



Welcome to [E-XFL.COM](https://www.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

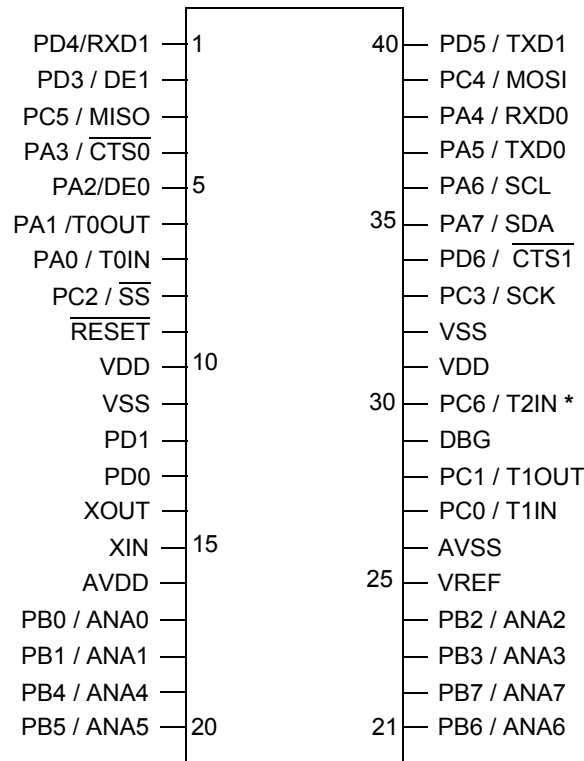
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f3221an020ec00tr



Flash Status Register	190
Page Select Register	191
Flash Sector Protect Register	192
Flash Frequency High and Low Byte Registers	192
Option Bits	195
Overview	195
Operation	195
Option Bit Configuration By Reset	195
Option Bit Address Space	195
Flash Memory Address 0000H	196
Flash Memory Address 0001H	197
On-Chip Debugger	199
Overview	199
Architecture	199
Operation	200
OCD Interface	200
DEBUG Mode	201
OCD Data Format	202
OCD Auto-Baud Detector/Generator	202
OCD Serial Errors	203
Breakpoints	203
On-Chip Debugger Commands	204
On-Chip Debugger Control Register Definitions	209
OCD Control Register	209
OCD Status Register	210
On-Chip Oscillator	211
Overview	211
Operating Modes	211
Crystal Oscillator Operation	211
Oscillator Operation with an External RC Network	213
Electrical Characteristics	215
Absolute Maximum Ratings	215
DC Characteristics	217
On-Chip Peripheral AC and DC Electrical Characteristics	226
AC Characteristics	231
General-Purpose I/O Port Input Data Sample Timing	232

Pin Configurations

Figure 2 through Figure 7 on page 13 display the pin configurations for all of the packages available in the Z8 Encore! XP 64K Series Flash Microcontrollers. For description of the signals, see Table 3 on page 14. Timer 3 is not available in the 40-pin and 44-pin packages.



Note: Timer 3 is not supported.

* T2OUT is not supported.

Figure 2. Z8 Encore! XP 64K Series Flash Microcontrollers in 40-Pin Dual Inline Package (PDIP)

Program Memory

The eZ8[™] CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP 64K Series Flash Microcontrollers contains 16 KB to 64 KB of on-chip Flash in the Program Memory address space, depending upon the device. Reading from Program Memory addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. [Table 5](#) describes the Program Memory maps for the 64K Series products.

