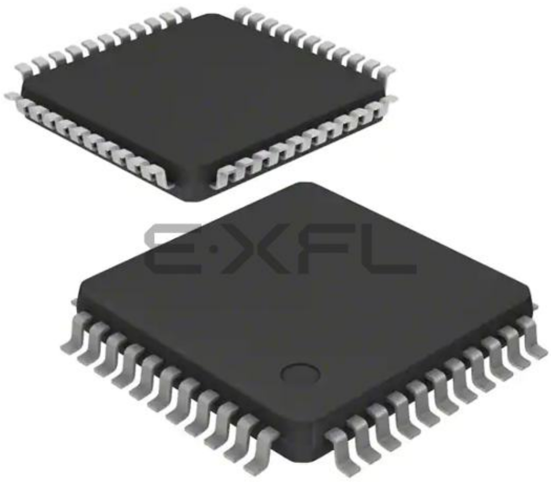

What is "Embedded - Microcontrollers"?

"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	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	24KB (24K 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	0°C ~ 70°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/z8f2421an020sc00tr

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

ISO 9001:2000
FS 507510

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

Use of All Uppercase Letters

The use of all uppercase letters designates the names of states, modes, and commands.

- Example 1: The bus is considered BUSY after the Start condition.
- Example 2: A START command triggers the processing of the initialization sequence.
- Example 3: STOP mode.

Bit Numbering

Bits are numbered from *0* to *n–1* where *n* indicates the total number of bits. For example, the 8 bits of a register are numbered from 0 to 7.

Safeguards

It is important that you understand the following safety terms, which are defined here.

Caution: *Indicates a procedure or file may become corrupted if you do not follow directions.*

Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FCD	Interrupt Edge Select	IRQES	00	78
FCE	Interrupt Port Select	IRQPS	00	78
FCF	Interrupt Control	IRQCTL	00	79
GPIO Port A				
FD0	Port A Address	PAADDR	00	61
FD1	Port A Control	PACTL	00	62
FD2	Port A Input Data	PAIN	XX	66
FD3	Port A Output Data	PAOUT	00	66
GPIO Port B				
FD4	Port B Address	PBADDR	00	61
FD5	Port B Control	PBCTL	00	62
FD6	Port B Input Data	PBIN	XX	66
FD7	Port B Output Data	PBOUT	00	66
GPIO Port C				
FD8	Port C Address	PCADDR	00	61
FD9	Port C Control	PCCTL	00	62
FDA	Port C Input Data	PCIN	XX	66
FDB	Port C Output Data	PCOUT	00	66
GPIO Port D				
FDC	Port D Address	PDADDR	00	61
FDD	Port D Control	PDCTL	00	62
FDE	Port D Input Data	PDIN	XX	66
FDF	Port D Output Data	PDOUT	00	66
GPIO Port E				
FE0	Port E Address	PEADDR	00	61
FE1	Port E Control	PECTL	00	62
FE2	Port E Input Data	PEIN	XX	66
FE3	Port E Output Data	PEOUT	00	66
GPIO Port F				
FE4	Port F Address	PFADDR	00	61
FE5	Port F Control	PFCTL	00	62
FE6	Port F Input Data	PFIN	XX	66
FE7	Port F Output Data	PFOUT	00	66
GPIO Port G				
FE8	Port G Address	PGADDR	00	61
FE9	Port G Control	PGCTL	00	62
FEA	Port G Input Data	PGIN	XX	66
FEB	Port G Output Data	PGOUT	00	66
GPIO Port H				
FEC	Port H Address	PHADDR	00	61
FED	Port H Control	PHCTL	00	62
FEE	Port H Input Data	PHIN	XX	66

UART0 Transmit Data
U0TXD (F40H - Write Only)
D7 D6 D5 D4 D3 D2 D1 D0

- D7–D0: UART0 transmitter data byte [7:0]

UART0 Receive Data
U0RXD (F40H - Read Only)
D7 D6 D5 D4 D3 D2 D1 D0

- D7–D0: UART0 receiver data byte [7:0]

