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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80f92az020eg

- Watchdog Timer (WDT)
- 24 bits of General-Purpose I/O and ZDI debug interfaces
- 100-pin LQFP package
- 3.0–3.6 V supply voltage with 5V tolerant inputs
- Operating Temperature Range:
  - Standard: 0 ºC to +70 ºC
  - Extended: –40 ºC to +105 ºC

Note: *All signals with an overline are active Low. For example, B/$\overline{W}$, for which WORD is active Low, and $\overline{B}$/W, for which BYTE is active Low.*

Power connections follow these conventional descriptions:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

Block Diagram

Figure 1 on page 3 displays the block diagram of the eZ80F92 processor.

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

Pin No	Symbol	Function	Signal Direction	Description
25	ADDR20	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
26	ADDR21	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
27	ADDR22	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
28	ADDR23	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
29	$\overline{CS0}$	Chip Select 0	Output, Active Low	$\overline{CS0}$ Low indicates that an access is occurring in the defined $\overline{CS0}$ memory or I/O address space.
30	$\overline{CS1}$	Chip Select 1	Output, Active Low	$\overline{CS1}$ Low indicates that an access is occurring in the defined $\overline{CS1}$ memory or I/O address space.
31	$\overline{CS2}$	Chip Select 2	Output, Active Low	$\overline{CS2}$ Low indicates that an access is occurring in the defined $\overline{CS2}$ memory or I/O address space.
32	$\overline{CS3}$	Chip Select 3	Output, Active Low	$\overline{CS3}$ Low indicates that an access is occurring in the defined $\overline{CS3}$ memory or I/O address space.

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
16	ADDR13	I/O	O	N/A	Yes	No	No	No
17	ADDR14	I/O	O	N/A	Yes	No	No	No
18	V_{DD}							
19	V_{SS}							
20	ADDR15	I/O	O	N/A	Yes	No	No	No
21	ADDR16	I/O	O	N/A	Yes	No	No	No
22	ADDR17	I/O	O	N/A	Yes	No	No	No
23	ADDR18	I/O	O	N/A	Yes	No	No	No
24	ADDR19	I/O	O	N/A	Yes	No	No	No
25	ADDR20	I/O	O	N/A	Yes	No	No	No
26	ADDR21	I/O	O	N/A	Yes	No	No	No
27	ADDR22	I/O	O	N/A	Yes	No	No	No
28	ADDR23	I/O	O	N/A	Yes	No	No	No
29	$\overline{CS0}$	O	O	Low	No	No	No	No
30	$\overline{CS1}$	O	O	Low	No	No	No	No
31	$\overline{CS2}$	O	O	Low	No	No	No	No
32	$\overline{CS3}$	O	O	Low	No	No	No	No
33	V_{DD}							
34	V_{SS}							
35	DATA0	I/O	I	N/A	Yes	No	No	No
36	DATA1	I/O	I	N/A	Yes	No	No	No
37	DATA2	I/O	I	N/A	Yes	No	No	No
38	DATA3	I/O	I	N/A	Yes	No	No	No
39	DATA4	I/O	I	N/A	Yes	No	No	No
40	DATA5	I/O	I	N/A	Yes	No	No	No
41	DATA6	I/O	I	N/A	Yes	No	No	No
42	DATA7	I/O	I	N/A	Yes	No	No	No
43	V_{DD}							

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.								

Port x Data Direction Registers

In conjunction with the other GPIO Control Registers, the Port *x* Data Direction registers, listed in Table 8, control the operating modes of the GPIO port pins. See Table 6 on page 39 for more information.

Table 8. Port *x* Data Direction Registers; (PB_DDR = 009Bh, PC_DDR = 009Fh, PD_DDR = 00A3h)

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

Port x Alternate Register 1

In conjunction with the other GPIO Control Registers, the Port *x* Alternate Register 1, listed in Table 9, control the operating modes of the GPIO port pins. See Table 6 on page 39 for more information.

Table 9. Port *x* Alternate Registers 1; (PB_ALT1 = 009Ch, PC_ALT1 = 00A0h, PD_ALT1 = 00A4h)

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/W	R/W
Note: R/W = Read/Write.								

Port x Alternate Register 2

In conjunction with the other GPIO Control Registers, the Port *x* Alternate Register 2, listed in Table 10, control the operating modes of the GPIO port pins. See Table 6 on page 39 for more information.

