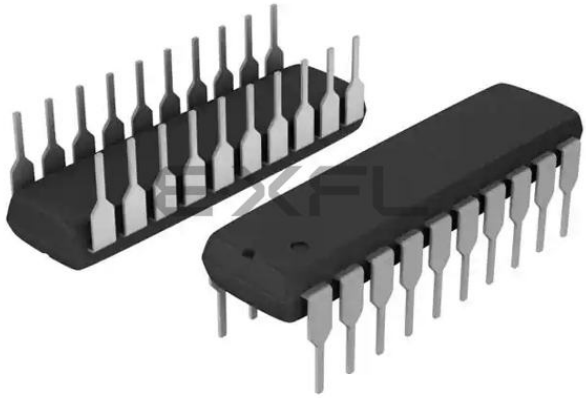

Welcome to E-XFL.COM

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, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
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	A/D 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/z8f0421ph020eg

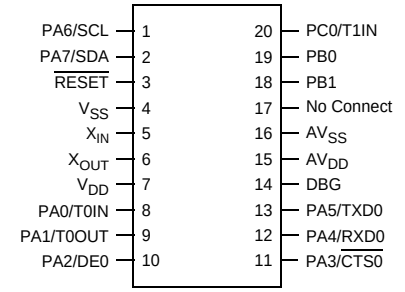

Figure 4. The Z8F0811 and Z8F0411 MCUs in 20-Pin SSOP and PDIP Packages

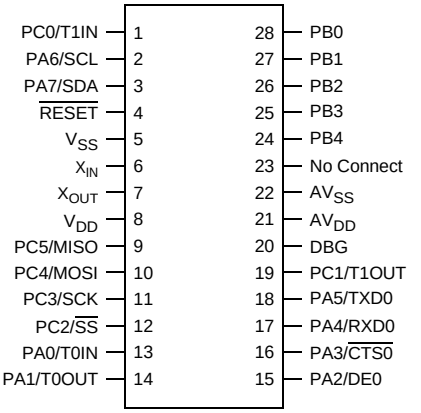

Figure 5. The Z8F0812 and Z8F0412 MCUs in 28-Pin SOIC and PDIP Packages

Address Space

The eZ8 CPU accesses three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, Peripheral, and GPIO 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 hold data only

These three address spaces are covered briefly in the following sections. For more information about the eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download at www.zilog.com.

Register File

The Register File address space in the Z8 Encore! XP® F0822 Series is 4KB (4096 bytes). It is composed of two sections: Control registers and General-Purpose registers. When instructions are executed, registers are read from when defined as sources and written to when defined as destinations. 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 1KB Register File address space is 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-byte Control Register section is reserved (unavailable). Reading from the reserved Register File addresses returns an undefined value. Writing to reserved Register File addresses is not recommended by Zilog because it can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. Z8 Encore! XP® F0822 Series contains 1KB 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.

General-Purpose Input/Output

Z8 Encore! XP® F0822 Series products support a maximum of 19 Port A–C pins for General-Purpose Input/Output (GPIO) operations. Each port consists Control and Data registers. The GPIO Control registers are used to determine data direction, open-drain, output drive current, programmable pull-ups, Stop Mode Recovery functionality, and alternate pin functions. Each port pin is individually programmable. Ports A and C support 5 V-tolerant inputs.

GPIO Port Availability by Device

Table 11 lists the port pins available with each device and package type.

Table 11. Port Availability by Device and Package Type

Devices	Package	Port A	Port B	Port C
Z8X0821, Z8X0811, Z8X0421, Z8X0411	20-pin	[7:0]	[1:0]	[0]
Z8X0822, Z8X0812, Z8X0422, Z8X0412	28-pin	[7:0]	[4:0]	[5:0]

Architecture

Figure 8 displays a simplified block diagram of a GPIO port pin. It does not display the ability to accommodate alternate functions, variable port current drive strength, and programmable pull-up.

