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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f2810fi20ag

IPO Trim Registers (Information Area Address 0021h and 0022h) 296
ADC Reference Voltage Trim (Information Area Address 0023h) 297
On-Chip Debugger 298
Architecture 298
Operation 299
On-Chip Debug Enable 299
Serial Interface 300
Serial Data Format 300
Baud Rate Generator 301
Auto-Baud Detector 302
Line Control 303
9-Bit Mode 303
Start Bit Flow Control 304
Initialization 304
Initialization during Reset 305
Debug Lock 305
Error Reset 305
DEBUG Halt Mode 306
Reading and Writing Memory 306
Reading Memory CRC 307
Breakpoints 307
Instruction Trace 308
On-Chip Debugger Commands 309
Cyclic Redundancy Check 313
Memory Cyclic Redundancy Check 313
UART Mode 313
Serial Errors 314
Interrupts 314
DBG Pin as a GPIO Pin 314
Control Register Definitions 315
Receive Data Register 315
Transmit Data Register 315
Baud Rate Reload Register 316
Line Control Register 316
Status Register 318
Control Register 319
OCD Control Register 321
OCD Status Register 323
Hardware Breakpoint Registers 324
Trace Control Register 325
Trace Address Register 326

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FF_E188	Port J Stop Mode Recovery Enable	PJSMRE	00	77
FF_E189-FF_E18F	Port J Reserved	—	—	—
GPIO Port K Base Address = FF_E190				
FF_E190	Port K Input Data	PKIN	XX	71
FF_E191	Port K Output Data	PKOUT	00	72
FF_E192	Port K Data Direction	PKDD	00	73
FF_E193	Port K High Drive Enable	PKHDE	00	74
FF_E194	Reserved	—	—	—
FF_E195	Port K Alternate Function Low	PKAFL	00	76
FF_E196	Port K Output Control	PKOC	00	76
FF_E197	Port K Pull-Up Enable	PKPUE	00	76
FF_E198	Port K Stop Mode Recovery Enable	PKSMRE	00	77
FF_E199-FF_E19F	Port K Reserved	—	—	—
Serial Channels Base Address = FF_E200				
LIN-UART 0 Base Address = FF_E200				
FF_E200	LIN-UART0 Transmit Data	U0TXD	XX	154
	LIN-UART0 Receive Data	U0RXD	XX	154
FF_E201	LIN-UART0 Status 0	U0STAT0	0000011Xb	155
FF_E202	LIN-UART0 Control 0	U0CTL0	00	160
FF_E203	LIN-UART0 Control 1	U0CTL1	00	164
FF_E204	LIN-UART0 Mode Select and Status	U0MDSTAT	00	162
FF_E205	LIN-UART0 Address Compare Register	U0ADDR	00	166
FF_E206	LIN-UART0 Baud Rate High Byte	U0BRH	FF	166
FF_E207	LIN-UART0 Baud Rate Low Byte	U0BRL	FF	167
FF_E208-FF_E20F	Reserved	—	XX	—
LIN-UART 1 Base Address = FF_E210				

XX = Undefined.

determination, be aware that the input data is captured on chip during the rising edge of the system clock prior to the RD signal deassertion. The Read signal ($\overline{RD}$) timing is shown for both NORMAL and ISA modes.

