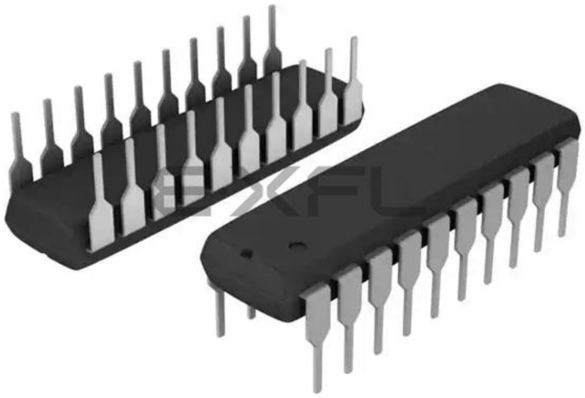

Welcome to E-XFL.COM

## 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 | 2KB (2K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 64 x 8 |
| RAM Size | 512 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 | Through Hole |
| Package / Case | 20-DIP (0.300", 7.62mm) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f022aph020sg |

# Pin Description

The Z8 Encore! XP F082A Series products are available in a variety of packages styles and pin configurations. This chapter describes the signals and available pin configurations for each of the package styles. For information about physical package specifications, see the Packaging chapter on page 245.

## Available Packages

The following package styles are available for each device in the Z8 Encore! XP F082A Series product line:

- SOIC: 8-, 20- and 28-pin
- PDIP: 8-, 20- and 28-pin
- SSOP: 20- and 28- pin
- QFN 8-pin (MLF-S, a QFN-style package with an 8-pin SOIC footprint)

In addition, the Z8 Encore! XP F082A Series devices are available both with and without advanced analog capability (ADC, temperature sensor and op amp). Devices Z8F082A, Z8F042A, Z8F022A and Z8F012A contain the advanced analog, while devices Z8F081A, Z8F041A, Z8F021A and Z8F011A do not have the advanced analog capability.

## Pin Configurations

Figure 2 through Figure 4 display the pin configurations for all the packages available in the Z8 Encore! XP F082A Series. See Table 2 on page 10 for a description of the signals. The analog input alternate functions (ANA*x*) are not available on the Z8F081A, Z8F041A, Z8F021A and Z8F011A devices. The analog supply pins ($AV_{DD}$ and $AV_{SS}$) are also not available on these parts and are replaced by PB6 and PB7.

At reset, all Port A, B and C pins default to an input state. In addition, any alternate functionality is not enabled, so the pins function as general purpose input ports until programmed otherwise. At powerup, the PD0 pin defaults to the $\overline{\text{RESET}}$ alternate function.

The pin configurations listed are preliminary and subject to change based on manufacturing limitations.