GPIO Alternate Functions

Many of the GPIO port pins are used as both general-purpose I/O and to provide access to on-chip peripheral functions such as timers and serial communication devices. The Port A–C Alternate Function subregisters configure these pins for either general-purpose I/O or alternate function operation. When a pin is configured for alternate function, control of the port-pin direction (input/output) is passed from the Port A–C Data Direction registers to the alternate function assigned to this pin. Table 12 lists the alternate functions associated with each port pin.

Architecture

Figure 9 displays a block diagram of the interrupt controller.

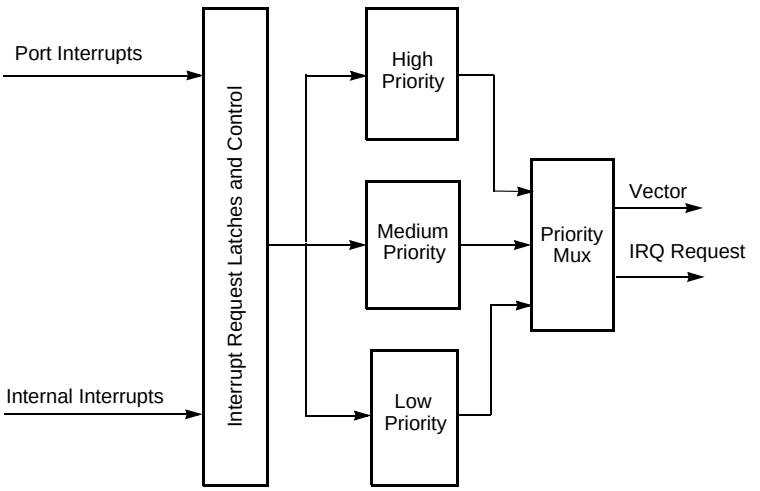

Figure 9. Interrupt Controller Block Diagram

Operation

This section describes the operational aspects of the following functions.

Master Interrupt Enable: see page 42

Interrupt Vectors and Priority: see page 43

Interrupt Assertion: see page 43

Software Interrupt Assertion: see page 44

Master Interrupt Enable

The master interrupt enable bit (IRQE) in the Interrupt Control Register globally enables and disables interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an Enable Interrupt (EI) instruction
- Execution of an Return from Interrupt (IRET) instruction

Bit	Description
[7:0] PWMH, PWML	**Pulse-Width Modulator High and Low Bytes** These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control (TxCTL) Register. The TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in CAPTURE or CAPTURE/COMPARE modes.

Timer 0–3 Control 0 Registers

The Timer 0–3 Control 0 (TxCTL0) registers, shown in Table 45, allow cascading of the Timers.

Table 45. Timer 0–3 Control 0 Registers (TxCTL0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			CSC	Reserved			
RESET	0							
R/W	R/W							
Address	F06H, F0EH, F16H, F1EH							

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 000.
[4] CSC	**Cascade Timers** 0 = Timer Input signal comes from the pin. 1 = For Timer 0, input signal is connected to Timer 1 output. For Timer 1, the input signal is connected to the Timer 0 output.
[3:0]	**Reserved** These bits are reserved and must be programmed to 0000.

UART Data and Error Handling Procedure

Figure 16 displays the recommended procedure for UART receiver ISRs.

Baud Rate Generator Interrupts

If the BRG interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the BRG to function as an additional counter if the UART functionality is not employed.

