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	-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/ez80f93az020eg

Architectural Overview

Zilog's eZ80F92 device is a high-speed single-cycle instruction-fetch microcontroller with a maximum clock speed of 20 MHz. It is the first member of Zilog's new eZ80Acclaim!® product family, which offers on-chip Flash program memory.
The eZ80F92 device can operate in Z80® compatible addressing mode (64 KB) or full 24-bit addressing mode (16 MB). The rich peripheral set of the eZ80F92 device makes it suitable for a variety of applications including industrial control, embedded communication, and point-of-sale terminals.

Note: *Additionally, Zilog offers the eZ80F93 device, which features scaled-down memory options. For clarity, this document refers to both devices collectively as the eZ80F92 device, unless otherwise specified.*

Features

The features of eZ80F92/eZ80F93 device include:

- Single-cycle instruction fetch, high-performance, pipelined eZ80® CPU core[1]
- eZ80F92 contains 128 KB Flash memory and 8 KB SRAM
- eZ80F93 contains 64 KB Flash memory and 4 KB SRAM
- Low power features including SLEEP mode, HALT mode, and selective peripheral power-down control
- Two UARTs with independent baud rate generators
- SPI with independent clock rate generator
- I^2C with independent clock rate generator
- IrDA-compliant infrared encoder/decoder
- New DMA-like CPU instructions for efficient block data transfer
- Glueless external peripheral interface with 4 Chip Selects, individual Wait State generators, and an external $\overline{WAIT}$ input pin—supports Z80-, Intel-, and Motorola-style buses
- Fixed-priority vectored interrupts (both internal and external) and interrupt controller
- Real-Time Clock with on-chip 32 kHz oscillator, selectable 50/60 Hz input, and separate V_{DD} pin for battery backup
- Six 16-bit Counter/Timers with clock dividers and direct input/output drive

1. For simplicity, the term *eZ80® CPU* is referred to as *CPU* for the bulk of this document.

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

Pin No	Symbol	Function	Signal Direction	Description
54	$\overline{\text{BUSACK}}$	Bus Acknowledge	Output, Active Low	The CPU responds to a Low on $\overline{\text{BUSREQ}}$, by tristating the address, data, and control signals, and by driving the $\overline{\text{BUSACK}}$ line Low. During bus acknowledge cycles ADDR[23:0], $\overline{\text{IORQ}}$, and $\overline{\text{MREQ}}$ are inputs.
55	$\overline{\text{HALT\_SLP}}$	HALT and SLEEP Indicator	Output, Active Low	A Low on this pin indicates that the CPU has entered either HALT or SLEEP mode because of execution of either a HALT or SLP instruction.
56	V_{DD}	Power Supply		Power Supply.
57	V_{SS}	Ground		Ground.
58	RTC_X_{IN}	Real-Time Clock Crystal Input	Input	This pin is the input to the low-power 32 kHz crystal oscillator for the Real-Time Clock.
59	RTC_X_{OUT}	Real-Time Clock Crystal Output	Bidirectional	This pin is the output from the low-power 32 kHz crystal oscillator for the Real-Time Clock. This pin is an input when the RTC is configured to operate from 50/60 Hz input clock signals and the 32 kHz crystal oscillator is disabled.
60	RTC_V_{DD}	Real-Time Clock Power Supply		Power supply for the Real-Time Clock and associated 32 kHz oscillator. Isolated from the power supply to the remainder of the chip. A battery can be connected to this pin to supply constant power to the Real-Time Clock and 32 kHz oscillator.
61	V_{SS}	Ground		Ground.
62	TMS	JTAG Test Mode Select	Input	JTAG Mode Select Input.
63	TCK	JTAG Test Clock	Input	JTAG and ZDI clock input.
64	TRIGOUT	JTAG Test Trigger Output	Output	Active High trigger event indicator.
65	TDI	JTAG Test Data In	Bidirectional	JTAG data input pin. Functions as ZDI data I/O pin when JTAG is disabled.
66	TDO	JTAG Test Data Out	Output	JTAG data output pin.