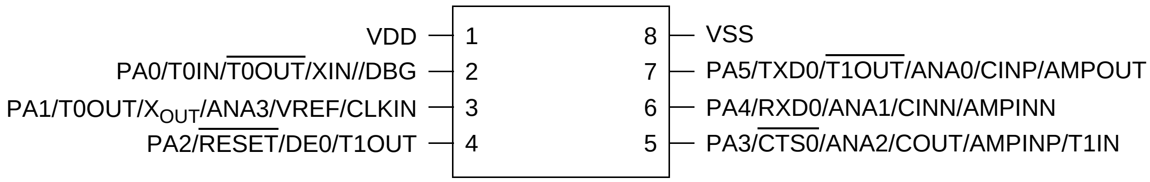


**Figure 2. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 8-Pin SOIC, QFN/MLF-S, or PDIP Package**

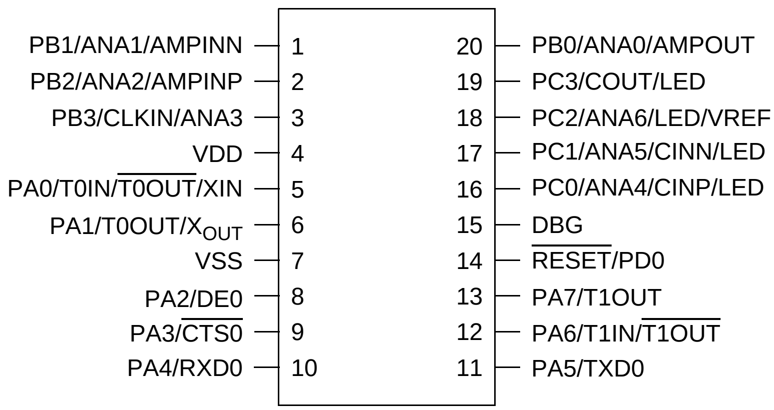


**Figure 3. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 20-Pin SOIC, SSOP or PDIP Package**

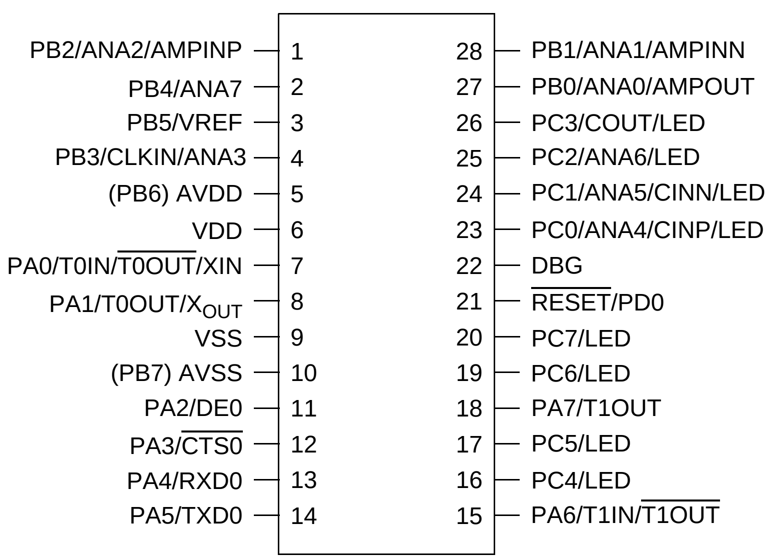


**Figure 4. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 28-Pin SOIC, SSOP or PDIP Package**

# Reset, Stop Mode Recovery and Low Voltage Detection

The Reset Controller within the Z8 Encore! XP F082A Series controls Reset and Stop Mode Recovery operation and provides indication of low supply voltage conditions. In typical operation, the following events cause a Reset:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO)
- Watchdog Timer time-out (when configured by the WDT_RES Flash option bit to initiate a reset)
- External $\overline{\text{RESET}}$ pin assertion (when the alternate RESET function is enabled by the GPIO Register)
- On-chip debugger initiated Reset (OCDCTL[0] set to 1)

When the device is in STOP Mode, a Stop Mode Recovery is initiated by either of the following occurrences:

- Watchdog Timer time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source

The low voltage detection circuitry on the device (available on the 8-pin product versions only) performs the following functions:

- Generates the VBO reset when the supply voltage drops below a minimum safe level.
- Generates an interrupt when the supply voltage drops below a user-defined level (8-pin devices only).

## Reset Types

The Z8 Encore! XP F082A Series provides several different types of Reset operation. Stop Mode Recovery is considered as a form of Reset. Table 8 lists the types of Reset and their operating characteristics. The System Reset is longer if the external crystal oscillator is enabled by the Flash option bits, allowing additional time for oscillator start-up.

operational amplifier (LPO) is OFF. To use the LPO, clear the LPO bit, turning it ON. Clearing this bit might interfere with normal ADC measurements on ANA0 (the LPO output). This bit enables the amplifier even in STOP Mode. If the amplifier is not required in STOP Mode, disable it. Failure to perform this results in STOP Mode currents greater than specified.

**Note:** This register is only reset during a POR sequence. Other system reset events do not affect it.

**Table 13. Power Control Register 0 (PWRCTL0)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | LPO | Reserved | | VBO | TEMP | ADC | COMP | Reserved |
| RESET | 1 | 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 | F80H | | | | | | | |

| Bit | Description |
|---|---|
| [7]<br>LPO | **Low-Power Operational Amplifier Disable**<br>0 = LPO is enabled (this applies even in STOP Mode).<br>1 = LPO is disabled. |
| [6:5] | **Reserved**<br>These bits are reserved and must be programmed to 00. |
| [4]<br>VBO | **Voltage Brown-Out Detector Disable**<br>This bit and the VBO_AO Flash option bit must both enable the VBO for the VBO to be active.<br>0 = VBO enabled.<br>1 = VBO disabled. |
| [3]<br>TEMP | **Temperature Sensor Disable**<br>0 = Temperature Sensor enabled.<br>1 = Temperature Sensor disabled. |
| [2]<br>ADC | **Analog-to-Digital Converter Disable**<br>0 = Analog-to-Digital Converter enabled.<br>1 = Analog-to-Digital Converter disabled. |
| [1]<br>COMP | **Comparator Disable**<br>0 = Comparator is enabled.<br>1 = Comparator is disabled. |
| [0] | **Reserved**<br>This bit is reserved and must be programmed to 0. |

**Table 44. IRQ2 Enable and Priority Encoding**

| IRQ2ENH[*x*] | IRQ2ENL[*x*] | Priority | Description |
|---|---|---|---|
| 0 | 0 | Disabled | Disabled |
| 0 | 1 | Level 1 | Low |
| 1 | 0 | Level 2 | Medium |
| 1 | 1 | Level 3 | High |

Note: *x* indicates register bits 0–7.