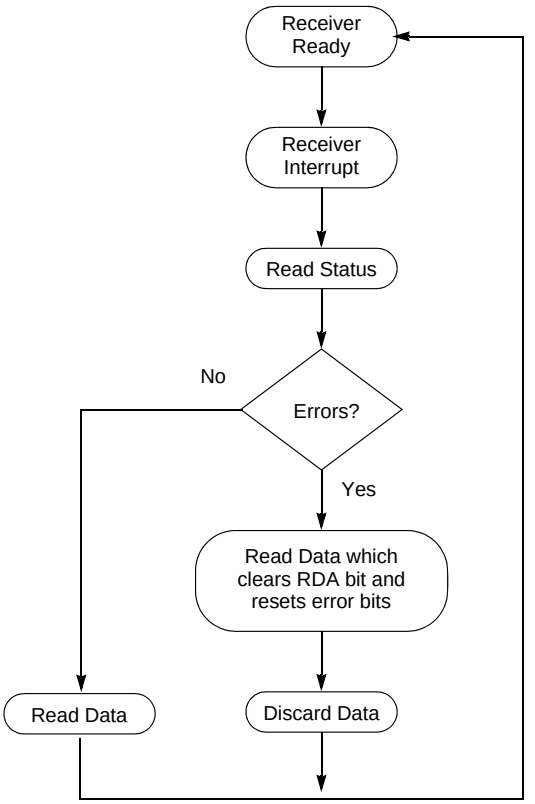

Figure 16. UART Receiver Interrupt Service Routine Flow

Bit	Description (Continued)
[2] BRGCTL	**Baud Rate Control** This bit causes different UART behavior depending on whether the UART receiver is enabled (REN = 1 in the UART Control 0 Register). When the UART receiver is not enabled, this bit determines whether the BRG will issue interrupts. 0 = Reads from the Baud Rate High and Low Byte registers return the BRG reload value 1 = The BRG generates a receive interrupt when it counts down to zero. Reads from the Baud Rate High and Low Byte registers return the current BRG count value. When the UART receiver is enabled, this bit allows reads from the Baud Rate registers to return the BRG count value instead of the reload value. 0 = Reads from the Baud Rate High and Low Byte registers return the BRG reload value. 1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the Timers, there is no mechanism to latch the High Byte when the Low Byte is read.
[1] RDAIRQ	**Receive Data Interrupt Enable** 0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller. 1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.
[0] IREN	**Infrared Encoder/Decoder Enable** 0 = Infrared Encoder/Decoder is disabled. UART operates normally operation. 1 = Infrared Encoder/Decoder is enabled. The UART transmits and receives data through the Infrared Encoder/Decoder.

UART Address Compare Register

The UART Address Compare Register, shown in Table 59, stores the multinode network address of the UART. When the MPMD[1] bit of UART Control Register 0 is set, all incoming address bytes will be compared to the value stored in the Address Compare Register. Receive interrupts and RDA assertions will only occur in the event of a match.

Table 59. UART Address Compare Register (U0ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
RESET	0							
R/W	R/W							
Address	F45H							

Bit	Description
[7:0] COMP_ADDR	**Compare Address** This 8-bit value is compared to the incoming address bytes.

SPI Control Register Definitions

This section defines the features of the following Serial Peripheral Interface registers.

SPI Data Register: see page 109

SPI Control Register: see page 110

SPI Status Register: see page 111

SPI Mode Register: see page 112

SPI Diagnostic State Register: see page 113

SPI Baud Rate High and Low Byte Registers: see page 114

SPI Data Register

The SPI Data Register, shown in Table 64, stores both the outgoing (transmit) data and the incoming (receive) data. Reads from the SPI Data Register always return the current contents of the 8-bit Shift Register. Data is shifted out starting with bit 7. The last bit received resides in bit position 0.

With the SPI configured as a Master, writing a data byte to this register initiates the data transmission. With the SPI configured as a Slave, writing a data byte to this register loads the shift register in preparation for the next data transfer with the external Master. In either the Master or Slave modes, if a transmission is already in progress, writes to this register are ignored and the Overrun error flag, OVR, is set in the SPI Status Register.

When the character length is less than 8 bits (as set by the NUMBITS field in the SPI Mode Register), the transmit character must be set as *left-justified* in the SPI Data Register. A received character of less than 8 bits is right justified (last bit received is in bit position 0). For example, if the SPI is configured for 4-bit characters, the transmit characters must be written to SPIDATA[7:4] and the received characters are read from SPIDATA[3:0].

Table 64. SPI Data Register (SPIDATA)

Bit	7	6	5	4	3	2	1	0
Field	DATA							
RESET	X							
R/W	R/W							
Address	F60H							

Bit	Description
[7:0] DATA	**SPI Data** Transmit and/or receive data.