UART0 Status 0
U0STAT0 (F41H - Read Only)
D7 D6 D5 D4 D3 D2 D1 D0

- D0: $\overline{\text{CTS}}$ signal
  Returns the level of the $\overline{\text{CTS}}$ signal
- D1: Transmitter Empty
  0 = Data is currently transmitting
  1 = Transmission is complete
- D2: Transmitter Data Register Empty
  0 = Transmit Data Register is full
  1 = Transmit Data register is empty
- D3: Break Detect
  0 = No break occurred
  1 = A break occurred
- D4: Framing Error
  0 = No framing error occurred
  1 = A framing occurred
- D5: Overrun Error
  0 = No overrun error occurred
  1 = An overrun error occurred
- D6: Parity Error
  0 = No parity error occurred
  1 = A parity error occurred
- D7: Receive Data Available
  0 = Receive Data Register is empty
  1 = A byte is available in the Receive Data Register

UART0 Control 0
U0CTL0 (F42H - Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

- D0: Loop Back Enable
  0 = Normal operation
  1 = Transmit data is looped back to the receiver
- D1: Stop Bit Select
  0 = Transmitter sends 1 Stop bit
  1 = Transmitter sends 2 Stop bits
- D2: Send Break
  0 = No break is sent
  1 = Output of the transmitter is zero
- D3: Parity Select
  0 = Even parity
  1 = Odd parity
- D4: Parity Enable
  0 = Parity is disabled
  1 = Parity is enabled
- D5: $\overline{\text{CTS}}$ Enable
  0 = $\overline{\text{CTS}}$ signal has no effect on the transmitter
  1 = UART recognizes $\overline{\text{CTS}}$ signal as
  a
  transmit enable control signal
- D6: Receive Enable
  0 = Receiver disabled
  1 = Receiver enabled
- D7: Transmit Enable
  0 = Transmitter disabled
  1 = Transmitter enabled

IRQ0 Enable Low Bit
IRQ0ENL (FC2H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D0: ADC IRQ Enable Hit Bit
- D1: SPI IRQ Enable Low Bit
- D2: I2C IRQ Enable Low Bit
- D3: UART 0 Transmitter IRQ Enable
- D4: UART 0 Receiver IRQ Enable Low
- D5: Timer 0 IRQ Enable Low Bit
- D6: Timer 1 IRQ Enable Low Bit
- D7: Timer 2 IRQ Enable Low Bit

Interrupt Request 1
IRQ1 (FC3H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D7–D0: Port A or D Pin Interrupt Request
  0 = IRQ from corresponding pin [7:0] is not pending
  1 = IRQ from corresponding pin [7:0] is awaiting service

IRQ1 Enable High Bit
IRQ1ENH (FC4H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D7–D0: Port A or D Pin IRQ Enable High Bit

IRQ1 Enable Low Bit
IRQ1ENL (FC5H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D7–D0: Port A or D Pin IRQ Enable Low Bit

Interrupt Request 2
IRQ2 (FC6H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D3–D0: Port C Pin Interrupt Request
  0 = IRQ from corresponding pin [3:0] is not pending
  1 = IRQ from corresponding pin [3:0] is awaiting service
- D4: DMA Interrupt Request
- D5: UART 1 Transmitter Interrupt
- D6: UART 1 Receiver Interrupt Request
- D7: Timer 3 Interrupt Request

For all of the above peripherals:
0 = Peripheral IRQ is not pending
1 = Peripheral IRQ is awaiting service

IRQ2 Enable High Bit
IRQ2ENH (FC7H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D3–D0: Port C Pin IRQ Enable High Bit
- D4: DMA IRQ Enable High Bit
- D5: UART 1 Transmitter IRQ Enable
- D6: UART 1 Receiver IRQ Enable High
- D7: Timer 3 IRQ Enable High Bit

IRQ2 Enable Low Bit
IRQ2ENL (FC8H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D3–D0: Port C Pin IRQ Enable Low Bit
- D4: DMA IRQ Enable Low Bit
- D5: UART 1 Transmitter IRQ Enable
- D6: UART 1 Receiver IRQ Enable Low
- D7: Timer 3 IRQ Enable Low Bit

Interrupt Edge Select
IRQES (FCDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D7–D0: Port A or D Interrupt Edge Select
  0 = Falling edge
  1 = Rising edge

Reset and Stop Mode Recovery

Overview

The Reset Controller within the Z8 Encore! XP 64K Series Flash Microcontrollers controls Reset and Stop Mode Recovery operation. In typical operation, the following events cause a Reset to occur:

- Power-On Reset
- Voltage Brownout
- Watchdog Timer time-out (when configured via the WDT_RES Option Bit to initiate a Reset)
- External $\overline{\text{RESET}}$ pin assertion
- On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)

When the 64K Series devices are in STOP mode, a Stop Mode Recovery is initiated by either of the following:

- Watchdog Timer time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source
- DBG pin driven Low

Reset Types

The 64K Series provides two different types of reset operation (system reset and Stop Mode Recovery). The type of Reset is a function of both the current operating mode of the 64K Series devices and the source of the Reset. Table 8 lists the types of Reset and their operating characteristics.

