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	eZ80
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	24
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80f93az020sg

Table 1. 100-Pin LQFP Pin Identification of the eZ80F92 Device (Continued)

Pin No	Symbol	Function	Signal Direction	Description
67	V_{DD}	Power Supply		Power Supply.
68	PD0	GPIO Port D	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	TxD0	UART Transmit Data	Output	This pin is used by the UART to transmit asynchronous serial data. This signal is multiplexed with PD0.
	IR_TxD	IrDA Transmit Data	Output	This pin is used by the IrDA encoder/ decoder to transmit serial data. This signal is multiplexed with PD0.
69	PD1	GPIO Port D	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	RxD0	Receive Data	Input	This pin is used by the UART to receive asynchronous serial data. This signal is multiplexed with PD1.
	IR_RxD	IrDA Receive Data	Input	This pin is used by the IrDA encoder/ decoder to receive serial data. This signal is multiplexed with PD1.
70	PD2	GPIO Port D	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	$\overline{RTS0}$	Request To Send	Output, Active Low	Modem control signal from UART. This signal is multiplexed with PD2.

Table 1. 100-Pin LQFP Pin Identification of the eZ80F92 Device (Continued)

Pin No	Symbol	Function	Signal Direction	Description
88	PB0	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	T0_IN	Timer 0 In	Input	Alternate clock source for Programmable Reload Timers 0 and 2. This signal is multiplexed with PB0.
89	PB1	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	T1_IN	Timer 1 In	Input	Alternate clock source for Programmable Reload Timers 1 and 3. This signal is multiplexed with PB1.
90	PB2	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	$\overline{SS}$	Slave Select	Input, Active Low	The slave select input line is used to select a slave device in SPI mode. This signal is multiplexed with PB2.
91	PB3	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	SCK	SPI Serial Clock	Bidirectional	SPI serial clock. This signal is multiplexed with PB3.

Table 2. Pin Characteristics of the eZ80F92 Device (Continued)

Pin No	Symbol	Direction	Reset Direction	Active Low/High	Tristate Output	Pull Up/Down	Schmitt Trigger Input	Open Drain/ Source
71	PD3	I/O	I	N/A	Yes	No	No	OD & OS
72	PD4	I/O	I	N/A	Yes	No	No	OD & OS
73	PD5	I/O	I	N/A	Yes	No	No	OD & OS
74	PD6	I/O	I	N/A	Yes	No	No	OD & OS
75	PD7	I/O	I	N/A	Yes	No	No	OD & OS
76	PC0	I/O	I	N/A	Yes	No	No	OD & OS
77	PC1	I/O	I	N/A	Yes	No	No	OD & OS
78	PC2	I/O	I	N/A	Yes	No	No	OD & OS
79	PC3	I/O	I	N/A	Yes	No	No	OD & OS
80	PC4	I/O	I	N/A	Yes	No	No	OD & OS
81	PC5	I/O	I	N/A	Yes	No	No	OD & OS
82	PC6	I/O	I	N/A	Yes	No	No	OD & OS
83	PC7	I/O	I	N/A	Yes	No	No	OD & OS
84	V_{SS}							
85	X_{IN}	I	I	N/A	N/A	No	No	N/A
86	X_{OUT}	O	O	N/A	No	No	No	No
87	V_{DD}							
88	PB0	I/O	I	N/A	Yes	No	No	OD & OS
89	PB1	I/O	I	N/A	Yes	No	No	OD & OS
90	PB2	I/O	I	N/A	Yes	No	No	OD & OS
91	PB3	I/O	I	N/A	Yes	No	No	OD & OS
92	PB4	I/O	I	N/A	Yes	No	No	OD & OS
93	PB5	I/O	I	N/A	Yes	No	No	OD & OS
94	PB6	I/O	I	N/A	Yes	No	No	OD & OS
95	PB7	I/O	I	N/A	Yes	No	No	OD & OS
96	V_{DD}							
97	V_{SS}							
98	SDA	I/O	I	N/A	Yes	Up	No	OD

an interrupt request signal to the CPU. Any time a port pin is configured for edge-triggered interrupt, writing a 1 to that pin's Port *x* Data register causes a reset of the edge-detected interrupt. The programmer must set the bit in the Port *x* Data register to 1 before entering either single or dual edge-triggered interrupt mode for that port pin.

When configured for dual edge-triggered interrupt mode (GPIO Mode 6), both a rising and a falling edge on the pin cause an interrupt request to be sent to the CPU.