SPI Baud Rate High and Low Byte Registers

The SPI Baud Rate High and Low Byte registers, shown in Tables 69 and 70, combine to form a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator. When configured as a general purpose timer, the interrupt interval is calculated using the following equation:

Interrupt Interval (s) = System Clock Period (s) × BRG[15:0]

Table 69. SPI Baud Rate High Byte Register (SPIBRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1							
R/W	R/W							
Address	F66H							

Bit	Description
[7:0] BRH	**SPI Baud Rate High Byte** Most significant byte, BRG[15:8], of the SPI Baud Rate Generator's reload value.

Table 70. SPI Baud Rate Low Byte Register (SPIBRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Address	F67H							

Bit	Description
[7:0] BRL	**SPI Baud Rate Low Byte** Least significant byte, BRG[7:0], of the SPI Baud Rate Generator's reload value.

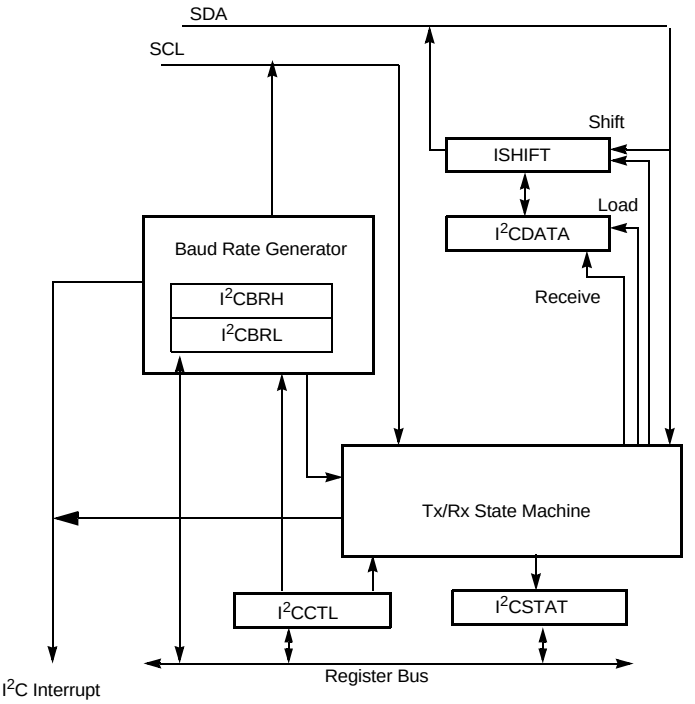

Figure 25. I²C Controller Block Diagram

Operation

The I²C Controller operates in MASTER Mode to transmit and receive data. Only a single master is supported. Arbitration between two masters must be accomplished in software. I²C supports the following operations:

- Master transmits to a 7-bit slave
- Master transmits to a 10-bit slave
- Master receives from a 7-bit slave
- Master receives from a 10-bit slave

5. The I2C Controller shifts the address and read bit out the SDA signal.
6. If the I2C Slave acknowledges the address by pulling the SDA signal Low during the next High period of SCL, the I2C Controller sets the ACK bit in the I2C Status Register. Continue to Step 7.

   If the slave does not acknowledge, the Not Acknowledge interrupt occurs (NCKI bit is set in the Status Register, ACK bit is cleared). Software responds to the Not Acknowledge interrupt by setting the stop bit and clearing the TXI bit. The I2C Controller sends the stop condition on the bus and clears the stop and NCKI bits. The transaction is complete; ignore the remainder of this sequence.
7. The I2C Controller shifts in the byte of data from the I2C Slave on the SDA signal. The I2C Controller sends a Not Acknowledge to the I2C Slave if the NAK bit is set (last byte), else it sends an Acknowledge.
8. The I2C Controller asserts the Receive interrupt (RDRF bit set in the Status Register).
9. Software responds by reading the I2C Data Register which clears the RDRF bit. If there is only one more byte to receive, set the NAK bit of the I2C Control Register.
10. If there are more bytes to transfer, return to Step 7.
11. After the last byte is shifted in, a Not Acknowledge interrupt is generated by the I2C Controller.
12. Software responds by setting the stop bit of the I2C Control Register.
13. A stop condition is sent to the I2C Slave, the stop and NCKI bits are cleared.