**Table 45. IRQ2 Enable High Bit Register (IRQ2ENH)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | Reserved | | | | C3ENH | C2ENH | C1ENH | C0ENH |
| 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 | FC7H | | | | | | | |

| Bit | Description |
|---|---|
| [7:4] | **Reserved**<br>These bits are reserved and must be programmed to 0000. |
| [3]<br>C3ENH | **Port C3 Interrupt Request Enable High Bit** |
| [2]<br>C2ENH | **Port C2 Interrupt Request Enable High Bit** |
| [1]<br>C1ENH | **Port C1 Interrupt Request Enable High Bit** |
| [0]<br>C0ENH | **Port C0 Interrupt Request Enable High Bit** |

# Watchdog Timer

The Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults and other system-level problems which may place the Z8 Encore! XP F082A Series devices into unsuitable operating states. The features of Watchdog Timer include:

- On-chip RC oscillator
- A selectable time-out response: reset or interrupt
- 24-bit programmable time-out value

## Operation

The Watchdog Timer is a one-shot timer that resets or interrupts the Z8 Encore! XP F082A Series devices when the WDT reaches its terminal count. The Watchdog Timer uses a dedicated on-chip RC oscillator as its clock source. The Watchdog Timer operates in only two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash option bit. The WDT_AO bit forces the Watchdog Timer to operate immediately upon reset, even if a WDT instruction has not been executed.

The Watchdog Timer is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is described by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10 kHz. The Watchdog Timer cannot be refreshed after it reaches `000002H`. The WDT reload value must not be set to values below `000004H`. Table 58 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

**Table 58. Watchdog Timer Approximate Time-Out Delays**

| WDT Reload Value (Hex) | WDT Reload Value (Decimal) | Approximate Time-Out Delay (with 10 kHz typical WDT oscillator frequency) | |
|---|---|---|---|
| | | Typical | Description |
| 000004 | 4 | 400 μs | Minimum time-out delay |
| FFFFFF | 16,777,215 | 28 minutes | Maximum time-out delay |

into the Watchdog Timer Reload registers results in a one-second time-out at room temperature and 3.3V supply voltage. Time-outs other than one second may be obtained by scaling the calibration values up or down as required.

**Note:** The Watchdog Timer accuracy still degrades as temperature and supply voltage vary. See Table 137 on page 235 for details.

# Watchdog Timer Control Register Definitions

This section defines the features of the following Watchdog Timer Control registers.

Watchdog Timer Control Register (WDTCTL): see page 96

Watchdog Timer Reload Upper Byte Register (WDTU): see page 97

Watchdog Timer Reload High Byte Register (WDTH): see page 97

Watchdog Timer Reload Low Byte Register (WDTL): see page 98

## Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) Register is a write-only control register. Writing the `55H`, `AAH` unlock sequence to the WDTCTL Register address 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. This register address is shared with the read-only Reset Status Register.

**Table 59. Watchdog Timer Control Register (WDTCTL)**

| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| **Field** | WDTUNLK | | | | | | | |
| **RESET** | X | X | X | X | X | X | X | X |
| **R/W** | W | W | W | W | W | W | W | W |
| **Address** | FF0H | | | | | | | |
| Note: X = Undefined. | | | | | | | | |

| **Bit** | **Description** |
|---|---|
| [7:0]<br>WDTUNLK | **Watchdog Timer Unlock**<br>The software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers. |

