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	eZ8
Core Size	8-Bit
Speed	24MHz
Connectivity	DALI, DMX, I²C, LINbus, SPI, UART/USART, USB
Peripherals	DMA, LCD, LVD, POR, PWM, WDT
Number of I/O	67
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.75K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-LQFP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f3282at024xk2246

Table 6. F6482 Series Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F1682 and Z8F1681 Products (Continued)	
0000–0001	Flash option bits
0002–0003	Reset vector
0004–0005	WDT interrupt vector
0006–0007	Illegal instruction trap
0008–0047	Interrupt vectors*
0048–004B	Oscillator fail traps*
004C–3FFF	Program Flash
F000	NVDS byte read
F3FD	NVDS byte write

Note: *See Table 51 on page 128 for a list of interrupt vectors and traps.

3.3. Data Memory

F6482 Series MCUs do not use the eZ8 CPU's 64KB Data Memory address space.

3.4. Flash Information Area

Table 7 lists the F6482 Series Flash Information Area. This 1KB space consists of two pages and is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into Program Memory and overlays the FC00h to FFFFh address range. When Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

Table 7. F6482 Series Flash Memory Information Area Map

Program Memory Address (Hex)	Function
FC00–FC3F	Zilog option bits.
FC40–FC53	Part number: a 20-character ASCII alphanumeric code, left-justified and filled with Fh.
FC54–FC5F	Reserved.
FC60–FC7F	Zilog calibration data.
FC80–FFFF	Reserved.

Table 21. Port Alternate Function Mapping, 80-Pin Parts (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Subregisters
Port C[2]	PC0	Reserved		AFS1[0]: 0, AFS2[0]: 0
		ANA4/VBIAS/ C0INP	ADC or Voltage Bias with low current drive capability or Comparator 0 Input (P)	AFS1[0]: 1, AFS2[0]: 0
		Reserved		AFS1[0]: x, AFS2[0]: 1
	PC1	MISO0	SPI 0 Master In/Slave Out	AFS1[1]: 0, AFS2[1]: 0
		ANA5/C0INN	ADC or Comparator 0 Input (N)	AFS1[1]: 1, AFS2[1]: 0
		Reserved		AFS1[1]: x, AFS2[1]: 1
	PC2	$\overline{SS0}$	SPI 0 Slave Select	AFS1[2]: 0, AFS2[2]: 0
		ANA3	ADC Analog Input	AFS1[2]: 1, AFS2[2]: 0
		Reserved		AFS1[2]: x, AFS2[2]: 1
	PC3	MISO0	SPI 0 Master In Slave Out	AFS1[3]: 0, AFS2[3]: 0
		ANA11/DAC	ADC or DAC	AFS1[3]: 1, AFS2[3]: 0
		Reserved		AFS1[3]: x, AFS2[3]: 1
	PC4	MOSI0	SPI 0 Master Out/Slave In	AFS1[4]: 0, AFS2[4]: 0
		T0IN/$\overline{T0OUT}$	Timer 0 Input/Timer 0 Output Complement	AFS1[4]: 1, AFS2[4]: 0
		SCL	I^2C Serial Clock	AFS1[4]: 0, AFS2[4]: 1
		$\overline{DE0}$	UART 0 Driver Enable	AFS1[4]: 1, AFS2[4]: 1
	PC5	SCK0	SPI 0 Serial Clock	AFS1[5]: 0, AFS2[5]: 0
		T0OUT	Timer 0 Output	AFS1[5]: 1, AFS2[5]: 0
		SDA	I^2C Serial Data	AFS1[5]: 0, AFS2[5]: 1
		$\overline{CTS0}$	UART 0 Clear to Send	AFS1[5]: 1, AFS2[5]: 1
	PC6	T2IN/$\overline{T2OUT}$	Timer 2 Input/Timer2 Output Complement	AFS1[6]: 0, AFS2[6]: 0
		SCKOUT	System Clock Out	AFS1[6]: 1, AFS2[6]: 0
		ESOUT[0]	Event System Output 0	AFS1[6]: 0, AFS2[6]: 1
		Reserved		AFS1[6]: 1, AFS2[6]: 1

Notes
1. Because there are at most two choices of alternate function for some pins of Ports A, B, D, E, F and G, the Alternate Function Set Subregister AFS2 is not implemented. Also, alternate function selection, as described in the Port A–J Alternate Function Subregisters (see page 88) must also be enabled.
2. The alternate function selection for Port C, as described in the Port A–J Alternate Function Subregisters (see page 88) must also be enabled.
3. Because there is only a single alternate function for each pin in ports H and J, the Alternate Function Set subregisters are not implemented for these two ports. Also, alternate function selection, as described in the Port A–J Alternate Function Subregisters (see page 88) must also be enabled.

9.4.3. Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) Register, shown in Table 54, stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 2 Register to determine if any interrupt requests are pending.