Table 10. Port x Alternate Registers 2; (PB_ALT2 = 009Dh, PC_ALT2 = 00A1h, PD_ALT2 = 00A5h)

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/W	R/W
Note: R/W = Read/Write.								

If all of the foregoing conditions are met to generate an I/O Chip Select, then the following actions occur:

- The appropriate Chip Select—$\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, or $\overline{CS3}$—is asserted (driven Low)
- $\overline{IORQ}$ is asserted (driven Low)
- Depending upon the instruction, either $\overline{RD}$ or $\overline{WR}$ is asserted (driven Low)

WAIT States

For each of the Chip Selects, programmable WAIT states can be asserted to provide external devices with additional clock cycles to complete their Read or Write operations. The number of WAIT states for a particular Chip Select is controlled by the 3-bit field CS*x*_WAIT (CS*x*_CTL[7:5]). The WAIT states can be independently programmed to provide 0 to 7 WAIT states for each Chip Select. The WAIT states idle the CPU for the specified number of system clock cycles.

$\overline{WAIT}$ Input Signal

Similar to the programmable WAIT states, an external peripheral can drive the $\overline{WAIT}$ input pin to force the CPU to provide additional clock cycles to complete its Read or Write operation. Driving the $\overline{WAIT}$ pin Low stalls the CPU. The CPU resumes operation on the first rising edge of the internal system clock following deassertion of the $\overline{WAIT}$ pin.

Caution: If the $\overline{WAIT}$ pin is to be driven by an external device, the corresponding Chip Select for the device must be programmed to provide at least one WAIT state. Due to input sampling of the $\overline{WAIT}$ input pin (displayed in Figure 7), one programmable WAIT state is required to allow the external peripheral sufficient time to assert the $\overline{WAIT}$ pin. It is recommended that the corresponding Chip Select for the external device be programmed to provide the maximum number of WAIT states (seven).

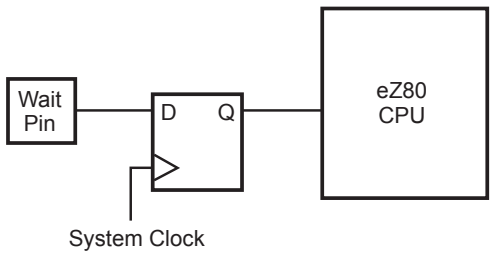

Figure 7.Wait Input Sampling Block Diagram

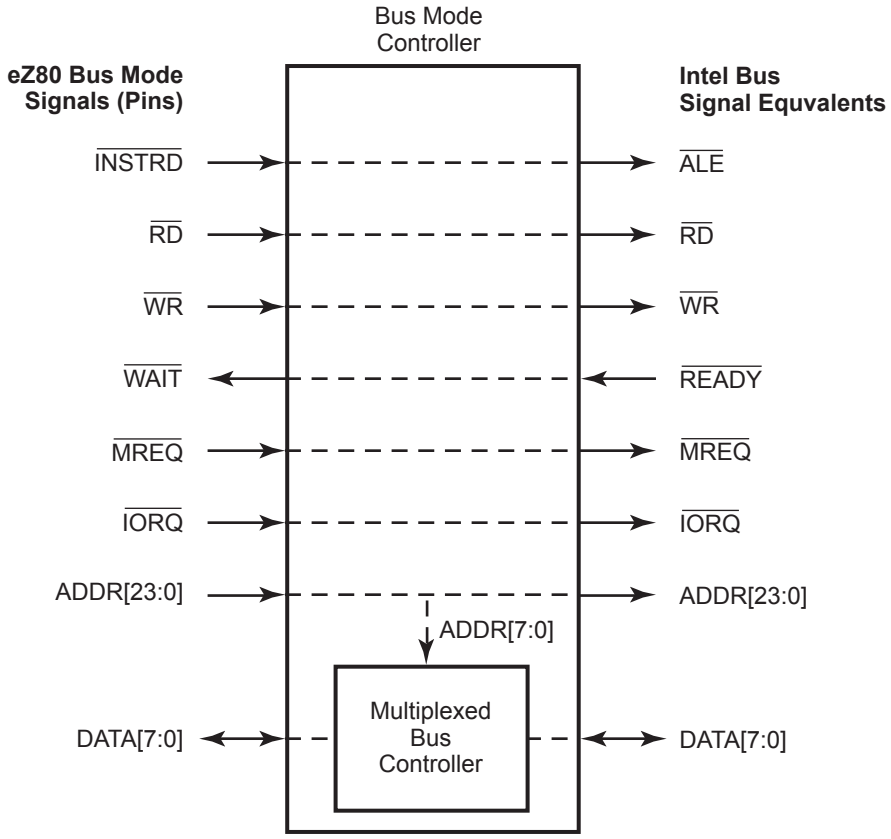

Figure 11. Intel™ Bus Mode Signal and Pin Mapping

Intel Bus Mode (Separate Address and Data Buses)