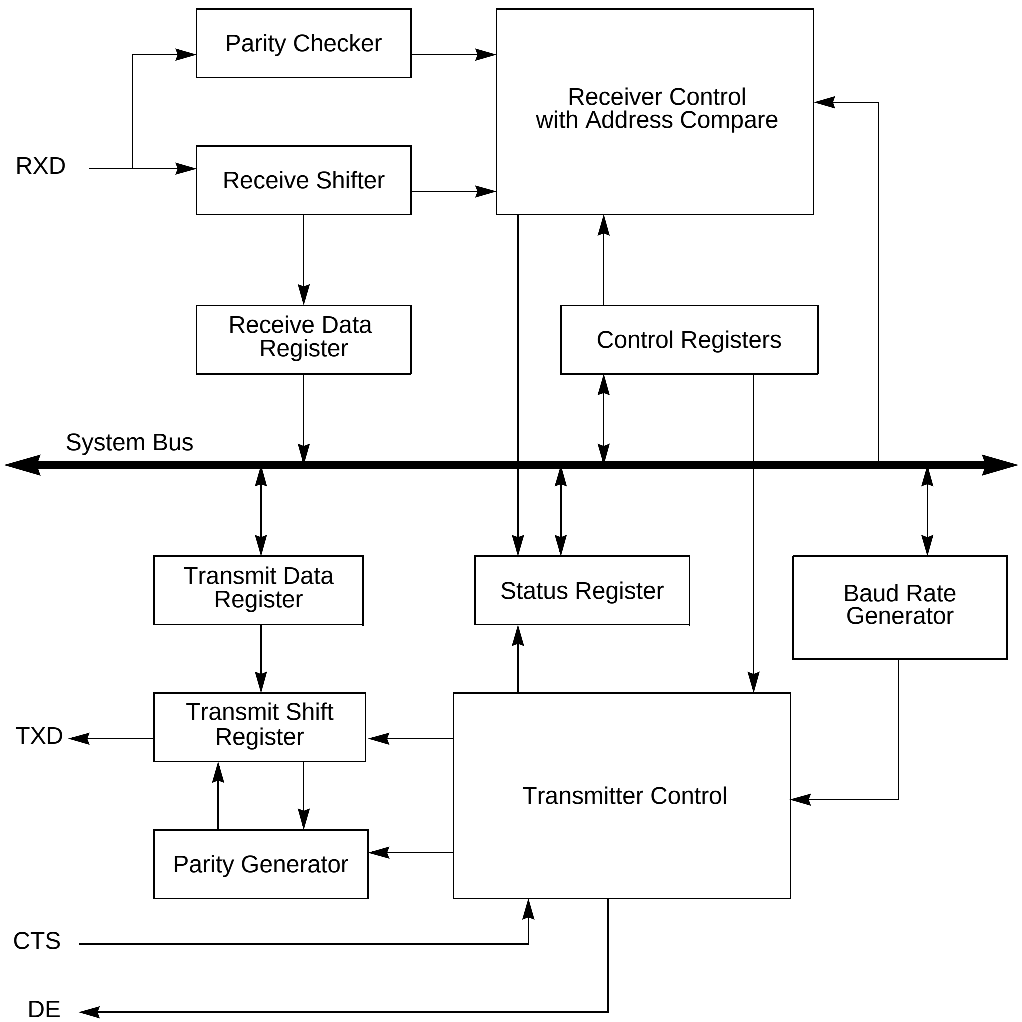


**Figure 10. UART Block Diagram**

# Operation

The UART always transmits and receives data in an 8-bit data format, least-significant bit first. An even or odd parity bit can be added to the data stream. Each character begins with an active Low start bit and ends with either 1 or 2 active High stop bits. Figures 11 and 12 display the asynchronous data format employed by the UART without parity and with parity, respectively.

   – Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin

6. Check the `TDRE` bit in the UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If empty, continue to Step 7. If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.
7. Write the UART Control 1 Register to select the outgoing address bit.
8. Set the Multiprocessor Bit Transmitter (`MPBT`) if sending an address byte, clear it if sending a data byte.
9. Write the data byte to the UART Transmit Data Register. The transmitter automatically transfers the data to the Transmit Shift Register and transmits the data.
10. Make any changes to the Multiprocessor Bit Transmitter (`MPBT`) value, if appropriate and MULTIPROCESSOR Mode is enabled.
11. To transmit additional bytes, return to Step 5.

## Transmitting Data using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Observe the following steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate 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 Transmitter interrupt and set the acceptable priority.
5. Write to the UART Control 1 Register to enable MULTIPROCESSOR (9-bit) Mode functions, if MULTIPROCESSOR Mode is appropriate.
6. Set the MULTIPROCESSOR Mode Select (`MPEN`) to Enable MULTIPROCESSOR Mode.
7. Write to the UART Control 0 Register to:
   – Set the transmit enable bit (`TEN`) to enable the UART for data transmission
   – Enable parity, if appropriate and if MULTIPROCESSOR Mode is not enabled and select either even or odd parity

**Table 67. UART Transmit Data Register (U0TXD)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | TXD | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | W | W | W | W | W | W | W | W |
| Address | F40H | | | | | | | |
| Note: X = Undefined. | | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>TXD | **Transmit Data**<br>UART transmitter data byte to be shifted out through the TXD*x* pin. |

## UART Receive Data Register

Data bytes received through the RXD*x* pin are stored in the UART Receive Data (UxRXD) Register, shown in Table 68. The read-only UART Receive Data Register shares a Register File address with the Write-only UART Transmit Data Register.

