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	Obsolete
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/ez80f92az020ec

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.

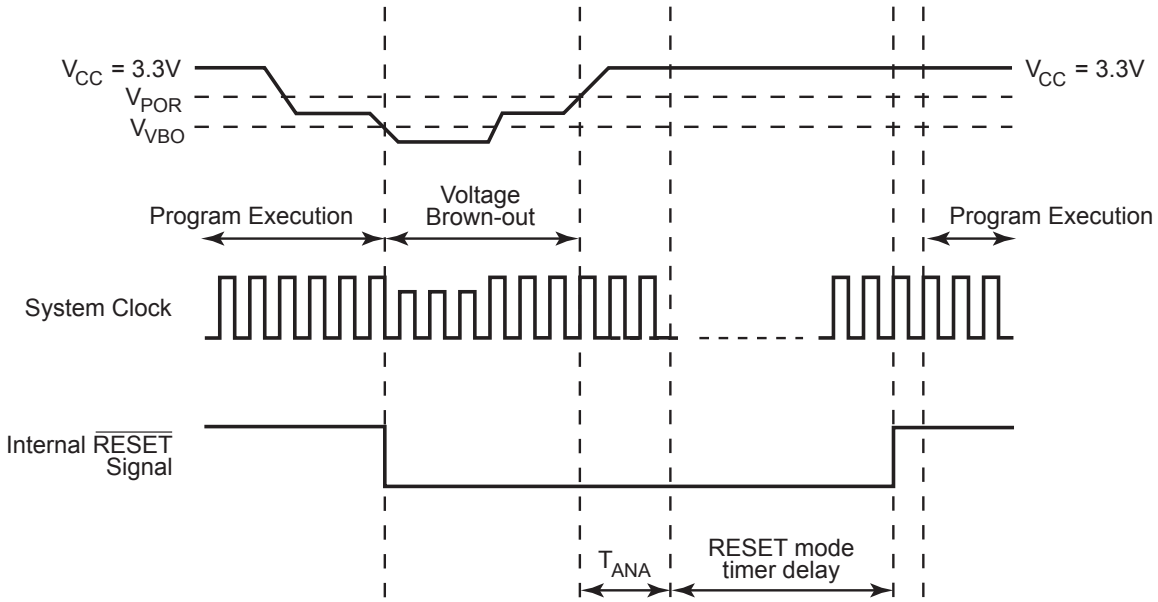

Figure 4. Voltage Brownout Reset Operation

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.

Chip Select *x* Bus Mode Control Register+The Chip Select Bus Mode register, listed in Table 25, configures the Chip Select for eZ80®, Z80®, Intel™, or Motorola bus modes. Changing the bus mode allows the eZ80F92 device to interface to peripherals based on the Z80-, Intel-, or Motorola-style asynchronous bus interfaces. When a bus mode other than CPU is programmed for a particular Chip Select, the CSx_WAIT setting in that Chip Select Control Register is ignored.

Table 25. Chip Select *x* Bus Mode Control Register(CS0_BMC = 00F0h, CS1_BMC = 00F1h, CS2_BMC = 00F2h, CS3_BMC = 00F3h)

Bit	7	6	5	4	3	2	1	0
CS0_BMC Reset	0	0	0	0	0	0	1	0
CS1_BMC Reset	0	0	0	0	0	0	1	0
CS2_BMC Reset	0	0	0	0	0	0	1	0
CS3_BMC Reset	0	0	0	0	0	0	1	0
CPU Access	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Note: R/W = Read/Write; R = Read Only.								

Bit Position	Value	Description
[7:6] BUS_MODE	00	eZ80® bus mode.
	01	Z80 bus mode.
	10	Intel™ bus mode.
	11	Motorola bus mode.
5 AD_MUX	0	Separate address and data.
	1	Multiplexed address and data—appears on data bus DATA[7:0].
4	0	Reserved.

[1:0] TMR0_IN	00	Timer counts at system clock divided by clock divider.
	01	Timer event input is Real-Time Clock source (32 kHz or 50/60 Hz—see Real-Time Clock on page 88 for details).
	10	The timer event input is the GPIO Port B pin 0.
	11	The timer event input is the GPIO Port B pin 0.

Real-Time Clock Minutes Register

This register contains the current minutes count. See Table 39.