Table 16. External Interface Timing for a Read Operation, Normal Mode

Parameter	Abbreviation	Delay (ns) Minimum	Delay (ns) Maximum
T_1	X_{IN} Rise to Address Valid Delay		10
T_2	X_{IN} Rise to Address Output Hold Time	3	
T_3	Data Input Valid to X_{IN} Rise Setup Time		3
T_4	$\overline{RD}$ Rise to Data Input Hold Time	0	
T_5	X_{IN} Rise to $\overline{CS}$ Assertion Delay		10
T_6	X_{IN} Rise to $\overline{CS}$ Deassertion Hold Time	3	
T_7	X_{IN} Rise to $\overline{RD}$ Assertion Delay		10
T_8	X_{IN} Rise to $\overline{RD}$ Deassertion Hold Time	3	
T_9	$\overline{WAIT}$ Input Pin Assertion to X_{IN} Rise Setup Time	1	
T_{10}	$\overline{WAIT}$ Input Pin Deassertion to X_{IN} Rise Setup Time	1	
T_{11}	X_{IN} Rise to $\overline{DMAACK}$ Assertion Delay		10
T_{12}	X_{IN} Rise to $\overline{DMAACK}$ Deassertion Hold Time	3	
T_{13}	X_{IN} Rise to $\overline{BHEN}$ or $\overline{BLEN}$ Assertion Delay		10
T_{14}	X_{IN} Rise to $\overline{BHEN}$ or $\overline{BLEN}$ Deassertion Hold Time	3	

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.

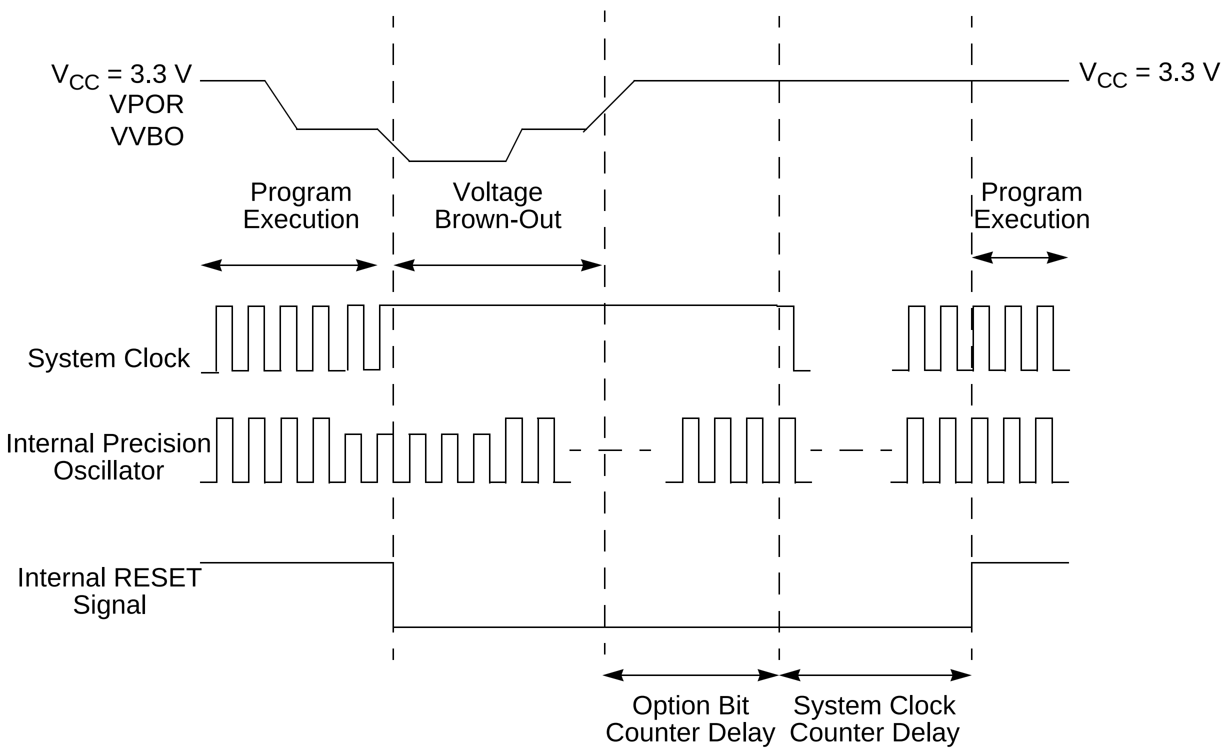

Figure 17. Voltage Brown-Out Reset Operation

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-

b. If parity is required and Multiprocessor Mode is not enabled, set the parity enable bit (PEN) and select either even- or odd parity (PSEL).

c. Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.

5. Check the TDRE bit in the LIN-UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If this register is empty, continue to Step 6. If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.

