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	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/ez80f92az020sg

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

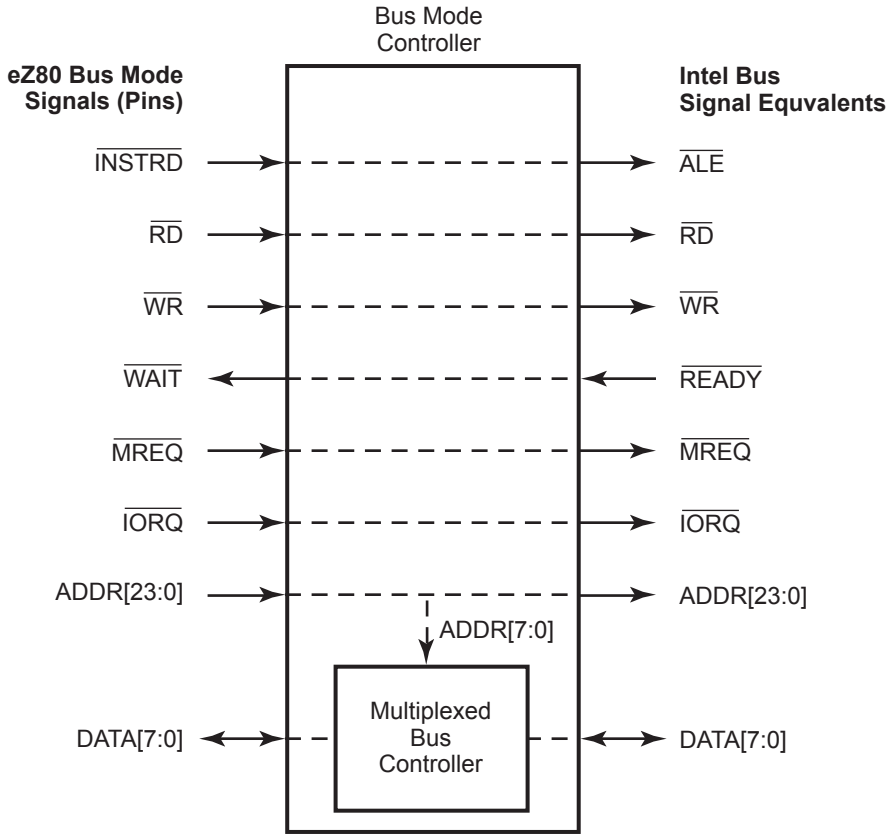

Figure 11. Intel™ Bus Mode Signal and Pin Mapping

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.