Table 39. Real-Time Clock Minutes Register; (RTC_MIN = 00E1h)

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_MIN	0–5	The tens digit of the current minutes count.
[3:0] MIN	0–9	The ones digit of the current minutes count.

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

Real-Time Clock Alarm Day-of-the-Week Register

This register contains the alarm day-of-the-week value. See Table 49.

Table 49. Real-Time Clock Alarm Day-of-the-Week Register; (RTC_ADOW = 00EBh)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	X	X	X	X
CPU Access	R	R	R	R	R/W*	R/W*	R/W*	R/W*

Note: X = Unchanged by RESET; R = Read Only; 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]	0000	Reserved.
[3:0] ADOW	1-7	The alarm day-of-the-week.value.

Binary Operation (BCD_EN = 0)

Bit Position	Value	Description
[7:4]	0000	Reserved.
[3:0] ADOW	01h–07h	The alarm day-of-the-week value.

UART Modem Control Register

This register is used to control and check the modem status, as listed in Table 63.

Table 63. UART Modem Control Registers(UART0_MCTL = 00C4h, UART1_MCTL = 00D4h)

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

Bit Position	Value	Description
[7:6]	00b	Reserved—must be 00b.
5 MDM	0	MULTIDROP mode disabled.
	1	MULTIDROP mode enabled. See Table 62 on page 118 for parity select definitions.
4 LOOP	0	LOOP BACK mode is not enabled.
	1	LOOP BACK mode is enabled. The UART operates in internal LOOP BACK mode. The transmit data output port is disconnected from the internal transmit data output and set to 1. The receive data input port is disconnected and internal receive data is connected to internal transmit data. The modem status input ports are disconnected and the four bits of the modem control register are connected as modem status inputs. The two modem control output ports (OUT1&2) are set to their inactive state.
3 OUT2	0–1	No function in normal operation. In LOOP BACK mode, this bit is connected to the DCD bit in the UART Status Register.
2 OUT1	0–1	No function in normal operation. In LOOP BACK mode, this bit is connected to the RI bit in the UART Status Register.
1 RTS	0–1	**Request To Send** In normal operation, the $\overline{\text{RTS}}$ output port is the inverse of this bit. In LOOP BACK mode, this bit is connected to the CTS bit in the UART Status Register.
0 DTR	0–1	**Data Terminal Ready** In normal operation, the $\overline{\text{DTR}}$ output port is the inverse of this bit. In LOOP BACK mode, this bit is connected to the DSR bit in the UART Status Register.

Setting the upper 4 bits of IR_CTL to 00h disables the frequency divider but not the IrDA receiver. In this mode, the IrDA receiver uses edge detection on the IR_RxD bit stream.

Jitter

Due to the inherent sampling of the received IR_RxD signal by the BIt Rate Clock, some jitter can be expected on the first bit in any sequence of data. However, all subsequent bits in the received data stream are a fixed 16 clock periods wide.

Infrared Encoder/Decoder Signal Pins

The IrDA endec signal pins (IR_TxD and IR_RxD) are multiplexed with General-Purpose I/O (GPIO) pins. These GPIO pins must be configured for alternate function operation for the endec to operate.

The remaining six UART0 pins ($\overline{\text{CTS0}}$, $\overline{\text{DCD0}}$, $\overline{\text{DSR0}}$, $\overline{\text{DTR0}}$, $\overline{\text{RTS}}$ and $\overline{\text{RI0}}$) are not required for use with the endec. The UART0 modem status interrupt should be disabled to prevent unwanted interrupts from these pins. The GPIO pins corresponding to these six unused UART0 pins can be used for inputs, outputs, or interrupt sources. Recommended GPIO Port D control register settings are provided in Table 69. See General-Purpose Input/Output on page 39 for additional information about setting the GPIO Port modes.

Table 69. GPIO Mode Selection when using the IrDA Encoder/Decoder

GPIO Port D Bits	Allowable GPIO Port Mode	Allowable Port Mode Functions
PD0	7	Alternate function.
PD1	7	Alternate function.
PD2–PD7	Any other than GPIO Mode 7 (1, 2, 3, 4, 5, 6, 8, or 9)	Output, input, open-drain, open-source, level-sensitive interrupt input, or edge-triggered interrupt input.