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.

Reset

Reset Operation

The Reset controller within the eZ80F92 device provides a consistent reset function for all types of resets that can affect the system. A system reset, referred in this document as RESET, returns the eZ80F92 device to a defined state. All internal registers affected by RESET return to their default conditions. RESET configures the GPIO port pins as inputs and clears the CPU's Program Counter to 000000h. Program code execution ceases during RESET.

The events that can cause a RESET are:

- Power-On Reset (POR)
- Low-Voltage Brownout (VBO)
- External $\overline{\text{RESET}}$ pin assertion
- Watchdog Timer (WDT) time-out when configured to generate a RESET
- Real-Time Clock alarm with the CPU in low-power SLEEP mode
- Execution of a debug reset command

During a RESET, an internal RESET mode timer holds the system in RESET mode for 257 system clock (SCLK) cycles. The RESET mode timer begins incrementing on the next rising edge of SCLK following deactivation of all RESET events.

Note: *You must determine if 257 SCLK cycles provides sufficient time for the primary crystal oscillator to stabilize.*

Power-On Reset

A Power-On Reset (POR) occurs each time the supply voltage to the part rises from below the Voltage Brownout threshold to above the POR voltage threshold (V_{POR}). The internal bandgap-referenced voltage detector sends a continuous RESET signal to the Reset controller until the supply voltage (V_{CC}) exceeds the POR voltage threshold. After V_{CC} rises above V_{POR}, an on-chip analog delay element briefly maintains the RESET signal to the Reset controller (T_{ANA}). After this analog delay, the eZ80F92 device is in RESET mode until the RESET mode timer expires. POR operation is displayed in Figure 3 on page 33. The signals in this figure are not drawn to scale and are for displaying purposes only.

Bus Mode Controller

The bus mode controller allows the address and data bus timing and signal formats of the eZ80F92 device to be configured to connect seamlessly with external eZ80®, Z80®-, Intel-, or Motorola-compatible devices. Bus modes for each of the chip selects can be configured independently using the Chip Select Bus Mode Control Registers. The number of CPU system clock cycles per bus mode state is also independently programmable. For Intel™ bus mode, multiplexed address and data can be selected in which the lower byte of the address and the data byte both use the data bus, DATA[7:0]. Each of the bus modes is explained in more detail in the following sections.

eZ80 Bus Mode

Chip selects configured for eZ80 bus mode do not modify the bus signals from the CPU. The timing diagrams for external Memory and I/O Read and Write operations are shown in the AC Characteristics section on page 229. The default mode for each chip select is eZ80 mode.

Z80 Bus Mode

Chip selects configured for Z80 mode modify the CPU bus signals to match the Z80 microprocessor address and data bus interface signal format and timing. During read operations, the Z80 bus mode employs three states (T1, T2, and T3) as listed in Table 14.

Table 14. Z80 Bus Mode Read States

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.
STATE T2	During State T2, the $\overline{RD}$ signal is asserted. Depending upon the instruction, either the $\overline{MREQ}$ or $\overline{IORQ}$ signal is asserted. If the external $\overline{WAIT}$ pin is driven Low at least one CPU system clock cycle prior to the end of State T2, additional WAIT states (T_{WAIT}) are asserted until the $\overline{WAIT}$ pin is driven High.
STATE T3	During State T3, no bus signals are altered. The data is latched by the eZ80F92 device at the rising edge of the CPU system clock at the end of State T3.

During Write operations, Z80 bus mode employs three states (T1, T2, and T3) as listed in Table 14.

Table 20. Motorola Bus Mode Read States (Continued)

STATE S4	During state S4, the CPU waits for a cycle termination signal $\overline{\text{DTACK}}$ ($\overline{\text{WAIT}}$), a peripheral signal. If the termination signal is not asserted at least one full CPU clock period prior to the rising clock edge at the end of S4, the CPU inserts WAIT (T_{WAIT}) states until $\overline{\text{DTACK}}$ is asserted. Each WAIT state is a full bus mode cycle.
STATE S5	During state S5, no bus signals are altered.
STATE S6	During state S6, data from the external peripheral device is driven onto the data bus.
STATE S7	On the rising edge of the clock entering state S7, the CPU latches data from the addressed peripheral device and deasserts $\overline{\text{AS}}$ and $\overline{\text{DS}}$. The peripheral device deasserts $\overline{\text{DTACK}}$ at this time.