Low-Power Modes

Overview

The 64K Series products contain power-saving features. The highest level of power reduction is provided by STOP mode. The next level of power reduction is provided by the HALT mode.

STOP Mode

Execution of the eZ8™ CPU's STOP instruction places the device into STOP mode. In STOP mode, the operating characteristics are:

- Primary crystal oscillator is stopped; the XIN pin is driven High and the XOUT pin is driven Low.
- System clock is stopped.
- eZ8 CPU is stopped.
- Program counter (PC) stops incrementing.
- The Watchdog Timer and its internal RC oscillator continue to operate, if enabled for operation during STOP mode.
- The Voltage Brownout protection circuit continues to operate, if enabled for operation in STOP mode using the associated Option Bit.
- All other on-chip peripherals are idle.

To minimize current in STOP mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails (V_{CC} or GND), the Voltage Brownout protection must be disabled, and the Watchdog Timer must be disabled. The devices can be brought out of STOP mode using Stop Mode Recovery. For more information on Stop Mode Recovery, see Reset and Stop Mode Recovery on page 47.

Caution: *STOP mode must not be used when driving the 64K Series devices with an external clock driver source.*

Watchdog Timer

Overview

The Watchdog Timer (WDT) helps protect against corrupt or unreliable software, power faults, and other system-level problems which may place the Z8 Encore! XP into unsuitable operating states. The features of Watchdog Timer include:

- On-chip RC oscillator.
- A selectable time-out response.
- WDT Time-out response: Reset or interrupt.
- 24-bit programmable time-out value.

Operation

The Watchdog Timer (WDT) is a retriggerable one-shot timer that resets or interrupts the 64K Series devices when the WDT reaches its terminal count. The Watchdog Timer uses its own dedicated on-chip RC oscillator as its clock source. The Watchdog Timer has only two modes of operation—ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. An enable can be performed by executing the WDT instruction or by setting the `WDT_AO` Option Bit. The `WDT_AO` bit enables the Watchdog Timer to operate all the time, even if a WDT instruction has not been executed.

The Watchdog Timer is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8™ CPU register space to set the reload value. The nominal WDT time-out period is given by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10 kHz. The Watchdog Timer cannot be refreshed once it reaches `000002H`. The WDT Reload Value must not be set to values below `000004H`. Table 47 provides information on approximate time-out delays for the minimum and maximum WDT reload values.

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

5. Check the TDRE bit in the UART Status 0 register to determine if the Transmit Data register is empty (indicated by a 1). If 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. Write the UART Control 1 register to select the outgoing address bit.
7. Set the MULTIPROCESSOR Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
8. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.
9. If desired and MULTIPROCESSOR mode is enabled, make any changes to the MULTIPROCESSOR Bit Transmitter (MPBT) value.
10. To transmit additional bytes, return to step 5.

Transmitting Data using the Interrupt-Driven Method

The UART transmitter interrupt indicates the availability of the Transmit Data register to accept new data for transmission. Follow the steps below to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the desired baud rate.
2. Enable the 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 UART Transmitter interrupt and set the desired priority.
5. If MULTIPROCESSOR mode is desired, write to the UART Control 1 register to enable MULTIPROCESSOR (9-bit) mode functions.
6. Set the MULTIPROCESSOR Mode Select (MPEN) to Enable MULTIPROCESSOR mode.
7. Write to the UART Control 0 register to:
   – Set the transmit enable bit (TEN) to enable the UART for data transmission.
   – Enable parity, if desired and if MULTIPROCESSOR mode is not enabled, and select either even or odd parity.
   – Set or clear the CTSE bit to enable or disable control from the remote receiver via the $\overline{CTS}$ pin.