Table 54. Interrupt Request 2 Register (IRQ2)

Bit	7	6	5	4	3	2	1	0
Field	PA7VI	PA6CI	PA5CI	PAD4I	PAD3I	PAD2I	PAD1I	PA0I
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC6h							

Bit	Description
[7] PA7VI	**Port A7 or LVD Interrupt Request** 0: No interrupt request is pending for GPIO Port A7 or LVD. 1: An interrupt request from GPIO Port A7 or LVD.
[6] PA6CI	**Port A6 or Comparator 0 Interrupt Request** 0: No interrupt request is pending for GPIO Port A6 or Comparator 0. 1: An interrupt request from GPIO Port A6 or Comparator 0.
[5] PA5CI	**Port A5 or Comparator 1 Interrupt Request** 0: No interrupt request is pending for GPIO Port A5 or Comparator 1. 1: An interrupt request from GPIO Port A5 or Comparator 1.
[4:1] PAD*x*I	**Port A*x* or Port D*x* Interrupt Request** 0: No interrupt request is pending for GPIO Port A*x* or Port D*x*; *x* indicates the specific Port A or D number (4–1). 1: An interrupt request from GPIO Port A*x* or Port D*x* is awaiting service.
[0] PA0I	**Port A0 Interrupt Request** For a description of the interrupt source select feature, see the Shared Interrupt Select Register 0 section on page 146. 0: No interrupt request is pending for GPIO Port A0. 1: An interrupt request from GPIO Port A0 is awaiting service.

not recommend DMA for One-Shot Compare operation, because upon a channel interrupt, the channel is disabled, thereby clearing its DMA request.

- DMA request behavior is a function of channel mode. The behavior in the first paragraph under DMA would pertain to all modes except Capture Mode.

11.5. Low-Power Modes

11.5.1. Operation in Halt Mode

When the eZ8 CPU is operating in Halt Mode, the Multi-Channel Timer will continue to operate if enabled. To minimize current in Halt Mode, the Multi-Channel Timer must be disabled by clearing the TEN control bit.

11.5.2. Operation in Stop Mode

When the eZ8 CPU is operating in Stop Mode, the Multi-Channel Timer ceases to operate as the System Clock is stopped. The registers are not reset and operation will resume after Stop-Mode Recovery occurs.

11.5.3. Power Reduction During Operation

Deassertion of the TEN bit will inhibit clocking of the entire Multi-Channel Timer block. Deassertion of the CHEN bit of individual channels will inhibit clocking of channel specific logic to minimize power consumption of unused channels. The CPU can still read/write registers when the enable bit(s) are deasserted.

11.6. Multi-Channel Timer Application Examples

11.6.1. PWM Programmable Deadband Generation

The Count Up/Down Mode supports motor control applications that require dead time between output signals. Figure 24 shows dead-time generation between two channels operating in Count Up/Down Mode.

Bit	Description (Continued)
[3:0] CHyEF	**Channel y Event Flag** This bit indicates whether a Capture/Compare event occurred for this channel. Software can use this bit to determine the channel(s) responsible for generating the MCT channel interrupt. This event flag is cleared by writing a 1 to the bit. These bits will be set when an event occurs independently of the setting of the CHIEN bit. This bit is cleared when TEN=0 (TEN is the MSB of MCTCTL1). 0: No Capture/Compare event occurred for this channel. 1: A Capture/Compare event occurred for this channel.

Note: *y* = A, B, C, D.

11.7.8. Multi-Channel Timer Channel-y Control Registers

Each channel in the Multi-Channel Timer Channel-y Control registers, shown in Table 100, has a control register to enable the channel, select the input/output polarity, enable channel interrupts, and select the channel mode of operation.

Table 100. Multi-Channel Timer Channel Control Register (MCTCHyCTL)

Bit	7	6	5	4	3	2	1	0
Field	CHEN	CHPOL	CHIEN	CHUE	Reserved	CHOP		
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Address	02h, 03h, 04h, 05h in Subaddress Register, accessible through Subregister 2							

Bit	Description
[7] CHEN	**Channel Enable** 0: Channel is disabled. 1: Channel is enabled.

Note: *y* = A, B, C, D.

b. If Multiprocessor Mode is not enabled, then enable parity (if appropriate) and select either even or odd parity.

9. Execute an EI instruction to enable interrupts.

The UART-LDD is now configured for interrupt-driven data reception. When the UART-LDD Receiver interrupt is detected, the associated ISR performs the following:

1. Checks the UART-LDD Status 0 Register to determine the source of the interrupt-error, break, or received data.
2. If the interrupt is due to data available, read the data from the UART-LDD Receive Data Register. If operating in Multiprocessor (9-bit) Mode, further actions may be required depending on the Multiprocessor Mode bits MPMD[1:0].
3. Execute the IRET instruction to return from the ISR and await more data.