Loopback Testing

Both internal and external loopback testing can be accomplished with the IrDA endec on the eZ80F92 device. Setting the LOOP_BACK bit to 1 enables internal loopback testing. During internal loopback, the IR_TxD output signal is inverted and connected on-chip to the IR_RxD input. External loopback testing of the off-chip IrDA transceiver can be accomplished by transmitting data from the UART while the receiver is enabled (IR_RxEN set to 1).

is transmitted, the IFLG is set and the I2C_SR register contains C8h and the I2C returns to the idle state. The AAK bit must be set to 1 before reentering SLAVE mode.

If no acknowledge is received after transmitting a byte, the IFLG is set and the I2C_SR register contains C0h. The I2C then returns to the idle state.

If a STOP condition is detected after an acknowledge bit, the I2C returns to the idle state.

Slave Receive

In SLAVE RECEIVE mode, a number of data bytes are received from a master transmitter.

The I2C enters SLAVE RECEIVE mode when it receives its own slave address and a Write bit (lsb = 0) after a START condition. The I2C transmits an acknowledge bit and sets the IFLG bit in the I2C_CTL register and the I2C_SR register contains the status code 60h. The I2C also enters SLAVE RECEIVE mode when it receives the general call address 00h (if the GCE bit in the I2C_SAR register is set). The status code is then 70h.

Note: *When the I2C contains a 10-bit slave address (signified by F0h–F7h in the I2C_SAR register), it transmits an acknowledge after the first address byte is received but no interrupt is generated. IFLG is not set and the status does not change. The I2C generates an interrupt only after the second address byte is received. The I2C sets the IFLG bit and loads the status code as described above.*

I2C goes from MASTER mode to SLAVE RECEIVE mode when arbitration is lost during the transmission of an address, and the slave address and Write bit (or the general call address if the CGE bit in the I2C_SAR register is set to 1) are received. The status code in the I2C_SR register is 68h if the slave address is received or 78h if the general call address is received. The IFLG bit must be cleared to 0 to allow data transfer to continue.

If the AAK bit in the I2C_CTL register is set to 1 then an acknowledge bit (Low level on SDA) is transmitted and the IFLG bit is set after each byte is received. The I2C_SR register contains the status code 80h or 90h if SLAVE RECEIVE mode is entered with the general call address. The received data byte can be read from the I2C_DR register and the IFLG bit must be cleared to allow the transfer to continue. If a STOP condition or a repeated START condition is detected after the acknowledge bit, the IFLG bit is set and the I2C_SR register contains status code A0h.

If the AAK bit is cleared to 0 during a transfer, the I2C transmits a not-acknowledge bit (High level on SDA) after the next byte is received, and set the IFLG bit. The I2C_SR register contains the status code 88h or 98h if SLAVE RECEIVE mode is entered with the general call address. The I2C returns to the idle state when the IFLG bit is cleared to 0.

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.

Table 89. I²C Status Codes (Continued)

Code	Status
40h	Address and Read bit transmitted, ACK received
48h	Address and Read bit transmitted, ACK not received
50h	Data byte received in MASTER mode, ACK transmitted
58h	Data byte received in MASTER mode, NACK transmitted
60h	Slave address and Write bit received, ACK transmitted
68h	Arbitration lost in address as master, slave address and Write bit received, ACK transmitted
70h	General Call address received, ACK transmitted
78h	Arbitration lost in address as master, General Call address received, ACK transmitted
80h	Data byte received after slave address received, ACK transmitted
88h	Data byte received after slave address received, NACK transmitted
90h	Data byte received after General Call received, ACK transmitted
98h	Data byte received after General Call received, NACK transmitted
A0h	STOP or repeated START condition received in SLAVE mode
A8h	Slave address and Read bit received, ACK transmitted
B0h	Arbitration lost in address as master, slave address and Read bit received, ACK transmitted
B8h	Data byte transmitted in SLAVE mode, ACK received
C0h	Data byte transmitted in SLAVE mode, ACK not received
C8h	Last byte transmitted in SLAVE mode, ACK received
D0h	Second Address byte and Write bit transmitted, ACK received
D8h	Second Address byte and Write bit transmitted, ACK not received
F8h	No relevant status information, IFLG = 0

If an illegal condition occurs on the I²C bus, the bus error state is entered (status code 00h). To recover from this state, the STP bit in the I2C_CTL register must be set and the IFLG bit cleared. The I²C then returns to the idle state. No STOP condition is transmitted on the I²C bus.

