Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ZNEO
Core Size	16-Bit
Speed	20MHz
Connectivity	EBI/EMI, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z16f3211fi20sg

Table 6. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FF_E210	LIN-UART1 Transmit Data	U1TXD	XX	154
	LIN-UART1 Receive Data	U1RXD	XX	154
FF_E211	LIN-UART1 Status 0	U1STAT0	0000011Xb	155
FF_E212	LIN-UART1 Control 0	U1CTL0	00	160
FF_E213	LIN-UART1 Control 1	U1CTL1	00	164
FF_E214	LIN-UART1 Mode Select and Status	U1MDSTAT	00	162
FF_E215	LIN-UART1 Address Compare Register	U1ADDR	00	166
FF_E216	LIN-UART1 Baud Rate High Byte	U1BRH	FF	166
FF_E217	LIN-UART1 Baud Rate Low Byte	U1BRL	FF	167
FF_E218-FF_E23F	Reserved	—	XX	—
I^2C Base Address = FF_E240				
FF_E240	I^2C Data	I2CDATA	00	227
FF_E241	I^2C Interrupt Status	I2CISTAT	80	228
FF_E242	I^2C Control	I2CCTL	00	229
FF_E243	I^2C Baud Rate High Byte	I2CBRH	FF	231
FF_E244	I^2C Baud Rate Low Byte	I2CBRL	FF	231
FF_E245	I^2C State	I2CSTATE	C0	232
FF_E246	I^2C Mode	I2CMODE	00	236
FF_E247	I^2C Slave Address	I2CSLVAD	00	237
FF_E248-FF_E25F	Reserved	—	XX	—
Enhanced Serial Peripheral Interface Base Address = FF_E260				
FF_E260	ESPI Data	ESPIDATA	XX	193
FF_E261	Reserved	—	XX	
FF_E262	ESPI Control	ESPICTL	00	194
FF_E263	ESPI Mode	ESPIMODE	00	197
FF_E264	ESPI Status	ESPISTAT	01	198
FF_E265	ESPI State	ESPISTATE	00	200
FF_E266	ESPI Baud Rate High Byte	ESPIBRH	FF	202
FF_E267	ESPI Baud Rate Low Byte	ESPIBRL	FF	202

XX = Undefined.

Table 7. External Interface Signals Description (Continued)

External Interface Signal	Direction
$\overline{CS1}$	Output
$\overline{CS2}$	Output
$\overline{CS3}$	Output
$\overline{CS4}$	Output
$\overline{CS5}$	Output

Chip Selects

The chip selects support connection of multiple memories and peripherals to the external interface. Figure 9 displays the memory map of the chip selects. The chip select boundaries are at fixed addresses. On-chip memory always have priority over external memory. Chip select 0 has the lowest priority and chip select 5 has the highest priority.