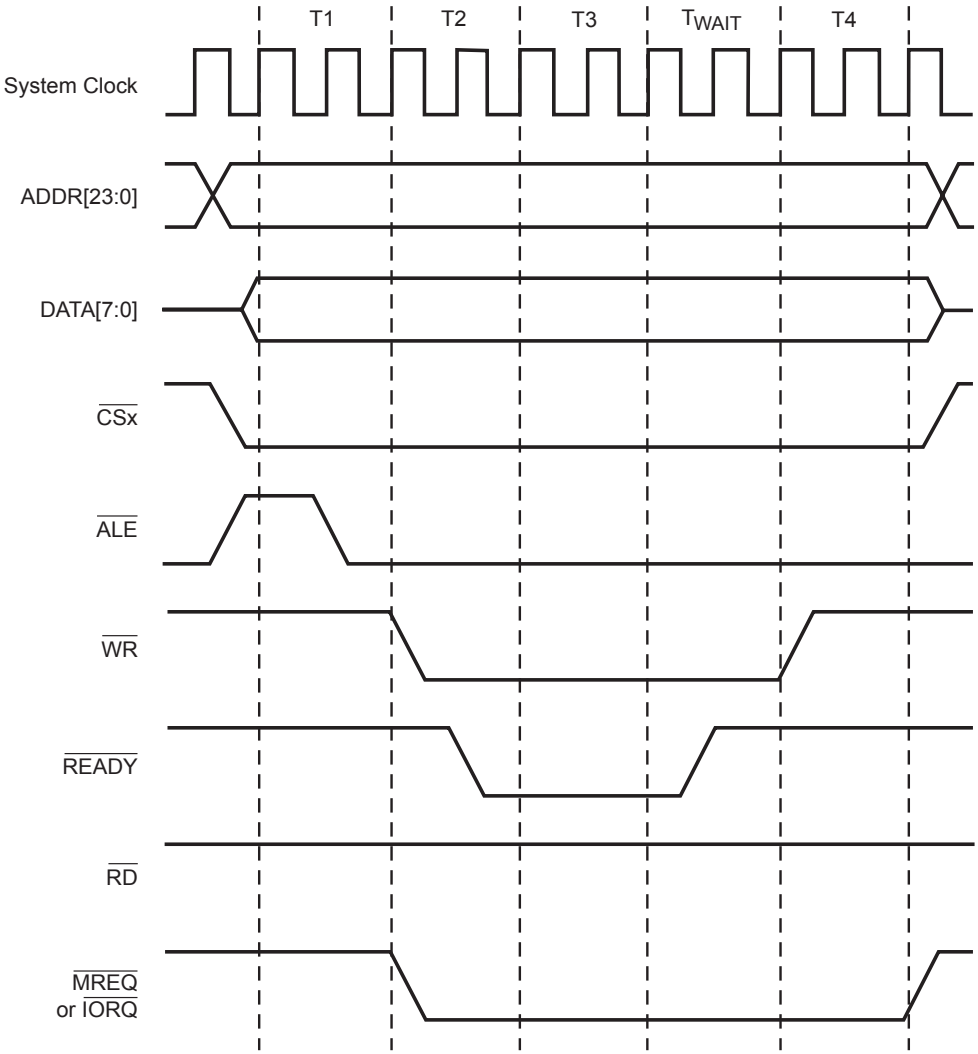

Figure 13. Example: Intel™ Bus Mode Write Timing—Separate Address and Data Buses

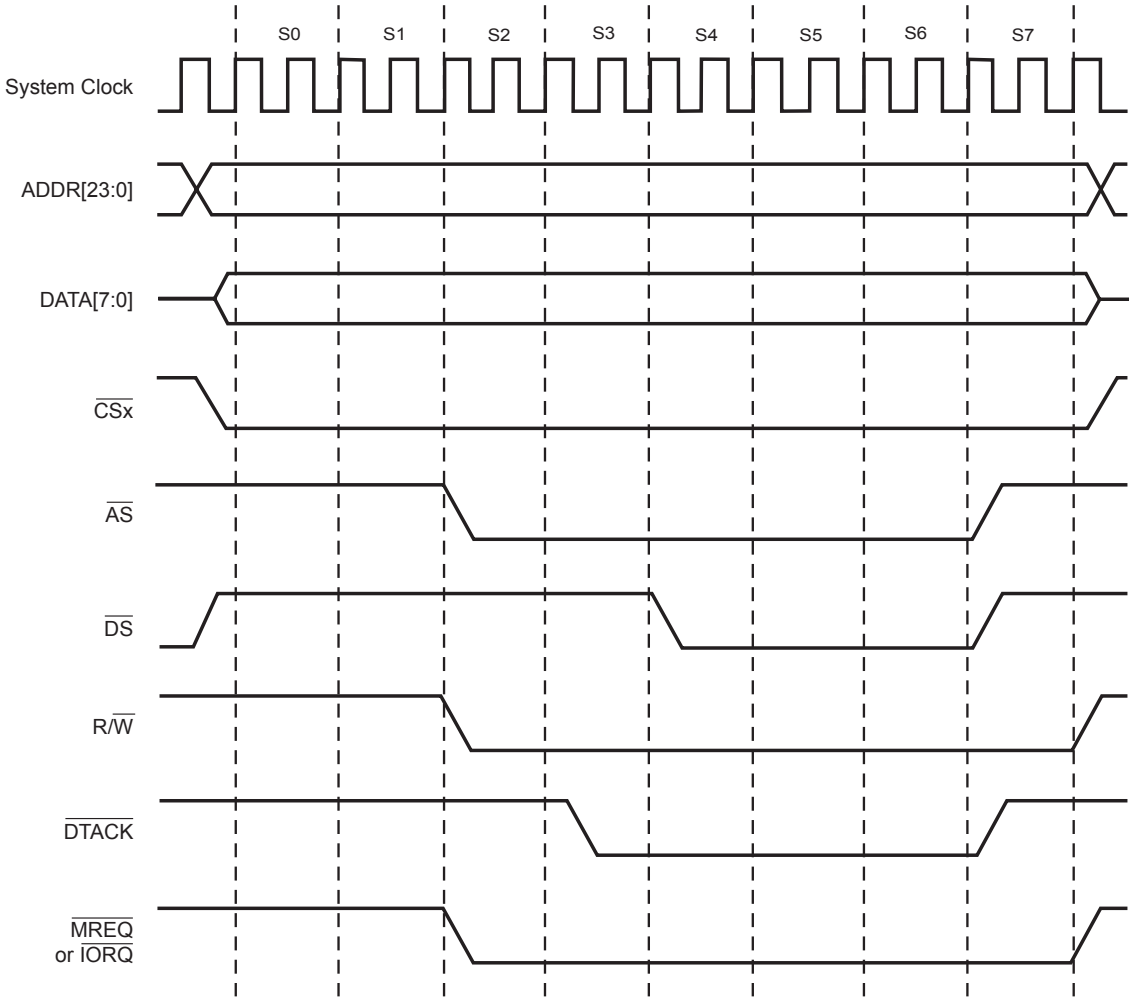

Figure 18. Example: Motorola Bus Mode Write Timing

Switching Between Bus Modes

Each time the bus mode controller must switch from one bus mode to another, there is a one-cycle CPU system clock delay. An extra clock cycle is not required for repeated access in any of the bus modes; nor is it required when the eZ80F92 device switches to eZ80® bus mode. The extra clock cycles are not shown in the timing examples. Due to the