Table 61. UART Baud Rates

20.0 MHz System Clock				18.432 MHz System Clock			
Desired Rate	BRG Divisor	Actual Rate	Error	Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)	(kHz)	(Decimal)	(kHz)	(%)
1250.0	1	1250.0	0.00	1250.0	1	1152.0	-7.84%
625.0	2	625.0	0.00	625.0	2	576.0	-7.84%
250.0	5	250.0	0.00	250.0	5	230.4	-7.84%
115.2	11	113.6	-1.36	115.2	10	115.2	0.00
57.6	22	56.8	-1.36	57.6	20	57.6	0.00
38.4	33	37.9	-1.36	38.4	30	38.4	0.00
19.2	65	19.2	0.16	19.2	60	19.2	0.00
9.60	130	9.62	0.16	9.60	120	9.60	0.00
4.80	260	4.81	0.16	4.80	240	4.80	0.00
2.40	521	2.40	-0.03	2.40	480	2.40	0.00
1.20	1042	1.20	-0.03	1.20	960	1.20	0.00
0.60	2083	0.60	0.02	0.60	1920	0.60	0.00
0.30	4167	0.30	-0.01	0.30	3840	0.30	0.00

16.667 MHz System Clock				11.0592 MHz System Clock			
Desired Rate	BRG Divisor	Actual Rate	Error	Desired Rate	BRG Divisor	Actual Rate	Error
(kHz)	(Decimal)	(kHz)	(%)	(kHz)	(Decimal)	(kHz)	(%)
1250.0	1	1041.69	-16.67	1250.0	N/A	N/A	N/A
625.0	2	520.8	-16.67	625.0	1	691.2	10.59
250.0	4	260.4	4.17	250.0	3	230.4	-7.84
115.2	9	115.7	0.47	115.2	6	115.2	0.00
57.6	18	57.87	0.47	57.6	12	57.6	0.00
38.4	27	38.6	0.47	38.4	18	38.4	0.00
19.2	54	19.3	0.47	19.2	36	19.2	0.00
9.60	109	9.56	-0.45	9.60	72	9.60	0.00
4.80	217	4.80	-0.83	4.80	144	4.80	0.00
2.40	434	2.40	0.01	2.40	288	2.40	0.00

Follow the steps below for setting up the ADC and initiating continuous conversion:

1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This disables the digital input and output driver.
2. Write to the ADC Control register to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
   – Write to the ANAIN[3:0] field to select one of the 12 analog input sources.
   – Set CONT to 1 to select continuous conversion.
   – Write to the $\overline{\text{VREF}}$ bit to enable or disable the internal voltage reference generator.
   – Set CEN to 1 to start the conversions.
3. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
   – CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation.
   – An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete.
4. Thereafter, the ADC writes a new 10-bit data result to {ADCD_H[7:0], ADCD_L[7:6]} every 256 system clock cycles. An interrupt request is sent to the Interrupt Controller when each conversion is complete.
5. To disable continuous conversion, clear the CONT bit in the ADC Control register to 0.

DMA Control of the ADC

The Direct Memory Access (DMA) Controller can control operation of the ADC including analog input selection and conversion enable. For more information on the DMA and configuring for ADC operations, see Direct Memory Access Controller on page 165.

Table 88. ADC Data Low Bits Register (ADCD_L)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_L		Reserved					
RESET	X							
R/W	R							
ADDR	F73H							

ADCD_L—ADC Data Low Bits
These are the least significant two bits of the 10-bit ADC output. These bits are undefined after a Reset.

Reserved
These bits are reserved and are always undefined.

Flash Memory Address 0000H

Table 98. Flash Option Bits At Flash Memory Address 0000H

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	RP	Reserved	FWP
RESET	U							
R/W	R/W							
ADDR	Program Memory 0000H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

WDT_RES—Watchdog Timer Reset
0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request.
1 = Watchdog Timer time-out causes a Short Reset. This setting is the default for unprogrammed (erased) Flash.

WDT_AO—Watchdog Timer Always On
0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled except during STOP Mode (if configured to power down during STOP Mode).
1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

OSC_SEL[1:0]—Oscillator Mode Selection
00 = On-chip oscillator configured for use with external RC networks (<4 MHz).
01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).
10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).
11 = Maximum power for use with high frequency crystals (8.0 MHz to 20.0 MHz). This setting is the default for unprogrammed (erased) Flash.

