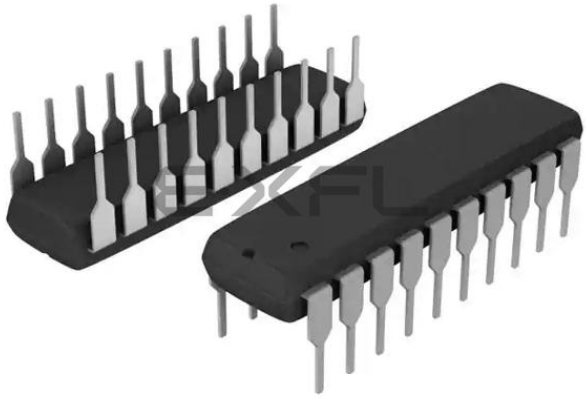


Welcome to E-XFL.COM

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	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0880ph020sg

List of Figures

On Reset, control registers within the Register File that have a defined Reset value are loaded with their Reset values. Other control registers (including the Stack Pointer, Register Pointer and Flags) and general-purpose RAM are not initialized and undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address.

Because the control registers are reinitialized by a System Reset, the system clock after reset is always the 11MHz IPO. User software must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

5.2. Reset Sources

Table 10 lists the possible sources of a System Reset.

Table 10. Reset Sources and Resulting Reset Type

Operating Mode	Reset Source	Special Conditions
NORMAL or HALT Mode	Power-On Reset	Reset delay begins after supply voltage exceeds POR level
	Voltage Brown-Out	Reset delay begins after supply voltage exceeds VBO level
	Watchdog Timer time-out when configured for Reset	None
	$\overline{\text{RESET}}$ pin assertion	All reset pulses less than three system clocks in width are ignored, see the Electrical Characteristics chapter on page 349.
	On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)	System Reset, except the OCD is unaffected by reset
STOP Mode	Power-On Reset	Reset delay begins after supply voltage exceeds POR level
	Voltage Brown-Out	Reset delay begins after supply voltage exceeds VBO level
	$\overline{\text{RESET}}$ pin assertion	All reset pulses less than the specified analog delay is ignored, see the Electrical Characteristics chapter on page 349.
	DBG pin driven Low	None

Table 19. Port Alternate Function Mapping, 40-/44-Pin Parts[1,2] (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port C	PC0	Reserved		AFS1[0]: 0
		ANA4/C0INP/LED	ADC or Comparator 0 Input (P), or LED drive	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/C0INN/LED	ADC or Comparator 0 Input (N), or LED Drive	AFS1[1]: 1
	PC2	$\overline{SS}$	SPI Slave Select	AFS1[2]: 0
		ANA6/LED	ADC Analog Input or LED Drive	AFS1[2]: 1
	PC3	MISO	SPI Master In Slave Out	AFS1[3]: 0
		LED	LED drive	AFS1[3]: 1
	PC4	MOSI	SPI Master Out Slave In	AFS1[4]: 0
		LED	LED Drive	AFS1[4]: 1
	PC5	SCK	SPI Serial Clock	AFS1[5]: 0
		LED	LED Drive	AFS1[5]: 1
	PC6	T2IN/$\overline{T2OUT}$	Timer 2 Input/Timer2 Output Complement	AFS1[6]: 0
		LED	LED Drive	AFS1[6]: 1
	PC7	T2OUT	Timer 2 Output	AFS1[7]: 0
		LED	LED Drive	AFS1[7]: 1

Notes:
1. Because there are at most two choices of alternate functions for some pins in Ports A–C, the Alternate Function Set Register (AFS2) is implemented but not used to select the function. The alternate function selection must also be enabled, as described in the Port A–E Alternate Function Subregisters section on page 61.
2. Because there is only one alternate function for each Port D and Port E pin, the Alternate Function Set registers are not implemented for Ports D and E. Enabling the alternate function selections automatically enables the associated alternate function, as described in the Port A–E Alternate Function Subregisters section on page 61.
3. This timer function is only available in the 44-pin package; its alternate functions are reserved in the 40-pin package.