The eight states for a Write operation in Motorola bus mode are listed in Table 21.

Table 21. Motorola Bus Mode Write States

STATE S0	The Write cycle starts in S0. The CPU drives R/$\overline{\text{W}}$ High (if a preceding Write cycle leaves R/$\overline{\text{W}}$ Low).
STATE S1	Entering S1, the CPU drives a valid address on the address bus.
STATE S2	On the rising edge of S2, the CPU asserts $\overline{\text{AS}}$ and drives R/$\overline{\text{W}}$ Low.
STATE S3	During S3, the data bus is driven out of the high-impedance state as the data to be written is placed on the bus.
STATE S4	At the rising edge of S4, the CPU asserts $\overline{\text{DS}}$. The CPU waits for a cycle termination signal $\overline{\text{DTACK}}$ ($\overline{\text{WAIT}}$). If the termination signal is not asserted at least one full CPU clock period prior to the rising clock edge at the end of S4, the CPU inserts WAIT (T_{WAIT}) states until $\overline{\text{DTACK}}$ is asserted. Each WAIT state is a full bus mode cycle.
STATE S5	During S5, no bus signals are altered.
STATE S6	During S6, no bus signals are altered.
STATE S7	Upon entering S7, the CPU deasserts $\overline{\text{AS}}$ and $\overline{\text{DS}}$. As the clock rises at the end of S7, the CPU drives R/$\overline{\text{W}}$ High. The peripheral device deasserts $\overline{\text{DTACK}}$ at this time.

Chip Select Registers

Chip Select *x* Lower Bound Register

For Memory Chip Selects, the Chip Select *x* Lower Bound register, listed in Table 22, defines the lower bound of the address range for which the corresponding Memory Chip Select (if enabled) can be active. For I/O Chip Selects, this register defines the address to which ADDR[15:8] is compared to generate an I/O Chip Select. All Chip Select lower bound registers reset to 00h.

Table 22. Chip Select x Lower Bound Register(CS0_LBR = 00A8h, CS1_LBR = 00ABh, CS2_LBR = 00AEh, CS3_LBR = 00B1h)

Bit	7	6	5	4	3	2	1	0
CS0_LBR Reset	0	0	0	0	0	0	0	0
CS1_LBR Reset	0	0	0	0	0	0	0	0
CS2_LBR Reset	0	0	0	0	0	0	0	0
CS3_LBR 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.								

Bit Position	Value	Description
[7:0] CSx_LBR	00h–FFh	**For Memory Chip Selects (CSX_IO = 0)** This byte specifies the lower bound of the Chip Select address range. The upper byte of the address bus, ADDR[23:16], is compared to the values contained in these registers for determining whether a Memory Chip Select signal should be generated.
		For I/O Chip Selects (CSX_IO = 1) This byte specifies the Chip Select address value. ADDR[15:8] is compared to the values contained in these registers for determining whether an I/O Chip Select signal should be generated.

If the NMI_OUT bit in the WDT_CTL register is set to 1, then upon time-out, the WDT asserts an NMI for CPU processing. The NMI_FLAG bit can be polled by the CPU to determine the source of the NMI event.

Watchdog Timer Registers

Watchdog Timer Control Register

The Watchdog Timer Control register, listed in Table 27, is an 8-bit Read/Write register used to enable the Watchdog Timer, set the time-out period, indicate the source of the most recent RESET, and select the required operation upon WDT time-out.