During Read operations with separate address and data buses, the Intel bus mode employs 4 states (T1, T2, T3, and T4) as listed in Table 16.

Table 16. Intel Bus Mode Read States (Separate Address and Data Buses

STATE T1	The Read cycle begins in State T1. The CPU drives the address onto the address bus and the associated Chip Select signal is asserted. The CPU drives the ALE signal High at the beginning of T1. During the middle of T1, the CPU drives ALE Low to facilitate the latching of the address.
STATE T2	During State T2, the CPU asserts the $\overline{RD}$ signal. Depending on the instruction, either the $\overline{MREQ}$ or $\overline{IORQ}$ signal is asserted.

9-bit data, the host processor programs the parity bit generator so that it marks the byte as either address (mark parity) or data (space parity).

UART Receiver

The receiver block controls the data reception from the RxD signal. The receiver block implements a receiver shift register, receiver line error condition monitoring logic and Receiver Data Ready logic. It also implements the parity checker.

The UARTx_RBR is a Read Only register of the module. The processor reads received data from this register. The condition of the UARTx_RBR register is monitored by the DR bit (bit 0 of the UARTx_LSR register). The DR bit is 1 when a data byte is received and transferred to the UARTx_RBR register from the receiver shift register. The DR bit is reset only when the processor reads all of the received data bytes. If the number of bits received is less than eight, the unused MSbs of the data byte Read are 0

For 9-bit data, the receiver checks incoming bytes for space parity. This check routine generates a line status interrupt when an address byte is received, because address bytes contain mark parity bits. The processor clears the interrupt, determines if the address matches its own, then configures the receiver to either accept the subsequent data bytes if the address matches, or ignore the data if it does not.

The receiver uses the clock from the BRG for receiving the data. This clock must be 16 times the appropriate baud rate. The receiver synchronizes the shift clock on the falling edge of the RxD input start bit. It then receives a complete byte according to the set parameters. The receiver also implements logic to detect framing errors, parity errors, overrun errors, and break signals.

UART Modem Control

The modem control logic provides two outputs and four inputs for handshaking with the modem. Any change in the modem status inputs, except $\overline{RI}$, is detected and an interrupt can be generated. For $\overline{RI}$, an interrupt is generated only when the trailing edge of the $\overline{RI}$ is detected. The module also provides LOOP mode for self-diagnostics.

UART Interrupts

There are six different sources of interrupts from the UART.

The six sources of interrupts are:

- Transmitter (two different interrupts)
- Receiver (three different interrupts)
- Modem status

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.