Example 1. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
AND r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 2 to clear bits in the Interrupt Request 0 Register:

Example 2. A good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

8.3.4. Software Interrupt Assertion

Program code can generate interrupts directly. Writing a 1 to the correct bit in the Interrupt Request Register triggers an interrupt (assuming that the interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the Interrupt Request Register is automatically cleared to 0.

! **Caution:** Zilog recommends not using a coding style to generate software interrupts by setting bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 3, which follows.

Example 3. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 4 to set bits in the Interrupt Request registers:

Example 4. A good coding style that avoids lost interrupt requests:

```
ORX IRQ0, MASK
```

8.4. Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the Primary Oscillator Fail Trap and the Watchdog Oscillator Fail Trap, the Interrupt Control registers enable individual interrupts, set interrupt priorities and indicate interrupt requests.

3. Upon reaching the reload value, the timer outputs a pulse on the Timer Output pin, generates an interrupt and resets the count value in the Timer High and Low Byte registers to 0001H. The period of the output pulse is a single timer clock. The TPOL bit also sets the polarity of the output pulse.
4. The Timer now idles until the next trigger event.

In TRIGGERED ONE-SHOT Mode, the timer clock always provides the timer input. The timer period is shown in the following equation:

$$\text{Triggered ONE-SHOT Mode Time-Out Period (s)} = \frac{(\text{Reload Value - Start Value}) \times \text{Prescale}}{\text{Timer Clock Frequency (Hz)}}$$

Table 53 provides an example initialization sequence for configuring Timer 0 in TRIGGERED ONE-SHOT Mode and initiating operation.

Table 53. TRIGGERED ONE-SHOT Mode Initialization Example

Register	Value	Comment
T0CTL0	E0H	TMODE[3:0] = 1011B selects TRIGGERED ONE-SHOT Mode. TICONFIG[1:0] = 11B enables interrupts on Timer reload only. CSC = 0 selects the Timer Input (Trigger) from the GPIO pin. PWMD[2:0] = 000B has no effect. INPCAP = 0 has no effect. TEN = 0 disables the timer. TPOL = 0 enables triggering on rising edge of Timer. Input and sets Timer Out signal to 0. PRES[2:0] = 000B sets prescaler to divide by 1. TCLKS = 1 sets 32kHz peripheral clock as the Timer clock source.
T0CTL1	03H	
T0CTL2	01H	
T0H	00H	Timer starting value = 0001H.
T0L	01H	
T0RH	ABH	Timer reload value = ABCDH.
T0RL	CDH	
PAADDR	02H	Selects Port A Alternate Function control register.
PACTL[1:0]	11B	PACTL[0] enables Timer 0 Input Alternate function. PACTL[1] enables Timer 0 Output Alternate function.
IRQ0ENH[5]	0B	Disables the Timer 0 interrupt.
IRQ0ENL[5]	0B	

In CONTINUOUS Mode, the timer clock always provides the timer input. The timer period is calculated using the following equation:

$$\text{CONTINUOUS Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{Timer Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT Mode equation must be used to determine the first time-out period.

9.2.3.4. COUNTER Mode

In COUNTER Mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO port pin Timer Input alternate function. The TPOL bit in the Timer Control 1 Register selects whether the count occurs on the rising edge or the falling edge of the Timer Input signal. In COUNTER Mode, the prescaler is disabled.

! Caution: The input frequency of the Timer Input signal must not exceed one-fourth the timer clock frequency.

Upon reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, 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.

Observe the following steps to configure a timer for COUNTER Mode and initiate the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer.
   - Configure the timer for COUNTER Mode.
   - Select either the rising edge or falling edge of the Timer Input signal for the count. This also sets the initial logic level (High or Low) for the Timer Output Alternate Function. However, the Timer Output function is not required to be enabled.
2. Write to the Timer Control 2 Register to choose the timer clock source.

12.3.6.1. Multiprocessor Control Register

When MSEL = 000b, the Multiprocessor Control Register, shown in Table 90, provides control for UART MULTIPROCESSOR Mode, IRDA Mode and Baud Rate Timer Mode as well as other features that can apply to multiple modes.