Table 5. Z8 Encore! XP 64K Series Flash Microcontrollers Program Memory Maps

Program Memory Address (Hex)	Function
Z8F162x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFF	Program Memory
Z8F242x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFF	Program Memory
Z8F322x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFF	Program Memory
Z8F482x Products	

Port C Control

PCCTL (FD9H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port C Control[7:0]
Provides Access to Port Sub-Registers

Port C Input Data

PCIN (FDAH - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port C Input Data [7:0]

Port C Output Data

PCOUT (FDBH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port C Output Data [7:0]

Port D Address

PDADDR (FDCH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port D Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port D Control

PDCTL (FDDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port D Control[7:0]
Provides Access to Port Sub-Registers

Port D Input Data

PDIN (FDE H- Read Only)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port D Input Data [7:0]

Port D Output Data

PDOUT (FDFH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port D Output Data [7:0]

Port E Address

PEADDR (FE0H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port E Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port E Control

PECTL (FE1H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port E Control[7:0]
Provides Access to Port Sub-Registers

Port E Input Data

PEIN (FE2H - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port E Input Data [7:0]

Port E Output Data

PEOUT (FE3H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port E Output Data [7:0]

Port F Address

PFADDR (FE4H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

Port F Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

GPIO Control Register Definitions

Four registers for each Port provide access to GPIO control, input data, and output data. [Table 13](#) lists these Port registers. Use the Port A–H Address and Control registers together to provide access to sub-registers for Port configuration and control.

Table 13. GPIO Port Registers and Sub-Registers

Port Register Mnemonic	Port Register Name
PxADDR	Port A–H Address Register (Selects sub-registers)
PxCTL	Port A–H Control Register (Provides access to sub-registers)
PxIN	Port A–H Input Data Register
PxOUT	Port A–H Output Data Register
Port Sub-Register Mnemonic	Port Register Name
PxDD	Data Direction
PxAF	Alternate Function
PxOC	Output Control (Open-Drain)
PxDD	High Drive Enable
PxSMRE	Stop Mode Recovery Source Enable

Port A–H Address Registers

The Port A–H Address registers select the GPIO Port functionality accessible through the Port A–H Control registers. The Port A–H Address and Control registers combine to provide access to all GPIO Port control ([Table 14](#)).

Table 14. Port A–H GPIO Address Registers (PxADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	PADDR[7:0]							
RESET	00H							
R/W	R/W							
ADDR	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

T1I—Timer 1 Interrupt Request

0 = No interrupt request is pending for Timer 1.

1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request

0 = No interrupt request is pending for Timer 0.

1 = An interrupt request from Timer 0 is awaiting service.

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.

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.

I²CI— I²C Interrupt Request

0 = No interrupt request is pending for the I²C.

1 = An interrupt request from the I²C is awaiting service.

SPII—SPI Interrupt Request

0 = No interrupt request is pending for the SPI.

1 = An interrupt request from the SPI is awaiting service.

ADCI—ADC Interrupt Request

0 = No interrupt request is pending for the Analog-to-Digital Converter.

1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register ([Table 25](#)) 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 eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 1 register to determine if any interrupt requests are pending.

Table 25. Interrupt Request 1 Register (IRQ1)

BITS	7	6	5	4	3	2	1	0
FIELD	PAD7I	PAD6I	PAD5I	PAD4I	PAD3I	PAD2I	PAD1I	PAD0I
RESET	0							
R/W	R/W							
ADDR	FC3H							

One-Shot time-out, first set the TPOL bit in the Timer Control 1 Register to the start value before beginning ONE-SHOT mode. Then, after starting the timer, set TPOL to the opposite bit value.

Follow the steps below for configuring a timer for ONE-SHOT mode and initiating the count:

1. Write to the Timer Control 1 register to:
 - Disable the timer
 - Configure the timer for ONE-SHOT mode
 - Set the prescale value
 - If using the Timer Output alternate function, set the initial output level (High or Low)
2. Write to the Timer High and Low Byte registers to set the starting count value
3. Write to the Timer Reload High and Low Byte registers to set the Reload value
4. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function
6. Write to the Timer Control 1 register to enable the timer and initiate counting

In ONE-SHOT mode, the system clock always provides the timer input. The timer period is given by the following equation:

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

CONTINUOUS Mode

In CONTINUOUS mode, the timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Reload value, 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 from High to Low) upon timer Reload.