14.1.6. Clear To Send Operation

The Clear To Send ($\overline{\text{CTS}}$) pin, if enabled by the CTSE bit of the UART-LDD Control 0 Register, performs flow control on the outgoing transmit data stream. The Clear To Send ($\overline{\text{CTS}}$) input pin is sampled one system clock before any new character transmission begins. To delay transmission of the next data character, an external receiver must reduce $\overline{\text{CTS}}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this operation is typically performed during the stop bit transmission. If $\overline{\text{CTS}}$ stops in the middle of a character transmission, the current character is sent completely.

14.1.7. External Driver Enable

The UART-LDD provides a Driver Enable (DE) signal for off-chip bus transceivers. This feature reduces the software overhead associated using a GPIO pin to control the transceiver when communicating on a multitransceiver bus, such as RS-485.

Driver Enable is a programmable polarity signal which envelopes the entire transmitted data frame including parity and stop bits, as illustrated in Figure 30. The Driver Enable signal asserts when a byte is written to the UART-LDD Transmit Data Register. The Driver Enable signal asserts at least one bit period, and no greater than two bit periods, before the start bit is transmitted. This assertion allows a set-up time to enable the transceiver. The Driver Enable signal deasserts one system clock period after the last stop bit is transmitted. This system clock delay allows both time for data to clear the transceiver before disabling it, as well as the ability to determine if another character follows the current character. In the event of back-to-back characters (new data must be written to the Transmit Data Register before the previous character is completely transmitted), the DE signal is not deasserted between characters. The DEPOL bit in the UART-LDD Control Register 1 sets the polarity of the Driver Enable signal.

Transmit Data Register is written or the ESPI block is disabled. After the Transmit Data Register is loaded into the Shift Register to start a new transfer, the TDRE bit will be set again, causing a new transmit interrupt. In Master or Slave modes, if information is being received but not transmitted, the transmit interrupts can be eliminated by selecting Receive Only Mode (ESPIEN1,0=01).

A receive interrupt is generated by the RDRNE status bit when the ESPI block is enabled, the DIRQS bit is set, and a character transfer completes. At the end of the character transfer, the contents of the Shift Register are transferred into the Receive Data Register, causing the RDRNE bit to assert. The RDRNE bit is cleared when the Data Buffer is read as empty. If information is being transmitted but not received by the software application, the receive interrupt can be eliminated by selecting Transmit Only Mode (ESPIEN1,0=10) in either Master or Slave modes. When information is being sent and received under interrupt control, RDRNE and TDRE will both assert simultaneously at the end of a character transfer. In this case, RDRNE and TDRE can be serviced in either order.

ESPI error interrupts occur if any of the TUND, COL, ABT and ROVR bits in the ESPI Status Register are set. These bits are cleared by writing a 1. If the ESPI is disabled (ESPIEN1, 0=00), an ESPI interrupt can be generated by a Baud Rate Generator time-out. This timer function must be enabled by setting the BRGCTL bit in the ESPICTL Register. This timer interrupt does not set any of the bits of the ESPI Status Register.

15.3.7. ESPI and DMA

The ESPI will assert a DMA RX request whenever the receive data register is not empty (RDRNE=1), and will deassert a DMA RX request whenever the Receive Data Register is read by the DMA or software.

The EPSI will assert a DMA TX request whenever the Transmit Data Register is empty (TDRE=1), and will deassert a DMA TX request whenever the Transmit Data Register is written by the DMA or software.

When using DMA, it can be desirable to clear DIRQS so that interrupts occur on errors, but not upon data requests. If the software application is moving data in only one direction, the ESPIEN1,ESPIEN0 bits are set to 10 or 01, allowing a single DMA channel to control the ESPI data transfer. When operating in Receive Only Mode, transmit data will be all 1s.

15.3.8. ESPI Baud Rate Generator

In ESPI Master Mode, the Baud Rate Generator creates a lower frequency serial clock (SCK) for data transmission synchronization between the master and the external slave. The input to the Baud Rate Generator is the system clock. The ESPI Baud Rate High and Low Byte registers combine to form a 16-bit reload value, BRG[15:0], for the ESPI Baud Rate Generator. The ESPI baud rate is calculated using the following equation:

USB host by the USB Module is placed in a USB Module IN endpoint buffer space prior to transmission. These endpoint buffer spaces can be accessed by software or by DMA.

Upon reset, the size of each endpoint buffer memory is configured to be 64 bytes. If it is desire that any endpoint buffer memory be less than 64 bytes, prior to enabling the USB, the size of each endpoint buffer memory must be configured.

