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	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f012ahh020sc

Interrupt Controller

The Z8 Encore! XP® F082A Series products support up to 20 interrupts. These interrupts consist of 8 internal peripheral interrupts and 12 general-purpose I/O pin interrupt sources. The interrupts have three levels of programmable interrupt priority.

Reset Controller

The Z8 Encore! XP F082A Series products can be reset using the $\overline{\text{RESET}}$ pin, Power-On Reset, Watchdog Timer (WDT) time-out, STOP mode exit, or Voltage Brownout (VBO) warning signal. The $\overline{\text{RESET}}$ pin is bi-directional, that is, it functions as reset source as well as a reset indicator.

Table 8. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	Control Registers	eZ8 CPU	Reset Latency (Delay)
System Reset	Reset (as applicable)	Reset	66 Internal Precision Oscillator Cycles
System Reset with Crystal Oscillator Enabled	Reset (as applicable)	Reset	5000 Internal Precision Oscillator Cycles
Stop Mode Recovery	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	66 Internal Precision Oscillator Cycles + IPO startup time
Stop Mode Recovery with Crystal Oscillator Enabled	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	5000 Internal Precision Oscillator Cycles

During a System Reset or Stop Mode Recovery, the Internal Precision Oscillator requires 4 µs to start up. Then the Z8 Encore! XP F082A Series device is held in Reset for 66 cycles of the Internal Precision Oscillator. If the crystal oscillator is enabled in the Flash option bits, this reset period is increased to 5000 IPO cycles. When a reset occurs because of a low voltage condition or Power-On Reset (POR), this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled, except PD0 (or PA2 on 8-pin devices), which is shared with the reset pin. On reset, the PD0 is configured as a bidirectional open-drain reset. The pin is internally driven low during port reset, after which the user code may reconfigure this pin as a general purpose output.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address.

As the control registers are reinitialized by a system reset, the system clock after reset is always the IPO. The software must reconfigure the oscillator control block, such that the correct system clock source is enabled and selected.

PAFS1[7:0]—Port Alternate Function Set 1
0 = Port Alternate Function selected as defined in Table 14 and Table 15 on page 44.
1 = Port Alternate Function selected as defined in Table 14 and Table 15 on page 44.

Port A–D Alternate Function Set 2 Sub-Registers

The Port A–D Alternate Function Set 2 sub-registers (Table 26) is accessed through the Port A–D Control register by writing 8H to the Port A–D Address register. The Alternate Function Set 2 sub-registers selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register is defined in Table 15.

Note: Alternate function selection on port pins must also be enabled as described in Port A–D Alternate Function Sub-Registers on page 47.

Table 26. Port A–D Alternate Function Set 2 Sub-Registers (PxAFS2)

BITS	7	6	5	4	3	2	1	0
FIELD	PAFS27	PAFS26	PAFS25	PAFS24	PAFS23	PAFS22	PAFS21	PAFS20
RESET	00H (all ports of 20/28 pin devices); 04H (Port A of 8-pin device)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	If 08H in Port A–D Address Register, accessible through the Port A–D Control Register							

PAFS2[7:0]—Port Alternate Function Set 2
0 = Port Alternate Function selected as defined in Table 15.
1 = Port Alternate Function selected as defined in Table 15.

Port A–C Input Data Registers

Reading from the Port A–C Input Data registers (Table 27) returns the sampled values from the corresponding ports pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Table 27. Port A–C Input Data Registers (PxIN)

BITS	7	6	5	4	3	2	1	0
FIELD	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	FD2H, FD6H, FDAH							
X = Undefined.								

Table 32. Trap and Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt or Trap Source
Highest	0002H	Reset (not an interrupt)
	0004H	Watchdog Timer (see Watchdog Timer on page 91)
	003AH	Primary Oscillator Fail Trap (not an interrupt)
	003CH	Watchdog Oscillator Fail Trap (not an interrupt)
	0006H	Illegal Instruction Trap (not an interrupt)
	0008H	Reserved
	000AH	Timer 1
	000CH	Timer 0
	000EH	UART 0 receiver
	0010H	UART 0 transmitter
	0012H	Reserved
	0014H	Reserved
	0016H	ADC
	0018H	Port A Pin 7, selectable rising or falling input edge or LVD (see Reset, Stop Mode Recovery, and Low Voltage Detection on page 23)
	001AH	Port A Pin 6, selectable rising or falling input edge or Comparator Output
	001CH	Port A Pin 5, selectable rising or falling input edge
	001EH	Port A Pin 4, selectable rising or falling input edge
	0020H	Port A Pin 3, selectable rising or falling input edge
	0022H	Port A Pin 2, selectable rising or falling input edge
	0024H	Port A Pin 1, selectable rising or falling input edge
	0026H	Port A Pin 0, selectable rising or falling input edge
	0028H	Reserved
	002AH	Reserved
	002CH	Reserved
	002EH	Reserved
	0030H	Port C Pin 3, both input edges
	0032H	Port C Pin 2, both input edges