6. If in Multiprocessor Mode, write the LIN-UART Control 1 Register to select the outgoing address bit by setting the multiprocessor bit transmitter (MPBT) if sending an address byte; clear it if sending a data byte.

7. Write the data byte to the LIN-UART Transmit Data Register. The transmitter automatically transfers the data to the transmit shift register and transmits the data.

8. If Multiprocessor Mode is required and Multiprocessor Mode is enabled, make any changes to the multiprocessor bit transmitter (MPBT) value.

9. To transmit additional bytes, return to Step 4.

Transmitting Data Using Interrupt-Driven Method

The LIN-UART transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Observe the following steps to configure the LIN-UART for interrupt-driven data transmission:

1. Write to the LIN-UART Baud Rate High and Low Byte registers to set the appropriate baud rate.

2. Enable the LIN-UART pin functions by configuring the associated GPIO port pins for alternate function operation.

3. Execute a DI instruction to disable interrupts.

4. Write to the interrupt control registers to enable the LIN-UART transmitter interrupt and set the appropriate priority.

5. If Multiprocessor Mode is required, write to the LIN-UART Control 1 Register to enable MULTIPROCESSOR (9-Bit) Mode functions by setting the Multiprocessor Mode select bit (MPEN) to enable Multiprocessor Mode.

6. Write to the LIN-UART Control 0 Register to:

   a. Set the transmit enable bit (TEN) to enable the LIN-UART for data transmission

   b. Enable parity, if Multiprocessor Mode is not enabled and select either even or odd parity.

6. Read data from the LIN-UART Receive Data Register. If operating in MULTIPROCESSOR (9-Bit) Mode, further actions are required depending on the Multiprocessor Mode bits MPMD[1:0].
7. Return to Step 5 to receive additional data.

Receiving Data Using the Interrupt-Driven Method

The LIN-UART receiver interrupt indicates the availability of new data (as well as error conditions). Observe the following steps to configure the LIN-UART receiver for interrupt-driven operation:

1. Write to the LIN-UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the LIN-UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the interrupt control registers to enable the LIN-UART receiver interrupt and set the appropriate priority.
5. Clear the LIN-UART receiver interrupt in the applicable interrupt request register.
6. Write to the LIN-UART Control 1 Register to enable MULTIPROCESSOR (9-Bit) Mode functions:
    a. Set the Multiprocessor Mode select (MPEN) to enable Multiprocessor Mode.
    b. Set the Multiprocessor Mode bits, MPMD[1:0], to select the appropriate address matching scheme.
    c. Configure the LIN-UART to interrupt on received data and errors or errors only (interrupt on errors only is unlikely to be useful for ZNEO devices without a DMA block).
7. Write the device address to the Address Compare Register (automatic multiprocessor modes only).
8. Write to the LIN-UART Control 0 Register to:
    a. Set the receive enable bit (REN) to enable the LIN-UART for data reception
    b. Enable parity, if Multiprocessor Mode is not enabled and select either even- or odd-parity.
9. Execute an EI instruction to enable interrupts.

The LIN-UART is now configured for interrupt-driven data reception. When the LIN-UART receiver interrupt is detected, the associated ISR performs the following operations:

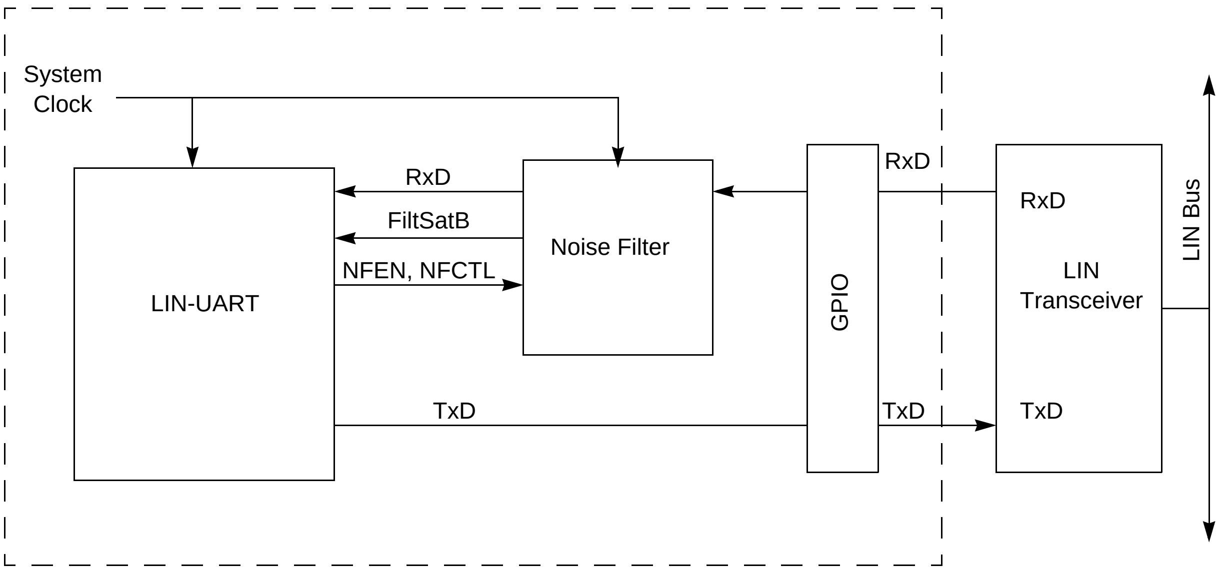

Figure 30. Noise Filter System Block Diagram

Operation

Figure 31 displays the operation of the noise filter with and without noise. The noise filter in this example is a 2-bit up/down counter, which saturates at `00b` and `11b`. A 2-bit counter is shown for convenience, the operation of wider counters is similar. The output of the filter switches from 1 to 0 when the counter counts down from `01b` to `00b` and switches from 0 to 1 when the counter counts up from `10b` to `11b`. The noise filter delays the received data by three system clock cycles.

The FiltSatB signal is checked when the filtered RxD is sampled in the center of the bit time. The presence of noise (FiltSatB = 1 at center of bit time) does not mean the sampled data is incorrect, just that the filter is not in its saturated state of all 1's or all 0's. If FiltSatB = 1 when RxD is sampled during a receive character, the NE bit in the ModeStatus[4:0] field is set. An indication of the level of noise in the network is obtained by observing this bit.

Bit	Description (Continued)
[5] OE	**Receive Data and Autobaud Overrun Error** This bit is set just as in normal UART operation if a receive data overrun error occurs. This bit is also set during LIN slave autobaud if the BRG counter overflows before the end of the autobaud sequence, indicating that the receive activity was not an autobaud character or the master baud rate is too slow. The ATB status bit will also be set in this case. This bit is cleared by reading the Receive Data Register. 0 = No autobaud or data overrun error occurred. 1 = An autobaud or data overrun error occurred.
[4] FE	**Framing Error** This bit indicates that a framing error (no Stop bit following data reception) is detected. Reading the Receive Data Register clears this bit. 0 = No framing error occurred. 1 = A framing error occurred.
[3] BRKD	**Break Detect** This bit is set in LIN Mode if (a) in LinSleep state and a break of at least 4 bit times occurred (Wake-up event) or (b) in Slave Wait Break state and a break of at least 11 bit times occurred (Break event), or (c) in Slave Active state and a break of at least 10 bit times occurs. Reading the Status 0 Register or the Receive Data Register clears this bit. 0 = No LIN break occurred. 1 = A LIN break occurred.
[2] TDRE	**Transmitter Data Register Empty** This bit indicates that the Transmit Data Register is empty and ready for additional data. Writing to the Transmit Data Register resets this bit. 0 = Do not write to the Transmit Data Register. 1 = The Transmit Data Register is ready to receive an additional byte to be transmitted.
[1] TXE	**Transmitter Empty** This bit indicates that the transmit shift register is empty and character transmission is finished. 0 = Data is currently transmitting. 1 = Transmission is complete.
[0] ATB	**LIN Slave AutoBaud Complete** This bit is set in LIN Slave Mode when an autobaud character is received. If the ABIEN bit is set in the LIN Control Register then a receive interrupt is generated when this bit is set. Reading the Status 0 Register clears this bit. This bit will be 0 in LIN Master Mode.