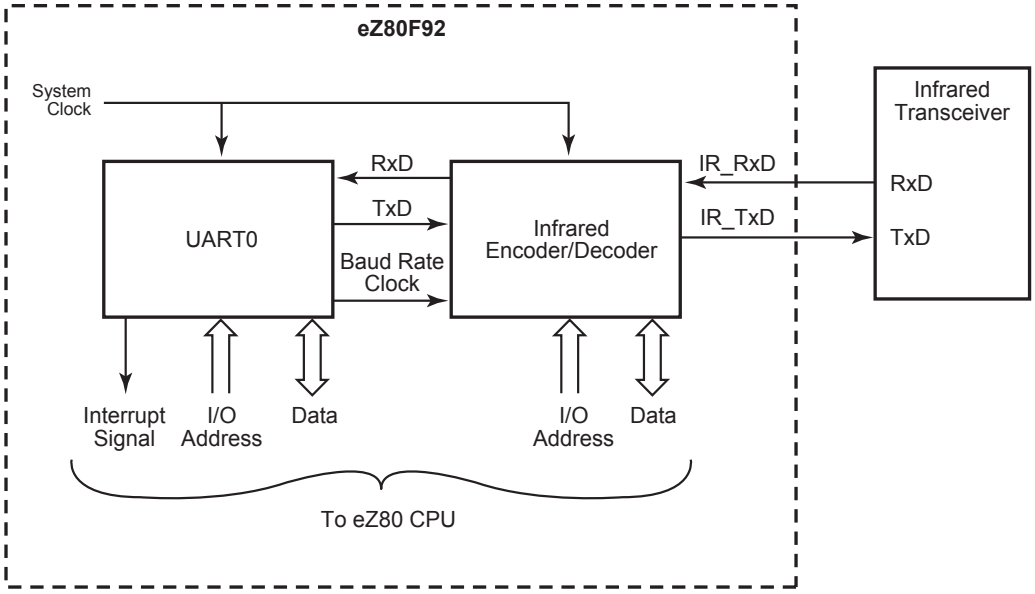

Figure 26. Infrared System Block Diagram

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

endec. See Universal Asynchronous Receiver/Transmitter on page 104 for more information about the UART and its Baud Rate Generator.

Transmit

The data to be transmitted via the IR transceiver is first sent to UART0. The UART transmit signal (TxD) and Baud Rate Clock are used by the IrDA endec to generate the modulation signal (IR_TxD) that drives the infrared transceiver. To enable transmit encoding, the IR_RxEN bit in the IR_CTL register must be set to 0.

Each UART bit is 16-clocks wide. If the data to be transmitted is a logical 1 (High), the IR_TxD signal remains Low (0) for the full 16-clock period. If the data to be transmitted is a logical 0, a 3-clock High (1) pulse is output following a 7-clock Low (0) period. Following the 3-clock High pulse, a 6-clock Low pulse completes the full 16-clock data period. Data transmission is displayed in Figure 27. During data transmission, the IR receive function should be disabled by clearing the IR_RxEN bit in the IR_CTL reg to 0. The SIR data format uses half-duplex communication; the UART does not transmit data while the receiver decoder is enabled.

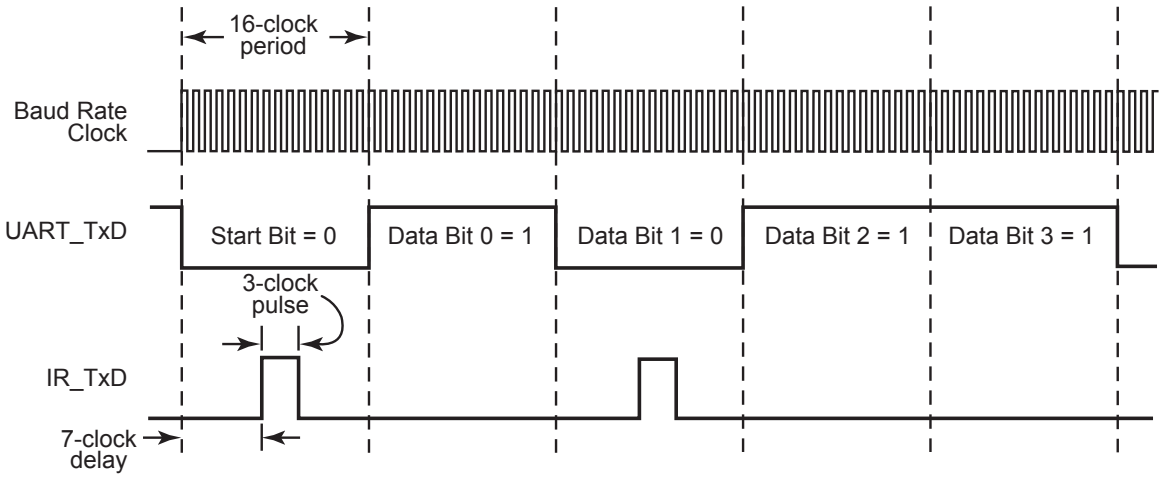

Figure 27.Infrared Data Transmission

Receive

Data is received from the IR transceiver via the IR_RxD signal and decoded by the IrDA endec. This decoded data is passed from the endec to UART0. To enable receiver decode, the IR_RxEN bit in the IR_CTL register must be set to 1. The SIR data format uses half-duplex communication; therefore, the UART should not transmit data during normal operation while the receiver decoder is enabled.