OUT endpoint buffer memory sizes are used to derive the endpoint buffer memory allocation loaded into the USBOxADDR and USBISTADDR subregisters. IN endpoint buffer memory sizes are used to derive the endpoint buffer memory allocation loaded into the USBIxADDR and USBISPADDR subregisters. OUT endpoint 0 buffer memory starts at address 000h. USBOxADDR, USBISTADDR and USBIxADDR subregisters contain start address information for the other IN and OUT endpoints in buffer memory. The USBISPADDR Subregister contains the stop (upper) address for the highest number IN endpoint used. The size of an endpoint buffer space is a multiple of 2 bytes.

The equations to determine values for the endpoint buffer memory allocation subregisters are shown in Table 165. Endpoint buffer size is determined by subtracting consecutive values of endpoint start addresses, as shown in Table 165. If an OUT endpoint does not exist (or is not used), USBOxADDR should be cleared to 00h for that OUT endpoint. If an IN endpoint does not exist (or is not used), USBIxADDR should be cleared to 00h for that IN endpoint. Table 166 shows an example of endpoint buffer memory allocation when IN and OUT endpoints 3 are not used. Example register values are shown in Figure 55.

Endpoint buffer memory can be accessed using software as described in the Overview of USB Registers and Subregisters section on page 341 or using DMA as described in the DMA section on page 355.

Table 165. Determining USB Endpoint Buffer Memory Allocation with All Endpoints Used

Subregister	Subregister Value	Endpoint Buffer Size (Bytes)
USBO1ADDR	OUT_EP0_SIZE ÷ 2	OUT_EP0_SIZE=2 * USBO1ADDR
USBO2ADDR	USBO1ADDR + (OUT_EP1_SIZE ÷ 2)	OUT_EP1_SIZE=2 * (USBO2ADDR –USBO1ADDR)
USBO3ADDR	USBO2ADDR + (OUT_EP2_SIZE ÷ 2)	OUT_EP2_SIZE=2 * (USBO3ADDR –USBO2ADDR)
USBISTADDR	(USBO3ADDR + (OUT_EP3_SIZE ÷ 2)) ÷ 2	OUT_EP3_SIZE=2 * ((2 * USBISTADDR) –USBO3ADDR)
USBI1ADDR	IN_EP0_SIZE ÷ 2	IN_EP0_SIZE=2 * USBI1ADDR
USBI2ADDR	USBI1ADDR + (IN_EP1_SIZE ÷ 2)	IN_EP1_SIZE=2 * (USBI2ADDR – USBI1ADDR)
USBI3ADDR	USBI2ADDR + (IN_EP2_SIZE ÷ 2)	IN_EP2_SIZE=2 * (USBI3ADDR – USBI2ADDR)
USBISPADDR	((USBI3ADDR + (IN_EP3_SIZE ÷ 2)) ÷ 8) + (USBISTADDR ÷ 4)	IN_EP3_SIZE=2 * ((8 * USBISPADDR) –USBI3ADDR – (2 * USBISTADDR))

17.3.13. USB OUT Interrupt Request Subregister

The USB OUT Interrupt Request Subregister, shown in Table 182, indicates the OUT endpoint interrupt requests. The USB Module sets OUTxIRQ when it receives an error free OUT endpoint x data packet. The interrupt is cleared by writing a 1 to the corresponding register position.

Table 182. USB OUT Interrupt Request Subregister (USBOUTIRQ)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				OUT3IRQ	OUT2IRQ	OUT1IRQ	OUT0IRQ
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W1*	R/W1*	R/W1*	R/W1*
Address	If USBSA = 2Ah in the USB Subaddress Register, accessible through the USB Subdata Register							

Note: *R/W1 = Writing a 1 clears this bit.

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] OUT3IRQ	**OUT Endpoint 3 Interrupt Request** 0: No interrupt request from OUT endpoint 3. 1: Interrupt request from OUT endpoint 3.
[2] OUT2IRQ	**OUT Endpoint 2 Interrupt Request** 0: No interrupt request from OUT endpoint 2. 1: Interrupt request from OUT endpoint 2.
[1] OUT1IRQ	**OUT Endpoint 1 Interrupt Request** 0: No interrupt request from OUT endpoint 1. 1: Interrupt request from OUT endpoint 1.
[0] OUT0IRQ	**OUT Endpoint 0 Interrupt Request** 0: No interrupt request from OUT endpoint 0. 1: Interrupt request from OUT endpoint 0.

18.3.9. DMA 0–3 Linked List Descriptor Address High and Low Subregisters

The DMA Linked List Descriptor Address High and Low (DMAxLAH and DMAxLAL) subregisters, shown in Tables 215 and 216, contain the 12-bit linked list descriptor address. These registers have an effect only during linked list operation. Writing DMAx-LAL automatically starts the Linked List DMAx even if DMAxLAH is not written.

Table 215. DMA 0–3 Linked List Descriptor Address High Subregister (DMAxLAH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				LAH			
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Address	If DMASA = 8h in the DMAxSA Register, accessible through the DMA 0–3 Subregister							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] LAH	**Linked List Descriptor Address High** LAH and LAL together form the 12-bit address that points to the current linked list descriptor. 0–F: The upper 4 bits of the linked list descriptor address.