Table 90. Multiprocessor Control Register (U0CTL1 = F43H with MSEL = 000b)

Bit	7	6	5	4	3	2	1	0
Field	MPMD1	MPEN	MPMD0	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
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
Address	F43H, F4BH							

Note: R/W = Read/Write.

Bit Position	Value	Description
[7,5] MPMD[1:0]	**MULTIPROCESSOR (9-bit) Mode**	
	00	The LIN-UART generates an interrupt request on all data and address bytes.
	01	The LIN-UART generates an interrupt request only on received address bytes.
	10	The LIN-UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs.
	11	The LIN-UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register.
[6] MPEN	**Multiprocessor Enable** This bit is used to enable MULTIPROCESSOR (9-bit) Mode.	
	0	Disable MULTIPROCESSOR (9-bit) Mode.
	1	Enable MULTIPROCESSOR (9-bit) Mode.
[4] MPBT	**Multiprocessor Bit Transmit** This bit is applicable only when MULTIPROCESSOR (9-bit) Mode is enabled.	
	0	Send a 0 in the multiprocessor bit location of the data stream (9th bit).
	1	Send a 1 in the multiprocessor bit location of the data stream (9th bit).
[3] DEPOL	**Driver Enable Polarity**	
	0	DE signal is active High.
	1	DE signal is active Low.

Table 118. I²C Master/Slave Controller Registers (Continued)

Name	Abbreviation	Description
I²C Baud Rate High	I2CBRH	High byte of baud rate generator initialization value.
I²C Baud Rate Low	I2CBRL	Low byte of baud rate generator initialization value.
I²C State	I2CSTATE	State register.
I²C Mode	I2CMODE	Selects MASTER or SLAVE modes, 7-bit or 10-bit addressing; configure address recognition, define slave address bits [9:8].
I²C Slave Address	I2CSLVAD	Defines slave address bits [7:0].

17.2. Operation

The I²C Master/Slave Controller operates in MASTER/SLAVE Mode, SLAVE ONLY Mode, or with master arbitration. In MASTER/SLAVE Mode, it can be used as the only Master on the bus or as one of the several masters on the bus, with arbitration. In a Multi-Master environment, the controller switches from MASTER to SLAVE Mode on losing arbitration.

Though slave operation is fully supported in MASTER/SLAVE Mode, if a device is intended to operate only as a slave, then SLAVE ONLY mode can be selected. In SLAVE ONLY mode, the device will not initiate a transaction, even if the software inadvertently sets the start bit.

17.2.1. SDA and SCL Signals

The I²C circuit sends all addresses, Data and Acknowledge signals over the SDA line, with most-significant bit first. SCL is the clock for the I²C bus. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The Master is responsible for driving the SCL clock signal. During the Low period of the clock, a slave can hold the SCL signal Low to suspend the transaction if it is not ready to proceed. The Master releases the clock at the end of the Low period and notices that the clock remains Low instead of returning to a High level. When the slave releases the clock, the I²C master continues the transaction. All data is transferred in bytes; there is no limit to the amount of data transferred in one operation. When transmitting address, data, or an Acknowledge, the SDA signal changes in the middle of the Low period of SCL. When receiving address, Data, or an Acknowledge; the SDA signal is sampled in the middle of the High period of SCL.

A low-pass digital filter can be applied to the SDA and SCL receive signals by setting the Filter Enable (FILTEN) bit in the I²C Control Register. When the filter is enabled, any glitch that is less than a system clock period in width will be rejected. This filter should be

GCE bit = 1 in the I2CMODE Register. The software checks the RD bit in the I2CISTAT Register to determine if the transaction is a Read or Write transaction. The General Call Address and STARTBYTE address 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 via 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 first byte of the transaction is compared against {11110,SLA[9:8], R/$\overline{W}$} and the second byte is compared against SLA[7:0].

17.2.2.4. Arbitration Lost Interrupts

Arbitration Lost interrupts (ARBLST bit = 1 in I2CISTAT) occur when the I^2C controller is in MASTER Mode and loses arbitration (outputs 1 on SDA and receives 0 on SDA). The I^2C controller switches to SLAVE Mode when this instance occurs. This bit clears automatically when the I2CISTAT Register is read.