Bit	Description (Continued)
[2] BRGCTL	**Baud Rate Generator Control** This bit causes different LIN-UART behaviors, depending on whether the LIN-UART receiver is enabled (REN = 1 in the LIN-UART Control 0 Register). When the LIN-UART receiver is not enabled, this bit determines whether the baud rate generator issues interrupts. 0 = BRG is disabled. Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value. 1 = BRG is enabled and counting. The BRG generates a receive interrupt when it counts down to 0. Reads from the Baud Rate High and Low Byte registers return the current BRG count value. When the LIN-UART receiver is enabled, this bit allows reads from the baud rate registers to return the BRG count value instead of the Reload Value. 0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value. 1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the timers, there is no mechanism to latch the High byte when the Low byte is read.
[1] RDAIRQ	**Receive Data Interrupt Enable** 0 = Received data and receiver errors generates an interrupt request to the interrupt controller. 1 = Received data does not generate an interrupt request to the interrupt controller. Only receiver errors generate an interrupt request.
[0] IREN	**Infrared Encoder/Decoder Enable** 0 = Infrared encoder/decoder is disabled. LIN-UART operates normally. 1 = Infrared encoder/decoder is enabled. The LIN-UART transmits and receives data through the Infrared encoder/decoder.

2. Software asserts the TXI bit of the I^2C Control Register to enable Transmit interrupts.
3. The I^2C interrupt asserts because the I^2C Data Register is empty.
4. Software responds to the TDRE interrupt by writing the first Slave address byte (11110xx0). The least-significant bit must be 0 for the write operation.
5. Software asserts the Start bit of the I^2C Control Register.
6. The I^2C Controller sends the Start condition to the I^2C Slave.
7. The I^2C Controller loads the I^2C Shift Register with the contents of the I^2C Data Register.
8. When one bit of address is shifted out by the SDA signal, the Transmit interrupt asserts.
9. Software responds by writing the second byte of address into the contents of the I^2C Data Register.
10. The I^2C Controller shifts the rest of the first byte of address and write bit out the SDA signal.
11. The I^2C Slave sends an acknowledge by pulling the SDA signal Low during the next High period of SCL. The I^2C Controller sets the ACK bit in the I^2C Status Register.

    If the slave does not acknowledge the first address byte, the I^2C Controller sets the NCKI bit in the I^2C Status Register, sets the ACKV bit and clears the ACK bit in the I^2C State Register. Software responds to the Not Acknowledge interrupt by setting the Stop bit and clearing the TXI bit. The I^2C Controller flushes the second address byte from the data register, sends the Stop condition on the bus and clears the Stop and NCKI bits. The transaction is complete; ignore the following steps.
12. The I^2C Controller loads the I^2C Shift Register with the contents of the I^2C Data Register (2nd address byte).
13. The I^2C Controller shifts the second address byte out the SDA signal. When the first bit is sent, the Transmit interrupt asserts.
14. Software responds by writing the data to be written out to the I^2C Control Register.
15. The I^2C Controller shifts out the rest of the second byte of Slave address (or ensuing data bytes if looping) by the SDA signal.
16. The I^2C Slave sends an acknowledge by pulling the SDA signal Low during the next High period of SCL. The I^2C Controller sets the ACK bit in the I^2C Status Register.

    If the slave does not acknowledge, return to the second paragraph of Step 11.
17. The I^2C Controller shifts the data out by the SDA signal. After the first bit is sent, the Transmit interrupt asserts.
18. If more bytes remain to be sent, return to step 14.

ADC0 Data Low Bits Register

The ADC0 Data Low Bits Register contains the lower bits of the ADC0 output. Access to the ADC0 Data Low Bits Register is Read-Only.