Follow the steps below for configuring a timer for CONTINUOUS mode and initiating the count:

1. Write to the Timer Control 1 register to:
 - Disable the timer
 - Configure the timer for CONTINUOUS mode
 - Set the prescale value
 - If using the Timer Output alternate function, set the initial output level (High or Low)

1 = Timer Output is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon Reload.

CAPTURE mode

0 = Count is captured on the rising edge of the Timer Input signal.
1 = Count is captured on the falling edge of the Timer Input signal.

COMPARE mode

When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

GATED mode

0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input.
1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.

CAPTURE/COMPARE mode

0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal.
1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.



Caution: *When the Timer Output alternate function TxOUT on a GPIO port pin is enabled, TxOUT will change to whatever state the TPOL bit is in. The timer does not need to be enabled for that to happen. Also, the Port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL bit with the timer enabled and running does not immediately change the TxOUT.*

PRES—Prescale value.

The timer input clock is divided by 2^{PRES} , where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This insures proper clock division each time the Timer is restarted.

000 = Divide by 1
001 = Divide by 2
010 = Divide by 4
011 = Divide by 8
100 = Divide by 16
101 = Divide by 32

UART

Overview

The Universal Asynchronous Receiver/Transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two Stop bits
- Separate transmit and receive interrupts
- Framing, parity, overrun and break detection
- Separate transmit and receive enables
- 16-bit Baud Rate Generator (BRG)
- Selectable MULTIPROCESSOR (9-bit) mode with three configurable interrupt schemes
- Baud Rate Generator timer mode
- Driver Enable output for external bus transceivers

Architecture

The UART consists of three primary functional blocks: Transmitter, Receiver, and Baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. [Figure 13](#) on page 104 displays the UART architecture.

configuration bits. In general, the address compare feature reduces the load on the CPU, since it does not need to access the UART when it receives data directed to other devices on the multi-node network. The following three MULTIPROCESSOR modes are available in hardware:

- Interrupt on all address bytes.
- Interrupt on matched address bytes and correctly framed data bytes.
- Interrupt only on correctly framed data bytes.

These modes are selected with MPMD[1 : 0] in the UART Control 1 Register. For all MULTIPROCESSOR modes, bit MPEN of the UART Control 1 Register must be set to 1.

The first scheme is enabled by writing 01b to MPMD[1 : 0]. In this mode, all incoming address bytes cause an interrupt, while data bytes never cause an interrupt. The interrupt service routine must manually check the address byte that caused triggered the interrupt. If it matches the UART address, the software clears MPMD[0]. At this point, each new incoming byte interrupts the CPU. The software is then responsible for determining the end of the frame. It checks for end-of-frame by reading the MPRX bit of the UART Status 1 Register for each incoming byte. If MPRX=1, a new frame has begun. If the address of this new frame is different from the UART's address, then set MPMD[0] to 1 causing the UART interrupts to go inactive until the next address byte. If the new frame's address matches the UART's, the data in the new frame is processed as well.

The second scheme is enabled by setting MPMD[1 : 0] to 10b and writing the UART's address into the UART Address Compare Register. This mode introduces more hardware control, interrupting only on frames that match the UART's address. When an incoming address byte does not match the UART's address, it is ignored. All successive data bytes in this frame are also ignored. When a matching address byte occurs, an interrupt is issued and further interrupts now occur on each successive data byte. The first data byte in the frame contains the NEWFRM=1 in the UART Status 1 Register. When the next address byte occurs, the hardware compares it to the UART's address. If there is a match, the interrupts continue and the NEWFRM bit is set for the first byte of the new frame. If there is no match, then the UART ignores all incoming bytes until the next address match.

The third scheme is enabled by setting MPMD[1 : 0] to 11b and by writing the UART's address into the UART Address Compare Register. This mode is identical to the second scheme, except that there are no interrupts on address bytes. The first data byte of each frame is still accompanied by a NEWFRM assertion.