Table 27. Watchdog Timer Control Register; (WDT_CTL = 0093h)

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

Bit Position	Value	Description
7 WDT_EN	0	WDT is disabled.
	1	WDT is enabled. When enabled, the WDT cannot be disabled without a RESET.
6 NMI_OUT	0	WDT time-out resets the CPU.
	1	WDT time-out generates a nonmaskable interrupt (NMI) to the CPU.
5 RST_FLAG*	0	RESET caused by external full-chip reset or ZDI reset.
	1	RESET caused by WDT time-out. This flag is set by the WDT time-out, even if the NMI_OUT flag is set to 1. The CPU can poll this bit to determine the source of the RESET or NMI.
[4:3] WDT_CLK	00	WDT clock source is system clock.
	01	WDT clock source is Real-Time Clock source (32 kHz on-chip oscillator or 50/60 Hz input as set by RTC_CTRL[4]).
	10	Reserved.
	11	Reserved.
2	0	Reserved.

Note: *RST_FLAG is only cleared by a non-WDT RESET.

- Write values to the RTC alarm registers to set the appropriate alarm conditions
- Write to RTC_CTRL to clear the RTC_UNLOCK bit; clearing the RTC_UNLOCK bit resets and enables the clock divider

Real-Time Clock Registers

The Real-Time Clock registers are accessed via the address and data bus using I/O instructions. RTC_UNLOCK controls access to the RTC count registers. When unlocked (RTC_UNLOCK = 1), the RTC count is disabled and the count registers are Read/Write. When locked (RTC_UNLOCK = 0), the RTC count is enabled and the count registers are Read Only. The default, at RESET, is for the RTC to be locked.

Real-Time Clock Seconds Register

This register contains the current seconds count. The value in the RTC_SEC register is unchanged by a RESET. The current setting of BCD_EN determines whether the values in this register are binary (BCD_EN = 0) or binary-coded decimal (BCD_EN = 1). Access to this register is Read Only if the RTC is locked and Read/Write if the RTC is unlocked. See Table 38.

Table 38. Real-Time Clock Seconds Register; (RTC_SEC = 00E0h)

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] TEN_SEC	0–5	The tens digit of the current seconds count.
[3:0] SEC	0–9	The ones digit of the current seconds count.

Binary Operation (BCD_EN = 0)		
Bit Position	**Value**	**Description**
[7:0] SEC	00h–3Bh	The current seconds count.

Real-Time Clock Alarm Seconds Register

This register contains the alarm seconds value. See Table 46.

Table 46. Real-Time Clock Alarm Seconds Register; (RTC_ASEC = 00E8h)

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_SEC	0–5	The tens digit of the alarm seconds value.
[3:0] ASEC	0–9	The ones digit of the alarm seconds value.

Binary Operation (BCD_EN = 0)

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

Real-Time Clock Alarm Control Register

This register contains alarm enable bits for the Real-Time Clock. The RTC_ACTRL register is cleared by a RESET. See Table 50.

Table 50. Real-Time Clock Alarm Control Register; (RTC_ACTRL = 00ECh)

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/W	R/W	R/W	R/W
Note: X = Unchanged by RESET; R/W = Read/Write; R = Read Only.								

Bit Position	Value	Description
[7:4]	0000	Reserved.
3 ADOW_EN	0	The day-of-the-week alarm is disabled.
	1	The day-of-the-week alarm is enabled.
2 AHRS_EN	0	The hours alarm is disabled.
	1	The hours alarm is enabled.
1 AMIN_EN	0	The minutes alarm is disabled.
	1	The minutes alarm is enabled.
0 ASEC_EN	0	The seconds alarm is disabled.
	1	The seconds alarm is enabled.

Real-Time Clock Control Register

This register contains control and status bits for the Real-Time Clock. Some bits in the RTC_CTRL register are cleared by a RESET. The ALARM flag and associated interrupt (if INT_EN is enabled) are cleared by reading this register. The ALARM flag is updated by clearing (locking) the RTC_UNLOCK bit or by an increment of the RTC count. Writing to the RTC_CTRL register also resets the RTC clock divider allowing the RTC to be synchronized to another time source.