asynchronous nature of these bus protocols, the extra delay does not impact peripheral communication.

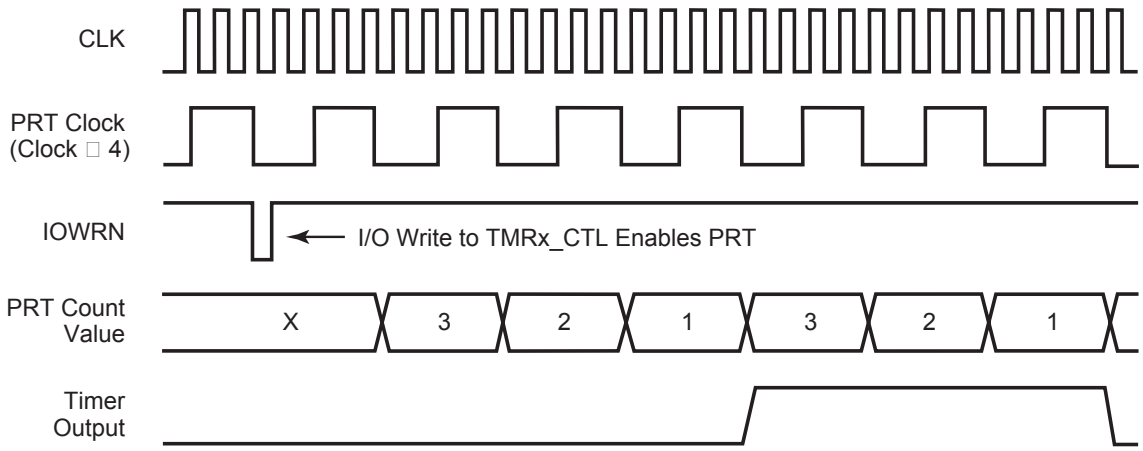

Figure 23. PRT Timer Output Operation Example

Table 31. PRT Timer Out 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 Reload Value	{TMRx_RR_H, TMRx_RR_L}	0003h

Programmable Reload Timer Registers

Each programmable reload timer is controlled using five 8-bit registers. These registers are the Timer Control register, Timer Reload Low Byte register, Timer Reload High Byte register, Timer Data Low Byte register, and Timer Data High Byte register.