Table 216. DMA 0–3 Linked List Descriptor Address Low Subregister (DMAxLAL)

Bit	7	6	5	4	3	2	1	0
Field	LAL							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
Address	If DMASA = 9h in the DMAxSA Register, accessible through the DMA 0–3 Subregister							

Bit	Description
[7:0] LAL	**Linked List Descriptor Address Low** LAH and LAL together form the a 12-bit address that points to the current linked list descriptor. As descriptors are aligned on 8-byte address boundaries, the lower three bits of LAL cannot be written, and are always zero. Writing to this DMAxLAL Register automatically starts the linked list DMA operation. A write to DMAxLAH is not required to start linked list operation. 00–F8: The lower 8 bits of the linked list descriptor address.

19.6.2. Event System Source Subdata Register

The Event System Source Subdata Register (ESSSD), shown in Table 221, sets Channel Source Subregister (ESCHxSRC) operation. The value in the ESSSA Register determines which ESCHxSRC subregister is accessed.

Table 221. Event System Source Subdata Register (ESSSD)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	ESSSD						
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F99h							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6:0] ESSSD	**Event System Source Subregister Data**

Bit	Description (Continued)
RESOLUT = 1 (14-bit)	
[7:0]	00–FF: The 8 LSBs of the last conversion result are latched into this data register whenever the ADC Data High Byte register is read.

21.3.9. Sample Time Register

The Sample Time Register, shown in Table 243, is used to program the length of the sampling time and sample settling time after a conversion begins by setting the START=01 in the ADC Control 0 Register or is initiated by the Event System. The number of ADC clock cycles required for sample time varies from system to system, depending on the impedance of the external source and the ADC clock period used. The system designer should program this register to contain the number of ADC clocks required to meet accuracy requirements as described in the ADC Timing section on page 446.

Table 243. Sample Time (ADCST)

Bits	7	6	5	4	3	2	1	0
Field	ST				SST			
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F78h							

Bit	Description
[7:4] ST	**Sampling Time** 0000: 2 ADC clocks. 0001: 2 ADC clocks. 0010: 4 ADC clocks. 0011: 8 ADC clocks. 0100: 16 ADC clocks. 0101: 32 ADC clocks. 0110: 64 ADC clocks. 0111: 96 ADC clocks. 1000: 128 ADC clocks. 1001: 192 ADC clocks. 1010: 256 ADC clocks. 1011: 320 ADC clocks. 1100: 384 ADC clocks. 1101: 512 ADC clocks. 1110: 768 ADC clocks. 1111: 1024 ADC clocks.

Bit	Description
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:4] IRESSEL	**Current Source Resistor Select** RESSEL is meaningful only when MODE=1 00: External resistor, connected to the GPIO used as Op Amp B negative input, AMPBINN, forms current drive set point. 01: Internal 31.25KΩ (nominal) resistor forms current drive set point and provides 10µA*. 10: Internal 3.125KΩ (nominal) resistor forms current drive set point and provides 100µA*. 11: Internal 312.5Ω (nominal) resistor forms current drive set point and provides 1.0mA*. The Op Amp B IRESSEL = 11 setting is not recommended for AV_{DD} < 3V
[3] OPOWER	**Op Amp Power/Speed Select** 0: Low power, 1µA current, 40kHz unity gain bandwidth (nominal values). 1: Normal power, 30µA current, 620kHz unity gain bandwidth (nominal values). Note: This bit is OR'ed with the Op Amp A OPOWER bit such that if either OPOWER bit is set, both op amps will operate with normal power.
[2]	**Reserved** This bit is reserved and must be programmed to 0.
[1] INNSEL	**Negative Input Signal Select** INNSEL is dependent upon the value of MODE **MODE=0** 0: GPIO pin used as Op Amp B negative input, AMPBINN. 1: Op Amp B output, unity gain configuration. **MODE=1** x: Connection to current drive set point resistor selected by IRESSEL.
[0] MODE	**Mode** 0=Normal mode, INNSEL bit selects the Op Amp B negative input. 1=Current drive mode, INNSEL has no effect.

Note: *Assumes the Programmable Reference 1 level is 0.3125V.

24.4. Comparator and Reference System Register Definitions

This section defines the features of the following Comparator and Reference System registers:

Comparator Control Register (CMPCTL) at address F8Fh

Comparator 0 Control 0 Register (CMP0CTL0) at address F90h

Comparator 0 Control 1 Register (CMP0CTL1) at address F91h

Comparator 1 Control 0 Register (CMP1CTL0) at address F92h

Comparator 1 Control 1 Register (CMP1CTL1) at address F93h

Bit	Description (Continued)
[1:0] CSTATUS	**Comparator Status** Status is dependent upon the state of WINEN. **WINEN=0** 0x: Comparator 0 Output (C0OUT) is Low*. 1x: Comparator 0 Output (C0OUT) is High*. x0: Comparator 1 Output (C1OUT) is Low*. x1: Comparator 1 Output (C1OUT) is High*. **WINEN=1** Window state naming is from the perspective of noninverted polarity (POLSEL = 0) for both comparators 00: Inside Window (C0OUT ≠ C1OUT)*. 01: Below Window (C0OUT=C1OUT=0)*. 10: Above Window (C0OUT=C1OUT=1)*. 11: Reserved.