**Table 68. UART Receive Data Register (U0RXD)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | RXD | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R | R | R | R | R | R | R | R |
| Address | F40H | | | | | | | |
| Note: X = Undefined. | | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>RXD | **Receive Data**<br>UART receiver data byte from the RXD*x* pin. |

## UART Address Compare Register

The UART Address Compare (UxADDR) Register stores the multi-node network address of the UART (see Table 69). When the MPMD[1] bit of UART Control Register 0 is set, all incoming address bytes are compared to the value stored in the Address Compare Register. Receive interrupts and RDA assertions only occur in the event of a match.

# Infrared Encoder/Decoder

Z8 Encore! XP F082A Series products contain a fully-functional, high-performance UART to Infrared Encoder/Decoder (endec). The infrared endec is integrated with an on-chip UART to allow easy communication between the Z8 Encore! XP MCU and IrDA Physical Layer Specification, Version 1.3-compliant infrared transceivers. Infrared communication provides secure, reliable, low-cost, point-to-point communication between PCs, PDAs, cell phones, printers and other infrared enabled devices.

## Architecture

Figure 16 displays the architecture of the infrared endec.

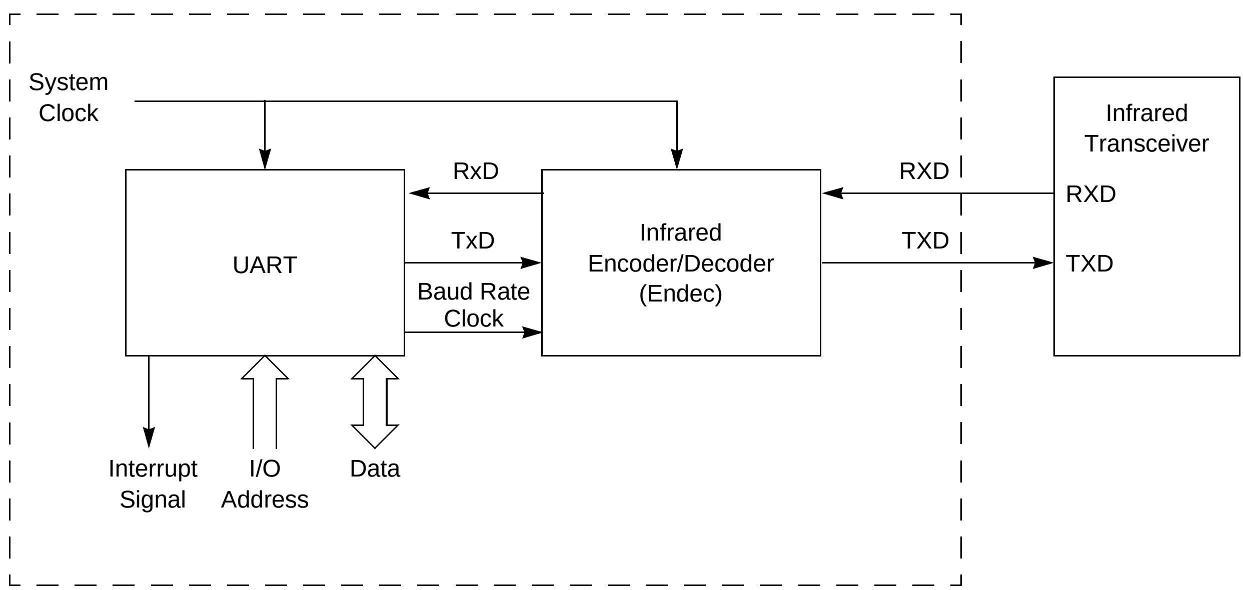


**Figure 16. Infrared Data Communication System Block Diagram**

## Operation

When the infrared endec is enabled, the transmit data from the associated on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver through the TXD pin. Likewise, data received from the infrared transceiver is passed to the infrared endec through the RXD pin, decoded by the infrared endec and passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