The Timer Control register can be read or written to. The timer reload registers are Write Only and are located at the same I/O address as the timer data registers, which are Read Only.

Timer Control Register

The Timer Control register, listed in Table 32 on page 82, is used to control operation of the timer, including enabling the timer, selecting the clock divider, enabling the interrupt, selecting between CONTINUOUS and SINGLE PASS modes, and enabling the auto-reload feature.

Real-Time Clock Day-of-the-Month Register

This register contains the current day-of-the-month count. The RTC_DOM register begins counting at 01h. See Table 42.

Table 42. Real-Time Clock Day-of-the-Month Register; (RTC_DOM = 00E4h)

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 = Unchanged by RESET; R/W* = Read Only if RTC locked, Read/Write if RTC unlocked.								

Binary-Coded-Decimal Operation (BCD_EN = 1)

Bit Position	Value	Description
[7:4] TENS_DOM	0–3	The tens digit of the current day-of-the-month count.
[3:0] DOM	0–9	The ones digit of the current day-of-the-month count.

Binary Operation (BCD_EN = 0)

Bit Position	Value	Description
[7:0] DOM	01h–1Fh	The current day-of-the-month count.

Real-Time Clock Century Register

This register contains the current century count. See Table 45.

Table 45. Real-Time Clock Century Register; (RTC_CEN = 00E7h)

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 = Unchanged by RESET; R/W* = Read Only if RTC locked, Read/Write if RTC unlocked.								

Binary-Coded-Decimal Operation (BCD_EN = 1)		
Bit Position	**Value**	**Description**
[7:4] TENS_CEN	0–9	The tens digit of the current century count.
[3:0] CEN	0–9	The ones digit of the current century count.

Binary Operation (BCD_EN = 0)		
Bit Position	**Value**	**Description**
[7:0] CEN	00h–63h	The current century count.

Real-Time Clock Alarm Minutes Register

This register contains the alarm minutes value. See Table 47.

Table 47. Real-Time Clock Alarm Minutes Register; (RTC_AMIN = 00E9h)

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 = Unchanged by RESET; R/W = Read/Write.								

Binary-Coded-Decimal Operation (BCD_EN = 1)

Bit Position	Value	Description
[7:4] ATEN_MIN	0–5	The tens digit of the alarm minutes value.
[3:0] AMIN	0–9	The ones digit of the alarm minutes value.

Binary Operation (BCD_EN = 0)

Bit Position	Value	Description
[7:0] AMIN	00h–3Bh	The alarm minutes value.

UART Functional Description

The UART function implements:

- The transmitter and associated control logic
- The receiver and associated control logic
- The modem interface and associated logic

UART Functions

The UART function implements:

- The transmitter and associated control logic
- The receiver and associated control logic
- The modem interface and associated logic

UART Transmitter

The transmitter block controls the data transmitted on the TxD output. It implements the FIFO, accessed through the UARTx_THR register, the transmit shift register, the parity generator, and control logic for the transmitter to control parameters for the asynchronous communication protocol.

The UARTx_THR is a Write Only register. The processor writes the data byte to be transmitted into this register. In the FIFO mode, up to 16 data bytes can be written via the UARTx_THR register. The data byte from the FIFO is transferred to the transmit shift register at the appropriate time and transmitted out on TxD output. After SYNC_RESET, the UARTx_THR register is empty. Therefore, the Transmit Holding Register Empty (THRE) bit (bit 5 of the UARTx_LSR register) is 1 and an interrupt is sent to the processor (if interrupts are enabled). The processor can reset this interrupt by loading data into the UARTx_THR register, which clears the transmitter interrupt.