When configured for single edge-triggered interrupt mode (GPIO Mode 9), the value in the Port *x* Data register determines if a positive or negative edge causes an interrupt request. A 0 in the Port *x* Data register bit sets the selected pin to generate an interrupt request for falling edges. A 1 in the Port *x* Data register bit sets the selected pin to generate an interrupt request for rising edges.

GPIO Control Registers

The 12 GPIO Control Registers operate in groups of four with a set for each Port (B, C, and D). Each GPIO port features a Port Data register, Port Data Direction register, Port Alternate register 1, and Port Alternate register 2.

Port *x* Data Registers

When the port pins are configured for one of the output modes, the data written to the Port *x* Data registers, listed in Table 7, are driven on the corresponding pins. In all modes, reading from the Port *x* Data registers always returns the current sampled value of the corresponding pins.

When the port pins are configured as edge-triggered interrupt sources, writing a 1 to the corresponding bit in the Port *x* Data register clears the interrupt signal that is sent to the CPU. When the port pins are configured for edge-selectable interrupts or level-sensitive interrupts, the value written to the Port *x* Data register bit selects the interrupt edge or interrupt level. See Table 6 on page 39 for more information.

Table 7. Port *x* Data Registers; (PB_DR = 009Ah, PC_DR = 009Eh, PD_DR = 00A2h)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Note: X = Undefined; R/W = Read/Write.								

Signal timing for Intel™ bus mode with multiplexed address and data is displayed for a Read operation in Figure 14 and for a Write operation in Figure 15 on page 62. In the figures, each Intel bus mode state is 2 CPU system clock cycles in duration. Figure 14 and Figure 15 on page 62 also display the assertion of one wait state (T_{WAIT}) by the selected peripheral.