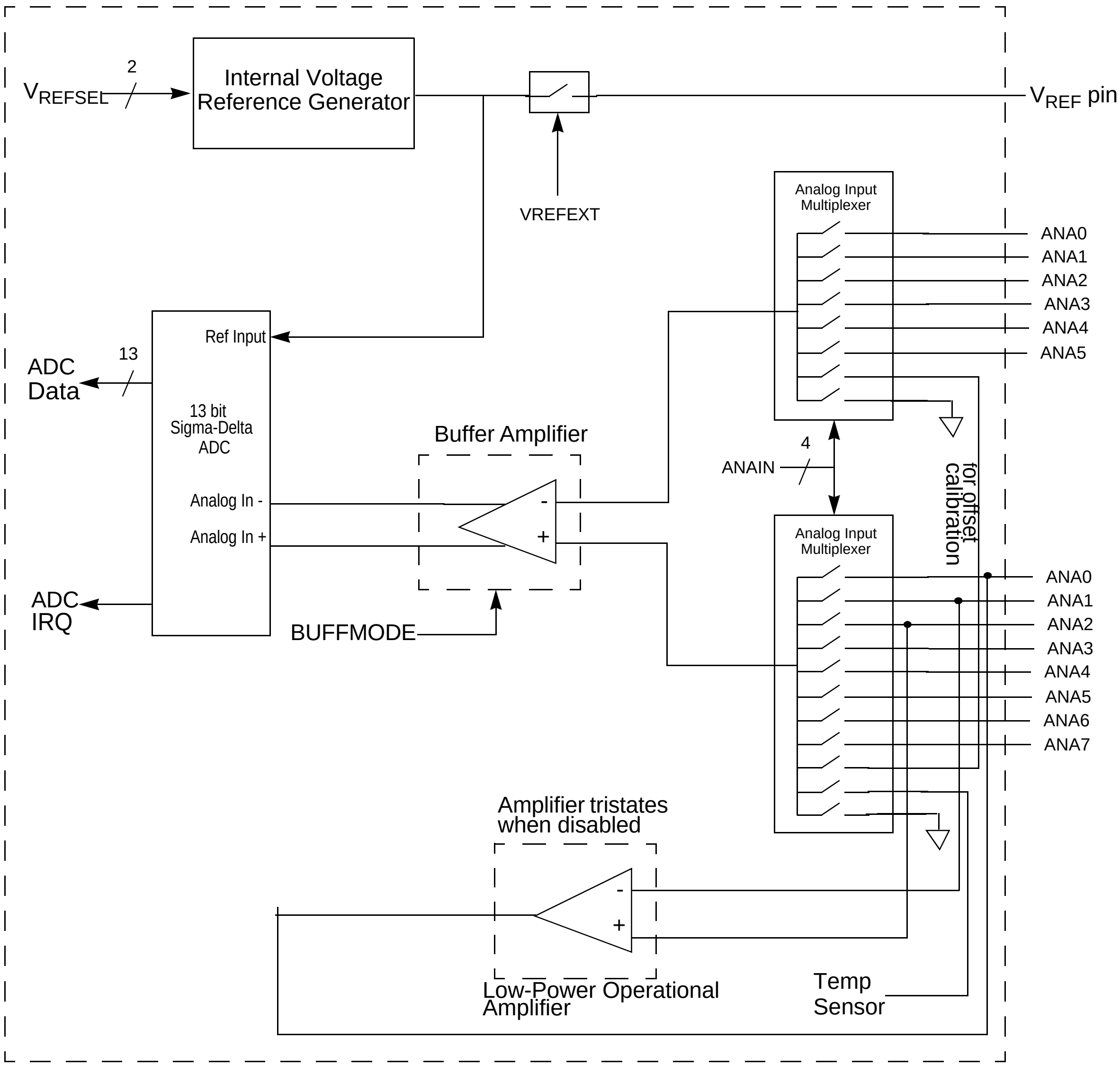


**Figure 19. Analog-to-Digital Converter Block Diagram**

# Operation

In both SINGLE-ENDED and DIFFERENTIAL modes, the effective output of the ADC is an 11-bit, signed, two's complement digital value. In DIFFERENTIAL Mode, the ADC can output values across the entire 11-bit range, from –1024 to +1023. In SINGLE-ENDED Mode, the output generally ranges from 0 to +1023, but offset errors can cause small negative values.

## Trim Bit Address 0004H

**Table 95. Trim Option Bits at 0004H**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Field** | Reserved | | | | | | | |
| **RESET** | U | U | U | U | U | U | U | U |
| **R/W** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Address** | Information Page Memory 0024H | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|---|---|
| [7:0] | **Reserved**<br>These bits are reserved; altering this register may result in incorrect device operation. |

# Zilog Calibration Data

This section briefly describes the features of the following Flash option bit calibration registers.