The transmit shift register places the byte to be transmitted on the TxD signal serially. The lsb of the byte to be transmitted is shifted out first and the msb is shifted out last. The control logic within the block adds the asynchronous communication protocol bits to the data byte being transmitted. The transmitter block obtains the parameters for the protocol from the bits programmed via the UARTx_LCTL register. When enabled, an interrupt is generated after the most recent protocol bit is transmitted, which the processor may reset by loading data into the UARTx_THR register. The TxD output is set to 1 if the transmitter is idle (it does not contain any data to be transmitted).

The transmitter operates with the Baud Rate Generator (BRG) clock. The data bits are placed on the TxD output one time every 16 BRG clock cycles. The transmitter block also implements a parity generator that attaches the parity bit to the byte, if programmed. For

Bit Position	Value	Description
[7:0] UART_BRG_L	00h–FFh	These bits represent the Low byte of the 16-bit Baud Rate Generator divider value. The complete BRG divisor value is returned by {UART_BRG_H, UART_BRG_L}.

Table 53. UART Baud Rate Generator Register—High Bytes(UART0_BRG_H = 00C1h, UART1_BRG_H = 00D1h)

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 = Read only; R/W = Read/Write.								

Bit Position	Value	Description
[7:0] UART_BRG_H	00h–FFh	These bits represent the High byte of the 16-bit Baud Rate Generator divider value. The complete BRG divisor value is returned by {UART_BRG_H, UART_BRG_L}.

UART Registers

After a RESET, all UART registers are set to their default values. Any writes to unused registers or register bits are ignored and reads return a value of 0. For compatibility with future revisions, unused bits within a register should always be written with a value of 0. Read/Write attributes, reset conditions, and bit descriptions of all of the UART registers are provided in this section.

UART Transmit Holding Register

If less than eight bits are programmed for transmission, the lower bits of the byte written to this register are selected for transmission. The transmit FIFO is mapped at this address. The user can write up to 16 bytes for transmission at one time to this address if the FIFO is enabled by the application. If the FIFO is disabled, this buffer is only one byte deep.

These registers share the same address space as the UARTx_RBR and UARTx_BRG_L registers. See Table 54 on page 112.

Bit Position	Value	Description
6 SB	0	Do not send a BREAK signal.
	1	Send Break UART sends continuous zeroes on the transmit output from the next bit boundary. The transmit data in the transmit shift register is ignored. After forcing this bit High, the TxD output is 0 only after the bit boundary is reached. Just before forcing TxD to 0, the transmit FIFO is cleared. Any new data written to the transmit FIFO during a break should be written only after the THRE bit of UARTx_LSR register goes High. This new data is transmitted after the UART recovers from the break. After the break is removed, the UART recovers from the break for the next BRG edge.
5 FPE	0	Do not force a parity error.
	1	Force a parity error. When this bit and the party enable bit (PEN) are both 1, an incorrect parity bit is transmitted with the data byte.
4 EPS	0	Use odd parity for transmit and receive. The total number of 1 bits in the transmit data plus parity bit is odd. Use as a SPACE bit in MULTIDROP mode. See Table 62 on page 118 for parity select definitions.*
	1	Use even parity for transmit and receive. The total number of 1 bits in the transmit data plus parity bit is even. Use as a MARK bit in MULTIDROP mode. See Table 62 on page 118 for parity select definitions.
3 PEN	0	Parity bit transmit and receive is disabled.
	1	Parity bit transmit and receive is enabled. For transmit, a parity bit is generated and transmitted with every data character. For receive, the parity is checked for every incoming data character. In MULTIDROP mode, receive parity is checked for space parity.
[2:0] CHAR	000–111	UART Character Parameter Selection—see Table 61 on page 118 for a description of the values.