Read Transaction with a 10-Bit Address

Figure 31 displays the read transaction format for a 10-bit addressed slave. The shaded regions indicate data transferred from the I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

S	Slave Address 1st 7 bits	W=0	A	Slave Address 2nd Byte	A	S	Slave Address 1st 7 bits	R=1	A	Data	A	Data	A	P

Figure 31. Receive Data Format for a 10-Bit Addressed Slave

The first seven bits transmitted in the first byte are 11110XX. The two XX bits are the two most significant bits of the 10-bit address. The lowest bit of the first byte transferred is the write control bit.

Observe the following procedure for the data transfer procedure for a read operation to a 10-bit addressed slave:

Table 83. Z8 Encore! XP® F0822 Series Information Area Map

Flash Memory Address (Hex)	Function
FE00H–FE3FH	Reserved
FE40H–FE53H	**Part Number** 20-character ASCII alphanumeric code Left-justified and filled with zeros
FE54H–FFFFH	Reserved

Operation

The Flash Controller provides the proper signals and timing for the Byte Programming, Page Erase, and Mass Erase functions within Flash memory. The Flash Controller contains a protection mechanism, using the Flash Control Register (FCTL), to prevent accidental programming or erasure. The following subsections provide details about the various operations (Lock, Unlock, Sector Protect, Byte Programming, Page Erase and Mass Erase).

Timing Using the Flash Frequency Registers

Before performing a program or erase operation in Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasure of Flash memory with system clock frequencies ranging from 20kHz through 20MHz (the valid range is limited to the device operating frequencies).

The Flash Frequency High and Low Byte registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit Flash Frequency value must contain the system clock frequency in kHz. This value is calculated using the following equation:

$$\text{FFREQ[15:0]} = \frac{\text{System Clock Frequency (Hz)}}{1000}$$

Caution: Flash programming and erasure are not supported for system clock frequencies below 20kHz, above 20MHz, or outside of the device operating frequency range. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper Flash programming and erase operations.

For more information about bypassing the Flash Controller, refer to the Third Party Flash Programming Support for Z8 Encore! MCU Application Note (AN0117), available for download at www.zilog.com.

Flash Controller Behavior in Debug Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the OCD:

- The Flash Write Protect option bit is ignored
- The Flash Sector Protect Register is ignored for programming and erase operations
- Programming operations are not limited to the page selected in the Page Select Register
- Bits in the Flash Sector Protect Register can be written to 1 or 0
- The second write of the Page Select Register to unlock the Flash Controller is not necessary
- The Page Select Register is written when the Flash Controller is unlocked
- The Mass Erase command is enabled

Flash Control Register Definitions

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

Flash Control Register: see page 150

Flash Status Register: see page 151

Page Select Register: see page 152

Flash Sector Protect Register: see page 152

Flash Frequency High and Low Byte Registers: see page 153

Bit	Description (Continued)
[1]	**Reserved** This bit is reserved and must always be 1.
[0] FWP	**Flash Write Protect*** These two option bits combine to provide three levels of Program memory protection. 0 = Programming, Page Erase, and Mass Erase using User Code is disabled. Mass Erase is available through the OCD. 1 = Programming and Page Erase are enabled for all of Flash program memory.

Note: *Applies only to the Flash versions of the F0822 Series of devices.

Flash Memory Address 0001H

Table 91. Options Bits at Flash Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U							
R/W	R/W							
Address	Program Memory 0001H							

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

Bit	Description
[7:0]	**Reserved** These option bits are reserved and must always be 1. This setting is the default for unprogrammed (erased) Flash.

