescaler} \times \text{Reload Value}}{f_{PWMclk}}$$

Table 66. PWM Reload High Byte Register (PWMRH)

Bits	7	6	5	4	3	2	1	0
Field	Reserved				PWMRH			
RESET	0h				Fh			
R/W	R/W				R/W			
Addr	FF_E38EH							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] PWMRH	**PWM Reload Register High and Low** These two bytes form the 12-bit reload value, {PWMRH[3:0], PWMRL[7:0]}. This value sets the PWM period.

Table 67. PWM Reload Low Byte Register (PWMRL)

Bits	7	6	5	4	3	2	1	0
Field	PWMRL							
RESET	FF							
R/W	R/W							
Addr	FF_E38Fh							

Bit	Description
[7:0] PWMRL	**PWM Reload Register High and Low** These two bytes form the 12-bit reload value, {PWMRH[3:0], PWMRL[7:0]}. This value sets the PWM period.

2. Load the appropriate 16-bit count value into the LIN-UART Baud Rate High and Low Byte registers.
3. Enable the BRG timer function and associated interrupt by setting the BRGCTL bit in the LIN-UART Control 1 Register to 1. Enable the UART receive interrupt in the interrupt controller.

When configured as a general purpose timer, the BRG interrupt interval is calculated using the following equation:

UART BRG Interrupt Interval (s) = System Clock Period (s) × BRG[15:0]

Noise Filter

A noise filter circuit is included to filter noise on a digital input signal, such as UART receive data before the data is sampled by the block. This filter is a requirement for protocols in a noisy environment.

The noise filter includes following features:

- Synchronizes the receive input data to the system clock.
- Noise filter enable (NFEN)input selects whether the noise filter is bypassed (NFEN = 0) or included (NFEN = 1) in the receive data path.
- Noise filter control (NFCTL[2:0])input selects the width of the up/down saturating counter digital filter. The available widths range is from 4 to11 bits.
- The digital filter output has hysteresis.
- Provides an active low saturated state output (FiltSatB), used to indicate presence of noise.

Architecture

Figure 30 displays how the noise filter is integrated with the LIN-UART for use on a LIN network.

  - STR bit on the SPI module replaced with ESPIEN1. SPIEN replaced with ESPIEN0. This enhancement allows unidirectional transfers which minimizes software or DMA overhead.
  - BIRQ replaced with BRGCTL.
- Mode register:
  - Added SSMD field which adds support for loop back and I2S modes.
  - Moved SSV bit to the transmit data command register as described above.
  - Added slave select polarity (SSPO) to support active High and Low slave select on $\overline{SS}$ pin.
- Status register:
  - IRQ split into TDRE and RDRF (separate transmit and receive interrupts).
  - Replace overrun error with separate transmit under-run and receive overrun.
- State register.
  - Replaced SCKEN bit with SCKI.
  - Replaced TCKEN with SDI.

Operation

During transfer, data is sent and received simultaneously by both master and slave devices. Separate signals are required to transmit data, receive data and the serial clock. When a transfer occurs, a multibit (typically 8-bit) character is shifted out one data pin and a multi-bit character is simultaneously shifted in on a second data pin. An 8-bit shift register in the master and an 8-bit shift register in the slave is connected as a circular buffer. The ESPI shift register is buffered to support back-to-back character transfers in high performance applications.

A transaction is initiated when the Transmit Data Register is written in the master device. The value from the data register is transferred into the shift register and the transaction begins. After the transmit data is loaded into the shift register, the Transmit Data register Empty (TDRE) status bit asserts, indicating that Transmit Data Register is written with the next value. At the end of each character transfer, the shift register value (receive data) is loaded into the Receive Data Register. At that point the Receive Data register Full (RDRF) status bit asserts. When software or DMA reads the receive data from the Receive Data Register, the RDRF signal deasserts.

The master sources the SCK and $\overline{SS}$ signal during the transfer.