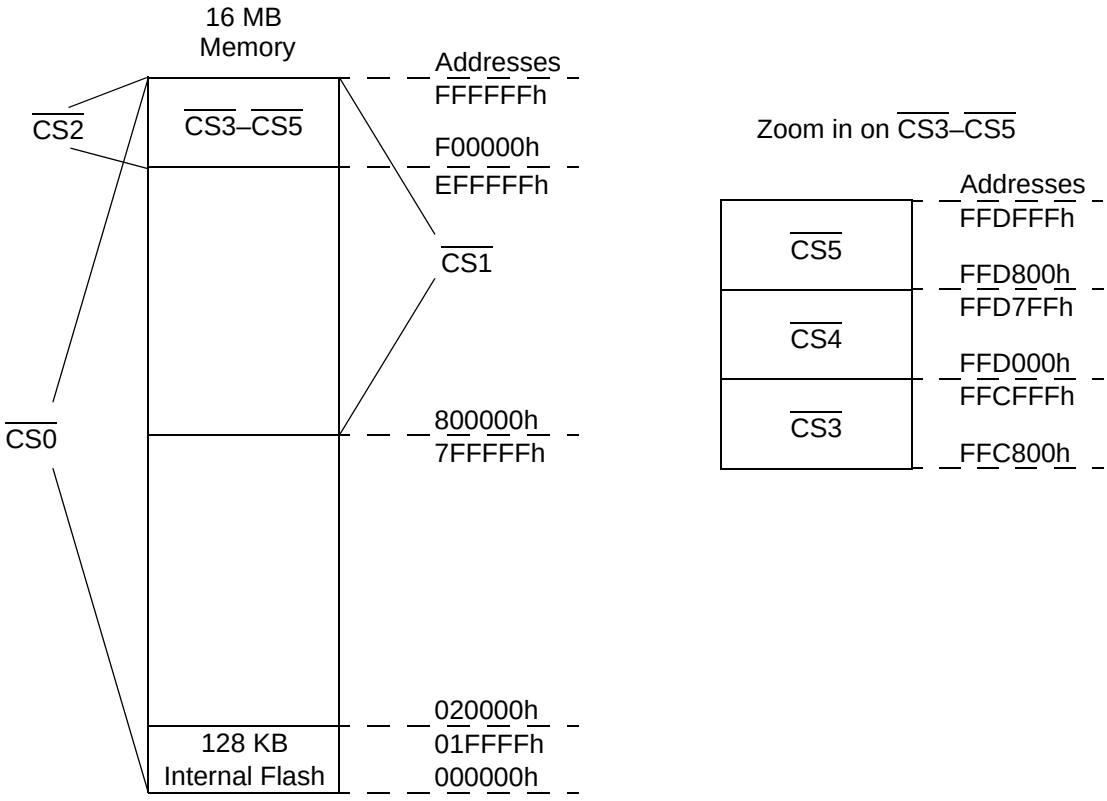


Figure 9. Chip Select Boundary Addressing with 128 KB Internal Flash

(V_{POR}) and has stabilized, the POR counter is enabled and counts 50 cycles of the IPO. At this point, the system clock is enabled and the POR counter counts a total of 16 system clock pulses. The device is held in the Reset state until the second POR counter sequence has timed out. After the ZNEO Z16F Series exits the POR state, the ZNEO CPU fetches the Reset vector. Following POR, the POR status bit in the Reset Status and Control Register is set to 1.

Figure 16 displays Power-On Reset operation. For the POR threshold voltage (V_{POR}), see Table 75 on page 343.