Note: *C0OUT and C1OUT include the effect of POLSEL.

24.4.2. Comparator 0 Control 0 Register

The Comparator 0 Control 0 Register is shown in Table 258.

Table 258. Comparator 0 Control 0 Register (CMP0CTL0)

Bit	7	6	5	4	3	2	1	0
Field	CPOWER		HYST		INNSEL		INPSEL	
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F90h							

Bit	Description
[7:6] CPOWER	**Comparator Power/Speed Select** 00: Ultra-low power, current= 200nA, Tpd=10µs (nominal values). 01: Low power, current=1µA, Tpd=1.5µs (nominal values). 10: Normal, current =4µA, Tpd= 700ns (nominal values). 11: High Speed/Power, current = 27µA, Tpd =150ns (nominal values).
[5:4] HYST	**Hysteresis Level Select** 00: None, 0mV. 01: 15mV (nominal). 10: Reserved. 11: 40mv (nominal).

Figures 104 and 105 provide information about each of the eZ8 CPUinstructions.

Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1.1 **BRK**	2.2 **SRP** IM	2.3 **ADD** r1,r2	2.4 **ADD** r1,Ir2	3.3 **ADD** R2,R1	3.4 **ADD** IR2,R1	3.3 **ADD** R1,IM	3.4 **ADD** IR1,IM	4.3 **ADDX** ER2,ER1	4.3 **ADDX** IM,ER1	2.3 **DJNZ** r1,X	2.2 **JR** cc,X	2.2 **LD** r1,IM	3.2 **JP** cc,DA	1.2 **INC** r1	1.2 **NOP**
1	2.2 **RLC** R1	2.3 **RLC** IR1	2.3 **ADC** r1,r2	2.4 **ADC** r1,Ir2	3.3 **ADC** R2,R1	3.4 **ADC** IR2,R1	3.3 **ADC** R1,IM	3.4 **ADC** IR1,IM	4.3 **ADCX** ER2,ER1	4.3 **ADCX** IM,ER1						See 2nd Op Code Map
2	2.2 **INC** R1	2.3 **INC** IR1	2.3 **SUB** r1,r2	2.4 **SUB** r1,Ir2	3.3 **SUB** R2,R1	3.4 **SUB** IR2,R1	3.3 **SUB** R1,IM	3.4 **SUB** IR1,IM	4.3 **SUBX** ER2,ER1	4.3 **SUBX** IM,ER1						1, 2 **ATM**
3	2.2 **DEC** R1	2.3 **DEC** IR1	2.3 **SBC** r1,r2	2.4 **SBC** r1,Ir2	3.3 **SBC** R2,R1	3.4 **SBC** IR2,R1	3.3 **SBC** R1,IM	3.4 **SBC** IR1,IM	4.3 **SBCX** ER2,ER1	4.3 **SBCX** IM,ER1						
4	2.2 **DA** R1	2.3 **DA** IR1	2.3 **OR** r1,r2	2.4 **OR** r1,Ir2	3.3 **OR** R2,R1	3.4 **OR** IR2,R1	3.3 **OR** R1,IM	3.4 **OR** IR1,IM	4.3 **ORX** ER2,ER1	4.3 **ORX** IM,ER1						
5	2.2 **POP** R1	2.3 **POP** IR1	2.3 **AND** r1,r2	2.4 **AND** r1,Ir2	3.3 **AND** R2,R1	3.4 **AND** IR2,R1	3.3 **AND** R1,IM	3.4 **AND** IR1,IM	4.3 **ANDX** ER2,ER1	4.3 **ANDX** IM,ER1						1.2 **WDT**
6	2.2 **COM** R1	2.3 **COM** IR1	2.3 **TCM** r1,r2	2.4 **TCM** r1,Ir2	3.3 **TCM** R2,R1	3.4 **TCM** IR2,R1	3.3 **TCM** R1,IM	3.4 **TCM** IR1,IM	4.3 **TCMX** ER2,ER1	4.3 **TCMX** IM,ER1						1.2 **STOP**
7	2.2 **PUSH** R2	2.3 **PUSH** IR2	2.3 **TM** r1,r2	2.4 **TM** r1,Ir2	3.3 **TM** R2,R1	3.4 **TM** IR2,R1	3.3 **TM** R1,IM	3.4 **TM** IR1,IM	4.3 **TMX** ER2,ER1	4.3 **TMX** IM,ER1						1.2 **HALT**
8	2.5 **DECW** RR1	2.6 **DECW** IRR1	2.5 **LDE** r1,Irr2	2.9 **LDEI** Ir1,Irr2	3.2 **LDX** r1,ER2	3.3 **LDX** Ir1,ER2	3.4 **LDX** IRR2,R1	3.5 **LDX** IRR2,IR1	3.4 **LDX** r1,Irr2,X	3.4 **LDX** rr1,r2,X						1.2 **DI**
9	2.2 **RL** R1	2.3 **RL** IR1	2.5 **LDE** r2,Irr1	2.9 **LDEI** Ir2,Irr1	3.2 **LDX** r2,ER1	3.3 **LDX** Ir2,ER1	3.4 **LDX** R2,IRR1	3.5 **LDX** IR2,IRR1	3.3 **LEA** r1,r2,X	3.5 **LEA** rr1,rr2,X						1.2 **EI**
A	2.5 **INCW** RR1	2.6 **INCW** IRR1	2.3 **CP** r1,r2	2.4 **CP** r1,Ir2	3.3 **CP** R2,R1	3.4 **CP** IR2,R1	3.3 **CP** R1,IM	3.4 **CP** IR1,IM	4.3 **CPX** ER2,ER1	4.3 **CPX** IM,ER1						1.4 **RET**
B	2.2 **CLR** R1	2.3 **CLR** IR1	2.3 **XOR** r1,r2	2.4 **XOR** r1,Ir2	3.3 **XOR** R2,R1	3.4 **XOR** IR2,R1	3.3 **XOR** R1,IM	3.4 **XOR** IR1,IM	4.3 **XORX** ER2,ER1	4.3 **XORX** IM,ER1						1.5 **IRET**
C	2.2 **RRC** R1	2.3 **RRC** IR1	2.5 **LDC** r1,Irr2	2.9 **LDCI** Ir1,Irr2	2.3 **JP** IRR1	2.9 **LDC** Ir1,Irr2		3.4 **LD** r1,r2,X	3.2 **PUSHX** ER2							1.2 **RCF**
D	2.2 **SRA** R1	2.3 **SRA** IR1	2.5 **LDC** r2,Irr1	2.9 **LDCI** Ir2,Irr1	2.6 **CALL** IRR1	2.2 **BSWAP** R1	3.3 **CALL** DA	3.4 **LD** r2,r1,X	3.2 **POPX** ER1							1.2 **SCF**
E	2.2 **RR** R1	2.3 **RR** IR1	2.2 **BIT** p,b,r1	2.3 **LD** r1,Ir2	3.2 **LD** R2,R1	3.3 **LD** IR2,R1	3.2 **LD** R1,IM	3.3 **LD** IR1,IM	4.2 **LDX** ER2,ER1	4.2 **LDX** IM,ER1						1.2 **CCF**
F	2.2 **SWAP** R1	2.3 **SWAP** IR1	2.6 **TRAP** Vector	2.3 **LD** Ir1,r2	2.8 **MULT** RR1	3.3 **LD** R2,IR1	3.3 **BTJ** p,b,r1,X	3.4 **BTJ** p,b,Ir1,X								