The UART baud rate clock is used by the IrDA endec to generate the demodulated signal (RxD) that drives the UART. Each UART bit period is sixteen baud-clocks wide. Each IR_RXD bit is encoded during a bit period such that a 0 is represented by a pulse and a 1 is represented by no pulse. The IrDA Physical Layer Specification describes a nominal pulse as being $^{3}/_{16}$ of a bit period wide. In this case, if the data to be received is a logical 0 (Low), a 3-clock-wide Low (0) pulse is received following a 7-clock High (1) period. Following the 3-clock Low pulse is a 6-clock High pulse to complete the full 16-clock data period. If the data to be received is a logical 1 (High), the IR_RxD signal is held High (1) for the full 16-clock period. Data reception is displayed in Figure 28.

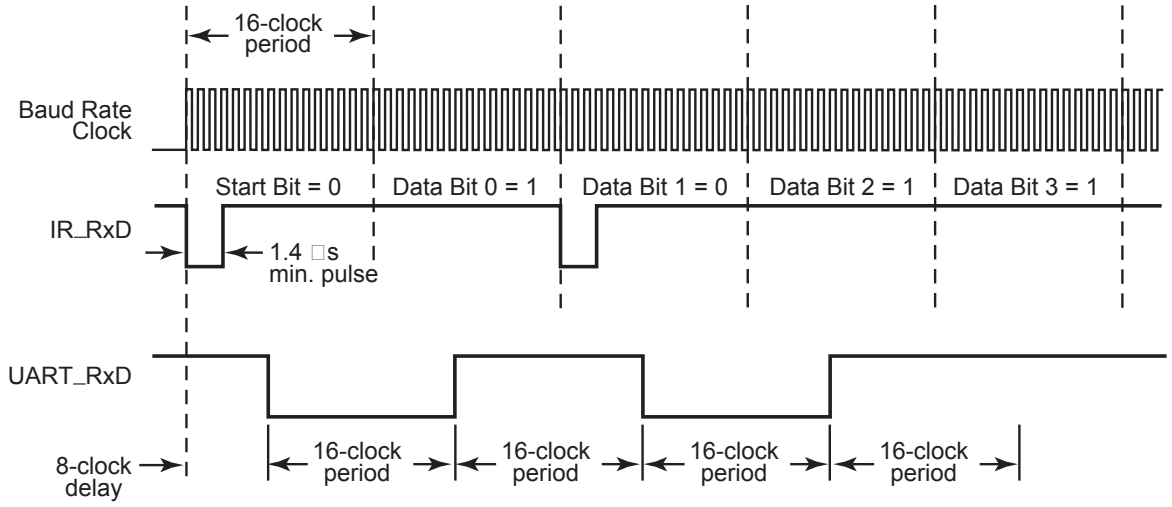

Figure 28.Infrared Data Reception

The IrDA Physical Layer Specification allows for a minimum signal width as well as the nominal signal width described above. By definition, the received pulse duration can be as small as 1.41 seconds for all baud rates up to 115.2 kbps. Table 67 outlines the minimum and maximum pulse durations for all baud rates supported by the eZ80® CPU. A receiver frequency divider based upon the system clock frequency measures this time limit and allows legal signals to pass to UART0.

Table 67. IrDA Physical Layer 1.4 Pulse Durations Specifications

Baud Rate	Minimum Pulse Width	Maximum Pulse Width
9600	1.41 s	22.13 s
19200	1.41 s	11.07 s
38400	1.41 s	5.96 s

Serial Peripheral Interface

The Serial Peripheral Interface (SPI) is a synchronous interface allowing several SPI-type devices to be interconnected. The SPI is a full-duplex, synchronous, character-oriented communication channel that employs a four-wire interface. The SPI block consists of a transmitter, receiver, baud rate generator, and control unit. During an SPI transfer, data is sent and received simultaneously by both the master and the slave SPI devices.

In a serial peripheral interface, separate signals are required for data and clock. The SPI may be configured as either a master or a slave. The connection of two SPI devices (one master and one slave) and the direction of data transfer is displayed in Figure 29 and Figure 30 on page 131.