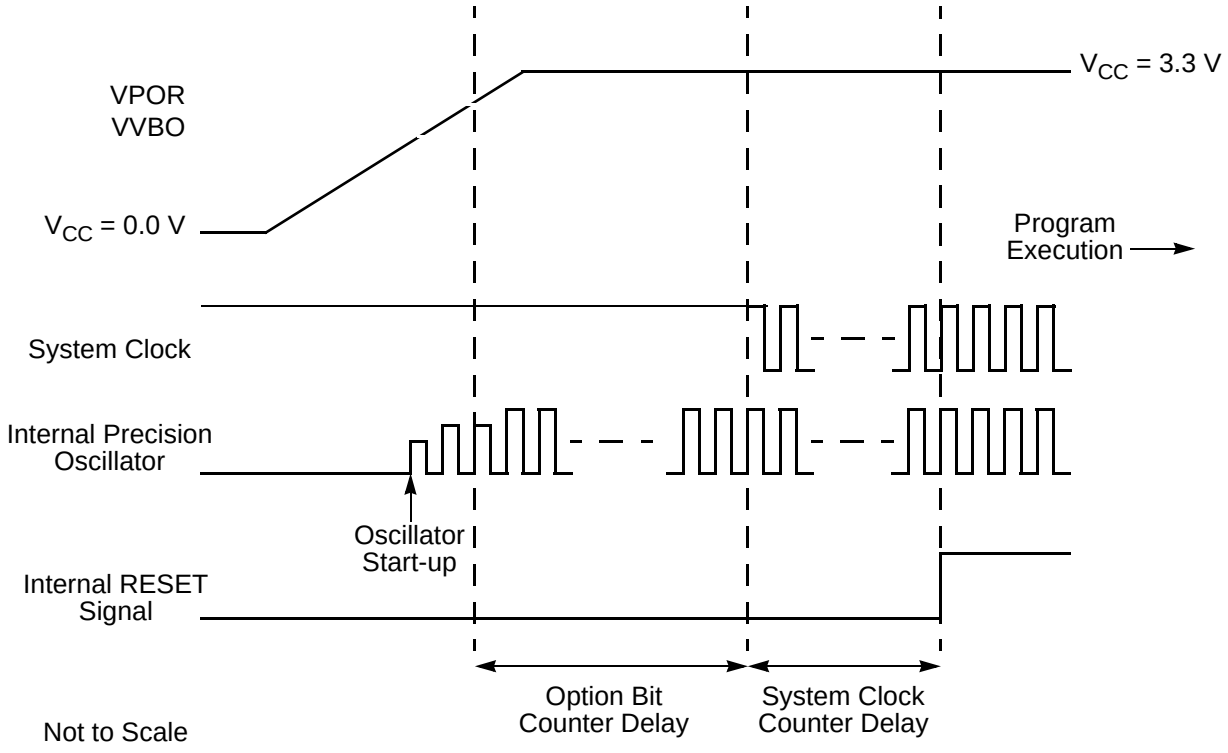


Figure 16. Power-On Reset Operation

Voltage Brown-Out Reset

The ZNEO Z16F Series provides Low Voltage Brown-Out (VBO) protection. The VBO circuit senses the supply voltage when it drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage remains below the POR voltage threshold (V_{POR}), the VBO holds the device in the Reset state.

When the supply voltage exceeds the V_{POR} and is stabilized, the device progresses through a full System Reset sequence, as described in the section <CrossRef> Power-On

Table 24. Port Alternate Function Mapping

Port	Pin	Alternate Function 1	Alternate Function 2	Alternate Function 3	External Interface
Port A	PA0	T0IN/$\overline{\text{T0OUT}}$	DMA0REQ	T0INPB	
	PA1	T0OUT	DMA0ACK		
	PA2	DE0	FAULTY		
	PA3	CTS0	FAULT0		
	PA4	RXD0	$\overline{\text{CS1}}$		
	PA5	TXD0	$\overline{\text{CS2}}$		
	PA6	SCL	$\overline{\text{CS3}}$		
	PA7	SDA	$\overline{\text{CS4}}$		
Port B	PB0/T0IN0	ANA0			
	PB1/T0IN1	ANA1			
	PB2/T0IN2	ANA2			
	PB3	ANA3/OPOUT			
	PB4	ANA4			
	PB5	ANA5			
	PB6	ANA6/OPINP/CINN			
	PB7	ANA7/OPINN			
Port C	PC0	T1IN/$\overline{\text{T1OUT}}$	DMA1REQ	CINN	
	PC1	T1OUT	DMA1ACK	COMPOUT	
	PC2	SS	$\overline{\text{CS4}}$		
	PC3	SCK	DMA2REQ		
	PC4	MOSI	DMA2ACK		
	PC5	MISO	$\overline{\text{CS5}}$		
	PC6	T2IN/$\overline{\text{T2OUT}}$	PWMH0		
	PC7	T2OUT	PWML0		
Port D	PD0	PWMH1	ADDR[20]		
	PD1	PWML1	ADDR[21]		
	PD2	PWMH2	ADDR[22]		
	PD3	DE1	ADDR[16]		
	PD4	RXD1	ADDR[18]		
	PD5	TXD1	ADDR[19]		
	PD6	CTS1	ADDR[17]		
	PD7	PWML2	ADDR[23]		

Port A-K Output Data Registers

The Port A-K Output Data registers, shown in Table 26, write output data to the pins.

Table 26. Port A-K Output Data Registers (P*x*OUT)

Bits	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
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_E101, FF_E111, FF_E121, FF_E131, FF_E141, FF_E151, FF_E161, FF_E171, FF_E181, FF_E191							

Bit	Description
[7:0] POUT[7:0]	**Port Output Data** These bits contain the data to be driven out from the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation. 0 = Drive a logical 0 (Low). 1= Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding port output control register bit to 1.

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.