Internal data movement (either by software or DMA) to/from the ESPI block is controlled by the Transmit Data Register empty (TDRE) and Receive Data Register full (RDRF) signals. These signals are read only bits in the ESPI Status Register. When either the TDRE or RDRF bits assert, an interrupt is sent to the interrupt controller if the data interrupt

Bit	Description (Continued)
[2] WOR	**Wire-OR (Open-Drain) Mode Enabled** 0 = ESPI signal pins not configured for open-drain. 1 = All four ESPI signal pins (SCK, $\overline{SS}$, MISO, MOSI) configured for open-drain function. This setting is used for Multi-Master and/or Multi-Slave configurations.
[1] MMEN	**ESPI Master Mode Enable** This bit controls the data I/O pin selection and SCK direction. 0 = Data-out on MISO, data-in on MOSI (used in SPI Slave Mode), SCK is an input. 1 = Data-out on MOSI, data-in on MISO (used in SPI Master Mode), SCK is an output.

Caution: If reading the counter one byte at a time while the BRG is counting keep in mind that the values will not be in sync. It is recommended to read the counter using word (2-byte) reads.

ESPI Mode Register

The ESPI Mode Register, shown in Table 105, configures the character bit width and mode of the ESPI I/O pins.

Table 105. ESPI Mode Register (ESPIMODE)

Bits	7	6	5	4	3	2	1	0
Field	SSMD			NUMBITS[2:0]			SSIO	SSPO
RESET	000			000			0	0
R/W	R/W			R/W			R/W	R/W
Addr	FF_E263h							

Bit	Description
[7:5] SSMD	**Slave Select Mode** This field selects the behavior of $\overline{SS}$ as a framing signal. For a detailed description of these modes, see the Slave Select section on page 179. **000 = SPI Mode** When SSIO = 1, the $\overline{SS}$ pin is driven directly from the SSV bit in the Transmit Data Command register. The Master software or DMA must set SSV (or a GPIO output if the $\overline{SS}$ pin is not connected to the appropriate Slave) to the asserted state prior to or on the same clock cycle with which the Transmit Data Register is written with the initial byte. At the end of a frame (after the last RDRF event), SSV is deasserted by software. Alternatively, SSV is automatically deasserted by hardware if the TEOF bit in the Transmit Data Command register is set when the last transmit byte is loaded. In SPI Mode, SCK is active only for data transfer (one clock cycle per bit transferred). **001 = LOOPBACK Mode** When ESPI is configured as Master (MMEN = 1) the outputs are deasserted and data is looped from shift register out to shift register in. When ESPI is configured as a Slave (MMEN = 0) and $\overline{SS}$ in asserts, MISO (Slave output) is tied to MOSI (Slave input) to provide an a remote loop back (echo) function. **010 = I^2S Mode** In this mode, the value from SSV will be output by the Master on the $\overline{SS}$ pin one SCK period before the data and will remain in that state until the start of the next frame. Typically this mode is used to send back-to-back frames with $\overline{SS}$ alternating on each frame. A frame boundary is indicated in the Master when SSV changes. A frame boundary is detected in the Slave by $\overline{SS}$ changing state. The $\overline{SS}$ framing signal will lead the frame by one SCK period. In this mode SCK will run continuously, starting with the initial $\overline{SS}$ assertion. Frames will run back-to-back as long as software/DMA continue to provide data. The I^2S protocol (Inter IC Sound) is used to carry left and right channel audio data with the $\overline{SS}$ signal indicating which channel is being sent. In Slave Mode, the change in state of $\overline{SS}$ (Low to High or High to Low) will trigger the start of a transaction on the next SCK cycle.

Each interrupt source other than the baud rate generator interrupt has an associated bit in the I2CISTAT Register, which clears automatically when software reads the register or performs some other task such as reading or writing the data register.

Transmit Interrupts

Transmit interrupts (TDRE bit = 1 in I2CISTAT) occur under the following conditions:

- The Transmit Data Register is empty and the TXI bit = 1 in the I^2C Control Register
- The I^2C Controller is enabled, with any one of the following operations:
  - The first bit of a 10-bit address is shifted out
  - The first bit of the final byte of an address is shifted out and the RD bit is deasserted
  - The first bit of a data byte is shifted out

Writing to the I^2C Data Register always clears the TRDE bit to 0.