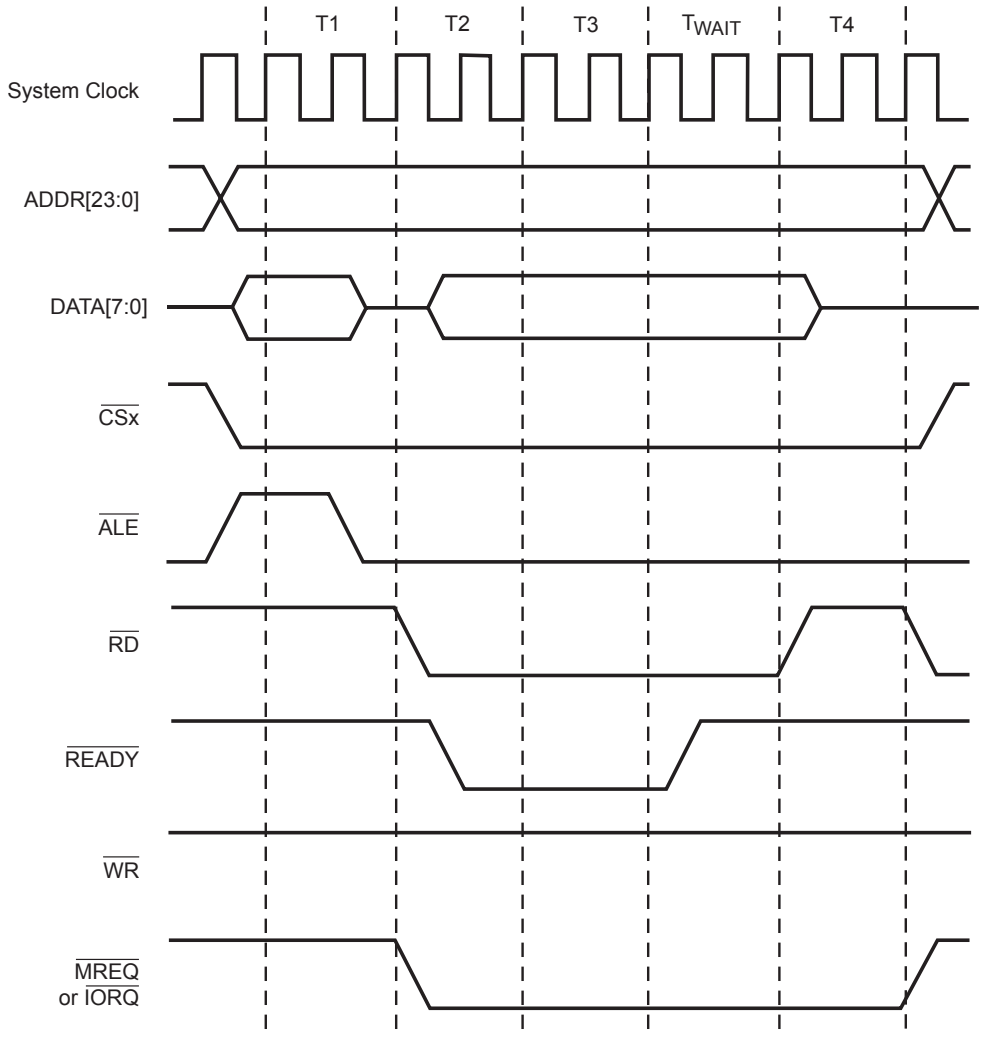


Figure 14. Example: Intel™ Bus Mode Read Timing—Multiplexed Address and Data Bus

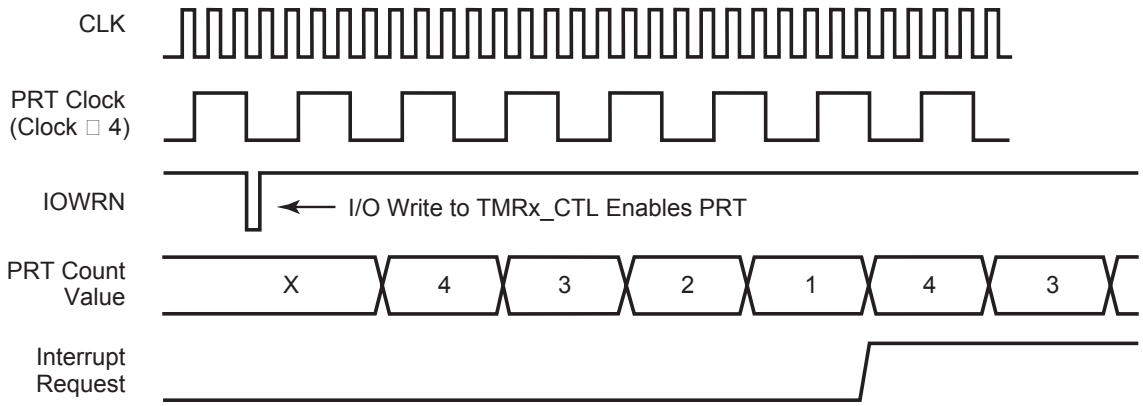


Figure 22. PRT CONTINUOUS Mode Operation Example

Table 30. PRT CONTINUOUS Mode Operation Example

Parameter	Control Register(s)	Value
PRT Enabled	TMRx_CTL[0]	1
Reload and Restart Enabled	TMRx_CTL[1]	1
PRT Clock Divider = 4	TMRx_CTL[3:2]	00b
CONTINUOUS Mode	TMRx_CTL[4]	1
PRT Interrupt Enabled	TMRx_CTL[6]	1
PRT Reload Value	{TMRx_RR_H, TMRx_RR_L}	0004h

Reading the Current Count Value

The CPU is capable of reading the current count value while the timer is running. This Read event does not affect timer operation. The High byte of the current count value is latched during a Read of the Low byte.

Timer Interrupts

The timer interrupt flag, PRT_IRQ, is set to 1 whenever the timer reaches its end-of-count value, 0000h, in SINGLE PASS mode, or when the timer reloads the start value in CONTINUOUS mode. The interrupt flag is only set when the timer reaches 0000h (or reloads) from 0001h. The timer interrupt flag is not set to 1 when the timer is loaded with the value 0000h, which selects the maximum time-out period.

The CPU can be programmed to poll the PRT_IRQ bit for the time-out event. Alternatively, an interrupt service request signal can be sent to the CPU by setting IRQ_EN to 1.

Universal Asynchronous Receiver/Transmitter

The UART module implements all of the logic required to support several asynchronous communications protocols. The module also implements two separate 16-byte-deep FIFOs for both transmission and reception. A block diagram of the UART is displayed in Figure 25.