17.2.2.5. Stop/Restart Interrupts

A Stop/Restart event interrupt (SPRS bit = 1 in I2CISTAT) occurs when the I^2C controller is operating in SLAVE Mode and a stop or restart condition is received, indicating the end of the transaction. The RSTR bit in the I^2C State Register indicates whether the bit is set due to a stop or restart condition. When a restart occurs, a new transaction by the same master is expected to follow. This bit is cleared automatically when the I2CISTAT Register is read. The Stop/Restart interrupt occurs only on a selected (address match) slave.

17.2.2.6. Not Acknowledge Interrupts

Not Acknowledge interrupts (NCKI bit = 1 in I2CISTAT) occur in MASTER Mode when Not Acknowledge is received or sent by the I^2C controller and the start or stop bit is not set in the I^2C Control Register. In MASTER Mode, the Not Acknowledge interrupt clears by setting the start or stop bit. When this interrupt occurs in MASTER Mode, the I^2C controller waits until it is cleared before performing any action. In SLAVE Mode, the Not Acknowledge interrupt occurs when a Not Acknowledge is received in response to data sent. The NCKI bit clears in SLAVE Mode when software reads the I2CISTAT Register.

17.2.2.7. General Purpose Timer Interrupt from Baud Rate Generator

If the I^2C controller is disabled (IEN bit in the I2CCTL Register = 0) and the BIRQ bit in the I2CCTL Register = 1, an interrupt is generated when the baud rate generator (BRG) counts down to 1. The baud rate generator reloads and continues counting, providing a periodic interrupt. None of the bits in the I2CISTAT Register are set, allowing the BRG in the I^2C Controller to be used as a general-purpose timer when the I^2C Controller is disabled.

```
nop      ; wait for output to settle
ldx IRQ0,#0 ; clear any spurious interrupts pending
ei
```

18.2. Comparator Control Register Definitions

This section defines the features of the following Comparator Control registers.

Comparator 0 Control Register: see page 257

Comparator 1 Control Register: see page 258

18.2.1. Comparator 0 Control Register

The Comparator 0 Control Register (CMP0), shown in Table 130, configures the Comparator 0 inputs and sets the value of the internal voltage reference.

Table 130. Comparator 0 Control Register (CMP0)

Bits	7	6	5	4	3	2	1	0
Field	INPSEL	INNSEL	REFLVL				TIMTRG	
Reset	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F90H							

Bit	Description
[7] INPSEL	**Signal Select for Positive Input** 0 = GPIO pin used as positive comparator 0 input. 1 = Temperature sensor used as positive comparator 0 input.
[6] INNSEL	**Signal Select for Negative Input** 0 = Internal reference disabled, GPIO pin used as negative comparator 0 input. 1 = Internal reference enabled as negative comparator 0 input.

Table 141. Flash Option Bits at Program Memory Address 0001H

Bits	7	6	5	4	3	2	1	0
Field	EXTLTMG		FLASH_WR_PRO_EN	EXTL_AO	Reserved		X2_Mode	X2TL_AO
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Program Memory 0001H							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7:6] EXTLTMG	**External Crystal Reset Timing** 00 = 500 Internal Precision Oscillator Cycles. 01 = 1000 Internal Precision Oscillator Cycles. 10 = 5000 Internal Precision Oscillator Cycles. 11 = 10,000 Internal Precision Oscillator Cycles.
[5] FLASH_WR_PRO_EN	**Flash Write Operation Protect** 0 = Flash write protect disable. 1 = Flash write protect with internal LVD.
[4] EXTL_AO	**External Crystal Always ON** 0 = Crystal oscillator is enabled during Reset, resulting in longer reset timing. 1 = Crystal oscillator is disabled during Reset, resulting in shorter reset timing. Note: This bit determines the state of the external crystal oscillator at Reset. Its selection as system clock must be performed in the Oscillator Control Register (OSCCTL0).
[3:2]	Reserved; must be 00.
[1] X2_Mode	**Secondary Crystal Mode Select** 0 = External clock input. 1 = External 32kHz watch crystal.
[0] X2TL_AO	**Secondary Crystal Always ON** 0 = Secondary Crystal Oscillator is enabled during reset. 1 = Secondary Crystal Oscillator is disabled during reset.