Receive Interrupts

Receive interrupts (RDRF bit = 1 in I2CISTAT) occur when a byte of data has been received by the I^2C Controller. The RDRF bit is cleared by reading from the I^2C Data Register. If the RDRF interrupt is not serviced prior to the completion of the next receive byte, the I^2C Controller holds SCL Low during the last data bit of the next byte until RDRF is cleared to prevent receive overruns. A receive interrupt does not occur when a Slave receives an address byte or for data bytes following a Slave address that did not match. An exception is if the interactive receive mode (IRM) bit is set in the I2CMODE Register in which case receive interrupts occur for all receive address and data bytes in Slave Mode.

Slave Address Match Interrupts

Slave address match interrupts (SAM bit = 1 in I2CISTAT) occur when the I^2C Controller is in Slave Mode and an address is received which matches the unique Slave address. The General Call Address (0000_0000) and STARTBYTE (0000_0001) are recognized if the GCE bit = 1 in the I2CMODE Register. Software verifies the RD bit in the I2CISTAT Register to determine if the transaction is a read or write transaction. The General Call Address and STARTBYTE addresses are also distinguished by the RD bit. The general call address (GCA) bit of the I2CISTAT Register indicates whether the address match occurred on the unique Slave address or the General Call/STARTBYTE address. The SAM bit clears automatically when the I2CISTAT Register is read.

If configured using the MODE[1:0] field of the I^2C Mode Register for 7-bit slave addressing, the most significant 7 bits of the first byte of the transaction are compared against the SLA[6:0] bits of the Slave Address Register. If configured for 10-bit slave addressing, the

Bit	Description (Continued)
[1] SPRS	**Stop/Restart Condition Interrupt** This bit is set when the I2C Controller is enabled in Slave Mode and detects a Stop or Restart condition during a transaction directed to this slave. This bit clears when the I2CISTAT Register is read. Read the RSTR bit of the I2CSTATE Register to determine whether the interrupt was caused by a Stop or Restart condition.
[0] NCKI	**NAK Interrupt** In Master Mode, this bit is set when a Not Acknowledge condition is received or sent and neither the Start nor the Stop bit is active. In Master Mode, this bit is cleared only by setting the Start or Stop bits. In Slave Mode, this bit is set when a Not Acknowledge condition is received (Master reading data from Slave), indicating the Master is finished reading. A Stop or Restart condition follows. In Slave Mode this bit clears when the I2CISTAT Register is read.

I2C Control Register

The I2C Control Register, shown in Table 114, enables and configures the I2C operation.

Table 114. I2C Control Register (I2CCTL)

Bits	7	6	5	4	3	2	1	0
Field	IEN	START	STOP	BIRQ	TXI	NAK	FLUSH	FILTEN
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W1	R/W1	R/W	R/W	R/W1	R/W	R/W
Addr	FF–E242h							

Note: R/W1 = The bit is set (write 1) but not cleared.

Bit	Description
[7] IEN	**I2C Enable** This bit enables the I2C Controller.
[6] START	**Send Start Condition** When set, this bit causes the I2C Controller (when configured as the Master) to send the Start condition. After it is asserted, this bit is cleared by the I2C Controller after it sends the Start condition or by deasserting the IEN bit. If this bit is 1, it cannot be cleared by writing to the bit. After this bit is set, the Start condition is sent if there is data in the I2CDATA or I2CSHIFT Register. If there is no data in one of these registers, the I2C Controller waits until data is loaded. If this bit is set while the I2C Controller is shifting out data, it generates a Restart condition after the byte shifts and the acknowledge phase completes. If the Stop bit is also set, it also waits until the Stop condition is sent before the Start condition. If Start is set while a Slave Mode transaction is underway to this device, the Start bit is cleared and ARBLST bit in the Interrupt Status Register will be set.

Sample Time Register

The Sample Time Register is used to program the length of active time for the sample after a conversion has begun by setting the Start bit in the ADC Control Register or initiated by the PWM. The number of system clock cycles required for sample time varies from system to system depending on the clock period used. This register must be programmed to contain the number of system clocks required to meet a 1 µs minimum sample time.