Table 128. ADC0 Data Low Bits Register (ADC0D_L)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	ADC0D_L		Reserved					
RESET	X		X					
R/W	R		R					
Addr	FF–E503h							

Bit	Description
[7:6] ADC0D_L	**ADC0 Low Bits** 00–11b = These bits are the 2 least significant bits of the 10-bit ADC0 output. These bits are undefined after a Reset.
[5:0]	**Reserved** These bits are reserved and must be programmed to 0.

Sample Settling Time Register

The Sample Settling Time Register is used to program the length of time from the SAMPLE/$\overline{\text{HOLD}}$ signal to the Start signal, when the conversion begins. The number of clock cycles required for settling varies from system to system depending on the system clock period used. This register must be programmed to contain the number of clocks required to meet a 0.5 µs minimum settling time.

Table 129. Sample and Settling Time (ADCSST)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved			SST				
RESET	0	0	0	1	1	1	1	1
R/W	R			R/W				
Addr	FF–E504h							

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 0.
[4:0] SST	**Sample Settling Time** 00h–1Fh = Sample settling time in number of system clock periods to meet 0.5 µs minimum.

The operational amplifier is a two-input, one-output operational amplifier with a typical open loop gain of 10,000 (80 dB). The general-purpose input pin (OPINP) provides the noninverting amplifier input, while general-purpose input pin (OPINN) provides the inverting amplifier input. The output is available at the output pin (OPOUT).

The key operating characteristics of the operational amplifier are:

- Frequency compensated for unity gain stability
- Input common-mode-range from GND (0.0 V) to V_{DD} – 1 V
- Input offset voltage less than 15 mV
- Output voltage swing from GND + 0.1 V to V_{DD} – 0.1 V
- Input bias current less than 1 µA
- Operating the operational amplifier open loop (no feedback) effectively provides another on-chip comparator

Comparator Operation

The comparator output reflects the relationship between the noninverting input and the inverting (reference) input. If the voltage on the noninverting input is higher than the voltage on the inverting input, the comparator output is at a high state. If the voltage on the noninverting input is lower than the voltage on the inverting input, the comparator output is at a low state.

To operate, the comparator must be enabled by setting the CMPEN bit in the comparator and op-amp register to 1. In addition the CINP and CINN comparator input alternate functions must be enabled on their respective GPIO pins. For more information, see the GPIO Alternate Functions section on page 67.

The comparator does not automatically power-down. To reduce operating current when not in use, the comparator is disabled by clearing the CMPEN bit to 0.

Operational Amplifier Operation

To operate, the operational amplifier must be enabled by setting the OPEN bit in the comparator and op-amp register to 1. In addition, the OPINP, OPINN and OPOUT alternate functions must be enabled on their respective general-purpose I/O pins. For more information, see GPIO Alternate Functions.

The logical value of the operational amplifier output (OPOUT) is read from the Port 3 data input register if both the operational amplifier and input pin Schmitt trigger are enabled. For more information, see GPIO Alternate Functions. The operational amplifier generates an interrupt via the GPIO Port B3 input interrupt, if enabled.

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.

Table 145. Linked List Descriptor

Address	Even
LAR	CONTROL
LAR + 02h	TXLN
LAR + 04h	DAR High
LAR + 08h	SAR High
LAR + 0Ch	LAR High

DMA Control Bit Definitions

The following paragraphs explain the control bits of each DMA Channel.

DMAxEN

This bit if set by the CPU enables the DMA Channel for direct operation. Direct operation uses the addresses and transfer length, which has been directly written to the DMA Channel by software.

If this bit is set by a descriptor read then LINKED LIST Mode is enabled. Linked list operation starts when an address is written to the DMAxLAR. This write causes the DMA to read in the descriptor control value and addresses and place them in the DMA Channel.

LOOP

If the DMA is in LINKED LIST Mode and this bit is set to 1, it prevents the DMA from updating the descriptor when the buffer is closed. This bit is set to allow lists to loop on themselves without software intervention.