CAPTURE RESTART mode
0 = Count is captured on the rising edge of the Timer Input signal.
1 = Count is captured on the falling edge of the Timer Input signal.

COMPARATOR COUNTER mode
When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload. Also:

0 = Count is captured on the rising edge of the comparator output.
1 = Count is captured on the falling edge of the comparator output.

Caution: When the Timer Output alternate function TxOUT on a GPIO port pin is enabled, TxOUT changes to whatever state the TPOL bit is. The timer does not need to be enabled for that to happen. Also, the Port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL with the timer enabled and running does not immediately change the TxOUT.

PRES—Prescale value
The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This reset ensures proper clock division each time the Timer is restarted.

000 = Divide by 1
001 = Divide by 2
010 = Divide by 4
011 = Divide by 8
100 = Divide by 16
101 = Divide by 32
110 = Divide by 64
111 = Divide by 128

TMODE—Timer mode
This field along with the TMODEHI bit in TxCTL0 register determines the operating mode of the timer. TMODEHI is the most significant bit of the Timer mode selection value. The entire operating mode bits are expressed as {TMODEHI, TMODE[2:0]}. The TMODEHI is bit 7 of the TxCTL0 register while TMODE[2:0] is the lower 3 bits of the TxCTL1 register.

0000 = ONE-SHOT mode
0001 = CONTINUOUS mode
0010 = COUNTER mode
0011 = PWM SINGLE OUTPUT mode
0100 = CAPTURE mode
0101 = COMPARE mode
0110 = GATED mode
0111 = CAPTURE/COMPARE mode

WDT Reset in Normal Operation

If configured to generate a Reset when a time-out occurs, the Watchdog Timer forces the device into the System Reset state. The WDT status bit in the Reset Status (RSTSTAT) register is set to 1. For more information on system reset, see Reset, Stop Mode Recovery, and Low Voltage Detection on page 23.

WDT Reset in STOP Mode

If configured to generate a Reset when a time-out occurs and the device is in STOP mode, the Watchdog Timer initiates Stop Mode Recovery. Both the WDT status bit and the STOP bit in the Reset Status (RSTSTAT) register are set to 1 following WDT time-out in STOP mode.

Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer Control register address (WDTCTL) unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL register address produce no effect on the bits in the WDTCTL register. The locking mechanism prevents spurious writes to the Reload registers. Follow these steps below to unlock the Watchdog Timer Reload Byte registers (WDTU, WDTH, and WDTL) for write access.

1. Write 55H to the Watchdog Timer Control register (WDTCTL).
2. Write AAH to the Watchdog Timer Control register (WDTCTL).
3. Write the Watchdog Timer Reload Upper Byte register (WDTU) with the desired time-out value.
4. Write the Watchdog Timer Reload High Byte register (WDTH) with the desired time-out value.
5. Write the Watchdog Timer Reload Low Byte register (WDTL) with the desired time-out value.

All three Watchdog Timer Reload registers must be written in the order just listed. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes can occur unless the sequence is restarted. The value in the Watchdog Timer Reload registers is loaded into the counter when the Watchdog Timer is first enabled and every time a WDT instruction is executed.

Watchdog Timer Calibration

Due to its extremely low operating current, the Watchdog Timer oscillator is somewhat inaccurate. This variation can be corrected using the calibration data stored in the Flash Information Page (see Table 97 and Table 98 on page 165). Loading these values into the

Receiving Data using the Interrupt-Driven Method

The UART Receiver interrupt indicates the availability of new data (as well as error conditions). Follow the steps below to configure the UART receiver for interrupt-driven operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the acceptable baud rate.
2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the acceptable priority.
5. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
6. Write to the UART Control 1 Register to enable Multiprocessor (9-bit) mode functions, if appropriate.
   - Set the Multiprocessor Mode Select (MPEN) to Enable MULTIPROCESSOR mode.
   - Set the Multiprocessor Mode Bits, MPMD[1:0] to select the acceptable address matching scheme.
   - Configure the UART to interrupt on received data and errors or errors only (interrupt on errors only is unlikely to be useful for Z8 Encore® devices without a DMA block)
7. Write the device address to the Address Compare Register (automatic MULTIPROCESSOR modes only).
8. Write to the UART Control 0 register to:
   - Set the receive enable bit (REN) to enable the UART for data reception
   - Enable parity, if appropriate and if multiprocessor mode is not enabled, and select either even or odd parity.
9. Execute an EI instruction to enable interrupts.