Table 130. Sample Time (ADCST)

Bits	7	6	5	4	3	2	1	0
Field	Reserved		ST					
RESET	0		1	1	1	1	1	1
R/W	R		R/W					
Addr	FF–E505h							

Bit	Description
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:0] SHT	**Sample Hold Time** 00h–3Fh = Sample Hold time in number of system clock periods to meet 1 µs minimum.

ADC Clock Prescale Register

The ADC Clock Prescale Register is used to provide a divided system clock to the ADC. When this register is programmed with 0h, the system clock is used for the ADC Clock.

Table 131. ADC Clock Prescale Register (ADCCP)

Bits	7	6	5	4	3	2	1	0
Field	Reserved				DIV16	DIV8	DIV4	DIV2
RESET	0				0	0	0	0
R/W	R				R/W			
Addr	FF–E506h							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] DIV16	**DIV16** 0 = Clock is not divided. 1 = System Clock is divided by 16 for ADC Clock.

Bit	Description (Continued)
[4] CPISEL	**Comparator Input Select** 0 = PortB6 provides the comparator - input. 1 = PortC0 provides the comparator - input.
[3] CMPIRQ	**Comparator Interrupt Edge Select** 0 = Interrupt Request on Comparator Rising Edge. 1 = Interrupt Request on Comparator Falling Edge.
[2] CMPIV	**PWM Fault Comparator Polarity** 0 = PWM Fault is active when cp+ > cp- 1 = PWM Fault is active when cp- > cp+
[1] CMPOUT	**Comparator Output Value** 0 = Comparator output is logical 0. 1 = Comparator output is logical 1.
[0] CMPEN	**Comparator Enable** 0 = Comparator is disabled. 1 = Comparator is enabled.

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.

Bit	Description (Continued)
[2] TXCOL	**Transmit Collision** This bit is set when a Transmit Collision occurs. This bit is cleared by writing a one to this bit. 0 = No collision has been detected. 1 = Transmit Collision has been detected.
[1] RXBUSY	**Receiver Busy** This bit is set when the receiver is receiving the data. Multi-master systems uses this bit to ensure the line is idle before sending the data. 0 = Receiver is idle. 1 = Receiver is receiving data.
[0] TXBUSY	**Transmitter Busy** This bit is set when the transmitter is sending the data. This bit is used to determine when to turn off a transceiver for RS-485 applications. 0 = Transmitter is idle. 1 = Transmitter is sending the data.

Control Register

The Control Register (DBGCTL), shown in Table 175, sets the mode of the serial interface.

Table 175. Control Register (DBGCTL)

Bits	7	6	5	4	3	2	1	0
Field	OCDLOCK	OCDEN	Reserved		CRCEN	UARTEN	ABCHAR	ABSRCH
RESET	1	1	00		1	0	0	1
R/W	R/W	R/W	R		R/W	R/W	R/W	R/W
Addr	FF_E086							

Bit	Description
[7] OCDLOCK	**On-Chip Debug Lock** This bit locks the Debug Control register so it cannot be written by the CPU. This bit is automatically set if the DBGUART option bit is in its default erased state (one). 0 = Debug Control register unlocked. 1 = Debug Control register locked.
[6] OCDEN	**On-Chip Debug Enable** This bit is set when the OCD is enabled. When this bit is set, received data is interpreted as debug command. To use the DBG pin as a UART or GPIO pin, this bit must be cleared to 0 by software. This bit cannot be written by the CPU if OCDLOCK is set. 0 = OCD is disabled. 1 = OCD is enabled.

Bit	Description (Continued)
[23:16] ADDR[23:16]	**Breakpoint Address** The address to match when generating a breakpoint.
[15:0] ADDR[15:0]	

Trace Control Register

The Trace Control Register (TRACECTL), shown in Table 179, is used to enable the Trace operation. It also selects the size of the trace buffer.