Table 97. Absolute Maximum Ratings (Continued)

Parameter	Minimum	Maximum	Units	Notes
20-pin SSOP Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		250	mW	
Maximum current into V_{DD} or out of V_{SS}		69	mA	
20-pin PDIP Package Maximum Ratings at –40°C to 70°C				
Total power dissipation		775	mW	
Maximum current into V_{DD} or out of V_{SS}		215	mA	
20-pin PDIP Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		285	mW	
Maximum current into V_{DD} or out of V_{SS}		79	mA	
28-pin SOIC Package Maximum Ratings at –40°C to 70°C				
Total power dissipation		450	mW	
Maximum current into V_{DD} or out of V_{SS}		125	mA	
28-pin SOIC Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		260	mW	
Maximum current into V_{DD} or out of V_{SS}		73	mA	
28-pin PDIP Package Maximum Ratings at –40°C to 70°C				
Total power dissipation		1100	mW	
Maximum current into V_{DD} or out of V_{SS}		305	mA	
28-pin PDIP Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		400	mW	
Maximum current into V_{DD} or out of V_{SS}		110	mA	

Note: This voltage applies to all pins except the following: V_{DD}, AV_{DD}, V_{REF}, pins that support analog input (Port B), and where otherwise noted.

DC Characteristics

Table 98 lists the DC characteristics of the Z8 Encore! XP® F0822 Series products. All voltages are referenced to V_{SS}, the primary system ground.

Table 98. DC Characteristics

		T_A = –40°C to 105°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
V_{DD}	Supply Voltage	2.7	–	3.6	V	
V_{IL1}	Low Level Input Voltage	-0.3	–	0.3*V_{DD}	V	For all input pins except $\overline{\text{RESET}}$, DBG, and X_{IN}.
V_{IL2}	Low Level Input Voltage	–0.3	–	0.2*V_{DD}	V	For $\overline{\text{RESET}}$, DBG, and X_{IN}.
V_{IH1}	High Level Input Voltage	0.7*V_{DD}	–	5.5	V	Ports A and C pins when their programmable pull-ups are disabled.
V_{IH2}	High Level Input Voltage	0.7*V_{DD}	–	V_{DD}+0.3	V	Port B pins. Ports A and C pins when their programmable pull-ups are enabled.
V_{IH3}	High Level Input Voltage	0.8*V_{DD}	–	V_{DD}+0.3	V	$\overline{\text{RESET}}$, DBG, and X_{IN} pins.
V_{OL1}	Low Level Output Voltage	–	–	0.4	V	I_{OL} = 2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OH1}	High Level Output Voltage	2.4	–	–	V	I_{OH} = –2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 20 mA; V_{DD} = 3.3V High Output Drive enabled T_A = –40°C to +70°C
V_{OH2}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = –20 mA; V_{DD} = 3.3V High Output Drive enabled; T_A = –40°C to +70°C
V_{OL3}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 15mA; V_{DD} = 3.3V High Output Drive enabled; T_A = +70°C to +105°C
V_{OH3}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = 15 mA; V_{DD} = 3.3V High Output Drive enabled; T_A = +70°C to +105°C
V_{RAM}	RAM Data Retention	0.7	–	–	V	
I_{IL}	Input Leakage Current	–5	–	+5	µA	V_{DD} = 3.6V; V_{IN} = V_{DD} or V_{SS}[1]

I²C Timing

Figure 53 and Table 111 provide timing information for I²C pins.