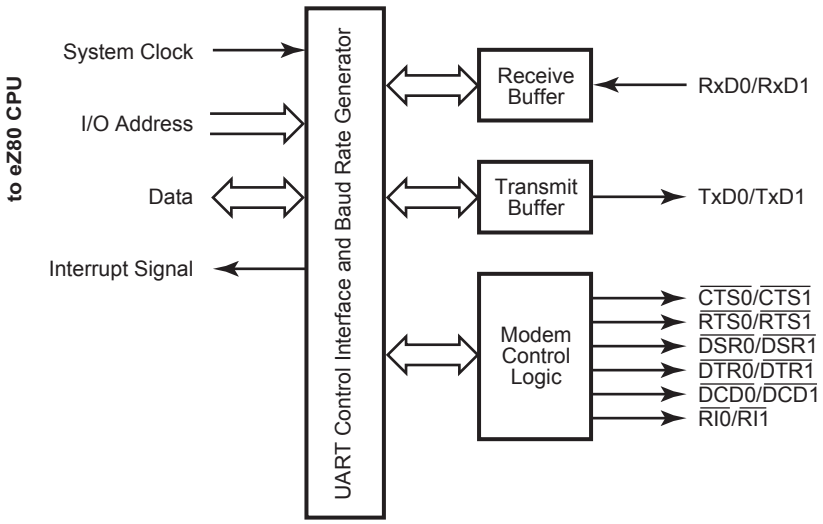


Figure 25.UART Block Diagram

The UART module provides the following asynchronous communication protocol-related features and functions:

- 5-, 6-, 7-, 8- or 9-bit data transmission
- Even/odd, space/mark, or no parity bit generation and detection
- Start and stop bit generation and detection
- Supports up to two stop bits
- Line break detection and generation
- Receiver overrun and framing errors detection
- Logic and associated I/O to provide modem handshake capability

UART Interrupt Identification Register

The Read Only UARTx_IIR register allows the user to check whether the FIFO is enabled and the status of interrupts. These registers share the same I/O addresses as the UARTx_FCTL registers. See Table 57 and Table 58.

Table 57. UART Interrupt Identification Registers(UART0_IIR = 00C2h, UART1_IIR = 00D2h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	1
CPU Access	R	R	R	R	R	R	R	R
Note: R = Read only.								

Bit Position	Value	Description
[7:6] FSTS	00	FIFO is disabled.
	10	Receive FIFO is disabled (MULTIDROP mode).
	11	FIFO is enabled.
[5:4]	00	Reserved.
[3:1] INSTS	000–110	**Interrupt Status Code** The code indicated in these three bits is valid only if INTBIT is 1. If two internal interrupt sources are active and their respective enable bits are High, only the higher priority interrupt is seen by the application. The lower-priority interrupt code is indicated only after the higher-priority interrupt is serviced. Table 58 lists the interrupt status codes.
0 INTBIT	0	There is an active interrupt source within the UART.
	1	There is not an active interrupt source within the UART.

Table 58. UART Interrupt Status Codes

INSTS Value	Priority	Interrupt Type
011	Highest	Receiver Line Status
010	Second	Receiver Data Ready or Trigger Level
110	Third	Character Time-out
101	Fourth	Transmission Complete

Table 65. UART Modem Status Registers(UART0_MSR = 00C6h, UART1_MSR = 00D6h)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	R	R	R	R	R	R	R	R
Note: R = Read only.								

Bit Position	Value	Description
7 DCD	0–1	**Data Carrier Detect** In NORMAL mode, this bit reflects the inverted state of the $\overline{\text{DCD}}$x input pin. In LOOP BACK mode, this bit reflects the value of the UARTx_MCTL[3] = out2.
6 RI	0–1	**Ring Indicator** In NORMAL mode, this bit reflects the inverted state of the $\overline{\text{RI}}$x input pin. In LOOP BACK mode, this bit reflects the value of the UARTx_MCTL[2] = out1.
5 DSR	0–1	**Data Set Ready** In NORMAL mode, this bit reflects the inverted state of the $\overline{\text{DSR}}$x input pin. In LOOP BACK mode, this bit reflects the value of the UARTx_MCTL[0] = DTR.
4 CTS	0–1	**Clear To Send** In NORMAL mode, this bit reflects the inverted state of the $\overline{\text{CTS}}$x input pin. In LOOP BACK mode, this bit reflects the value of the UARTx_MCTL[1] = RTS.
3 DDCD	0–1	**Delta Status Change of $\overline{\text{DCD}}$** This bit is set to 1 whenever the $\overline{\text{DCD}}$x pin changes state. This bit is reset to 0 when the UARTx_MSR register is read.
2 TERI	0–1	**Trailing Edge Change on $\overline{\text{RI}}$** This bit is set to 1 whenever a falling edge is detected on the $\overline{\text{RI}}$x pin. This bit is reset to 0 when the UARTx_MSR register is read.
1 DDSR	0–1	Delta Status Change of $\overline{\text{DSR}}$ This bit is set to 1 whenever the $\overline{\text{DSR}}$x pin changes state. This bit is reset to 0 when the UARTx_MSR register is read.
0 DCTS	0–1	Delta Status Change of $\overline{\text{CTS}}$ This bit is set to 1 whenever the $\overline{\text{CTS}}$x pin changes state. This bit is reset to 0 when the UARTx_MSR register is read.