Figure 104. First Op Code Map

33.4.9. Comparator

Table 339 presents electrical and timing data for the F6482 Series' Comparator function.

Table 339. Comparator Electrical Characteristics

		T_A = –40°C to +85°C V_{DD} = 1.8V to 3.6V				
Symbol	**Parameter**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
I_{DD}COMP	Comparator Active Current		0.2		µA	CPOWER=00
			1		µA	CPOWER=01
			4		µA	CPOWER=10
			27		µA	CPOWER=11
V_{ICM}	Input Common Mode Voltage	V_{SS}		V_{DD}	V	
V_{OS}	Input DC Offset		±2	±5	mV	
V_{HYS}	Input Hysteresis		0		mV	HYST=0x
			15		mV	HYST=10
			40		mV	HYST=11
T_{PROP}	Propagation Delay		5		µs	CPOWER=00
			1.5		µs	CPOWER=01
			700		ns	CPOWER=10
			150		ns	CPOWER=11
T_{WAKE}	Time for Wake up		2	5	µs	

Note: *Data in the Typical column is from characterization at 3.0V and 25°C. These values are provided for design guidance only and are not tested in production.

Table 340. Reference System Electrical Characteristics (Continued)

		T_A=–40°C to +85°C V_{DD}=1.8V to 3.6V				
Symbol	**Parameter**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
T_{WAKEF}	Time for Wake up, fixed references		2	5	µs	Upon selection of a fixed reference
T_{WAKEP}	Time for Wake up, VBIAS or programmable internal references			2	ms	Upon enable of VBIAS or Programmable Internal Reference

Notes:
1. Data in the Typical column is from characterization at 3.0V and 25°C. These values are provided for design guidance only and are not tested in production.
2. V_{PREF} is a user-set programmable reference voltage. See Tables 259 and 261 in the Comparators and Reference System chapter on page 484.