Note: *The STP and STA bits may be set to 1 at the same time to recover from the bus error. The I²C then sends a START condition.*

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 95. ZDI Address Match Registers(ZDI_ADDR0_L = 00h, ZDI_ADDR0_H = 01h, ZDI_ADDR0_U = 02h, ZDI_ADDR1_L = 04h, ZDI_ADDR1_H = 05h, ZDI_ADDR1_U = 06h, ZDI_ADDR2_L = 08h, ZDI_ADDR2_H = 09h, ZDI_ADDR2_U = 0Ah, ZDI_ADDR3_L = 0Ch, ZDI_ADDR3_H = 0Dh, and ZDI_ADDR3_U = 0Eh 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: W = Write Only.								

Bit Position	Value	Description
[7:0] ZDI_ADDR*x*_L, ZDI_ADDR*x*_H, or ZDI_ADDR*x*_U	00h–FFh	The four sets of ZDI address match registers are used for setting the addresses for generating BREAK points. The 24-bit addresses are supplied by {ZDI_ADDRx_U, ZDI_ADDRx_H, ZDI_ADDRx_L, where *x* is 0, 1, 2, or 3.

ZDI BREAK Control Register

The ZDI BREAK Control register is used to enable BREAK points. ZDI asserts a BREAK when the CPU instruction address, ADDR[23:0], matches the value in the ZDI Address Match 3 registers, {ZDI_ADDR3_U, ZDI_ADDR3_H, ZDI_ADDR3_L}. BREAKs can only occur on an instruction boundary. If the instruction address is not the beginning of an instruction (that is, for multibyte instructions), then the BREAK occurs at the end of the current instruction. The BRK_NEXT bit is set to 1. The BRK_NEXT bit must be reset to 0 to release the BREAK. See Table 96 on page 173.

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.

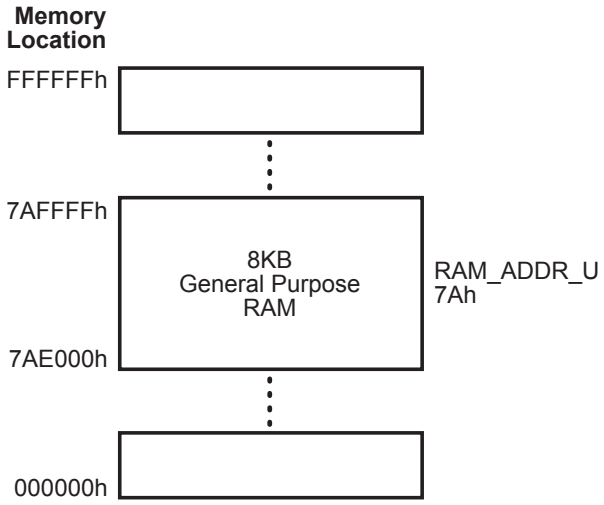

Figure 46.eZ80F92 On-Chip RAM Memory Addressing Example

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.

RAM Control Registers

RAM Control Register

The internal data RAM can be disabled by clearing the RAM_EN bit. The default, upon RESET, is for RAM to be enabled.

Table 111. RAM Control Register; (RAM_CTL=00B4h)

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

Bit Position	Value	Description
7 RAM_EN	0	On-chip general-purpose RAM is disabled
	1	On-chip general-purpose RAM is enabled
[6:0]	0000000	Reserved

RAM Address Upper Byte Register

The RAM_ADDR_U register defines the upper byte of the address for the on-chip RAM. If enabled, RAM addresses assume priority over all Chip Selects. The external Chip Select signals are not asserted if the corresponding RAM address is enabled.

Table 112. RAM Address Upper Byte Register; (RAM_ADDR_U=00B5h)

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.								

Bit Position	Value	Description
[7:0] RAM_ADDR_U	00h–FFh	This byte defines the upper byte of the RAM address. On-chip RAM is prioritized over all Memory Chip Selects. If the enabled RAM and Chip Select addresses overlap, the external Chip Select is not asserted.

Index

Numerics

A

B

C

J

L

M