



Welcome to [E-XFL.COM](http://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	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f0413sj005ec">https://www.e-xfl.com/product-detail/zilog/z8f0413sj005ec</a>

# Address Space

The eZ8 CPU can access three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral, and general-purpose I/O port control registers.
- The Program Memory contains addresses for all memory locations having executable code and/or data.
- The Data Memory contains addresses for all memory locations that contain data only.

These three address spaces are covered briefly in the following subsections. For more detailed information regarding the eZ8 CPU and its address space, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at [www.zilog.com](http://www.zilog.com).

## Register File

The Register File address space in the Z8 Encore! XP<sup>®</sup> MCU is 4 KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read, and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4 KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B control register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. Z8 Encore! XP F0823 Series devices contain 256 B-1 KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

## Program Memory

The eZ8 CPU supports 64 KB of Program Memory address space. Z8 Encore! XP F0823 Series devices contain 1 KB to 8 KB of on-chip Flash memory in the Program Memory address space. Reading from Program Memory addresses outside the available Flash

memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 6 describes the Program Memory maps for the Z8 Encore! XP<sup>®</sup> F0823 Series products.

**Table 6. Z8 Encore! XP F0823 Series Program Memory Maps**

<b>Program Memory Address (Hex)</b>	<b>Function</b>
<b>Z8F0823 and Z8F0813 Products</b>	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–003D	Oscillator Fail Traps*
003E–0FFF	Program Memory
<b>Z8F0423 and Z8F0413 Products</b>	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–003D	Oscillator Fail Traps*
003E–0FFF	Program Memory
<b>Z8F0223 and Z8F0213 Products</b>	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–003D	Oscillator Fail Traps*
003E–07FF	Program Memory
<b>Z8F0123 and Z8F0113 Products</b>	
0000–0001	Flash Option Bits

# Register Map

Table 8 lists the address map for the Register File of the Z8 Encore! XP<sup>®</sup> F0823 Series devices. Not all devices and package styles in the Z8 Encore! XP F0823 Series support the ADC, or all GPIO ports. Consider registers for unimplemented peripherals as reserved.

**Table 8. Register File Address Map**

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
<b>General-Purpose RAM</b>				
<b>Z8F0823/Z8F0813 Devices</b>				
000–3FF	General-Purpose Register File RAM	—	XX	
400–EFF	Reserved	—	XX	
<b>Z8F0423/Z8F0413 Devices</b>				
000–3FF	General-Purpose Register File RAM	—	XX	
400–EFF	Reserved	—	XX	
<b>Z8F0223/Z8F0213 Devices</b>				
000–1FF	General-Purpose Register File RAM	—	XX	
200–EFF	Reserved	—	XX	
<b>Z8F0123/Z8F0113 Devices</b>				
000–0FF	General-Purpose Register File RAM	—	XX	
100–EFF	Reserved	—	XX	
<b>Timer 0</b>				
F00	Timer 0 High Byte	T0H	00	80
F01	Timer 0 Low Byte	T0L	01	80
F02	Timer 0 Reload High Byte	T0RH	FF	81
F03	Timer 0 Reload Low Byte	T0RL	FF	81
F04	Timer 0 PWM High Byte	T0PWMH	00	81
F05	Timer 0 PWM Low Byte	T0PWML	00	82
F06	Timer 0 Control 0	T0CTL0	00	82
F07	Timer 0 Control 1	T0CTL1	00	83
<b>Timer 1</b>				
F08	Timer 1 High Byte	T1H	00	80
F09	Timer 1 Low Byte	T1L	01	80
F0A	Timer 1 Reload High Byte	T1RH	FF	81
F0B	Timer 1 Reload Low Byte	T1RL	FF	81

**STOP—Stop Mode Recovery Indicator**

If this bit is set to 1, a Stop Mode Recovery is occurred. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurred because of a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the Stop Mode Recovery was not caused by a WDT time-out. This bit is reset by a POR or a WDT time-out that occurred while not in STOP mode. Reading this register also resets this bit.

**WDT—Watchdog Timer time-out Indicator**

If this bit is set to 1, a WDT time-out occurred. A POR resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit. This read must occur before clearing the WDT interrupt.

**EXT—External Reset Indicator**

If this bit is set to 1, a Reset initiated by the external  $\overline{\text{RESET}}$  pin occurred. A Power-On Reset or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.