VBO_AO—Voltage Brownout Protection Always On
0 = Voltage Brownout Protection is disabled in STOP mode to reduce total power consumption.
1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

RP—Read Protect
0 = User program code is inaccessible. Limited control features are available through

- **Write OCD Control Register (04H)**—The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of putting the device back into normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

- **Read OCD Control Register (05H)**—The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

- **Write Program Counter (06H)**—The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

- **Read Program Counter (07H)**—The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

- **Write Register (08H)**—The Write Register command writes data to the Register File. Data can be written 1-256 bytes at a time (256 bytes can be written by setting size to zero). If the device is not in DEBUG mode, the address and data values are discarded. If the Read Protect Option Bit is enabled, then only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0,Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

- **Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1-256 bytes at a time (256 bytes can be read by setting size to zero). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
```

Oscillator Operation with an External RC Network

The External RC oscillator mode is applicable to timing insensitive applications. Figure 41 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

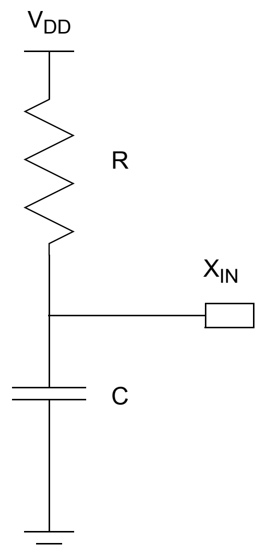

Figure 41. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45 kΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 kΩ. The typical oscillator frequency can be estimated from the values of the resistor (*R* in kΩ) and capacitor (*C* in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1 \times 10^6}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 42 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 45 kΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator XIN pin and the printed circuit board should be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20 pF are recommended.

[Figure 48](#) displays the maximum current consumption in STOP mode with the VBO disabled and Watchdog Timer enabled versus the power supply voltage. All GPIO pins are configured as outputs and driven High. Disabling the Watchdog Timer and its internal RC oscillator in STOP mode will provide some additional reduction in STOP mode current consumption. This small current reduction would be indistinquishable on the scale of Figure 48.