Bit	Description (Continued)
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for timer 0. 1 = An interrupt request from timer 0 is awaiting service. Writing 1 to this bit resets it to 0.
[4] U0RXI	**UART 0 Receiver Interrupt Request** 0 = No interrupt request is pending for the UART 0 receiver. 1 = An interrupt request from the UART 0 receiver is awaiting service. Writing 1 to this bit resets it to 0.
[3] U0TXI	**UART 0 Transmitter Interrupt Request** 0 = No interrupt request is pending for the UART 0 transmitter. 1 = An interrupt request from the UART 0 transmitter is awaiting service. Writing 1 to this bit resets it to 0.
[2] I2CI	**I^2C Interrupt Request** 0 = No interrupt request is pending for the I^2C. 1 = An interrupt request from the I^2C is awaiting service. Writing 1 to this bit resets it to 0.
[1] SPII	**SPI Interrupt Request** 0 = No interrupt request is pending for the SPI. 1 = An interrupt request from the SPI is awaiting service. Writing 1 to this bit resets it to 0.
[0] ADCI	**ADC Interrupt Request** 0 = No interrupt request is pending for ADC. 1 = An interrupt request from ADC is awaiting service. Writing 1 to this bit resets it to 0.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) Register, shown in Table 45, stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the ZNEO CPU. If interrupts are globally disabled (polled interrupts), the ZNEO CPU reads the Interrupt Request 1 Register to determine, if any interrupt requests are pending. Writing 1 to the bits in this register clears the interrupt. The bits of this register are set by writing 1 to the Interrupt Request 1 Set Register (IRQ1SET) at address FF_E035h.

Table 51. IRQ1 Enable High Bit Register (IRQ1ENH)

Bits	7	6	5	4	3	2	1	0
Field	PAD7ENH	PAD6ENH	PAD5ENH	PAD4ENH	PAD3ENH	PAD2ENH	PAD1ENH	PAD0ENH
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_E036h							

Bit	Description
[7:0] PAD*x*ENH	Port A/D Bit[*x*] Interrupt Request Enable High Bit.

Table 52. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bits	7	6	5	4	3	2	1	0
Field	PAD7ENL	PAD6ENL	PAD5ENL	PAD4ENL	PAD3ENL	PAD2ENL	PAD1ENL	PAD0ENL
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_E037h							

Bit	Description
[7:0] PAD*x*ENL	Port A/D Bit[*x*] Interrupt Request Enable Low Bit.

IRQ2 Enable High and Low Bit Registers

The IRQ2 Enable High and Low Bit registers, shown in Tables 54 and 55, form a priority encoded enabling for interrupts in the Interrupt Request 2 Register. Priority is generated by setting bits in each register. Table 53 describes the priority control for IRQ2.

Table 53. IRQ2 Enable and Priority Encoding

IRQ2ENH[*x*]	IRQ2ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: *x* indicates the register bits from 0 through 7.

or DMA request is being serviced (set TEOF before or simultaneously with writing the last data byte). When the last bit of the last character is transmitted, the hardware will automatically deassert the SSV and TEOF bits. The second method is for software to directly clear the SSV bit after the transaction completes. If software clears the SSV bit directly, it is not necessary for software to also set the TEOF bit on the last transmit byte. After writing the last transmit byte, the end of the transaction is detected by waiting for the last RDRF interrupt or monitoring the TFST bit in the ESPI Status register.

The transmit underrun and receive overrun errors do not occur in an SPI Mode master. If the RDRF and TDRE requests have not been serviced before the current byte transfer completes, SCLK is paused until the data register is read and written. The transmit underrun and receive overrun errors will occur in a slave if the slave's software/DMA does not keep up with the master data rate. If a transmit underrun occurs in Slave Mode, the shift register in the slave is loaded with all 1s.

In the SPI Mode, the SCK is active only for the data transfer with one SCK period per bit transferred. If the SPI bus has multiple slaves, the slave select lines to all or one of the slaves must be controlled independently by software using GPIO pins.

Figure 38 displays multiple character transfer in SPI Mode. Note that while character 'n' is being transferred using the shift register, software/DMA responds to the receive request for character n-1 and the transmit request for character n+1.

0 for General Call Address). For a General Call Address, the I2C Controller automatically responds during the address acknowledge phase with the value in the NAK bit of the I2CCTL Register. If software processes the data bytes associated with the GCA bit, the IRM bit is optionally set following the SAM interrupt to allow software to examine each received data byte before deciding to set or clear the NAK bit. A Start byte will not be acknowledged (requirement the I2C specification).