Infrared Encoder/Decoder

The eZ80F92 device contains a UART to infrared encoder/decoder (endec). The IrDA endec is integrated with the on-chip UART0 to allow easy communication between the CPU and IrDA Physical Layer Specification Version 1.4-compatible infrared transceivers, as displayed in Figure 26. Infrared communication provides secure, reliable, high-speed, low-cost, point-to-point communication between PCs, PDAs, mobile telephones, printers, and other infrared-enabled devices.

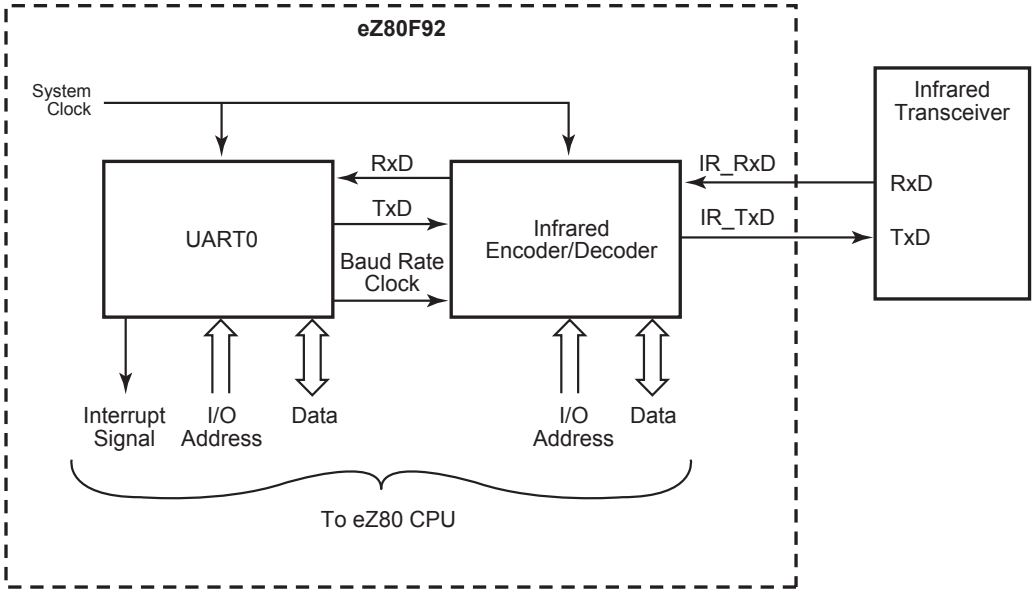


Figure 26. Infrared System Block Diagram

Functional Description

When the IrDA endec is enabled, the transmit data from the on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver. Likewise, data received from the infrared transceiver is decoded by the endec and passed to the UART. Communication is half-duplex, meaning that simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART Baud Rate Generator, and supports IrDA standard baud rates from 9600 bps to 115.2 kbps. Higher baud rates than 115.2 kbps are possible, but do not meet IrDA specifications for these data rates. The UART must be enabled to use the

Table 67. IrDA Physical Layer 1.4 Pulse Durations Specifications (Continued)

Baud Rate	Minimum Pulse Width	Maximum Pulse Width
57600	1.41 s	4.34 s
115200	1.41 s	2.23 s

Receiver Frequency Divider

The IrDA receiver uses a 6-bit frequency divider. The value is derived from the system clock to measure IR_RxD pulses. The IrDA endec detects pulses that are within the IrDA Physical Layer specified minimum and maximum ranges, with system clock frequencies from 5 MHz up to 50 MHz.

The upper four bits of the frequency divider factor are set via the FREQ_DIV bit in the IR_CTL register, based on the following equation:

$$\text{Frequency Divider Factor} = \frac{\text{System Clock Frequency (MHz)}}{\text{Target Frequency of 3.33 MHz}}$$

The remaining lower two bits of the divider are set to 03h. The target frequency corresponds to a period of 1.2 seconds. The FREQ_DIV value must be rounded to the nearest integer and the resulting period of the 6-bit frequency divider must not be larger than 1.4 seconds, which is the IrDA defined minimum pulse width. If the period is greater than 1.4 seconds, FREQ_DIV should be rounded to the next lower integer. The receiver frequency divider value versus the system clock frequency is shown in table, below.