TXSIZE

The TXSIZE bits set the width of the transfer, as follows:

00 8-bit bytes are transferred on each DMA transfer. The destination and source addresses increment or decrement by 1 for each transfer if the DSTCTL and/or SRCCTL is selected for increment or decrement. The transfer length is decremented by 1 to allow 64Kbytes to be transferred.

01 A 16-bit word is transferred on each DMA transfer. The destination and source addresses increment or decrement by 2 if the DSTCTL and/or SRCCTL is selected for increment or decrement. In Word Mode, the transfer length is still decremented by 1 to allow 64Kwords to be transferred.

10 A 32-bit quad is transferred on each DMA transfer. The destination and source addresses increment or decrement by 4 if the DSTCTL and/or SRCCTL is selected for increment or decrement. In Quad Mode, the transfer length is still decremented by 1 to allow 64Kquads to be transferred.

External DMA Signals

Two external pins are associated with each DMA Channel capable of external transfers (Channel 3 does not have external DMA capability). They are active Low DMAxREQ and DMAxACK signals. DMAxACK signals are outputs and DMAxREQ are inputs. DMAxREQ must be asserted for a minimum of one system clock period to generate one DMA transfer. DMAxREQ is left asserted for multiple transactions and deasserted after DMAxACK asserts for the last appropriate transfer.

DMA Timing

External DMA Transfer

Figure 60 shows the read and write timing of external DMA transfers.

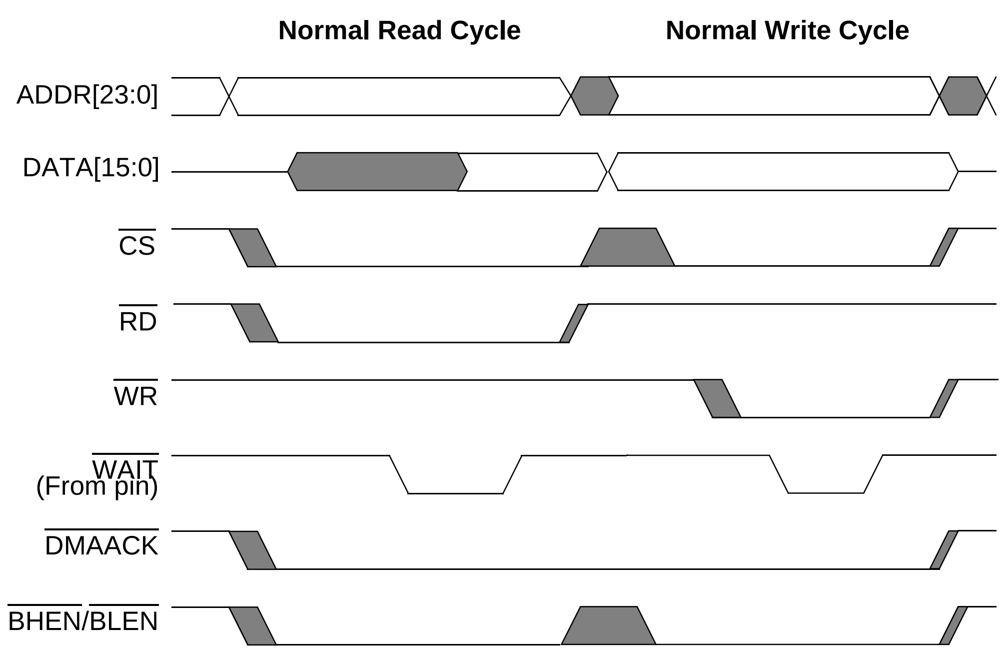

Figure 60. External DMA Transfer

On-Chip Debugger

The ZNEO® Z16F Series products have an integrated On-Chip Debugger (OCD) that provides the following features:

- Reading and writing memory
- Reading and writing CPU registers
- Execution of CPU instructions
- In-circuit programming and erasing of the Flash
- Unlimited number of software breakpoints
- Four hardware breakpoints
- Instruction execution trace
- Single-pin serial communication interface

Architecture

The OCD consists of two main blocks: the transmitter/receiver unit and the debug control logic. Figure 62 displays the architecture of the OCD.