External Driver Enable

The UART provides a Driver Enable (DE) signal for off-chip bus transceivers. This feature reduces the software overhead associated with using a GPIO pin to control the transceiver when communicating on a multi-transceiver bus, such as RS-485.

Driver Enable is an active High signal that envelopes the entire transmitted data frame including parity and Stop bits as displayed in [Figure 17](#). The Driver Enable signal asserts

repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal. This action allows the Endec to tolerate jitter and baud rate errors in the incoming data stream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined in [UART Control Register Definitions](#) on page 114.



Caution: *To prevent spurious signals during IrDA data transmission, set the `IREN` bit in the `UARTx Control 1` register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.*

Table 64. SPI Control Register (SPICTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN
RESET	0							
R/W	R/W							
ADDR	F61H							

IRQE—Interrupt Request Enable

0 = SPI interrupts are disabled. No interrupt requests are sent to the Interrupt Controller.

1 = SPI interrupts are enabled. Interrupt requests are sent to the Interrupt Controller.

STR—Start an SPI Interrupt Request

0 = No effect.

1 = Setting this bit to 1 also sets the **IRQ** bit in the SPI Status register to 1. Setting this bit forces the SPI to send an interrupt request to the Interrupt Control. This bit can be used by software for a function similar to transmit buffer empty in a UART.

Writing a 1 to the **IRQ** bit in the SPI Status register clears this bit to 0.

BIRQ—BRG Timer Interrupt Request

If the SPI is enabled, this bit has no effect. If the SPI is disabled:

0 = The Baud Rate Generator timer function is disabled.

1 = The Baud Rate Generator timer function and time-out interrupt are enabled.

PHASE—Phase Select

Sets the phase relationship of the data to the clock. For more information on operation of the **PHASE** bit, see [SPI Clock Phase and Polarity Control](#) on page 132.

CLKPOL—Clock Polarity

0 = SCK idles Low (0).

1 = SCK idle High (1).

WOR—Wire-OR (OPEN-DRAIN) Mode Enabled

0 = SPI signal pins not configured for open-drain.

1 = All four SPI signal pins (**SCK**, **SS**, **MISO**, **MOSI**) configured for open-drain function.

This setting is typically used for multi-master and/or multi-slave configurations.

MMEN—SPI Master Mode Enable

0 = SPI configured in Slave mode.

1 = SPI configured in Master mode.

SPIEN—SPI Enable

0 = SPI disabled.

1 = SPI enabled.

I²C Controller

Overview

The I²C Controller makes the 64K Series products bus-compatible with the I²C protocol. The I²C Controller consists of two bidirectional bus lines—a serial data signal (SDA) and a serial clock signal (SCL). Features of the I²C Controller include:

- Transmit and Receive Operation in MASTER mode
- Maximum data rate of 400 kbit/sec
- 7- and 10-bit addressing modes for Slaves
- Unrestricted number of data bytes transmitted per transfer

The I²C Controller in the 64K Series products does not operate in SLAVE mode.

I²C Baud Rate High and Low Byte Registers

The I²C Baud Rate High and Low Byte registers (Tables 73 and 73) combine to form a 16-bit reload value, BRG[15:0], for the I²C Baud Rate Generator.

When the I²C is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

1. Disable the I²C by clearing the IEN bit in the I²C Control register to 0.
2. Load the desired 16-bit count value into the I²C Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BIRQ bit in the I²C Control register to 1.

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

$$\text{Interrupt Interval (s)} = \text{System Clock Period (s)} \times \text{BRG}[15:0]$$

Table 73. I²C Baud Rate High Byte Register (I2CBRH)

BITS	7	6	5	4	3	2	1	0
FIELD	BRH							
RESET	FFH							
R/W	R/W							
ADDR	F53H							

BRH = I²C Baud Rate High Byte

Most significant byte, BRG[15:8], of the I²C Baud Rate Generator's reload value.