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

1. Checks the UART Status 0 register to determine the source of the interrupt - error, break, or received data.
2. Reads the data from the UART Receive Data register if the interrupt was because of data available. If operating in MULTIPROCESSOR (9-bit) mode, further actions may be required depending on the MULTIPROCESSOR mode bits MPMD[1:0].

send. This action provides 7 bit periods of latency to load the Transmit Data register before the Transmit shift register completes shifting the current character. Writing to the UART Transmit Data register clears the TDRE bit to 0.

Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte is received and is available in the UART Receive Data register. This interrupt can be disabled independently of the other receiver interrupt sources. The received data interrupt occurs after the receive character has been received and placed in the Receive Data register. To avoid an overrun error, software must respond to this received data available condition before the next character is completely received.

Note: In MULTIPROCESSOR mode (*MPEN = 1*), the receive data interrupts are dependent on the multiprocessor configuration and the most recent address byte.

- A break is received.
- An overrun is detected.
- A data framing error is detected.

UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the UART Status 0 register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data register contains a data byte. However, because the overrun error occurred, this byte may not contain valid data and must be ignored. The BRKD bit indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data register must be read again to clear the error bits in the UART Status 0 register. Updates to the Receive Data register occur only when the next data word is received.

UART Data and Error Handling Procedure

Figure 15 displays the recommended procedure for use in UART receiver interrupt service routines.

Endec, and passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART's Baud Rate Generator and supports IrDA standard baud rates from 9600 baud to 115.2 kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the Infrared Endec. The Infrared Endec data rate is calculated using the following equation:

$$\text{Infrared Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16 clocks wide. If the data bit to be transmitted is 1, the IR_TXD signal remains low for the full 16 clock period. If the data bit to be transmitted is 0, the transmitter first outputs a 7 clock low period, followed by a 3 clock high pulse. Finally, a 6 clock low pulse is output to complete the full 16 clock data period. Figure 17 displays IrDA data transmission. When the Infrared Endec is enabled, the UART's TXD signal is internal to the Z8 Encore! XP® F082A Series products while the IR_TXD signal is output through the TXD pin.