Note: This bit determines state of the Secondary Crystal Oscillator at Reset. Its selection as peripheral clock must be performed in the Peripheral Control (for example, Timer Control2) register.

21.2.4.7. Trim Bit Address 0007H

In the Trim Option Bits Register at address 0007H and shown in Table 155, all bits are reserved.

Table 155. Trim Option Bits at 0007H (TFilter0)

Bits	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0027H							

Note: U = Unchanged by Reset. R/W = Read/Write.

21.2.4.8. Trim Bit Address 0008H

In the Trim Option Bits Register at address 0008H and shown in Table 156, all bits are reserved.

Table 156. Trim Option Bits at 0008H (TFilter1)

Bits	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0028H							

Note: U = Unchanged by Reset. R/W = Read/Write.

22.2.1. Byte Write

To write a byte to the NVDS array, the user code must first push the address, then the data byte onto the stack. The user code issues a CALL instruction to the address of the Byte Write routine (0x43FD). At the return from the subroutine, the write status byte resides in working register R0. The bit fields of this status byte are defined in Table 159. Also, the user code should pop the address and data bytes off the stack.

The write routine uses 16 bytes of stack space in addition to the two bytes of address and data pushed by the user code. Sufficient memory must be available for this stack usage.

Because of the flash memory architecture, NVDS writes exhibit a nonuniform execution time. In general, a write takes 136µs (assuming a 20MHz system clock). For every 200 writes, however, a maintenance operation is necessary. In this rare occurrence, the write takes up to 58ms to complete. Slower system clock speeds result in proportionally higher execution times.

NVDS byte writes to invalid addresses (those exceeding the NVDS array size) have no effect. Illegal write operations have a 7µs execution time.

Table 159. Write Status Byte

Bits	7	6	5	4	3	2	1	0
Field	Reserved					FE	IGADDR	WE
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:3]	Reserved; must be 0.
[2] FE	**Flash Error** If Flash error is detected, this bit is set to 1.
[1] IGADDR	**Illegal Address** When NVDS byte writes to invalid addresses (those exceeding the NVDS array size) occur, this bit is set to 1. Note: When the NVDS array size is 256 bytes, there is no address exceeding the size; therefore the IGADDR bit cannot be used.
[0] WE	**Write Error** A failure occurs during writing data into Flash. When writing data into a certain address, a read back operation is performed. If the read back value is not the same as the value written, this bit is set to 1.

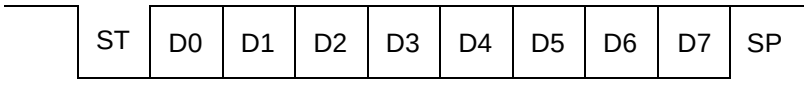
22.2.2. Byte Read

To read a byte from the NVDS array, user code must first push the address onto the stack. User code issues a CALL instruction to the address of the byte-read routine (0x4000). At

- Voltage Brown-Out reset
- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a Reset
- Driving the DBG pin Low when the device is in STOP Mode initiates a System Reset

23.2.2. OCD Data Format

The On-Chip Debugger (OCD) interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 start bit, 8 data bits (least-significant bit first) and 1 stop bit (see Figure 59).



ST = Start bit
SP = Stop bit
D0–D7 = Data bits

Figure 59. OCD Data Format

23.2.3. OCD Autobaud Detector/Generator

To run over a range of baud rates (bits per second) with various system clock frequencies, the On-Chip Debugger has an Autobaud Detector/Generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the host is the character 80H. The character 80H contains eight continuous bits Low (one start bit plus 7 data bits). The Autobaud Detector measures this period and sets the OCD Baud Rate Generator accordingly.

The Autobaud Detector/Generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. If the data can be synchronized with the system clock, the autobaud generator can run as high as the system clock frequency (1 clock/bit). The maximum recommended baud rate is the system clock frequency divided by 8. Table 162 lists the minimum and recommended maximum baud rates for sample crystal frequencies.

when it reaches the maximum count of FFFFH. The OCDCNTR Register automatically resets itself to 0000H when the OCD exits DEBUG mode if it is configured to count clock cycles between breakpoints.