Table 68. Frequency Divider Values

System Clock	FREQ_DIV
< 5.0 MHz	00h*
5.0–7.8 MHz	01h
7.8–10.8 MHz	02h
10.8–13.6 MHz	03h
13.6–25 MHz	FLOOR[4-bit Frequency Divider Factor]
25–50 MHz	ROUND[4-bit Frequency Divider Factor]

Note: *The frequency divider is disabled when set to 00h.

Table 87. I²C Control Registers(I2C_CTL = 00CBh)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Note: R/W = Read/Write; R = Read Only.								

Bit Position	Value	Description
7 IEN	0	I^2C interrupt is disabled.
	1	I^2C interrupt is enabled.
6 ENAB	0	The I^2C bus (SCL/SDA) is disabled and all inputs are ignored.
	1	The I^2C bus (SCL/SDA) is enabled.
5 STA	0	Master mode START condition is sent.
	1	Master mode start-transmit START condition on the bus.
4 STP	0	Master mode STOP condition is sent.
	1	Master mode stop-transmit STOP condition on the bus.
3 IFLG	0	I^2C interrupt flag is not set.
	1	I^2C interrupt flag is set.
2 AAK	0	Not Acknowledge.
	1	Acknowledge.
[1:0]	00	Reserved.

Because many register Read/Write operations exhibit secondary effects, such as clearing flags or causing operations to commence, the effects of the Read/Write operations during a ZDI BREAK must be taken into consideration.

Bus Requests During ZDI DEBUG Mode

The ZDI block on the eZ80F92 device allows an external device to take control of the address and data bus while the eZ80F92 device is in DEBUG mode. ZDI_BUSACK_EN causes ZDI to allow or prevent acknowledgement of bus requests by external peripherals. The bus acknowledge only occurs at the end of the current ZDI operation (indicated by a High during the single-bit byte separator). The default reset condition is for bus acknowledgement to be disabled. To allow bus acknowledgement, the ZDI_BUSACK_EN must be written.

When an external bus request ($\overline{\text{BUSREQ}}$ pin asserted) is detected, ZDI waits until completion of the current operation before responding. ZDI acknowledges the bus request by asserting the bus acknowledge ($\overline{\text{BUSACK}}$) signal. If the ZDI block is not currently shifting data, it acknowledges the bus request immediately. ZDI uses the single-bit byte separator of each data word to determine if it is at the end of a ZDI operation. If the bit is a logical 0, ZDI does not assert $\overline{\text{BUSACK}}$ to allow additional data Read or Write operations. If the bit is a logical 1, indicating completion of the ZDI commands, $\overline{\text{BUSACK}}$ is asserted.

Potential Hazards of Enabling Bus Requests During DEBUG Mode

There are some potential hazards that the user must be aware of when enabling external bus requests during ZDI DEBUG mode. First, when the address and data bus are used by an external source, ZDI must only access ZDI registers and internal CPU registers to prevent possible Bus contention. The bus acknowledge status is reported in the ZDI_BUS_STAT register. The $\overline{\text{BUSACK}}$ output pin also indicates the bus acknowledge state.

A second hazard is that when a bus acknowledge is granted, the ZDI is subject to any WAIT states that are assigned to the device currently accessed by the external peripheral. To prevent data errors, ZDI should avoid data transmission while another device is controlling the bus.

Finally, exiting ZDI DEBUG mode while an external peripheral controls the address and data buses, as indicated by $\overline{\text{BUSACK}}$ assertion, may produce unpredictable results.

ZDI Write Memory Register

A Write to the ZDI Write Memory register causes the eZ80F92 device to write the 8-bit data to the memory location specified by the current address in the program counter. In Z80® MEMORY mode, this address is {MBASE, PC[15:0]}. In ADL MEMORY mode, this address is PC[23:0]. The program counter, PC, increments after each data Write. However, the ZDI register address does not increment automatically when this register is accessed. As a result, the ZDI master is allowed to write any number of data bytes by writing to this address one time followed by any number of data bytes. See Table 102.