Software Address Recognition Mode. To disable the hardware address recognition, the IRM bit must be set = 1 prior to the reception of the address byte(s). When IRM = 1 each received byte generates a receive interrupt (RDRF = 1 in the I2CISTAT Register). Software must examine each byte and determine whether to set or clear the NAK bit. The Slave holds SCL Low during the acknowledge phase until software responds by writing to the I2CCTL Register. The value written to the NAK bit is used by the controller to drive the I2C Bus, then releasing the SCL. The SAM and GCA bits are not set when IRM = 1 during the address phase, but the RD bit is updated based on the first address byte.

Slave 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

Slave Receive Transaction with 7-Bit Address

The data transfer format for writing data from Master to Slave in 7-bit address Mode is shown in Figure 47. The following procedure describes the I2C Master/Slave Controller operating as a Slave in 7-bit address Mode, receiving data from the bus Master.

S	Slave Address	W=0	A	Data	A	Data	A	Data	A/$\overline{A}$	P/S

Figure 48. Data Transfer Format, Slave Receive Transaction with 7-Bit Addressing

1. Software configures the controller for operation as a Slave in 7-Bit Address Mode as follows.
   a. Initialize the MODE field in the I2C Mode Register for either Slave Only Mode or Master/Slave Mode with 7-Bit Addressing.

14. The Slave I2C Controller asserts the Stop/Restart interrupt (set the SPRS bit in the I2CISTAT Register).
15. Software responds to the Stop interrupt by reading the I2CISTAT Register, clearing the SPRS bit.

DMA Control of I2C Transactions

The DMA engine is configured to support transmit and receive DMA requests from the I2C Controller. The I2C data interrupt requests must be disabled by setting the DMAIF bit in the I2C Mode Register and clearing the TXI bit in the I2C Control Register. This allows error condition interrupts to be handled by software while data movement is handled by the DMA engine.

The DMA interface on the I2C Controller is intended to support data transfer but not Master Mode address byte transfer. The Start, Stop and NAK bits must be controlled by software.

A summary of the sequence of I2C data transfer using the DMA follows.

Master Write Transaction with Data DMA

1. Configure the selected DMA Channel for I2C transmit. The IEOB bit must be set in the DMACTL Register for the last buffer to be transferred.
2. The I2C interrupt must be enabled in the interrupt controller to alert software of any I2C error conditions. A Not Acknowledge interrupt occurs on the last byte transferred.
3. The I2C Master/Slave must be configured as defined in the sections above describing Master Mode transactions. The TXI bit in the I2CCTL Register must be cleared.
4. Initiate the I2C transaction as described in the Master Address Only Transactions section on page 210, using the ACKV and ACK bits in the I2CSTATE Register to determine if the slave acknowledges.
5. Set the DMAIF bit in the I2CMODE Register.
6. The DMA transfers the data, which is to be transmitted to the slave.
7. When the DMA interrupt occurs, poll the I2CSTAT Register until the TDRE bit = 1 to ensure that the I2C Master/Slave hardware has commenced transmitting the most recent byte written by the DMA.
8. Set the Stop bit in the I2CCTL Register. The Stop bit is polled by software to determine when the transaction is actually completed.
9. Clear the DMAIF bit in the I2CMODE Register.

The following section describes the I2C Master/Slave Controller operating as a Slave in 10-Bit Addressing Mode, transmitting data to the bus master.

version already in progress, the Start bit is read to indicate ADC operation status (busy or available).

ADC Timing

Each ADC measurement consists of three phases:

1. Input sampling (programmable, minimum of 1.0 µs).
2. Sample-and-hold amplifier settling (programmable, minimum of 0.5 µs).
3. Conversion is 12 ADCLK cycles.

Figure 53 displays the timing of an ADC conversion.