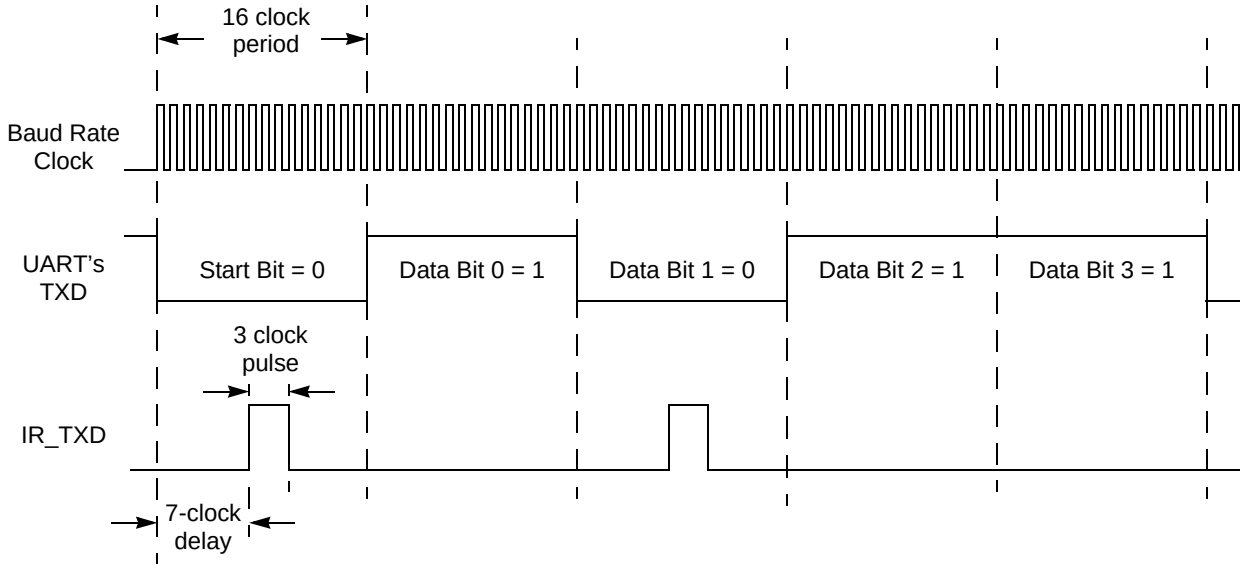

Figure 17. Infrared Data Transmission

baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window, the process is repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal, allowing the Endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined in Universal Asynchronous Receiver/Transmitter on page 97.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.

3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
   - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
   - Set CONT to 1 to select continuous conversion.
   - If the internal VREF must be output to a pin, set the VREFEXT bit to 1. The internal voltage reference must be enabled in this case.
   - Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or disable the internal reference. The REFSELH bit is contained in ADC Control/Status Register 1.
   - Set CEN to 1 to start the conversions.
4. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
   - CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation.
   - An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete.
5. The ADC writes a new data result every 256 system clock cycles. For each completed conversion, the ADC control logic performs the following operations:
   - Writes the 13-bit two's complement result to {ADCD_H[7:0], ADCD_L[7:3]}.
   - Sends an interrupt request to the Interrupt Controller denoting conversion complete.
6. To disable continuous conversion, clear the CONT bit in the ADC Control Register to 0.

Interrupts

The ADC is able to interrupt the CPU when a conversion has been completed. When the ADC is disabled, no new interrupts are asserted; however, an interrupt pending when the ADC is disabled is not cleared.

Calibration and Compensation

The Z8 Encore! XP® F082A Series ADC is factory calibrated for offset error and gain error, with the compensation data stored in Flash memory. Alternately, you can perform your own calibration, storing the values in Flash themselves. Thirdly, the user code can perform a manual offset calibration during DIFFERENTIAL mode operation.

Factory Calibration

Devices that have been factory calibrated contain 30 bytes of calibration data in the Flash option bit space. This data consists of 3 bytes for each input mode, one for offset and two for gain correction. For a list of input modes for which calibration data exists, see Zilog Calibration Data on page 161.

User Calibration

If you have precision references available, your own external calibration can be performed using any input modes. This calibration data takes into account buffer offset and non-linearity, so it is recommended that this calibration be performed separately for each of the ADC input modes planned for use.

Manual Offset Calibration

When uncalibrated, the ADC has significant offset (see Table 135 on page 231). Subsequently, manual offset calibration capability is built into the block. When the ADC Control Register 0 sets the input mode (ANAIN[2:0]) to MANUAL OFFSET CALIBRATION mode, the differential inputs to the ADC are shorted together by an internal switch. Reading the ADC value at this input produces 0 in an ideal system. The value actually read is the ADC offset. This value can be stored in non-volatile memory (see Non-Volatile Data Storage on page 169) and accessed by user code to compensate for the input offset error. There is no provision for manual gain calibration.

Software Compensation Procedure Using Factory Calibration Data

The value read from the ADC high and low byte registers is uncompensated. The user mode software must apply gain and offset correction to this uncompensated value for maximum accuracy. The following equation yields the compensated value:

$$ADC_{comp} = (ADC_{uncomp} - OFFCAL) + ((ADC_{uncomp} - OFFCAL) \times GAINCAL) / 2^{16}$$

where GAINCAL is the gain calibration value, OFFCAL is the offset calibration value and ADC_{uncomp} is the uncompensated value read from the ADC. All values are in two's complement format.

Note: The offset compensation is performed first, followed by the gain compensation. One bit of resolution is lost because of rounding in both the offset and gain computations. As a result the ADC registers read back 13:1 sign bit, two calibration bits lost to rounding and 10 data bits.

Also note that in the second term, the multiplication must be performed before the division by 2^{16}. Otherwise, the second term incorrectly evaluates to zero.

Compensation Steps:

1. Correct for Offset

ADC MSB	ADC LSB

-

Offset MSB	Offset LSB

=

#1 MSB	#1 LSB

2. Take absolute value of the offset corrected ADC value if negative—the gain correction factor is computed assuming positive numbers, with sign restoration afterward.

#2 MSB	#2 LSB

Also take absolute value of the gain correction word if negative

AGain MSB	AGain LSB

3. Multiply by Gain Correction Word. If in DIFFERENTIAL mode, there are two gain correction values: one for positive ADC values, another for negative ADC values. Based on the sign of #2, use the appropriate Gain Correction Word.