Table 102. ZDI Write Memory Register(ZDI_WR_MEM = 30h in the ZDI Register Write Only Address Space)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	W	W	W	W	W	W	W	W
Note: X = Undefined; W = Write.								

Bit Position	Value	Description
[7:0] ZDI_WR_MEM	00h–FFh	The 8-bit data that is transferred to the ZDI slave following a Write to this address is written to the address indicated by the current program counter. The program counter is incremented following each 8 bits of data. In Z80 MEMORY mode, ({MBASE, PC[15:0]}) ← 8 bits of transferred data. In ADL MEMORY mode, (PC[23:0]) ← 8 bits of transferred data.

Bit Position	Value	Description
4 ADL	0	The CPU is operating in Z80® MEMORY mode. (ADL bit = 0)
	1	The CPU is operating in ADL MEMORY mode. (ADL bit = 1)
3 MADL	0	The CPU's Mixed-Memory mode (MADL) bit is reset to 0.
	1	The CPU's Mixed-Memory mode (MADL) bit is set to 1.
2 IEF1	0	The CPU's Interrupt Enable Flag 1 is reset to 0. Maskable interrupts are disabled.
	1	The CPU's Interrupt Enable Flag 1 is set to 1. Maskable interrupts are enabled.
[1:0]	00	Reserved.

ZDI Read Register Low, High, and Upper

The ZDI register Read Only address space offers Low, High, and Upper functions, which contain the value read by a Read operation from the ZDI Read/Write Control register (ZDI_RW_CTL). This data is valid only while in ZDI BREAK mode and only if the instruction is read by a request from the ZDI Read/Write Control register. See Table 107.

Table 107. ZDI Read Register Low, High and Upper(ZDI_RD_L = 10h, ZDI_RD_H = 11h, and ZDI_RD_U = 12h in the ZDI Register Read Only Address Space)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R	R	R	R	R	R	R	R
Note: R = Read Only.								

Bit Position	Value	Description
[7:0] ZDI_RD_L, ZDI_RD_H, or ZDI_RD_U	00h–FFh	Values read from the memory location as requested by the ZDI Read Control register during a ZDI Read operation. The 24-bit value is supplied by {ZDI_RD_U, ZDI_RD_H, ZDI_RD_L}.

On-Chip Instrumentation

Introduction to On-Chip Instrumentation

On-Chip Instrumentation[1] (OCI™) for the eZ80® CPU core enables powerful debugging features. The OCI provides run control, memory and register visibility, complex break-points, and trace history features.

The OCI employs all of the functions of the ZDI as described in the Zilog Debug Interface section that starts on page 162. It also adds the following debug features:

- Control via a 4-pin Joint Test Action Group (JTAG)-standard port that conforms to the IEEE Standard 1149.1 (Test Access Port and Boundary Scan Architecture)[2]
- Complex break point trigger functions
- Break point enhancements, such as the ability to:
  - Define two break point addresses that form a range
  - Break on masked data values
  - Start or stop trace
  - Assert a trigger output signal
- Trace history buffer
- Software break point instruction

There are four sections to the OCI:

1. JTAG interface
2. ZDI debug control
3. Trace buffer memory
4. Complex triggers

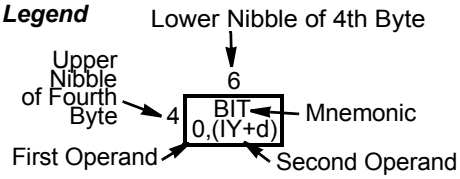
OCI Activation

OCI features clock initialization circuitry so that external debug hardware can be detected during power-up. The external debugger must drive the OCI clock pin (TCK) Low at least two system clock cycles prior to the end of the RESET to activate the OCI block. If TCK is High at the end of the RESET, the OCI block shuts down so that it does not draw power in normal product operation. When the OCI is shut down, ZDI is enabled directly and can

1. On-Chip Instrumentation and OCI are trademarks of First Silicon Solutions, Inc.
2. The eZ80F92 does not contain the boundary scan register required for 1149.1 compliance.