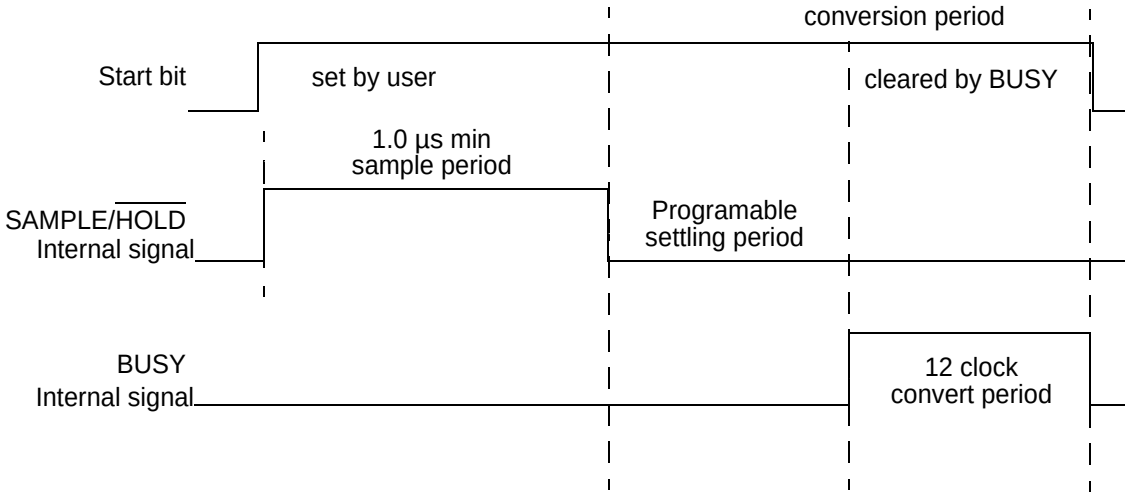


Figure 53. ADC Timing Diagram

Figure 54 displays the timing of convert period showing the 10 bit progression of the output.

ADC Reference Voltage Trim (Information Area Address 0023h)

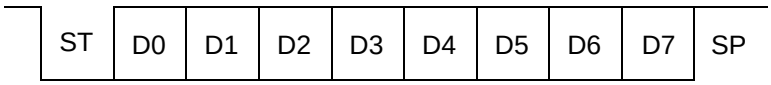
Table 167 defines the ADC Reference Voltage Trim settings, which are altered after reset by accessing the ADCTRIM Register.

Table 167. ADC Reference Voltage Trim (ADCTRIM)

Bits	7	6	5	4	3	2	1	0
Field	Reserved			ADCREF[4:0]				
RESET	L	L	L	L	L	L	L	L
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addr	FFFF_FF27							

Note: L = Loaded at Reset. R/W = Read/Write. This register is loaded from Information area on Reset.

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 1.
[4:0] ADCREF[4:0]	**ADC Reference Trim** These bits are used to trim the ADC reference voltage generator. If the part is not going to be trimmed, the value of this register must be F0h.



ST = Start Bit
SP = Stop Bit
D0–D7 = Data Bits

Figure 65. OCD Serial Data Format

Each bit time is of same length. The bit period is set by the baud rate generator.

When the transmitter sends a character, it first sends a Low start bit. The transmitter then waits one bit time. After the start bit is sent, the transmitter sends the next data bit. The transmitter sends each data bit in turn, waiting one full bit time before sending the next data bit. After the last data bit is sent, the transmitter sends a high stop bit for one bit time.

The receiver looks for the falling edge of the start bit. After the receiver sees the start bit is Low, it waits one half bit time and samples the middle of the start bit. If the middle of the start bit is High, the receiver considers this as a false start bit. The receiver ignores a false start bit and searches for another falling edge. If the middle of the start bit is Low, the receiver considers the start bit valid. The receiver will wait a full bit time from the middle of the start bit to sample the next data bit. The next data bit is sampled in the middle of the bit period. The receiver repeats this operation for each data bit, waiting one full bit time to between sampling each data bit.

After the receiver has sampled the last data bit, it waits one full bit time and sample the middle of the stop bit. If the stop bit is Low, the receiver detects a framing error.

If the stop bit is High, the data was correctly framed between a start and stop bit. After the receiver samples the middle of the stop bit, it begins searching for another start bit. To correct for any bit skew due to error between the transmit and receive baud rate clocks, the receiver does not wait for the full stop bit to be received before searching for the next start bit.