#2 MSB	#2 LSB

*

AGain MSB	AGain LSB

=

#3	#3	#3	#3

4. Round the result and discard the least significant two bytes (this is equivalent to dividing by 2^{16}).

#3	#3	#3	#3

-

0x00	0x00	0x80	0x00

=

#4 MSB	#4 LSB

5. Determine sign of the gain correction factor using the sign bits from Step 2. If the offset corrected ADC value AND the gain correction word have the same sign, then the factor is positive and is left unchanged. If they have differing signs, then the factor is negative and must be multiplied by -1.

ANAIN[3:0]—Analog Input Select
These bits select the analog input for conversion. Not all Port pins in this list are available in all packages for the Z8 Encore! XP ® F082A Series. For information on port pins available with each package style, see Pin Description on page 9. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1.

For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined.

SINGLE-ENDED:

0000 = ANA0 (transimpedance amp output when enabled)
0001 = ANA1 (transimpedance amp inverting input)
0010 = ANA2 (transimpedance amp non-inverting input)
0011 = ANA3
0100 = ANA4
0101 = ANA5
0110 = ANA6
0111 = ANA7
1000 = Reserved
1001 = Reserved
1010 = Reserved
1011 = Reserved
1100 = Hold transimpedance input nodes (ANA1 and ANA2) to ground.
1101 = Reserved
1110 = Temperature Sensor.
1111 = Reserved.

DIFFERENTIAL (non-inverting input and inverting input respectively):

0000 = ANA0 and ANA1
0001 = ANA2 and ANA3
0010 = ANA4 and ANA5
0011 = ANA1 and ANA0
0100 = ANA3 and ANA2
0101 = ANA5 and ANA4
0110 = ANA6 and ANA5
0111 = ANA0 and ANA2
1000 = ANA0 and ANA3
1001 = ANA0 and ANA4
1010 = ANA0 and ANA5
1011 = Reserved
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Manual Offset Calibration Mode

Table 94. ADC Calibration Data Location

Info Page Address	Memory Address	Compensation Usage	ADC Mode	Reference Type
60	FE60	Offset	Single-Ended Unbuffered	Internal 2.0 V
08	FE08	Gain High Byte	Single-Ended Unbuffered	Internal 2.0 V
09	FE09	Gain Low Byte	Single-Ended Unbuffered	Internal 2.0 V
63	FE63	Offset	Single-Ended Unbuffered	Internal 1.0 V
0A	FE0A	Gain High Byte	Single-Ended Unbuffered	Internal 1.0 V
0B	FE0B	Gain Low Byte	Single-Ended Unbuffered	Internal 1.0 V
66	FE66	Offset	Single-Ended Unbuffered	External 2.0 V
0C	FE0C	Gain High Byte	Single-Ended Unbuffered	External 2.0 V
0D	FE0D	Gain Low Byte	Single-Ended Unbuffered	External 2.0 V
69	FE69	Offset	Single-Ended 1x Buffered	Internal 2.0 V
0E	FE0E	Gain High Byte	Single-Ended 1x Buffered	Internal 2.0 V
0F	FE0F	Gain Low Byte	Single-Ended 1x Buffered	Internal 2.0 V
6C	FE6C	Offset	Single-Ended 1x Buffered	External 2.0 V
10	FE10	Gain High Byte	Single-Ended 1x Buffered	External 2.0 V
11	FE11	Gain Low Byte	Single-Ended 1x Buffered	External 2.0 V
6F	FE6F	Offset	Differential Unbuffered	Internal 2.0 V
12	FE12	Positive Gain High Byte	Differential Unbuffered	Internal 2.0 V
13	FE13	Positive Gain Low Byte	Differential Unbuffered	Internal 2.0 V
30	FE30	Negative Gain High Byte	Differential Unbuffered	Internal 2.0 V
31	FE31	Negative Gain Low Byte	Differential Unbuffered	Internal 2.0 V
72	FE72	Offset	Differential Unbuffered	Internal 1.0 V
14	FE14	Positive Gain High Byte	Differential Unbuffered	Internal 1.0 V
15	FE15	Positive Gain Low Byte	Differential Unbuffered	Internal 1.0 V
32	FE32	Negative Gain High Byte	Differential Unbuffered	Internal 1.0 V
33	FE33	Negative Gain Low Byte	Differential Unbuffered	Internal 1.0 V
75	FE75	Offset	Differential Unbuffered	External 2.0 V
16	FE16	Positive Gain High Byte	Differential Unbuffered	External 2.0 V
17	FE17	Positive Gain Low Byte	Differential Unbuffered	External 2.0 V

resides in working register R0. The bit definitions of this status byte are defined in Table 103. The contents of the status byte are undefined for write operations to illegal addresses. Also, user code must pop the address and data bytes off the stack.