► **Note:** *If the DIAG bit in the I²C Diagnostic Control Register is set to 1, a read of the I2CBRH register returns the current value of the I²C Baud Rate Counter[15:8].*

0 = DMA0 is not the source of the interrupt from the DMA Controller.

1 = DMA0 completed transfer of data to/from the End Address and generated an interrupt.

OCD Serial Errors

The On-Chip Debugger can detect any of the following error conditions on the DBG pin:

- Serial Break (a minimum of nine continuous bits Low).
- Framing Error (received Stop bit is Low).
- Transmit Collision (OCD and host simultaneous transmission detected by the OCD).

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a Serial Break 4096 system clock cycles long back to the host, and resets the Auto-Baud Detector/Generator. A Framing Error or Transmit Collision may be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the interface, returning a Serial Break back to the host only extends the length of the Serial Break if the host releases the Serial Break early.

The host transmits a Serial Break on the DBG pin when first connecting to the 64K Series devices or when recovering from an error. A Serial Break from the host resets the Auto-Baud Generator/Detector but does not reset the OCD Control register. A Serial Break leaves the device in DEBUG mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If Breakpoints are enabled, the OCD idles the eZ8 CPU and enters DEBUG mode. If Breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP.

If breakpoints are enabled, the OCD can be configured to automatically enter DEBUG mode, or to loop on the break instruction. If the OCD is configured to loop on the BRK instruction, then the CPU is still enabled to service DMA and interrupt requests.

The loop on BRK instruction can be used to service interrupts in the background. For interrupts to be serviced in the background, there cannot be any breakpoints in the interrupt service routine. Otherwise, the CPU stops on the breakpoint in the interrupt routine. For interrupts to be serviced in the background, interrupts must also be enabled. Debugging software should not automatically enable interrupts when using this feature, since interrupts are typically disabled during critical sections of code where interrupts should not occur (such as adjusting the stack pointer or modifying shared data).

Software can poll the IDLE bit of the OCDSTAT register to determine if the OCD is looping on a BRK instruction. When software wants to stop the CPU on the BRK instruction it is looping on, software should not set the DBGMODE bit of the OCDCTL register. The CPU may have vectored to and be in the middle of an interrupt service routine when this bit gets set. Instead, software must clear the BRKLP bit. This action allows the CPU to

On-Chip Oscillator

Overview

The products in the 64K Series feature an on-chip oscillator for use with external crystals with frequencies from 32 kHz to 20 MHz. In addition, the oscillator can support external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with oscillation frequencies up to 20 MHz. This oscillator generates the primary system clock for the internal eZ8[™] CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32 kHz–20 MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected.

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

Operating Modes

The 64K Series products support four different oscillator modes:

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

The oscillator mode is selected through user-programmable Option Bits. For more information, see [Option Bits](#) on page 195.

Crystal Oscillator Operation

[Figure 40](#) on page 212 displays a recommended configuration for connection with an external fundamental-mode, parallel-resonant crystal operating at 20 MHz. Recommended 20 MHz crystal specifications are provided in [Table 104](#) on page 212. Resistor R1 is optional and limits total power dissipation by the crystal. The printed circuit board layout

must add no more than 4 pF of stray capacitance to either the X_{IN} or X_{OUT} pins. If oscillation does not occur, reduce the values of capacitors C1 and C2 to decrease loading.