If the OCDCNTR Register is configured to generate a BRK when it counts down to zero, it will not be reset when the CPU starts running. The counter will start counting down toward zero after the On-Chip Debugger exits DEBUG mode. If the On-Chip Debugger enters DEBUG mode before the OCDCNTR Register counts down to zero, the OCDCNTR will stop counting.

If the OCDCNTR Register is configured to generate a BRK when the program counter matches the OCDCNTR Register, the OCDCNTR Register will not be reset when the CPU resumes executing and it will not be decremented when the CPU is running. A BRK will be generated when the program counter matches the value in the OCDCNTR Register before executing the instruction at the location of the program counter.

! Caution: The OCDCNTR Register is used by many of the OCD commands. It counts the number of bytes for the register and memory read/write commands. It retains the residual value when generating the CRC. If the OCDCNTR is used to generate a BRK, its value must be written as a final step before leaving DEBUG mode.

Because this register is overwritten by various OCD commands, it must only be used to generate temporary breakpoints, such as stepping over CALL instructions or running to a specific instruction and stopping.

When the OCDCNTR Register is read, it returns the inverse of the data in this register. The OCDCNTR Register is only decremented when counting. The mode where it counts the number of clock cycles in between execution is achieved by counting down from its maximum count. When the OCDCNTR Register is read, the counter appears to have counted up because its value is inverted. The value in this register is always inverted when it is read. If this register is used as a hardware breakpoint, the value read from this register will be the inverse of the data actually in the register.

23.3. On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In Debug mode, all OCD commands become available unless the user code is protected by programming the Flash Read Protect option bit (FRP). The Flash Read Protect option bit prevents the code in memory from being read out of the Z8 Encore! XP F1680 Series device. When this option is enabled, several of the OCD commands are disabled. When the Read Protect option bit is enabled, asserting the TESTMODE pad does NOT put the Z8 Encore! XP F1680 Series MCU in Flash Test mode.

Figures 67 and 68 provide information about each of the eZ8 CPUinstructions.

Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1.1 **BRK**	2.2 **SRP** IM	2.3 **ADD** r1,r2	2.4 **ADD** r1,Ir2	3.3 **ADD** R2,R1	3.4 **ADD** IR2,R1	3.3 **ADD** R1,IM	3.4 **ADD** IR1,IM	4.3 **ADDX** ER2,ER1	4.3 **ADDX** IM,ER1	2.3 **DJNZ** r1,X	2.2 **JR** cc,X	2.2 **LD** r1,IM	3.2 **JP** cc,DA	1.2 **INC** r1	1.2 **NOP**
1	2.2 **RLC** R1	2.3 **RLC** IR1	2.3 **ADC** r1,r2	2.4 **ADC** r1,Ir2	3.3 **ADC** R2,R1	3.4 **ADC** IR2,R1	3.3 **ADC** R1,IM	3.4 **ADC** IR1,IM	4.3 **ADCX** ER2,ER1	4.3 **ADCX** IM,ER1						See 2nd Op Code Map
2	2.2 **INC** R1	2.3 **INC** IR1	2.3 **SUB** r1,r2	2.4 **SUB** r1,Ir2	3.3 **SUB** R2,R1	3.4 **SUB** IR2,R1	3.3 **SUB** R1,IM	3.4 **SUB** IR1,IM	4.3 **SUBX** ER2,ER1	4.3 **SUBX** IM,ER1						1, 2 ATM
3	2.2 **DEC** R1	2.3 **DEC** IR1	2.3 **SBC** r1,r2	2.4 **SBC** r1,Ir2	3.3 **SBC** R2,R1	3.4 **SBC** IR2,R1	3.3 **SBC** R1,IM	3.4 **SBC** IR1,IM	4.3 **SBCX** ER2,ER1	4.3 **SBCX** IM,ER1						
4	2.2 **DA** R1	2.3 **DA** IR1	2.3 **OR** r1,r2	2.4 **OR** r1,Ir2	3.3 **OR** R2,R1	3.4 **OR** IR2,R1	3.3 **OR** R1,IM	3.4 **OR** IR1,IM	4.3 **ORX** ER2,ER1	4.3 **ORX** IM,ER1						
5	2.2 **POP** R1	2.3 **POP** IR1	2.3 **AND** r1,r2	2.4 **AND** r1,Ir2	3.3 **AND** R2,R1	3.4 **AND** IR2,R1	3.3 **AND** R1,IM	3.4 **AND** IR1,IM	4.3 **ANDX** ER2,ER1	4.3 **ANDX** IM,ER1						1.2 **WDT**
6	2.2 **COM** R1	2.3 **COM** IR1	2.3 **TCM** r1,r2	2.4 **TCM** r1,Ir2	3.3 **TCM** R2,R1	3.4 **TCM** IR2,R1	3.3 **TCM** R1,IM	3.4 **TCM** IR1,IM	4.3 **TCMX** ER2,ER1	4.3 **TCMX** IM,ER1						1.2 **STOP**
7	2.2 **PUSH** R2	2.3 **PUSH** IR2	2.3 **TM** r1,r2	2.4 **TM** r1,Ir2	3.3 **TM** R2,R1	3.4 **TM** IR2,R1	3.3 **TM** R1,IM	3.4 **TM** IR1,IM	4.3 **TMX** ER2,ER1	4.3 **TMX** IM,ER1						1.2 **HALT**
8	2.5 **DECW** RR1	2.6 **DECW** IRR1	2.5 **LDE** r1,Irr2	2.9 **LDEI** Ir1,Irr2	3.2 **LDX** r1,ER2	3.3 **LDX** Ir1,ER2	3.4 **LDX** IRR2,R1	3.5 **LDX** IRR2,IR1	3.4 **LDX** r1,rr2,X	3.4 **LDX** rr1,r2,X						1.2 **DI**
9	2.2 **RL** R1	2.3 **RL** IR1	2.5 **LDE** r2,Irr1	2.9 **LDEI** Ir2,Irr1	3.2 **LDX** r2,ER1	3.3 **LDX** Ir2,ER1	3.4 **LDX** R2,IRR1	3.5 **LDX** IR2,IRR1	3.3 **LEA** r1,r2,X	3.5 **LEA** rr1,rr2,X						1.2 **EI**
A	2.5 **INCW** RR1	2.6 **INCW** IRR1	2.3 **CP** r1,r2	2.4 **CP** r1,Ir2	3.3 **CP** R2,R1	3.4 **CP** IR2,R1	3.3 **CP** R1,IM	3.4 **CP** IR1,IM	4.3 **CPX** ER2,ER1	4.3 **CPX** IM,ER1						1.4 **RET**
B	2.2 **CLR** R1	2.3 **CLR** IR1	2.3 **XOR** r1,r2	2.4 **XOR** r1,Ir2	3.3 **XOR** R2,R1	3.4 **XOR** IR2,R1	3.3 **XOR** R1,IM	3.4 **XOR** IR1,IM	4.3 **XORX** ER2,ER1	4.3 **XORX** IM,ER1						1.5 **IRET**
C	2.2 **RRC** R1	2.3 **RRC** IR1	2.5 **LDC** r1,Irr2	2.9 **LDCI** Ir1,Irr2	2.3 **JP** IRR1	2.9 **LDC** Ir1,Irr2		3.4 **LD** r1,r2,X	3.2 **PUSHX** ER2							1.2 **RCF**
D	2.2 **SRA** R1	2.3 **SRA** IR1	2.5 **LDC** r2,Irr1	2.9 **LDCI** Ir2,Irr1	2.6 **CALL** IRR1	2.2 **BSWAP** R1	3.3 **CALL** DA	3.4 **LD** r2,r1,X	3.2 **POPX** ER1							1.2 **SCF**
E	2.2 **RR** R1	2.3 **RR** IR1	2.2 **BIT** p,b,r1	2.3 **LD** r1,Ir2	3.2 **LD** R2,R1	3.3 **LD** IR2,R1	3.2 **LD** R1,IM	3.3 **LD** IR1,IM	4.2 **LDX** ER2,ER1	4.2 **LDX** IM,ER1						1.2 **CCF**
F	2.2 **SWAP** R1	2.3 **SWAP** IR1	2.6 **TRAP** Vector	2.3 **LD** Ir1,r2	2.8 **MULT** RR1	3.3 **LD** R2,IR1	3.3 **BTJ** p,b,r1,X	3.4 **BTJ** p,b,Ir1,X								