SLP_WAKE indicates if an RTC alarm condition initiated the CPU recovery from SLEEP mode. This bit can be checked after RESET to determine if a sleep-mode recovery is caused by the RTC. SLP_WAKE is cleared by a Read of the RTC_CTRL register.

Setting BCD_EN causes the RTC to use BCD counting in all registers including the alarm set points.

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.

Table 78. I²C Master Transmit Status Codes

Code	I²C State	Microcontroller Response	Next I²C Action
18h	Addr+W transmitted, ACK received	For a 7-bit address: write byte to DATA, clear IFLG	Transmit data byte, receive ACK
		Or set STA, clear IFLG	Transmit repeated START
		Or set STP, clear IFLG	Transmit STOP
		Or set STA & STP, clear IFLG	Transmit STOP then START
		For a 10-bit address: write extended address byte to DATA, clear IFLG	Transmit extended address byte
20h	Addr+W transmitted, ACK not received	Same as code 18h	Same as code 18h
38h	Arbitration lost	Clear IFLG	Return to idle
		Or set STA, clear IFLG	Transmit START when bus is free
68h	Arbitration lost, +W received, ACK transmitted	Clear IFLG, AAK = 0	Receive data byte, transmit NACK
		Or clear IFLG, AAK = 1	Receive data byte, transmit ACK
78h	Arbitration lost, General call addr received, ACK transmitted	Same as code 68h	Same as code 68h
B0h	Arbitration lost, SLA+R received, ACK transmitted	Write byte to DATA, clear IFLG, clear AAK = 0	Transmit last byte, receive ACK
		Or write byte to DATA, clear IFLG, set AAK = 1	Transmit data byte, receive ACK

W = Write bit; that is, the lsb is cleared to 0.

If 10-bit addressing is being used, then the status code is `18h` or `20h` after the first part of a 10-bit address plus the Write bit are successfully transmitted.

After this interrupt is serviced and the second part of the 10-bit address is transmitted, the I2C_SR register contains one of the codes in Table 79 on page 148.

Table 81. I²C Master Receive Status Codes (Continued)

Code	I²C State	Microcontroller Response	Next I²C Action
48h	Addr + R transmitted, ACK not received	For a 7-bit address: Set STA, clear IFLG	Transmit repeated START
		Or set STP, clear IFLG	Transmit STOP
		Or set STA & STP, clear IFLG	Transmit STOP then START
		For a 10-bit address: Write extended address byte to DATA, clear IFLG	Transmit extended address byte
38h	Arbitration lost	Clear IFLG	Return to idle
		Or set STA, clear IFLG	Transmit START when bus is free
68h	Arbitration lost, SLA+W received, ACK transmitted	Clear IFLG, clear AAK = 0	Receive data byte, transmit NACK
		Or clear IFLG, set AAK = 1	Receive data byte, transmit ACK
78h	Arbitration lost, General call addr received, ACK transmitted	Same as code 68h	Same as code 68h
B0h	Arbitration lost, SLA+R received, ACK transmitted	Write byte to DATA, clear IFLG, clear AAK = 0	Transmit last byte, receive ACK
		Or write byte to DATA, clear IFLG, set AAK = 1	Transmit data byte, receive ACK

R = Read bit; that is, the lsb is set to 1.

If 10-bit addressing is being used, the slave is first addressed using the full 10-bit address plus the Write bit. The master then issues a restart followed by the first part of the 10-bit address again, but with the Read bit. The status code then becomes `40h` or `48h`. It is the responsibility of the slave to remember that it had been selected prior to the restart.

If a repeated START condition is received, the status code is `10h` instead of `08h`.

After each data byte is received, the IFLG is set and one of the status codes listed in Table 82 is in the I2C_SR register.