33.4.11. Temperature Sensor

Table 341 presents electrical and timing data for the F6482 Series' Temperature Sensor function.

Table 341. Temperature Sensor Electrical Characteristics

		T_A=–40°C to +85°C V_{DD}=1.8 to 3.6V				
Symbol	**Parameter**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
I_{DD}TEMP	Temperature Sensor Active Current			15	µA	
T_{AERR}	Temperature Sensor Output Error	–4		+4	°C	–40°C to +85°C (as measured by ADC)
		–1.5		+1.5	°C	+20°C to +30°C (as measured by ADC)
T_{WAKE}	Time for Wake up		0.9	2	ms	

Note: *Data in the Typical column is from characterization at 3.0V and 25°C. These values are provided for design guidance only and are not tested in production.

33.4.18. Low Frequency Crystal Oscillator

Table 348 presents electrical and timing data for the F6482 Series' Low Frequency Crystal Oscillator function.

Table 348. Low Frequency Oscillator (LFXO) Characteristics

		T_A=–40°C to +85°C V_{DD}=1.8V to 3.6V				
Symbol	**Parameter**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
I_{DD}LFXO	LFXO Active Current		0.13		µA	C_{LOAD}=7 pF
			0.3		µA	C_{LOAD}=12.5 pF
F_{XTAL2}	LFXO Frequency		32.768		kHz	
gm	Oscillator Transconductance	5			µA/V	
T_{WAKE}	Time for Wake up		400	1000	ms	
	SCKOUT Duty Cycle	40	50	60	%	

Note: *Data in the Typical column is from characterization at 3.0V and 25°C. These values are provided for design guidance only and are not tested in production.

33.4.19. Phase-Locked Loop Oscillator

Table 349 presents electrical and timing data for the F6482 Series' Phase-Locked Loop Oscillator (PLL) function.

Table 349. PLL Electrical Characteristics

		V_{DD}=1.8V to 3.6V T_A=–40°C to +85°C				
Symbol	**Parameter**	**Min**	**Typ***	**Max**	**Units**	**Conditions**
I_{DD}PLL	IPO Active Supply Current		2.5		mA	F_{PLLVCO}=96MHz
$F_{PLLCLKIN}$	PLL Input Clock Frequency	0.3125		24	MHz	
F_{PLLVCO}	PLL VCO Frequency	80		384	MHz	
F_{PLLCLK}	PLL Output Clock Frequency	10		48	MHz	
	Duty Cycle of Output	45		55	%	
T_{WAKE}	Time for Wake up		512		Clocks	PLL_{CLKIN} x (RDIV+1)

Note: *Data in the Typical column is from characterization at 3.0V and 25°C. These values are provided for design guidance only and are not tested in production.

signals 21
power-on reset (POR) 40
precharacterization 635
program control instructions 584
program counter 580
program memory 24
programmable gain 6, 472, 475, 478, 480
programming Flash memory 532
protocol configuration, ESPI 289
pseudo-operations, instruction set 577
PUSH 583
using extended addressing 583
PUSHX 583
PWM Mode 181, 182

R

R 579
r 579
RA
register address 579
RCF 582
real-time clock 210, 226
alarm 211
alarm control register 226
alarm day-of-the-week register 224, 226
alarm hours register 223
alarm minutes register 222
alarm seconds register 221
block diagram 210
calendar mode operation 211
control register 228
count register writing 212
counter mode operation 211
counts, synchronous reading of 212
day-of-the-month register 219
day-of-the-week register 216
enable 212
hours register 215
minutes register 214
month register 219
power-on reset 213, 214, 215, 216, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227
recommended operation 212
seconds register 213
source selection 211
timing register 227
year register 220
receive data register, UART-LDD 258
receiver interrupts, UART 251
receiving UART data-interrupt-driven method 235
receiving UART data-polled method 234
reference system
block diagram 486
electrical characteristics and timing 612
operation 489
references, ADC 445
register 120, 579
ADC control 0 451
ADC control 1 452
ADC control 2 453
ADC data high 458
ADC data low 458
ADC input select high 454
ADC input select low 455
ADC offset calibration 457
ADC sample time 459
ADC window lower threshold high 462
ADC window lower threshold low 463
ADC window upper threshold high 460
ADC window upper threshold low 461
AES data 434
baud low and high byte, I2C 332, 334, 339
baud rate high and low byte, ESPI 303
clock control 0 114
clock control 1 116
clock control 2 117
clock control 3 118
clock control 4 119
clock control 6 121
clock control 7 122
clock control 8 122
clock control 9 123
clock control A 123
clock control B 125
clock control C 126
comparator
0 control 0 493