Note: *When using the SPI module in the master mode, Port B2 must not be used as GPIO. If you do attempt to use it as GPIO, even though it is assigned as a standard Mode 1 output pin, outputting a logic low on the pin will cause the SPI to trigger a Mode Fault.*

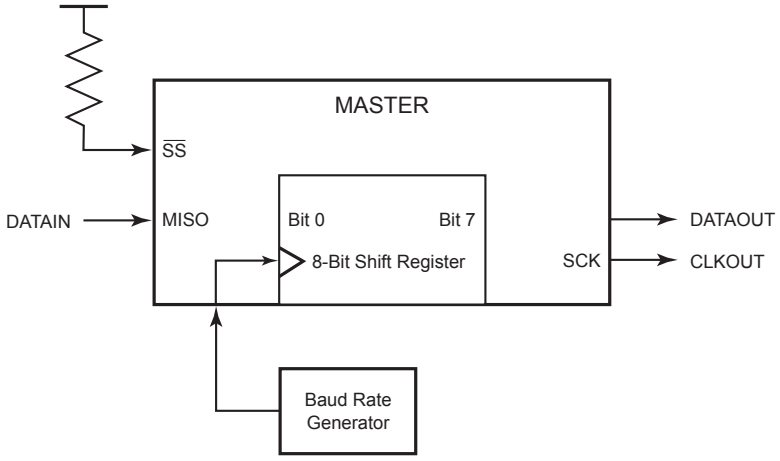

Figure 29.SPI Master Device

In other words, arbitration is not allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Clock Synchronization for Handshake

The Clock synchronizing mechanism can function as a handshake, enabling receivers to cope with fast data transfers, on either a byte or bit level. The byte level allows a device to receive a byte of data at a fast rate, but allows the device more time to store the received byte or to prepare another byte for transmission. Slaves hold the SCL line Low after reception and acknowledge the byte, forcing the master into a wait state until the slave is ready for the next byte transfer in a handshake procedure.

Operating Modes

Master Transmit

In MASTER TRANSMIT mode, the I^2C transmits a number of bytes to a slave receiver.

Enter MASTER TRANSMIT mode by setting the STA bit in the I2C_CTL register to 1. The I^2C then tests the I^2C bus and transmits a START condition when the bus is free. When a START condition is transmitted, the IFLG bit is 1 and the status code in the I2C_SR register is 08h. Before this interrupt is serviced, the I2C_DR register must be loaded with either a 7-bit slave address or the first part of a 10-bit slave address, with the lsb cleared to 0 to specify TRANSMIT mode. The IFLG bit should now be cleared to 0 to prompt the transfer to continue.

After the 7-bit slave address (or the first part of a 10-bit address) plus the Write bit are transmitted, the IFLG is set again. A number of status codes are possible in the I2C_SR register. See Table 78 on page 147.

I²C Registers

Addressing

The processor interface provides access to six 8-bit registers: four Read/Write registers, one Read Only register and two Write Only registers, as listed in Table 83.

Table 83. I²C Register Descriptions

Register	Description
I2C_SAR	Slave address register
I2C_XSAR	Extended slave address register
I2C_DR	Data byte register
I2C_CTL	Control register
I2C_SR	Status register (Read Only)
I2C_CCR	Clock Control register (Write Only)
I2C_SRR	Software reset register (Write Only)

Resetting the I²C Registers

Hardware Reset— When the I²C is reset by a hardware reset of the eZ80F92 device, the I2C_SAR, I2C_XSAR, I2C_DR and I2C_CTL registers are cleared to 00h; while the I2C_SR register is set to F8h.

Software Reset— Perform a software reset by writing any value to the I²C Software Reset Register (I2C_SRR). A software reset sets the I²C back to idle and the STP, STA, and IFLG bits of the I2C_CTL register to 0.

I²C Slave Address Register

The I2C_SAR register provides the 7-bit address of the I²C when in SLAVE mode and allows 10-bit addressing in conjunction with the I2C_XSAR register. I2C_SAR[7:1] = sla[6:0] is the 7-bit address of the I²C when in 7-bit SLAVE mode. When the I²C receives this address after a START condition, it enters SLAVE mode. I2C_SAR[7] corresponds to the first bit received from the I²C bus.

When the register receives an address starting with F7h to F0h (I2C_SAR[7:3] = 11110b), the I²C recognizes that a 10-bit slave addressing mode is selected. The I²C sends an ACK after receiving the I2C_SAR byte (the device does not generate an interrupt at this point). After the next byte of the address (I2C_XSAR) is received, the I²C generates an interrupt and goes into SLAVE mode. Then I2C_SAR[2:1] are used as the upper 2 bits for the 10-bit extended address. The full 10-bit address is supplied by {I2C_SAR[2:1], I2C_XSAR[7:0]}. See Table 84 on page 154.