Baud Rate Generator

The baud rate generator (BRG) is used to generate a bit clock for transmit and receive operations. The BRG reload register is automatically configured by the auto-baud detector, or it is written by software.

The value in the BRG reload register is calculated as:

$$\text{BAUD RELOAD VALUE} = \frac{\text{SYSTEM CLOCK}}{\text{BAUD RATE}} \times 8$$

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

Figure 73 displays the typical current consumption while operating at 3.3 V at 30 ºC versus the system clock frequency.

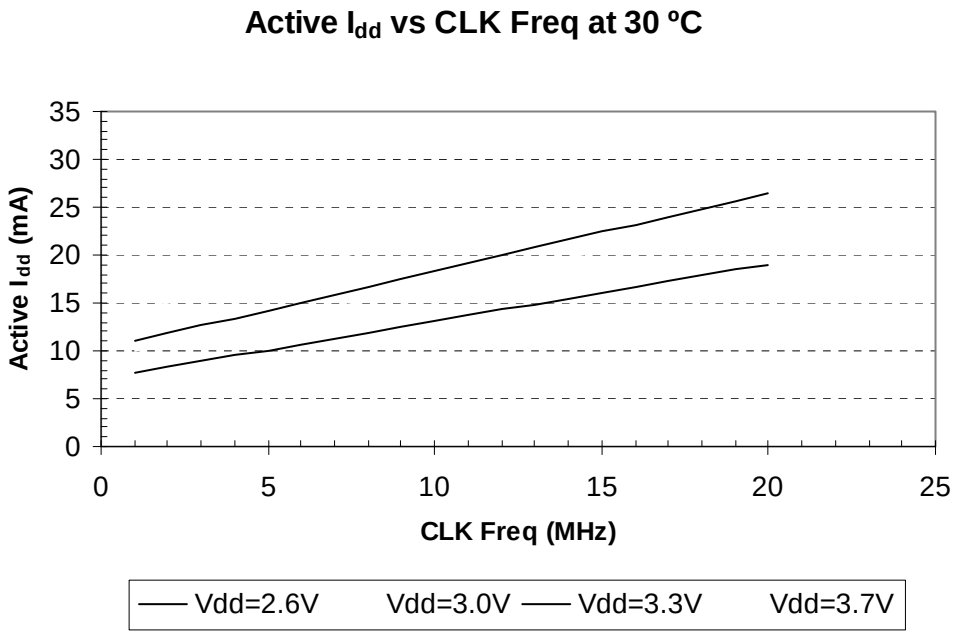