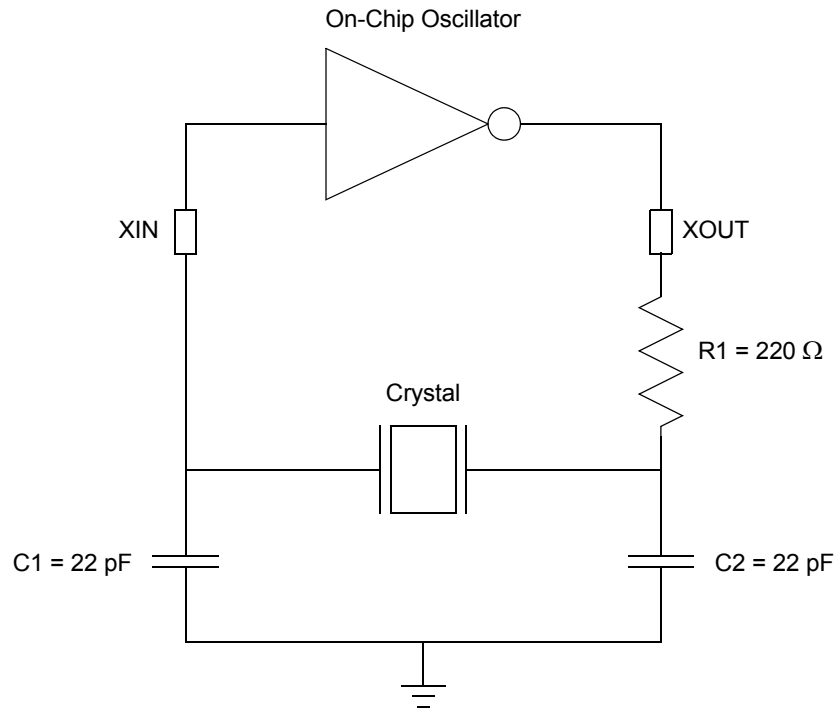


Figure 40. Recommended 20 MHz Crystal Oscillator Configuration

Table 104. Recommended Crystal Oscillator Specifications (20 MHz Operation)

Parameter	Value	Units	Comments
Frequency	20	MHz	
Resonance	Parallel		
Mode	Fundamental		
Series Resistance (R _S)	25	Ω	Maximum
Load Capacitance (C _L)	20	pF	Maximum
Shunt Capacitance (C ₀)	7	pF	Maximum
Drive Level	1	mW	Maximum

Figure 43 displays the typical active mode current consumption while operating at 25 °C versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