Initialization during Reset

The OCD is initialized during reset by asserting the reset pin, sending the auto-baud character, and then issuing the read revision command. When the OCD is initialized during reset, the DBGHALT bit in the OCDCTL Register is automatically set.

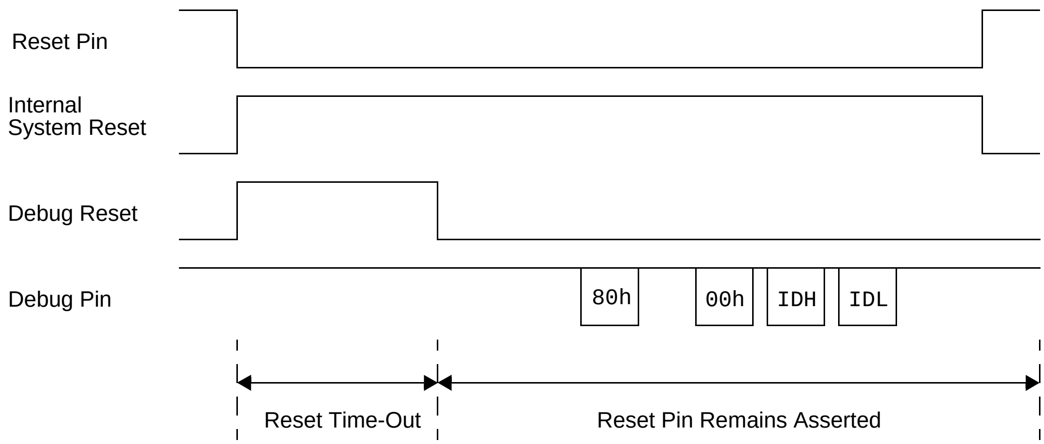

Figure 69. Initialization During Reset

Debug Lock

The interface has a locking mechanism to prevent user code from disabling the OCD and using the DBG pin as a UART or GPIO pin. The DBGLOCK bit in the DBGCTL Register prevents you from disabling the OCD and modifying any register that would inhibit communication with the OCD. The default state of the DBGLOCK bit is set accordingly to the DBGUART option bit.

In order to use the DBG pin as a UART or GPIO pin, you must program the DBGUART option bit to 0 so the OCDLOCK control bit is cleared after reset. After the control register is unlocked, software then clears the OCDEN control bit to use the DBG pin as aUART or GPIO pin.

If the DBGUART option bit is cleared and the OCDLOCK control bit is not set, the OCD is still locked before code has the chance to disable the OCD by initializing the Debugger during reset and writing the OCDLOCK control bit to 1.

Error Reset

The serial interface has an Auto-Reset mechanism that resets the serial interface when a Transmit Collision or Receive Framing Error is detected. When a Transmit Collision or Receive Framing Error is detected when OCDEN is set, the OCD aborts any command

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)

Oscillator Operation with an External RC Network

Figure 71 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

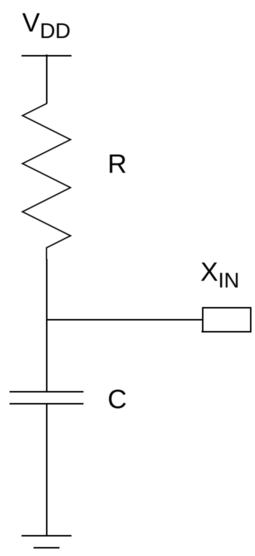

Figure 71. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 15 kΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 10 kΩ. The typical oscillator frequency is estimated from the values of the resistor (*R* in kΩ) and capacitor (*C* in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1 \times 10^6}{(1.5 \times R \times C)}$$

Figure 3 displays the typical (3.3 V and 25°C) oscillator frequency as a function of the capacitor (*C* in pF) employed in the RC network assuming a 15kΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board must be included in the estimation of the oscillator frequency.