Figure 73. Typical I_{DD} Versus System Clock Frequency

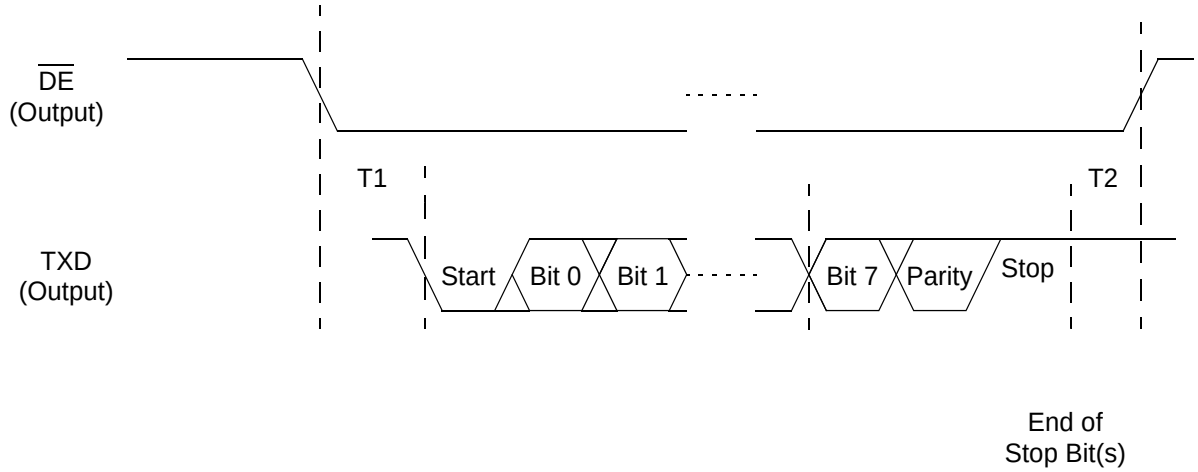
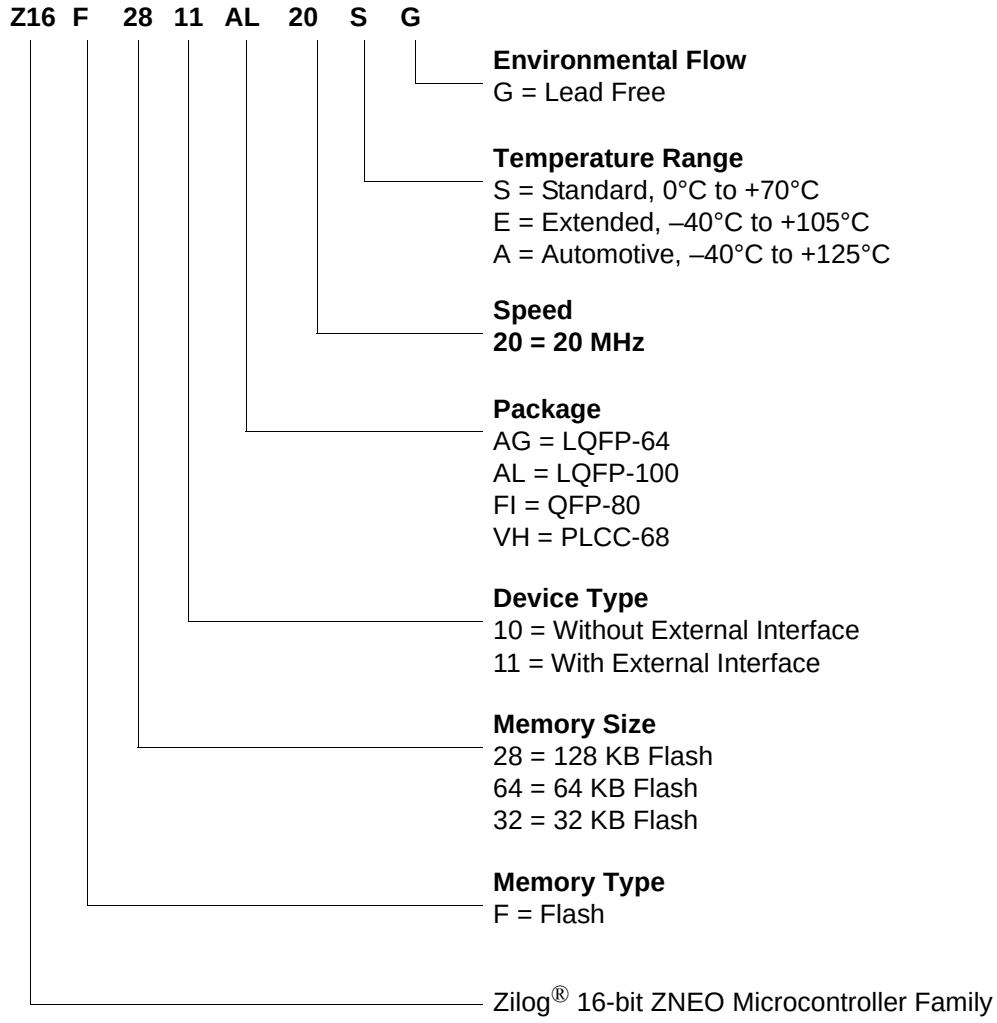


Figure 81. UART Timing without $\overline{CTS}$

Table 201. UART Timing without $\overline{CTS}$

		Delay (ns)	
Parameter	**Description**	**Min**	**Max**
T_1	$\overline{DE}$ Assertion to TXD Falling Edge (Start) Delay	1 Bit period	1 Bit period + 1 * X_{IN} period
T_2	End of Stop Bit(s) to $\overline{DE}$ Deassertion Delay	1 * X_{IN} period	2 * X_{IN} period

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.