Reserved—0 when read

**Table 15. Port Alternate Function Mapping (Non 8-Pin Parts)**

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
<b>Port A</b>	PA0	T0IN/T0OUT*	Timer 0 Input/Timer 0 Output Complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 Output	
		Reserved		
	PA2	DE0	UART 0 Driver Enable	
		Reserved		
	PA3	CTS0	UART 0 Clear to Send	
		Reserved		
	PA4	RXD0/IRRX0	UART 0 / IrDA 0 Receive Data	
		Reserved		
	PA5	TXD0/IRTX0	UART 0 / IrDA 0 Transmit Data	
		Reserved		
	PA6	T1IN/T1OUT*	Timer 1 Input/Timer 1 Output Complement	
		Reserved		
	PA7	T1OUT	Timer 1 Output	
		Reserved		

**Note:** Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in Port A–C Alternate Function Sub-Registers automatically enables the associated alternate function.

\* Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in Timer Pin Signal Operation on page 79.

PSMRE[7:0]—Port Stop Mode Recovery Source Enabled.

0 = The Port pin is not configured as a Stop Mode Recovery source. Transitions on this pin during STOP mode do not initiate Stop Mode Recovery.

1 = The Port pin is configured as a Stop Mode Recovery source. Any logic transition on this pin during STOP mode initiates Stop Mode Recovery.

### Port A–C Pull-up Enable Sub-Registers

The Port A–C Pull-up Enable sub-register (Table 25) is accessed through the Port A–C Control register by writing 06H to the Port A–C Address register. Setting the bits in the Port A–C Pull-up Enable sub-registers enables a weak internal resistive pull-up on the specified Port pins.

**Table 25. Port A–C Pull-Up Enable Sub-Registers (PxPUE)**

BITS	7	6	5	4	3	2	1	0
FIELD	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0
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	If 06H in Port A–C Address Register, accessible through the Port A–C Control Register							

PPUE[7:0]—Port Pull-up Enabled

0 = The weak pull-up on the Port pin is disabled.

1 = The weak pull-up on the Port pin is enabled.

### Port A–C Alternate Function Set 1 Sub-Registers

The Port A–C Alternate Function Set1 sub-register (Table 26) is accessed through the Port A–C Control register by writing 07H to the Port A–C Address register. The Alternate Function Set 1 sub-registers selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register are defined in GPIO Alternate Functions on page 36.

► **Note:** *Alternate function selection on port pins must also be enabled as described in Port A–C Alternate Function Sub-Registers on page 45.*

**!** **Caution:** *To avoid missing interrupts, use the following coding style to clear bits in the Interrupt Request 0 register:*

**Good coding style that avoids lost interrupt requests:**

ANDX IRQ0, MASK

## Software Interrupt Assertion

Program code generates interrupts directly. Writing a 1 to the correct bit in the Interrupt Request register triggers an interrupt (assuming that 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:** *The following coding style used to generate software interrupts by setting bits in the Interrupt Request registers is not recommended. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost.*

**Poor coding style that can result in lost interrupt requests:**

LDX r0, IRQ0

OR r0, MASK

LDX IRQ0, r0

**!** **Caution:** *To avoid missing interrupts, use the following coding style to set bits in the Interrupt Request registers:*

**Good coding style that avoids lost interrupt requests:**

ORX IRQ0, MASK

## Watchdog Timer Interrupt Assertion

The Watchdog Timer interrupt behavior is different from interrupts generated by other sources. The Watchdog Timer continues to assert an interrupt as long as the timeout condition continues. As it operates on a different (and usually slower) clock domain than the rest of the device, the Watchdog Timer continues to assert this interrupt for many system clocks until the counter rolls over.

**!** **Caution:** *To avoid re-triggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, it is recommended that the service routine continues to read from the RSTSTAT register until the WDT bit is cleared as given in the following coding sample:*

CLEARWDT:

LDX r0, RSTSTAT ; read reset status register to clear wdt  
bit

BTJNZ 5, r0, CLEARWDT ; loop until bit is cleared



## Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value 0001H into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value 0000H into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches FFFFH, the timer rolls over to 0000H and continues counting.

### Timer Operating Modes

The timers can be configured to operate in the following modes:

#### ONE-SHOT Mode

In ONE-SHOT 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 and the count value in the Timer High and Low Byte registers is reset to 0001H. The timer is automatically disabled and stops counting.

Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer Reload. If it is appropriate to have the Timer Output make a state change at a One-Shot time-out (rather than a single cycle pulse), first set the TPOL bit in the Timer Control register to the start value before enabling ONE-SHOT mode. 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 register to:
  - Disable the timer
  - Configure the timer for ONE-SHOT mode
  - Set the prescale value
  - Set the initial output level (High or Low) if using the Timer Output alternate function
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 appropriate, 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.

- Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function
- 2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
- 3. Write to the PWM High and Low Byte registers to set the PWM value.
- 4. Write to the PWM Control register to set the PWM dead band delay value. The deadband delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).
- 5. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
- 6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
- 8. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT mode equation determines the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

### **CAPTURE Mode**

In CAPTURE mode, the current timer count value is recorded when the appropriate external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control register determines if the Capture occurs on a rising edge or a falling edge

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data register is empty, an interrupt is generated immediately. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Write the UART Control 1 register to select the multiprocessor bit for the byte to be transmitted:  
Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
2. 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.
3. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
4. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.

### **Receiving Data using the Polled Method**

Follow the steps below to configure the UART for polled data reception:

1. Write to the UART Baud Rate High and Low Byte registers to set an acceptable baud rate for the incoming data stream.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Write to the UART Control 1 register to enable MULTIPROCESSOR mode functions, if appropriate.
4. Write to the UART Control 0 register to:
  - Set the receive enable bit (REN) to enable the UART for data reception
  - Enable parity, if appropriate and if Multiprocessor mode is not enabled, and select either even or odd parity
5. Check the RDA bit in the UART Status 0 register to determine if the Receive Data register contains a valid data byte (indicated by a 1). If RDA is set to 1 to indicate available data, continue to step 6. If the Receive Data register is empty (indicated by a 0), continue to monitor the RDA bit awaiting reception of the valid data.
6. Read data from the UART Receive Data register. If operating in MULTIPROCESSOR (9-bit) mode, further actions may be required depending on the MULTIPROCESSOR mode bits MPMD[1:0].
7. Return to step 4 to receive additional data.

Endec, and passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART's baud rate generator and supports IrDA standard baud rates from 9600 baud to 115.2 kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the Infrared Endec. The Infrared Endec data rate is calculated using the following equation:

$$\text{Infrared Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

## Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR\_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16 clocks wide. If the data to be transmitted is 1, the IR\_TXD signal remains low for the full 16 clock period. If the data to be transmitted is 0, the transmitter first outputs a 7 clock low period, followed by a 3 clock high pulse. Finally, a 6 clock low pulse is output to complete the full 16 clock data period. Figure 17 displays IrDA data transmission. When the Infrared Endec is enabled, the UART's TXD signal is internal to Z8 Encore! XP<sup>®</sup> F0823 Series products while the IR\_TXD signal is output through the TXD pin.

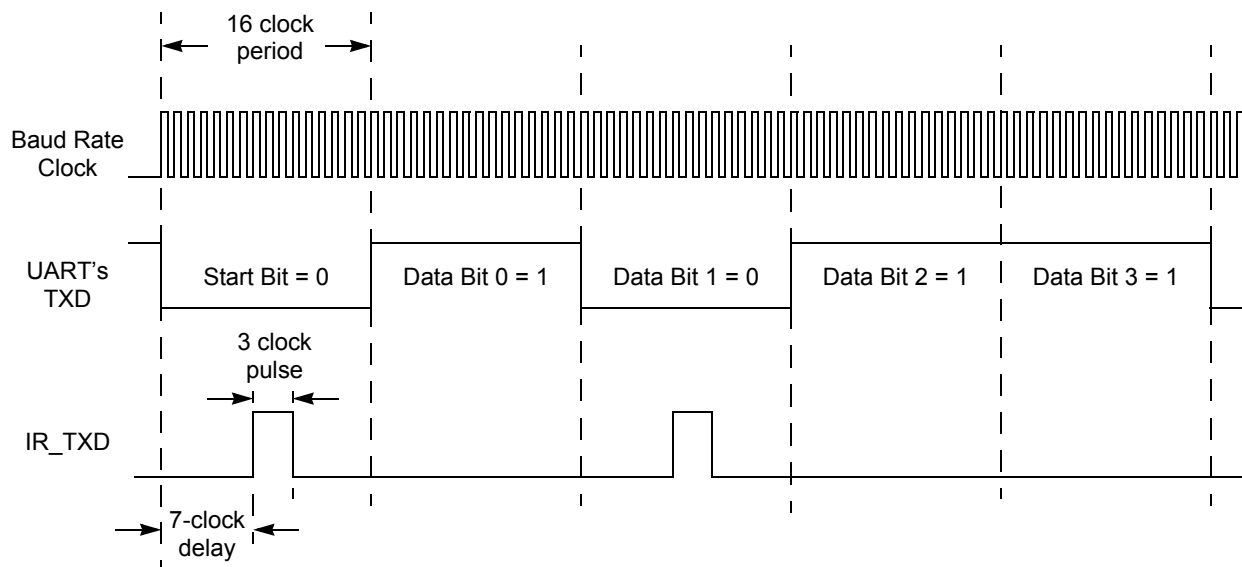


Figure 17. Infrared Data Transmission

**Figure 20. Flash Memory Arrangement**

## **Flash Information Area**

The Flash information area is separate from program memory and is mapped to the address range `FE00H` to `FFFFH`. Not all these addresses are accessible. Factory trim values for the analog peripherals are stored here. Factory calibration data for the ADC is also stored here.