The write routine uses 13 bytes of stack space in addition to the two bytes of address and data pushed by the user. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS writes exhibit a non-uniform execution time. In general, a write takes 251 µs (assuming a 20 MHz system clock). Every 400 to 500 writes, however, a maintenance operation is necessary. In this rare occurrence, the write takes up to 61 ms to complete. Slower system clock speeds result in proportionally higher execution times.

NVDS byte writes to invalid addresses (those exceeding the NVDS array size) have no effect. Illegal write operations have a 12 µs execution time.

Table 103. Write Status Byte

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				RCPY	PF	AWE	DWE
DEFAULT VALUE	0	0	0	0	0	0	0	0

Reserved—Must be 0.

RCPY—Recopy Subroutine Executed
A recopy subroutine was executed. These operations take significantly longer than a normal write operation.

PF—Power Failure Indicator
A power failure or system reset occurred during the most recent attempted write to the NVDS array.

AW—Address Write Error
An address byte failure occurred during the most recent attempted write to the NVDS array.

DWE—Data Write Error
A data byte failure occurred during the most recent attempted write to the NVDS array.

Byte Read

To read a byte from the NVDS array, user code must first push the address onto the stack. User code issues a CALL instruction to the address of the byte-read routine (0x1000). At the return from the sub-routine, the read byte resides in working register R0, and the read status byte resides in working register R1. The contents of the status byte are undefined for

High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

OCD Unlock Sequence (8-Pin Devices Only)

Because of pin-sharing on the 8-pin device, an unlock sequence must be performed to access the DBG pin. If this sequence is not completed during a system reset, then the PA0/DBG pin functions only as a GPIO pin.

The following sequence unlocks the DBG pin:

1. Hold PA2/$\overline{\text{RESET}}$ Low.
2. Wait 5ms for the internal reset sequence to complete.
3. Send the following bytes serially to the debug pin:

```
DBG ← 80H (autobaud)
DBG ← EBH
DBG ← 5AH
DBG ← 70H
DBG ← CDH (32-bit unlock key)
```

4. Release PA2/$\overline{\text{RESET}}$. The PA0/DBG pin is now identical in function to that of the DBG pin on the 20-/28-pin device. To enter DEBUG mode, re-autobaud and write 80H to the OCD control register (see On-Chip Debugger Commands on page 179).

Caution: Between Step 3 and Step 4, there is an interval during which the 8-pin device is neither in RESET nor DEBUG mode. If a device has been erased or has not yet been programmed, all program memory bytes contain FFH. The CPU interprets this as an illegal instruction, so some irregular behavior can occur before entering DEBUG mode, and the register values after entering DEBUG mode differs from their specified reset values. However, none of these irregularities prevent programming the Flash memory. Before beginning system debug, it is recommended that some legal code be programmed into the 8-pin device, and that a RESET occurs.

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If Breakpoints are enabled, the OCD enters DEBUG mode and halts the eZ8 CPU. If Breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

- **Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1–256 bytes at a time (256 bytes can be read by setting size to 0). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Program Memory (0BH)**—The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1–65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

- **Write Data Memory (0CH)**—The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
```

Table 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
COM dst	dst ← ~dst	R		60	–	*	*	0	–	–	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	–	–	2	3
		r	Ir	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	–	–	3	3
		r	Ir	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	–	–	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	–	–	4	3
		ER	IM	A9							4	3
DA dst	dst ← DA(dst)	R		40	*	*	*	X	–	–	2	2
		IR		41							2	3
DEC dst	dst ← dst - 1	R		30	–	*	*	*	–	–	2	2
		IR		31							2	3
DECW dst	dst ← dst - 1	RR		80	–	*	*	*	–	–	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	–	–	–	–	–	–	1	2
DJNZ dst, RA	dst ← dst – 1 if dst ≠ 0 PC ← PC + X	r		0A-FA	–	–	–	–	–	–	2	3
EI	IRQCTL[7] ← 1			9F	–	–	–	–	–	–	1	2
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							