Table 96. ZDI BREAK Control Register(ZDI_BRK_CTL = 10h in the ZDI Write Only Register Address Space)

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

Bit Position	Value	Description
7 brk_next	0	The ZDI BREAK on the next CPU instruction is disabled. Clearing this bit releases the CPU from its current BREAK condition.
	1	The ZDI BREAK on the next CPU instruction is enabled. The CPU can use multibyte Op Codes and multibyte operands. BREAK points only occur on the first Op Code in a multibyte Op Code instruction. If the ZCL pin is High and the ZDA pin is Low at the end of RESET, this bit is set to 1 and a BREAK occurs on the first instruction following the RESET. This bit is set automatically during ZDI BREAK on address match. A BREAK can also be forced by writing a 1 to this bit.
6 brk_addr3	0	The ZDI BREAK, upon matching BREAK address 3, is disabled.
	1	The ZDI BREAK, upon matching BREAK address 3, is enabled.
5 brk_addr2	0	The ZDI BREAK, upon matching BREAK address 2, is disabled.
	1	The ZDI BREAK, upon matching BREAK address 2, is enabled.
4 brk_addr1	0	The ZDI BREAK, upon matching BREAK address 1, is disabled.
	1	The ZDI BREAK, upon matching BREAK address 1, is enabled.
3 brk_addr0	0	The ZDI BREAK, upon matching BREAK address 0, is disabled.
	1	The ZDI BREAK, upon matching BREAK address 0, is enabled.

Instruction Store 4:0 Registers

The ZDI Instruction Store registers are located in the ZDI Register Write Only address space. They can be written with instruction data for direct execution by the CPU. When the ZDI_IS0 register is written, the eZ80F92 device exits the ZDI BREAK state and executes a single instruction. The Op Codes and operands for the instruction come from these Instruction Store registers. The Instruction Store Register 0 is the first byte fetched, followed by Instruction Store registers 1, 2, 3, and 4, as necessary. Only the bytes the processor requires to execute the instruction must be stored in these registers. Some CPU instructions, when combined with the MEMORY mode suffixes (.SIS, .SIL, .LIS, or .LIL), require 6 bytes to operate. These 6-byte instructions cannot be executed directly using the ZDI Instruction Store registers. See Table 101.

Note: *The Instruction Store 0 register is located at a higher ZDI address than the other Instruction Store registers. This feature allows the use of the ZDI auto-address increment function to load and execute a multibyte instruction with a single data stream from the ZDI master. Execution of the instruction commences with writing the most recent byte to ZDI_IS0.*

Table 101. Instruction Store 4:0 Registers(ZDI_IS4 = 21h, ZDI_IS3 = 22h, ZDI_IS2 = 23h, ZDI_IS1 = 24h, and ZDI_IS0 = 25h 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_IS4, ZDI_IS3, ZDI_IS2, ZDI_IS1, or ZDI_IS0	00h–FFh	These registers contain the Op Codes and operands for immediate execution by the CPU following a Write to ZDI_IS0. The ZDI_IS0 register contains the first Op Code of the instruction. The remaining ZDI_ISx registers contain any additional Op Codes or operand dates required for execution of the required instruction.

Flash Memory

Flash Memory Arrangement in eZ80F92

The eZ80F92 device features 128 KB (131,072 bytes) of non-volatile Flash memory with Read/Write/Erase capability. The main Flash memory array is arranged in 128 pages with eight rows per page and 128 bytes per row. In addition to main Flash memory, there are two separately-addressable rows which comprise a 256-byte Information Page.

The 128 KB of main storage can be protected in eight 16 KB blocks. Protecting a 16 KB block prevents Write or Erase operations. Figure 48 displays the Flash memory arrangement.