Note: *Receive Parity is set to SPACE in MULTIDROP mode.

Upon RESET, the 16-bit BRG divisor value resets to 0002h. When the SPI is operating as a Master, the BRG divisor value must be set to a value of 0003h or greater. When the SPI is operating as a Slave, the BRG divisor value must be set to a value of 0004h or greater. A software Write to either the Low- or High-byte registers for the BRG Divisor Latch causes both the Low and High bytes to load into the BRG counter, and causes the count to restart.

Data Transfer Procedure with SPI Configured as the Master

1. Load the SPI Baud Rate Generator Registers, SPI_BRG_H and SPI_BRG_L.
2. External device must deassert the $\overline{SS}$ pin if currently asserted.
3. Load the SPI Control Register, SPI_CTL.
4. Assert the $\overline{ENABLE}$ pin of the slave device using a GPIO pin.
5. Load the SPI Transmit Shift Register, SPI_TSR.
6. When the SPI data transfer is complete, deassert the $\overline{ENABLE}$ pin of the slave device.

Data Transfer Procedure with SPI Configured as a Slave

1. Load the SPI Baud Rate Generator Registers, SPI_BRG_H and SPI_BRG_L.
2. Load the SPI Transmit Shift Register, SPI_TSR. This load cannot occur while the SPI slave is currently receiving data.
3. Wait for the external SPI Master device to initiate the data transfer by asserting $\overline{SS}$.

SPI Registers

There are six registers in the Serial Peripheral Interface which provide control, status, and data storage functions. The SPI registers are described in the following paragraphs.

SPI Baud Rate Generator Registers—Low Byte and High Byte

These registers hold the Low and High bytes of the 16 bit divisor count loaded by the processor for baud rate generation. The 16 bit clock divisor value is returned by {SPI_BRG_H, SPI_BRG_L}. Upon RESET, the 16 bit BRG divisor value resets to 0002h. When configured as a Master, the 16 bit divisor value must be between 0003h and FFFFh, inclusive. When configured as a Slave, the 16 bit divisor value must be between 0004h and FFFFh, inclusive.

Transferring Data

Byte Format

Every character transferred on the SDA line must be a single 8-bit byte. The number of bytes that can be transmitted per transfer is unrestricted. Each byte must be followed by an Acknowledge (ACK)[1]. Data is transferred with the most-significant bit (msb) first. See Figure 34. A receiver can hold the SCL line Low to force the transmitter into a wait state. Data transfer then continues when the receiver is ready for another byte of data and releases SCL.

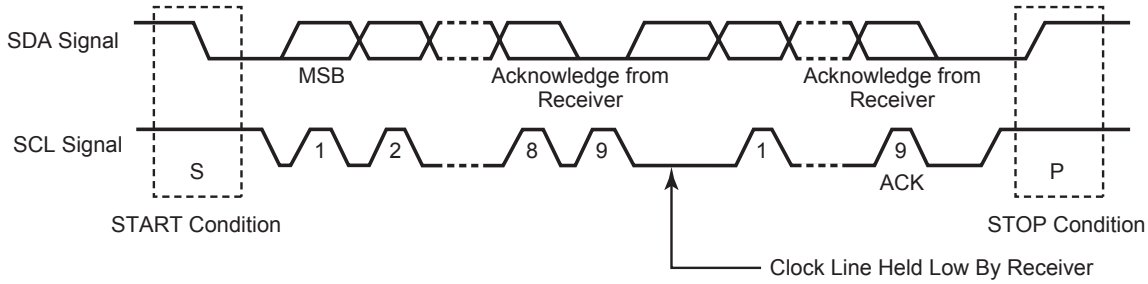

Figure 34. I²C Frame Structure

Acknowledge

Data transfer with an ACK function is obligatory. The ACK-related clock pulse is generated by the master. The transmitter releases the SDA line (High) during the ACK clock pulse. The receiver must pull down the SDA line during the ACK clock pulse so that it remains stable Low during the High period of this clock pulse. See Figure 35 on page 144.