Figure 67. First Op Code Map

The currents in Table 190 represent the power consumption without any peripherals active (unless otherwise noted). For design guidance, total power consumption will be the sum of all active peripheral currents plus the appropriate current characteristics shown below.

Table 190. Supply Current Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typical[1]**	**Max**	**Units**	**Conditions[2]**
I_{DDA1}	Active Mode Device Current Executing from Flash		8.5		mA	Typical: 20MHz[3, 4, 5, 6], V_{DD} = 3V, Flash, 25°C
I_{DDA2}	Active Mode Device Current Executing from PRAM		6		mA	Typical: 20MHz[3, 4, 5, 6], V_{DD} = 3V, PRAM, 25°C
I_{DDH}	Halt Mode Device Current		TBD		mA	Typical: 20MHz[3, 4, 5], V_{DD} typical, 25°C
I_{DDS1}	Stop Mode Device Current		2.5		µA	Typical: WDT, V_{DD} typical, 25 °C, all peripherals including VBO disabled[3, 4, 6]
I_{DDS2}	Stop Mode Device Current		<1		µA	Typical: V_{DD} typical, 25°C, all peripherals disabled including VBO and WDT[3, 4, 6]

Notes
1. These values are provided for design guidance only and are not tested in production.
2. Typical conditions are defined as 3.3 V at 25°C, unless otherwise noted.
3. All internal pull ups are disabled and all push-pull outputs are unloaded.
4. All open-drain outputs are pulled up to V_{DD}/AV_{DD} and are at a High state.
5. System clock source is an external square wave clock signal driven through the CLK-IN pin.
6. All inputs are at V_{DD}/AV_{DD} or V_{SS}/AV_{SS} as appropriate.

Figures 69 through 72 display the typical current consumption at voltages of 1.8V, 2.0V, 2.7V, 3.0V, 3.3V and 3.6V, respectively, versus different system clock frequencies while operating at a temperature of 25ºC.

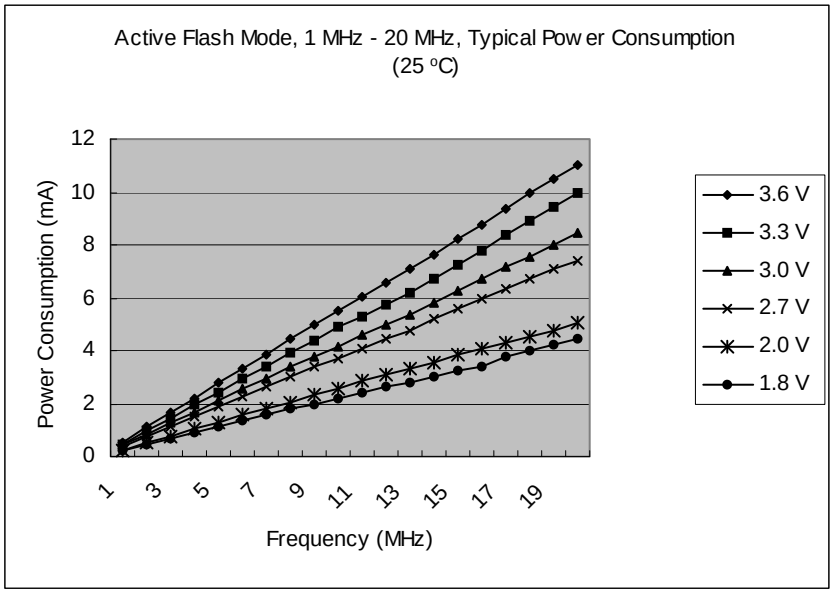


Figure 69. Typical Active Flash Mode Supply Current (1–20MHz)

V

W

X

Z