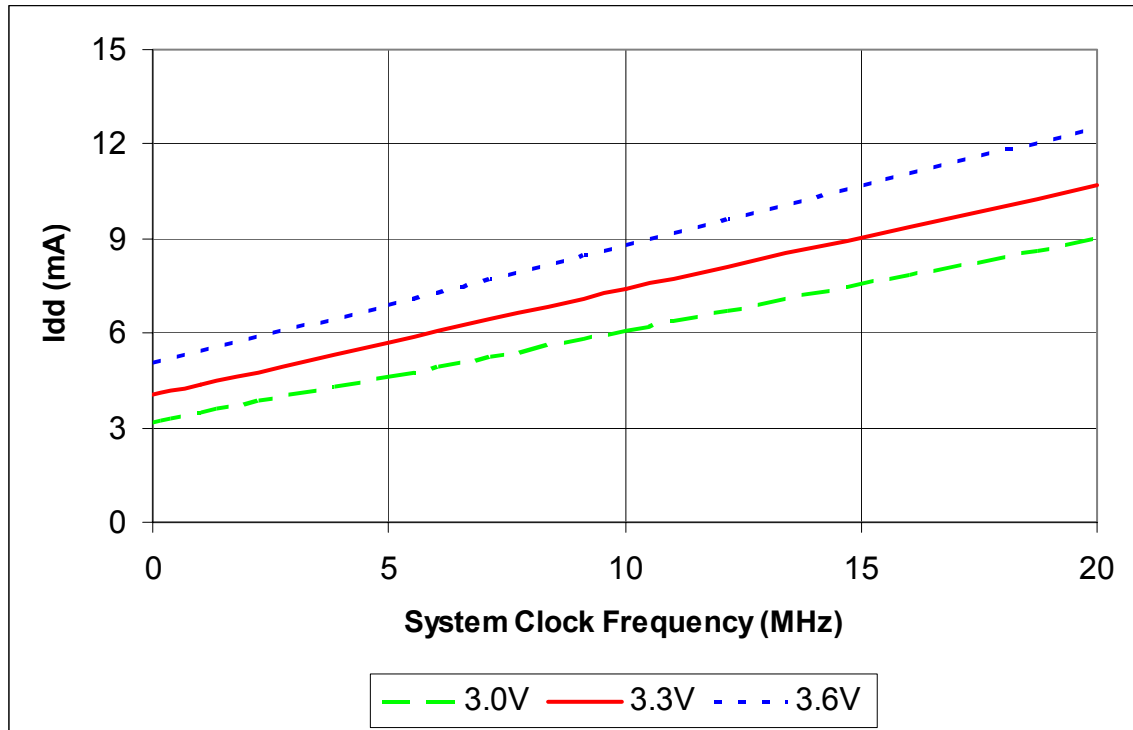


Figure 43. Typical Active Mode I_{dd} Versus System Clock Frequency

Figure 44 displays the maximum active mode current consumption across the full operating temperature range of the device and versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

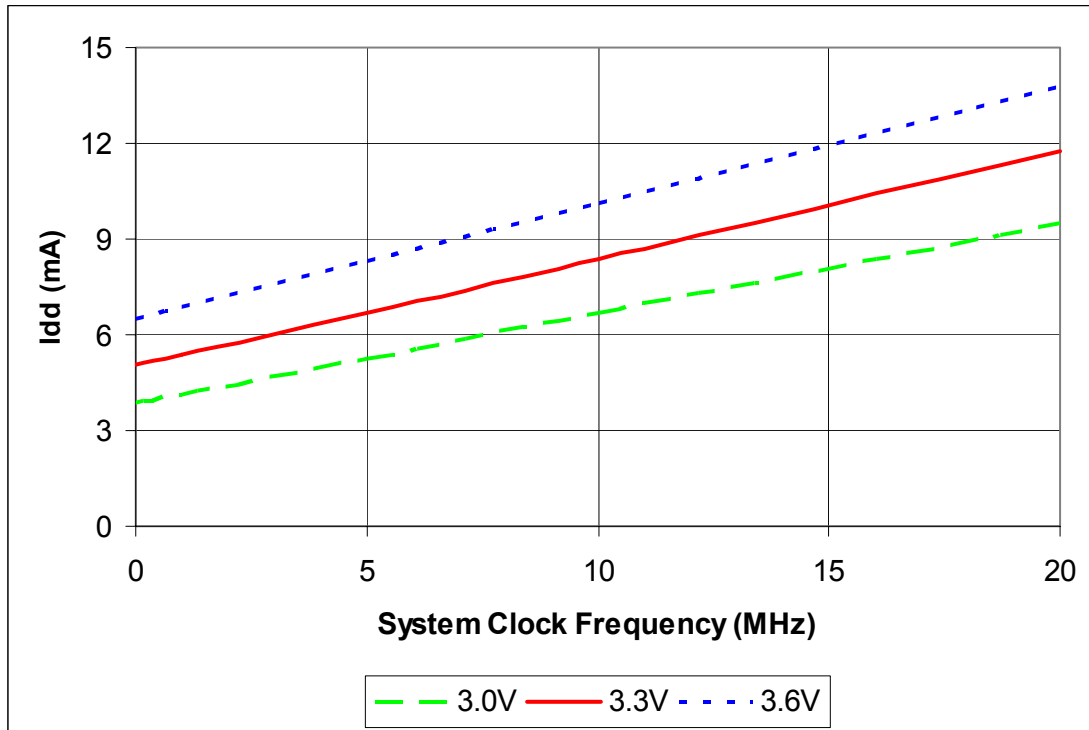


Figure 44. Maximum Active Mode I_{dd} Versus System Clock Frequency