Table 179. Trace Control Register (TRACECTL)

Bits	7	6	5	4	3	2	1	0
Field	TRACEEN	Reserved				TRACESEL		
RESET	0	0	0	0	0	000		
R/W	R/W	R	R	R	R	R/W		
Addr	FF_E013							

Bit	Description
[7] TRACEEN	**Trace Enable** 0 = Trace is disabled. 1 = Traces is enabled
[6:3]	**Reserved** This bit is reserved and must be programmed to 0000.
[2:0] TRACESEL	**Trace Size Select** 000 – 128 Bytes (16 Events) 001 – 256 Bytes (32 Events) 010 – 512 Bytes (64 Events) 011 – 1024 Bytes (128 Events) 100 – 2048 Bytes (256 Events) 101 – 4096 Bytes (512 Events) 110 – 8192 Bytes (1024 Events) 111 – 16384 Bytes (2048 Events)

On-Chip Peripheral AC and DC Electrical Characteristics

Table 187 lists the POR and VBO electrical characteristics and timing. Table 188 lists the Reset and Stop Mode Recovery pin timing.

Table 187. POR and VBO Electrical Characteristics and Timing

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
V_{POR}	Power-On Reset voltage threshold	2.20	2.45	2.70	V	$V_{DD} = V_{POR}$
V_{VBO}	Voltage Brown-Out reset voltage threshold	2.15	2.40	2.65	V	$V_{DD} = V_{VBO}$
	V_{POR}–V_{VBO}		50	100	mV	
	Starting V_{DD} voltage to ensure valid POR	—	V_{SS}	—	V	
T_{ANA}	Power-On Reset analog delay	—	50	—	ms	$V_{DD} > V_{POR}$; T_{POR} Digital Reset delay follows T_{ANA}
T_{POR}	Power-On Reset digital delay	—	12	—	µs	66 IPO cycles
T_{VBO}	Voltage Brown-Out pulse rejection period	—	10	—	ms	$V_{DD} < V_{VBO}$ to generate a Reset
T_{RAMP}	Time for V_{DD} to transition from V_{SS} to V_{POR} to ensure valid Reset	0.10	—	100	ms	
I_{CC}	Supply current		500		µA	V_{DD} = 3.3 V.

Note:
1. Data in the typical column is from characterization at 3.3 V and 0°C. These values are provided for design guidance only and are not tested in production.

Table 188. Reset and Stop Mode Recovery Pin Timing

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
T_{RESET}	$\overline{\text{RESET}}$ pin assertion to initiate a System Reset	4	—	—	T_{CLK}	Not in Stop Mode. T_{CLK} = System Clock period.
T_{SMR}	Stop Mode Recovery pin Pulse Rejection Period	10	20	40	ns	RESET, DBG and GPIO pins configured as SMR sources.

Table 203. ZNEO Z16F Series Part Numbering

Part Number	Flash (Kbytes)	RAM (Kbytes)	External Interface	I/O	Multi-Channel timers with PWM	Standard Timers with PWM	ADC Inputs	I^2C Master/Slave	UART with LIN and IrDA	ESPI	Package
Automotive Temperature: –40°C to +125°C											
Z16F2811AL20AG	128	4	Yes	76	1	3	12	1	2	1	100-pin LQFP
Z16F2811FI20AG	128	4	Yes	60	1	3	12	1	2	1	80-pin QFP
Z16F2810FI20AG	128	4	No	60	1	3	12	1	2	1	80-pin QFP
Z16F2810AG20AG	128	4	No	46	1	3	12	1	2	1	64-pin LQFP
Z16F2810VH20AG	128	4	No	46	1	3	12	1	2	1	68-pin PLCC
Z16F6411AL20AG	64	4	Yes	76	1	3	12	1	2	1	100-pin LQFP
Z16F6411FI20AG	64	4	Yes	60	1	3	12	1	2	1	80-pin QFP
Z16F3211AL20AG	32	2	Yes	76	1	3	12	1	2	1	100-pin LQFP
Z16F3211FI20AG	32	2	Yes	60	1	3	12	1	2	1	80-pin QFP

ZNEO Z16F Series Development Tools	
Z16F2800100ZCOG	ZNEO® Z16F Series Development Kit
ZUSBSC00100ZACG	USB Smart Cable Accessory Kit
ZUSBOPTSC01ZACG	Opto-Isolated USB Smart Cable Accessory Kit
ZENETSC0100ZACG	Ethernet Smart Cable Accessory Kit