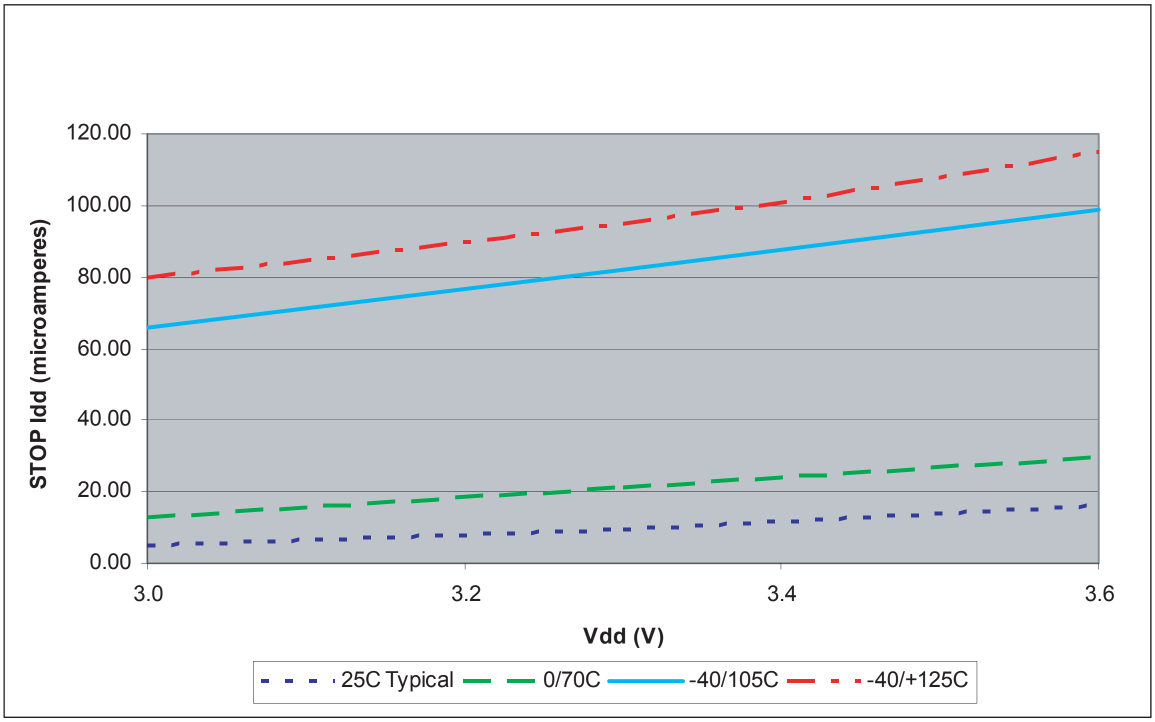

Figure 48. Maximum STOP Mode Idd with VBO Disabled versus Power Supply Voltage

Figure 57 and Table 121 provide timing information for UART pins for the case where the Clear To Send input signal ($\overline{CTS}$) is not used for flow control. In this example, it is assumed that the Driver Enable polarity has been configured to be Active Low and is represented here by $\overline{DE}$. $\overline{DE}$ asserts after the UART Transmit Data Register has been written. $\overline{DE}$ remains asserted for multiple characters as long as the Transmit Data register is written with the next character before the current character has completed.

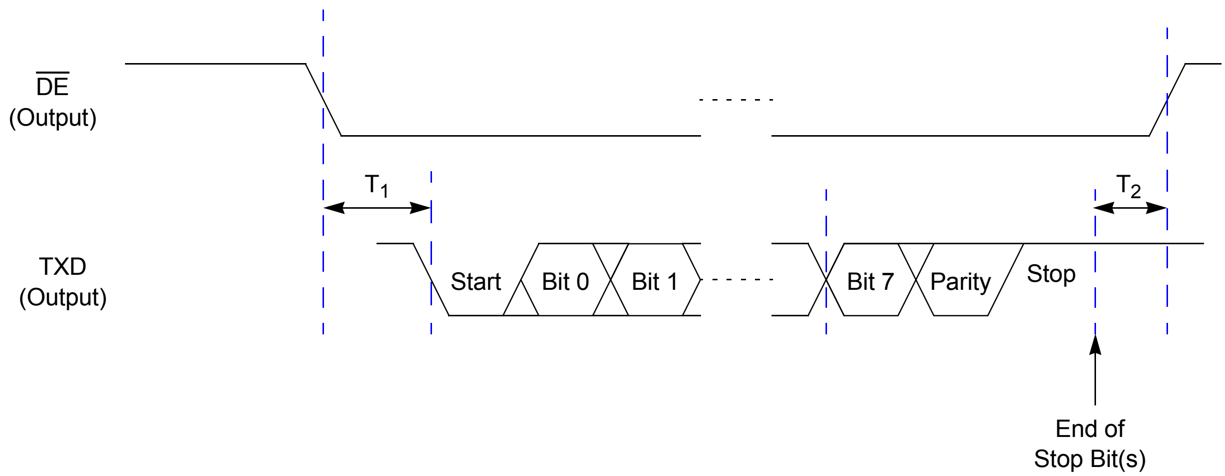

Figure 57. UART Timing without $\overline{CTS}$

Table 121. UART Timing without $\overline{CTS}$

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_1	$\overline{DE}$ Assertion to TXD Falling Edge (Start) Delay	1 Bit period	1 Bit period + 1 * XIN period
T_2	End of Stop Bit(s) to $\overline{DE}$ Deassertion Delay	1 * XIN period	2 * XIN period

; value `01H`, is the source. The value `01H` is written into the
; Register at address `234H`.

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed if you prefer manual program coding or intend to implement your own assembler.

Example 1: If the contents of Registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2: In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0 - R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

Refer to the device-specific Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status Flags, and address modes are represented by a notational shorthand that is described in Table 122.

Table 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
COM dst	dst ← ~dst	R		60	-	*	*	0	-	-	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	-	-	2	3
		r	Ir	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	-	-	3	3
		r	Ir	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	-	-	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	-	-	4	3
		ER	IM	A9							4	3
DA dst	dst ← DA(dst)	R		40	*	*	*	X	-	-	2	2
		IR		41							2	3
DEC dst	dst ← dst - 1	R		30	-	*	*	*	-	-	2	2
		IR		31							2	3
DECW dst	dst ← dst - 1	RR		80	-	*	*	*	-	-	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	-	-	-	-	-	-	1	2
DJNZ dst, RA	dst ← dst – 1 if dst ≠ 0 PC ← PC + X	r		0A-FA	-	-	-	-	-	-	2	3

V

W

X

Z