## **Operation**

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for Byte Programming, Page Erase, and Mass Erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

The Flowchart in Figure 21 displays basic Flash Controller operation. The following sub-sections provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase, and Mass Erase) displayed in Figure 21.

## Trim Bit Data Register

The Trim Bid Data (TRMDR) register contains the read or write data for access to the trim option bits.

**Table 86. Trim Bit Data Register (TRMDR)**

BITS	7	6	5	4	3	2	1	0
FIELD	TRMDR - Trim Bit Data							
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	FF7H							

## Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses 0000H and 0001H are reserved for the user-programmable Flash option bits.

## Flash Program Memory Address 0000H

**Table 87. Flash Option Bits at Program Memory Address 0000H**

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	Reserved		VBO_AO	FRP	Reserved	FWP
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
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 system 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.

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.

Reserved—R/W bits must be 1 during writes; 1 when read.

VBO\_AO—Voltage Brownout Protection Always ON

0 = Voltage Brownout Protection can be disabled in STOP mode to reduce total power consumption. For the block to be disabled, the power control register bit must also be written (see Power Control Register 0 on page 32).

1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

FRP—Flash Read Protect

0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger.

1 = User program code is accessible. All On-Chip Debugger commands are enabled. This setting is the default for unprogrammed (erased) Flash.

Reserved—Must be 1

FWP—Flash Write Protect

This Option Bit provides Flash Program Memory protection:

0 = Programming and erasure disabled for all of Flash Program Memory. Programming, Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available using the On-Chip Debugger.

1 = Programming, Page Erase, and Mass Erase are enabled for all of Flash program memory.

## Flash Program Memory Address 0001H

**Table 88. Flash Options Bits at Program Memory Address 0001H**

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved			XTLDIS	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
ADDR	Program Memory 0001H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Reserved—R/W must be 1 during writes; 1 when read

XTLDIS—State of Crystal Oscillator at Reset





Table 118. DC Characteristics (Continued)

Symbol	Parameter	T <sub>A</sub> = -40 °C to +105 °C (unless otherwise specified)			Units	Conditions
		Minimum	Typical	Maximum		
V <sub>OH2</sub>	High Level Output Voltage	2.4	–	–	V	I <sub>OH</sub> = -20 mA; V <sub>DD</sub> = 3.3 V High Output Drive enabled.
I <sub>IH</sub>	Input Leakage Current	–	±0.002	±5	μA	V <sub>IN</sub> = V <sub>DD</sub> V <sub>DD</sub> = 3.3 V;
I <sub>IL</sub>	Input Leakage Current	–	±0.007	±5	μA	V <sub>IN</sub> = V <sub>SS</sub> V <sub>DD</sub> = 3.3 V;
I <sub>TL</sub>	Tristate Leakage Current	–	–	±5	μA	
I <sub>LED</sub>	Controlled Current Drive	1.8	3	4.5	mA	{AFS2,AFS1} = {0,0}
		2.8	7	10.5	mA	{AFS2,AFS1} = {0,1}
		7.8	13	19.5	mA	{AFS2,AFS1} = {1,0}
		12	20	30	mA	{AFS2,AFS1} = {1,1}
C <sub>PAD</sub>	GPIO Port Pad Capacitance	–	8.0 <sup>2</sup>	–	pF	
C <sub>XIN</sub>	XIN Pad Capacitance	–	8.0 <sup>2</sup>	–	pF	
C <sub>XOUT</sub>	XOUT Pad Capacitance	–	9.5 <sup>2</sup>	–	pF	
I <sub>PU</sub>	Weak Pull-up Current	30	100	350	μA	V <sub>DD</sub> = 3.0 V–3.6 V
V <sub>RAM</sub>	RAM Data Retention Voltage	TBD			V	Voltage at which RAM retains static values; no reading or writing is allowed.
<b>Notes</b> 1. This condition excludes all pins that have on-chip pull-ups, when driven Low. 2. These values are provided for design guidance only and are not tested in production.						

UART Timing

Figure 32 and Table 130 provide timing information for UART pins for the case where CTS is used for flow control. The CTS to DE assertion delay (T1) assumes the transmit data register has been loaded with data prior to CTS assertion.