Table 141. Op Code Map—Fourth Byte After 0FDh, 0CBh, and dd*

Legend



Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0							RLC (IY+d)								RRC (IY+d)	
1							RL (IY+d)								RR (IY+d)	
2							SLA (IY+d)								SRA (IY+d)	
3															SRL (IY+d)	
4							BIT 0, (IY+d)								BIT 1, (IY+d)	
5							BIT 2, (IY+d)								BIT 3, (IY+d)	
6							BIT 4, (IY+d)								BIT 5, (IY+d)	
7							BIT 6, (IY+d)								BIT 7, (IY+d)	
8							RES 0, (IY+d)								RES 1, (IY+d)	
9							RES 2, (IY+d)								RES 3, (IY+d)	
A							RES 4, (IY+d)								RES 5, (IY+d)	
B							RES 6, (IY+d)								RES 7, (IY+d)	
C							SET 0, (IY+d)								SET 1, (IY+d)	
D							SET 2, (IY+d)								SET 3, (IY+d)	
E							SET 4, (IY+d)								SET 5, (IY+d)	
F							SET 6, (IY+d)								SET 7, (IY+d)	

Notes: d = 8-bit two's-complement displacement.

Table 145. DC Characteristics (Continued)

		T_A = 0 ºC to 70 ºC		T_A = 0 ºC to 105 ºC			
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Units**	**Conditions**
I_{RTC}	RTC Supply Current	2.5	10 Typical	2.5	10 Typical	µA	Supply current into RTC_V_{DD}; SLEEP mode[2].

Notes
1. This condition excludes all pins with on-chip pull-ups when driven Low.
2. RTC current increases when the eZ80F92 device is not in SLEEP mode as the RTC_VDD pin supplies power to system clock buffers within the Real-Time Clock circuit.

POR and VBO Electrical Characteristics

Table 146 lists the Power-On Reset and Voltage Brownout characteristics of the eZ80F92 device.

Table 146. POR and VBO Electrical Characteristics

		T_A = 0 ºC to +105 ºC				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Unit**	**Conditions**
V_{VBO}	VBO Voltage Threshold	2.40	2.55	2.85	V	$V_{CC} = V_{VBO}$
V_{POR}	POR Voltage Threshold	2.45	2.65	2.90	V	$V_{CC} = V_{POR}$
V_{HYST}	POR/VBO Hysteresis	50	100	150	mV	
T_{ANA}	POR/VBO analog RESET duration	40		100	µs	
T_{VBO_MIN}	VBO pulse reject period		10		µs	
V_{CCRAMP}	V_{CC} ramp rate requirements to guarantee proper RESET occurs	0.1		100	V/ms	

Typical Current Consumption Under Various Operating Conditions

In the following pages, Figure 53 on page 225 displays the typical current consumption of the eZ80F92 device versus the number of WAIT states while operating 25 ºC, 3.3 V, and with either a 5 MHz, 10 MHz, 15 MHz, or 20 MHz system clock. Figure 54 on page 226 displays the typical current consumption of the eZ80F92 device versus the system clock frequency while operating 25 ºC, 3.3 V, and using 0, 2, or 7 WAIT states. Figure 55 on page 227 displays the typical current consumption of the eZ80F92 device versus temperature while operating at 3.3 V, 7 WAIT states, and with either a 5 MHz, 10 MHz, 15 MHz or 20 MHz system clock. Figure 56 on page 228 displays the typical current consumption

Ordering Information

Table 156 lists a part name, a product specification index code, and a brief description of each part.

Table 156. Ordering Information;

Part Name	PSI	Description
eZ80F92	eZ80F92AZ020SC, eZ80F92AZ020SG	100-pin LQFP, 128 KB Flash memory, 8 KB SRAM, 20 MHz, Standard Temperature.
	eZ80F92AZ020EC, eZ80F92AZ020EG	100-pin LQFP, 128 KB Flash memory, 8 KB SRAM, 20 MHz, Extended Temperature.
eZ80F93	eZ80F93AZ020SC, eZ80F93AZ020SG	100-pin LQFP, 64 KB Flash memory, 4 KB SRAM, 20 MHz, Standard Temperature.
	eZ80F93AZ020EC, eZ80F93AZ020EG	100-pin LQFP, 64 KB Flash memory, 4 KB SRAM, 20 MHz, Extended Temperature.

Navigate your browser to Zilog's website to order the eZ80F92 or the eZ80F93. Or, contact your local Zilog Sales Office to order these devices. Zilog provides additional assistance on its Customer Service page, and is also here to help with technical support issues.

For Zilog's valuable software development tools and downloadable software, visit www.zilog.com.

Part Number Description

Zilog part numbers consist of a number of components, as listed in the following examples:

Zilog Base Products	
eZ80®	Zilog eZ80 CPU
F92	Product Number
AZ	Package
020	Speed
S or E	Temperature
C or G	Environmental Flow

W

X

Z