A receiver that is addressed is obliged to generate an ACK after each byte is received. When a slave-receiver does not acknowledge the slave address (for example, unable to receive because it's performing some real-time function), the data line must be left High by the slave. The master then generates a STOP condition to abort the transfer.

If a slave-receiver acknowledges the slave address, but cannot receive any more data bytes, the master must abort the transfer. The abort is indicated by the slave generating the Not Acknowledge (NACK) on the first byte to follow. The slave leaves the data line High and the master generates the STOP condition.

If a master-receiver is involved in a transfer, it must signal the end of data to the slave-transmitter by not generating an ACK on the final byte that is clocked out of the slave.

1. ACK is defined as a general Acknowledge bit. By contrast, the I²C Acknowledge bit is represented as AAK, bit 2 of the I²C Control Register, which identifies which ACK signal to transmit. See Table 87 on page 157.

OCI Information Requests

For additional information regarding On-Chip Instrumentation, or to order OCI debug tools, contact:

First Silicon Solutions, Inc.
5440 SW Westgate Drive, Suite 240
Portland, OR 97221
Phone: (503) 292-6730
Fax: (503) 292-5840
www.fs2.com

Random Access Memory

The eZ80F92 features 8KB (8192 bytes) single-port data Random Access Memory (RAM) for general-purpose use. The eZ80F93 features 4 KB (4096 bytes) general-purpose RAM. RAM can be enabled or disabled, and it can be relocated to the top of any 64 KB page in memory. Data is passed to and from RAM via the 8-bit data bus. On-chip RAM operates with zero WAIT states.

For the eZ80F92, RAM occupies memory addresses in the range {RAM_ADDR_U[7:0], E000h} to {RAM_ADDR_U[7:0], FFFFh}. Following a RESET, RAM is enabled with RAM_ADDR_U set to FFh. Figure 46 displays a memory map of on-chip RAM. In this example, the RAM Address Upper Byte register, RAM_ADDR_U, is set to 7Ah. Figure 46 is not drawn to scale, as RAM occupies only a very small fraction of the available 16 MB address space.

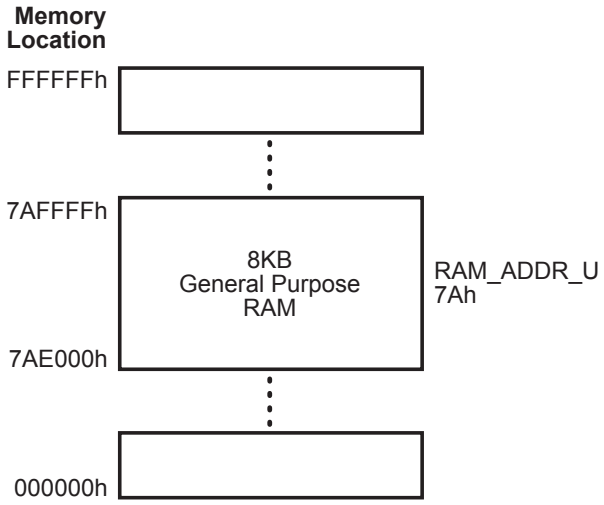

Figure 46.eZ80F92 On-Chip RAM Memory Addressing Example

For the eZ80F93 device, RAM occupies memory addresses in the range {RAM_ADDR_U[7:0], F000h} to {RAM_ADDR_U[7:0], F000h}. Following a RESET, RAM is enabled with RAM_ADDR_U set to FFh. Figure 47 on page 191 displays a memory map of on-chip RAM. In this example, the RAM Address Upper Byte register, RAM_ADDR_U, is set to 7Ah. Figure 46 is not drawn to scale, as RAM occupies only a very small fraction of the available 16 MB address space.