ADC Calibration Data: see page 169

Temperature Sensor Calibration Data: see page 171

Watchdog Timer Calibration Data: see page 172

Serialization Data: see page 173

Randomized Lot Identifier: see page 174

## Byte Write

To write a byte to the NVDS array, the user code must first push the address, then the data byte onto the stack. The user code issues a `CALL` instruction to the address of the byte-write routine (0x10B3). At the return from the sub-routine, the write status byte resides in working register R0. The bit fields of this status byte are defined in Table 106. 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 nonuniform 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 2 µs execution time.

**Table 106. Write Status Byte**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Field** | Reserved | | | | RCPY | PF | AWE | DWE |
| **Default Value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Description |
|---|---|
| [7:4] | **Reserved**<br>These bits are reserved and must be programmed to 0000. |
| [3]<br>RCPY | **Recopy Subroutine Executed**<br>A recopy subroutine was executed. These operations take significantly longer than a normal write operation. |
| [2]<br>PF | **Power Failure Indicator**<br>A power failure or system reset occurred during the most recent attempted write to the NVDS array. |
| [1]<br>AWE | **Address Write Error**<br>An address byte failure occurred during the most recent attempted write to the NVDS array. |
| [0]<br>DWE | **Data Write Error**<br>A data byte failure occurred during the most recent attempted write to the NVDS array. |

**Table 107. NVDS Read Time**

| Operation | Minimum Latency | Maximum Latency |
|---|---|---|
| Read (16 byte array) | 875 | 9961 |
| Read (64 byte array) | 876 | 8952 |
| Read (128 byte array) | 883 | 7609 |
| Write (16 byte array) | 4973 | 5009 |
| Write (64 byte array) | 4971 | 5013 |
| Write (128 byte array) | 4984 | 5023 |
| Illegal Read | 43 | 43 |
| Illegal Write | 31 | 31 |

If NVDS read performance is critical to your software architecture, you can optimize your code for speed. Try the first suggestion below before attempting the second.

1. Periodically refresh all addresses that are used. The optimal use of NVDS in terms of speed is to rotate the writes evenly among all addresses planned to use, bringing all reads closer to the minimum read time. Because the minimum read time is much less than the write time, however, actual speed benefits are not always realized.

2. Use as few unique addresses as possible to optimize the impact of refreshing, plus minimize the requirement for it.

Serial Break leaves the device in DEBUG Mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns 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, reautobaud and write `80H` to the OCD Control Register (see the On-Chip Debugger Commands section on page 186).

---

**! Caution:** Between Steps 3 and 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 value as an illegal instruction; therefore some irregular behavior can occur before entering DEBUG Mode, and the register values after entering DEBUG Mode will differ from their specified reset values. However, none of these irregularities prevent the programming of Flash memory. Before beginning system debug, Zilog recommends 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 idles the eZ8 CPU. If Breakpoints are not

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

**Read Data Memory (0DH).** The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1 to 65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG Mode, this command returns `FFH` for the data.

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

**Read Program Memory CRC (0EH).** The Read Program Memory CRC command computes and returns the Cyclic Redundancy Check (CRC) of Program Memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG Mode, this command returns `FFFFH` for the CRC value. Unlike most other OCD Read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads the Program Memory, calculates the CRC value and returns the result. The delay is a function of the Program Memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Program Memory.

```
DBG ← 0EH
DBG → CRC[15:8]
DBG → CRC[7:0]
```

**Step Instruction (10H).** The Step Instruction command steps one assembly instruction at the current Program Counter (PC) location. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 10H
```

**Stuff Instruction (11H).** The Stuff Instruction command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0-4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 11H
DBG ← opcode[7:0]
```

**Execute Instruction (12H).** The Execute Instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over Breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not

# eZ8 CPU Instruction Set

This chapter describes the following features of the eZ8 CPU instruction set:

Assembly Language Programming Introduction: see page 204

Assembly Language Syntax: see page 205

eZ8 CPU Instruction Notation: see page 206

eZ8 CPU Instruction Classes: see page 207

eZ8 CPU Instruction Summary: see page 212

## Assembly Language Programming Introduction

The eZ8 CPU assembly language provides a means for writing an application program without concern for actual memory addresses or machine instruction formats. A program written in assembly language is called a source program. Assembly language allows the use of symbolic addresses to identify memory locations. It also allows mnemonic codes (opcodes and operands) to represent the instructions themselves. The opcodes identify the instruction while the operands represent memory locations, registers, or immediate data values.