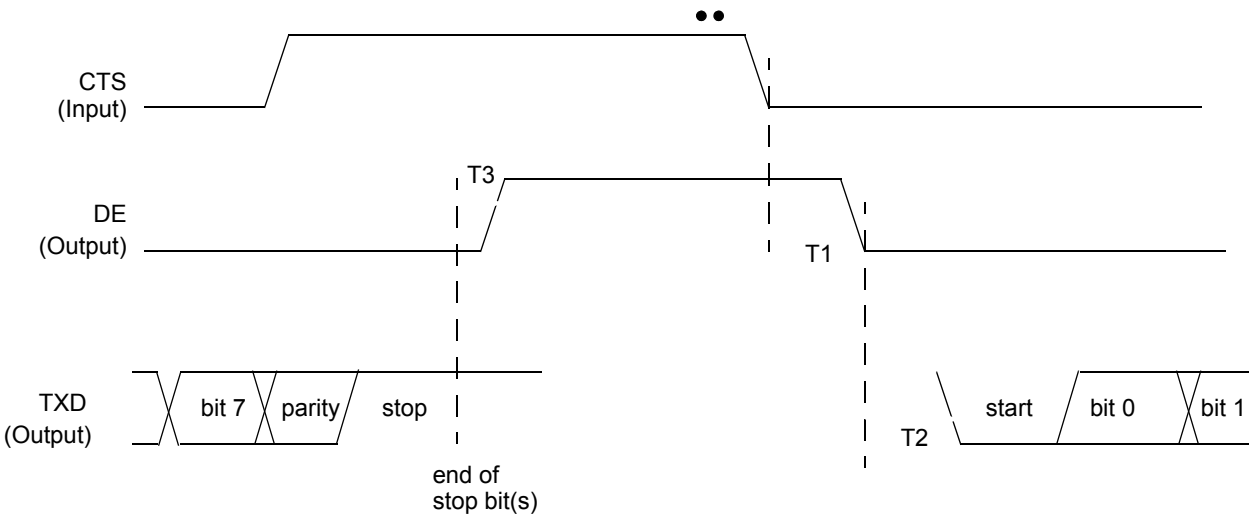


Figure 32. UART Timing With CTS

Table 130. UART Timing With CTS

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
UART			
T <sub>1</sub>	CTS Fall to DE output delay	2 * XIN period	2 * XIN period + 1 bit time
T <sub>2</sub>	DE assertion to TXD falling edge (start bit) delay	± 5	
T <sub>3</sub>	End of Stop Bit(s) to DE deassertion delay	± 5	

page erase 135  
page select register 138, 139  
FPS register 138, 139  
FSTAT register 137

## **G**

GATED mode 84  
general-purpose I/O 35  
GPIO 4, 35  
    alternate functions 36  
    architecture 36  
    control register definitions 43  
    input data sample timing 202  
    interrupts 43  
    port A-C pull-up enable sub-registers 48, 49  
    port A-H address registers 44  
    port A-H alternate function sub-registers 45  
    port A-H control registers 44  
    port A-H data direction sub-registers 45  
    port A-H high drive enable sub-registers 47  
    port A-H input data registers 49  
    port A-H output control sub-registers 46  
    port A-H output data registers 50  
    port A-H stop mode recovery sub-registers 47  
    port availability by device 35  
    port input timing 203  
    port output timing 204

## **H**

H 174  
HALT 176  
halt mode 32, 176  
hexadecimal number prefix/suffix 174

## **I**

I2C 4  
IM 173  
immediate data 173  
immediate operand prefix 174

INC 175  
increment 175  
increment word 175  
INCW 175  
indexed 173  
indirect address prefix 174  
indirect register 173  
indirect register pair 173  
indirect working register 173  
indirect working register pair 173  
infrared encoder/decoder (IrDA) 113  
Instruction Set 171  
instruction set, eZ8 CPU 171  
instructions  
    ADC 175  
    ADCX 175  
    ADD 175  
    ADDX 175  
    AND 177  
    ANDX 177  
    arithmetic 175  
    BCLR 176  
    BIT 176  
    bit manipulation 176  
    block transfer 176  
    BRK 178  
    BSET 176  
    BSWAP 176, 178  
    BTJ 178  
    BTJNZ 178  
    BTJZ 178  
    CALL 178  
    CCF 176  
    CLR 177  
    COM 177  
    CP 175  
    CPC 175  
    CPCX 175  
    CPU control 176  
    CPX 175  
    DA 175  
    DEC 175  
    DECW 175  
    DI 176