Bus Clock Speed

The I^2C bus is defined for bus clock speeds up to 100 kbps (400 kbps in FAST mode).

To ensure correct detection of START and STOP conditions on the bus, the I^2C must sample the I^2C bus at least ten times faster than the bus clock speed of the fastest master on the bus. The sampling frequency should therefore be at least 1 MHz (4 MHz in FAST mode) to guarantee correct operation with other bus masters.

The I^2C sampling frequency is determined by the frequency of the CPU system clock and the value in the I2C_CCR bits 2 to 0. The bus clock speed generated by the I^2C in MASTER mode is determined by the frequency of the input clock and the values in I2C_CCR[2:0] and I2C_CCR[6:3].

I^2C Software Reset Register

The I2C_SRR register is a Write Only register. Writing any value to this register performs a software reset of the I^2C module. See Table 91.

Table 91. I^2C Software Reset Register(I2C_SRR = 00CDh)

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: W = Write Only.								

Bit Position	Value	Description
[7:0] SRR	00h–FFh	Writing any value to this register performs a software reset of the I^2C module.

Table 99. ZDI Read/Write Control Register Functions(ZDI_RW_CTL = 16h in the ZDI Register Write Only Address Space (Continued))

Hex Value	Command	Hex Value	Command
0A	Exchange CPU register sets AF ← AF' BC ← BC' DE ← DE' HL ← HL'	8A	Reserved
0B	Read memory from current PC value, increment PC	8B	Write memory from current PC value, increment PC

The eZ80® CPU's alternate register set (A', F', B', C', D', E', HL') cannot be read directly. The ZDI programmer must execute the exchange instruction (EXX) to gain access to the alternate eZ80 CPU register set.

eZ80® Product ID Low and High Byte Registers

The eZ80 Product ID Low and High Byte registers combine to provide a means for an external device to determine the particular eZ80Acclaim!® product being addressed. See Table 103 and Table 104.

Table 103. eZ80 Product ID Low Byte Register(ZDI_ID_L = 00h in the ZDI Register Read Only Address Space)

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

Bit Position	Value	Description
[7:0] ZDI_ID_L	07h	{ZDI_ID_H, ZDI_ID_L} = {00h, 07h} indicates the eZ80F92 product.

Table 104. eZ80 Product ID High Byte Register(ZDI_ID_H = 01h 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_ID_H	00h	{ZDI_ID_H, ZDI_ID_L} = {00h, 07h} indicates the eZ80F92 product.

be accessed via the clock (TCK) and data (TDI) pins. See the Zilog Debug Interface section on page 162 for more information about ZDI.

OCI Interface

There are five dedicated pins on the eZ80F92 device for the OCI interface. Four pins—TCK, TMS, TDI, and TDO—are required for IEEE Standard 1149.1-compliant JTAG ports. The TRIGOUT pin provides additional testability features. These five OCI pins are listed in Table 110.

Table 110. OCI Pins

Symbol	Name	Type	Description
TCK	Clock	Input	Asynchronous to the primary CPU system clock. The TCK period must be at least twice the system clock period. During RESET, this pin is sampled to select either OCI or ZDI DEBUG modes. If Low during RESET, the OCI is enabled. If High during RESET, the OCI is powered down and ZDI DEBUG mode is enabled. When ZDI DEBUG mode is active, this pin is the ZDI clock. On-chip pull-up ensures a default value of 1 (High).
TMS	Test Mode Select	Input	This serial test mode input controls JTAG mode selection. On-chip pull-up ensures a default value of 1 (High). The TMS signal is sampled on the rising edge of the TCK signal.
TDI	Data In	Input (OCI enabled)	Serial test data input. On-chip pull-up ensures a default value of 1 (High). This pin is input-only when the OCI is enabled. The input data is sampled on the rising edge of the TCK signal.
		I/O (OCI disabled)	When the OCI is disabled, this pin functions as the ZDA (ZDI Data) I/O pin.
TDO	Data Out	Output	The output data changes on the falling edge of the TCK signal.
TRIGOUT	Trigger Output	Output	Generates an active High trigger pulse when valid OCI trigger events occur. Output is tristate when no data is driven out.

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