Each assembly language program consists of a series of symbolic commands called statements. Each statement can contain labels, operations, operands and comments.

Labels can be assigned to a particular instruction step in a source program. The label identifies that step in the program as an entry point for use by other instructions.

The assembly language also includes assembler directives that supplement the machine instruction. The assembler directives, or pseudo-ops, are not translated into a machine instruction. Rather, the pseudo-ops are interpreted as directives that control or assist the assembly process.

The source program is processed (assembled) by the assembler to obtain a machine language program called the object code. The object code is executed by the eZ8 CPU. An example segment of an assembly language program is detailed in the following example.

**Table 140. Low Power Operational Amplifier Electrical Characteristics**

| Symbol | Parameter | $V_{DD}$ = 2.7 V to 3.6 V $T_A$ = –40°C to +105°C Minimum | Typical | Maximum | Units | Conditions |
|---|---|---|---|---|---|---|
| Av | Open loop voltage gain | | 80 | | dB | |
| GBW | Gain/Bandwidth product | | 500 | | kHz | |
| PM | Phase Margin | | 50 | | deg | Assuming 13pF load capacitance. |
| $V_{osLPO}$ | Input Offset Voltage | | ±1 | ±4 | mV | |
| $V_{osLPO}$ | Input Offset Voltage (Temperature Drift) | | 1 | 10 | µV/C | |
| $V_{IN}$ | Input Voltage Range | 0.3 | | $V_{DD}$–1 | V | |
| $V_{OUT}$ | Output Voltage Range | 0.3 | | $V_{DD}$–1 | V | $I_{OUT}$ = 45µA. |

**Table 141. Comparator Electrical Characteristics**

| Symbol | Parameter | $V_{DD}$ = 2.7 V to 3.6 V $T_A$ = –40°C to +105°C Minimum | Typical | Maximum | Units | Conditions |
|---|---|---|---|---|---|---|
| $V_{OS}$ | Input DC Offset | | 5 | | mV | |
| $V_{CREF}$ | Programmable Internal Reference Voltage | | ±5 | | % | 20- and 28-pin devices. |
| | | | ±3 | | % | 8-pin devices. |
| $T_{PROP}$ | Propagation Delay | | 200 | | ns | |
| $V_{HYS}$ | Input Hysteresis | | 4 | | mV | |
| $V_{IN}$ | Input Voltage Range | $V_{SS}$ | | $V_{DD}$–1 | V | |

UARTx control 1 (UxCTL1) 112
UARTx receive data (UxRXD) 116
UARTx status 0 (UxSTAT0) 114
UARTx status 1 (UxSTAT1) 115
UARTx transmit data (UxTXD) 116
Watchdog Timer control (WDTCTL) 30, 96, 141, 196
Watchdog Timer reload high byte (WDTH) 97
Watchdog Timer reload low byte (WDTL) 98
Watchdog Timer reload upper byte (WDTU) 97
register file 15
register pair 206
register pointer 207
reset
and stop mode characteristics 23
and Stop Mode Recovery 22
carry flag 209
sources 24
RET 211
return 211
RL 211
RLC 211
rotate and shift instuctions 211
rotate left 211
rotate left through carry 211
rotate right 211
rotate right through carry 211
RP 207
RR 206, 211
rr 206
RRC 211

## S

SBC 208
SCF 209, 210
second opcode map after 1FH 225
set carry flag 209, 210
set register pointer 210
shift right arithmatic 211
shift right logical 211
signal descriptions 10
single-shot conversion (ADC) 126
software trap 211
source operand 207
SP 207
SRA 211
src 207
SRL 211
SRP 210
stack pointer 207
STOP 210
STOP mode 32
stop mode 210
Stop Mode Recovery
sources 27
using a GPIO port pin transition 28
using Watchdog Timer time-out 28
stop mode recovery
sources 29
using a GPIO port pin transition 29
SUB 208
subtract 208
subtract - extended addressing 208
subtract with carry 208
subtract with carry - extended addressing 208
SUBX 208
SWAP 211
swap nibbles 211
symbols, additional 207

## T

TCM 209
TCMX 209
test complement under mask 209
test complement under mask - extended addressing 209
test under mask 209
test under mask - extended addressing 209
timer signals 10
timers 70
architecture 70
block diagram 71
CAPTURE mode 79, 80, 87, 88
CAPTURE/COMPARE mode 83, 88
COMPARE mode 81, 87