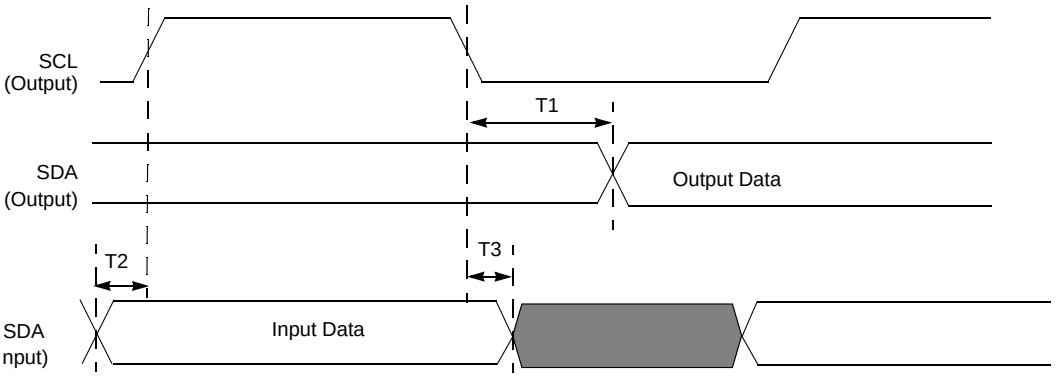

Figure 53. I²C Timing

Table 111. I²C Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
I²C			
T_1	SCL Fall to SDA output delay	SCL period/4	
T_2	SDA Input to SCL rising edge Setup Time	0	
T_3	SDA Input to SCL falling edge Hold Time	0	

resented 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.

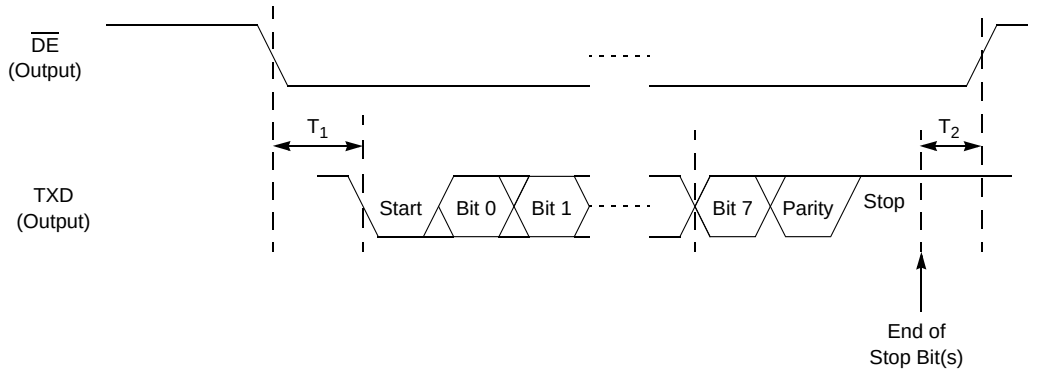

Figure 55. UART Timing without $\overline{CTS}$

Table 113. 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 * X_{IN} period
T_2	End of Stop Bit(s) to $\overline{DE}$ Deassertion Delay	1 * X_{IN} period	2 * X_{IN} period

Table 127. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
POPX dst	dst ← @SP SP ← SP + 1	ER		D8	-	-	-	-	-	-	3	2
PUSH src	SP ← SP – 1 @SP ← src	R		70	-	-	-	-	-	-	2	2
		IR		71							2	3
PUSHX src	SP ← SP – 1 @SP ← src	ER		C8	-	-	-	-	-	-	3	2
RCF	C ← 0			CF	0	-	-	-	-	-	1	2
RET	PC ← @SP SP ← SP + 2			AF	-	-	-	-	-	-	1	4
RL dst		R		90	*	*	*	*	-	-	2	2
		IR		91							2	3
RLC dst		R		10	*	*	*	*	-	-	2	2
		IR		11							2	3
RR dst		R		E0	*	*	*	*	-	-	2	2
		IR		E1							2	3
RRC dst		R		C0	*	*	*	*	-	-	2	2
		IR		C1							2	3
SBC dst, src	dst ← dst – src – C	r	r	32	*	*	*	*	1	*	2	3
		r	Ir	33							2	4
		R	R	34							3	3
		R	IR	35							3	4
		R	IM	36							3	3
		IR	IM	37							3	4
SBCX dst, src	dst ← dst – src – C	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39							4	3
SCF	C ← 1			DF	1	-	-	-	-	-	1	2
SRA dst		R		D0	*	*	*	0	-	-	2	2
		IR		D1							2	3

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

D

E

F