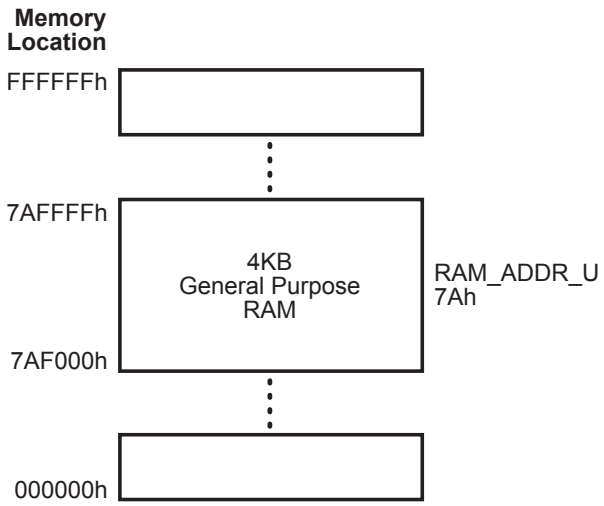

Figure 47.eZ80F93 On-Chip RAM Memory Addressing Example

When enabled, on-chip RAM assumes priority over on-chip Flash Memory and any Memory Chip Selects that can also be enabled in the same address space. If an address is generated in a range that is covered by both the RAM address space and a particular Memory Chip Select address space, the Memory Chip Select is not activated. On-chip RAM is not accessible by external devices during Bus Acknowledge cycles.

External Bus Acknowledge Timing

Table 153 lists information about the bus acknowledge timing. Once the external bus master detects BUSACK asserted and drives IORQN, MREQN, A[23:0] there is an asynchronous prop delay to the CS[3:0] outputs being valid.

Table 153. Bus Acknowledge Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
T_1	Clock Rise to $\overline{\text{BUSACK}}$ Assertion Delay	2.0	14.0
T_2	Clock Rise to $\overline{\text{BUSACK}}$ Deassertion Delay	2.0	14.0
T_3	IORQN, MREQN, A[23:0] input to CS[3:0] output prop delay	—	10.0

External System Clock Driver (PHI) Timing

Table 154 lists timing information for the PHI pin. The PHI pin allows external peripherals to synchronize with the internal system clock driver on the eZ80F92 device.

Table 154. PHI System Clock Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
T_1	Clock (XIN) Rise to PHI Rise	—	6.0
T_2	Clock (XIN) Fall to PHI Fall	—	6.0

Zilog Debug Interface Timing

Figure 65 and Table 155 display timing information for TCK, TDI, TDO, TMS pins.

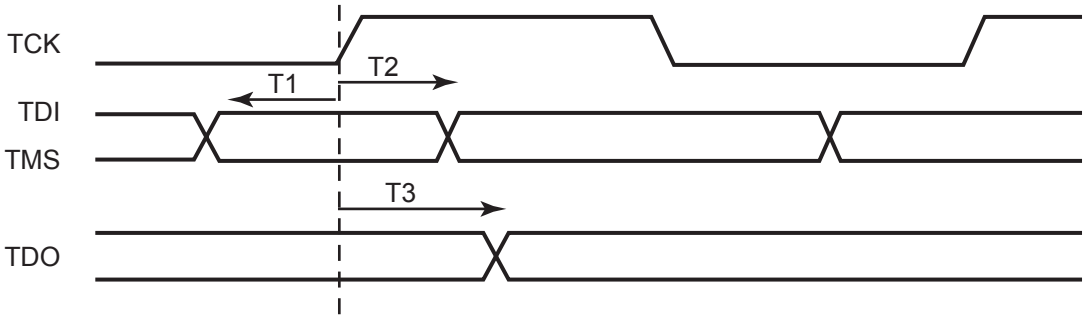

Figure 65.ZDI Timing

Table 155. ZDI Timing Specifications

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
T_{TCK}	TCK Period	2 x T_{XIN}	
T_1	TDI/TMS setup to TCK Rise	4	
T_2	TDI/TMS hold after TCK Rise Fall	4	
T_3	TCK Rise to TDO change		10

N

O

P