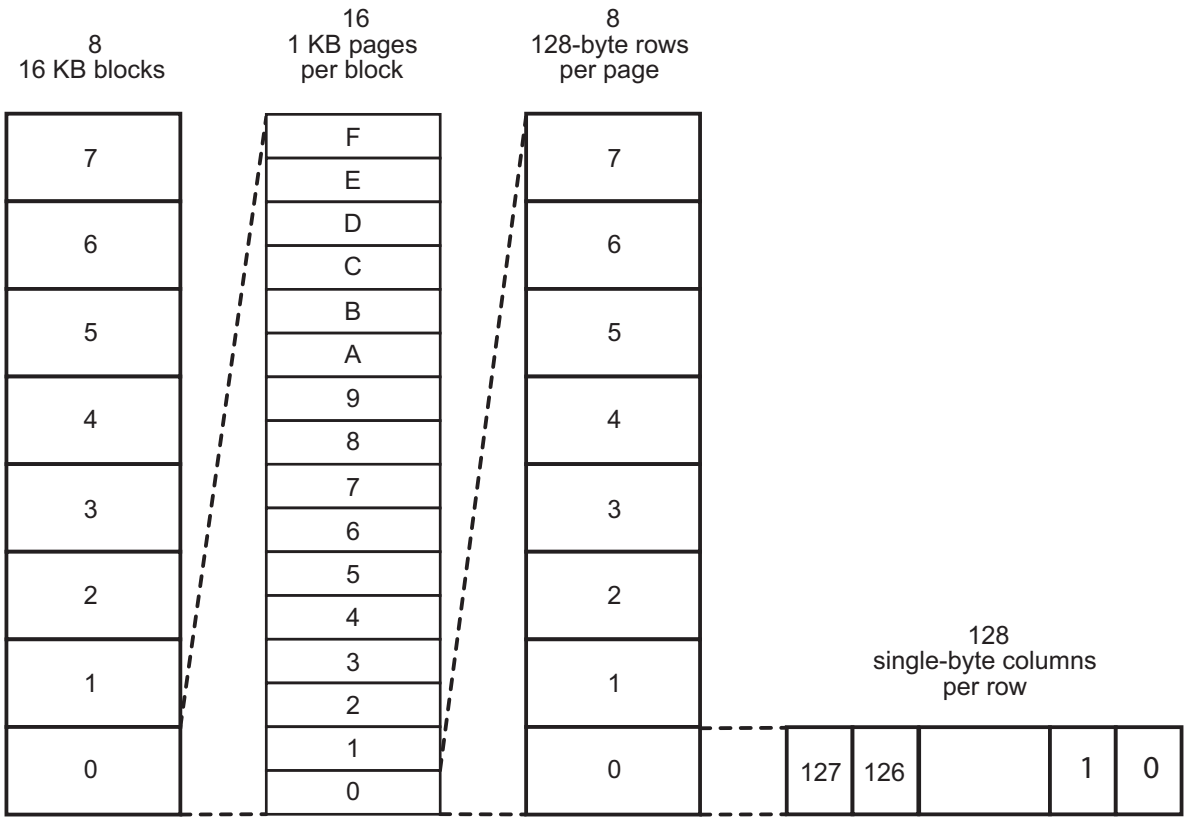

Figure 48.eZ80F92 Flash Memory Arrangement

Flash Frequency Divider Register

The 8-bit frequency divider allows programming to Flash over a range of system clock frequencies. Flash can programmed with system clock frequencies ranging from 154 kHz through 50 MHz. The Flash controller requires an input clock with a period that falls within the range of 5.1 µs to 6.5 µs. The period of the Flash controller clock is set through the Flash Frequency Divider register. Writes to this register are allowed only after it is unlocked via the FLASH_KEY register. The Frequency Divider register value required versus system clock frequency is listed in Table 117. System clock frequencies outside of the ranges shown in this table are not supported.

Table 117. Flash Frequency Divider Values

System Clock Frequency	Flash Frequency Divider Value
154–196 kHz	1
308–392 kHz	2
462–588 kHz	3
616 kHz–50 MHz	CEILING [System Clock Frequency (MHz) x 5.1 (µs)]*

Note: *The CEILING function rounds fractional values up to the next whole number, for example, CEILING(3.01) is 4.

Table 118. Flash Frequency Divider Register; (FLASH_FDIV=00F9h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	1
CPU Access	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W
Note: R/W = Read/Write, R = Read Only. *Key sequence required to enable Writes								

Bit Position	Value	Description
[7:0] FLASH_FDIV	01h–FFh	Divider value for generating the required 5.1–6.5 µs Flash controller clock period.

J

L

M

W

X

Z