



Welcome to [E-XFL.COM](https://www.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, WDT
Number of I/O	6
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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-QFN (5x6)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f011aqb020ec

Reset, Stop Mode Recovery, and Low Voltage Detection	23
Reset Types	23
Reset Sources	25
Power-On Reset	25
Voltage Brownout Reset	26
Watchdog Timer Reset	27
External Reset Input	27
External Reset Indicator	28
On-Chip Debugger Initiated Reset	28
Stop Mode Recovery	28
Stop Mode Recovery Using Watchdog Timer Time-Out	29
Stop Mode Recovery Using a GPIO Port Pin Transition	29
Stop Mode Recovery Using the External RESET Pin	30
Low Voltage Detection	30
Reset Register Definitions	30
Low-Power Modes	33
STOP Mode	33
HALT Mode	34
Peripheral-Level Power Control	34
Power Control Register Definitions	34
General-Purpose Input/Output	37
GPIO Port Availability By Device	37
Architecture	38
GPIO Alternate Functions	38
Direct LED Drive	39
Shared Reset Pin	39
Shared Debug Pin	39
Crystal Oscillator Override	40
5 V Tolerance	40
External Clock Setup	40
GPIO Interrupts	45
GPIO Control Register Definitions	45
Port A–D Address Registers	46
Port A–D Control Registers	46
Port A–D Data Direction Sub-Registers	47
Port A–D Alternate Function Sub-Registers	47

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	60
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	63
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	63
FC3	Interrupt Request 1	IRQ1	00	61
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	64
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	64
FC6	Interrupt Request 2	IRQ2	00	62
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	65
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	65
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	67
FCE	Shared Interrupt Select	IRQSS	00	67
FCF	Interrupt Control	IRQCTL	00	67
GPIO Port A				
FD0	Port A Address	PAADDR	00	45
FD1	Port A Control	PACTL	00	47
FD2	Port A Input Data	PAIN	XX	47
FD3	Port A Output Data	PAOUT	00	47
GPIO Port B				
FD4	Port B Address	PBADDR	00	45
FD5	Port B Control	PBCTL	00	47
FD6	Port B Input Data	PBIN	XX	47
FD7	Port B Output Data	PBOUT	00	47
GPIO Port C				
FD8	Port C Address	PCADDR	00	45
FD9	Port C Control	PCCTL	00	47
FDA	Port C Input Data	PCIN	XX	47
FDB	Port C Output Data	PCOUT	00	47
GPIO Port D				
FDC	Port D Address	PDADDR	00	45
FDD	Port D Control	PDCTL	00	47
FDE	Reserved	—	XX	
XX=Undefined				

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 μ s 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 re-initialized 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.

HALT Mode

Executing the eZ8 CPU's HALT instruction places the device into HALT mode, which powers down the CPU but leaves all other peripherals active. In HALT mode, the operating characteristics are:

- Primary oscillator is enabled and continues to operate.
- System clock is enabled and continues to operate.
- eZ8 CPU is stopped.
- Program counter (PC) stops incrementing.
- Watchdog Timer's internal RC oscillator continues to operate.
- If enabled, the Watchdog Timer continues to operate.
- All other on-chip peripherals continue to operate, if enabled.

The eZ8 CPU can be brought out of HALT mode by any of the following operations:

- Interrupt
- Watchdog Timer time-out (interrupt or reset)
- Power-On Reset
- Voltage Brownout reset
- External $\overline{\text{RESET}}$ pin assertion

To minimize current in HALT mode, all GPIO pins that are configured as inputs must be driven to one of the supply rails (V_{CC} or GND).

Peripheral-Level Power Control

In addition to the STOP and HALT modes, it is possible to disable each peripheral on each of the Z8 Encore! XP F082A Series devices. Disabling a given peripheral minimizes its power consumption.

Power Control Register Definitions

The following sections define the Power Control registers.

Power Control Register 0

Each bit of the following registers disables a peripheral block, either by gating its system clock input or by removing power from the block. The default state of the low-power

during STOP mode do not initiate Stop Mode Recovery.

1 = The Port pin is configured as a Stop Mode Recovery source. Any logic transition on this pin during STOP mode initiates Stop Mode Recovery.

Port A–D Pull-up Enable Sub-Registers

The Port A–D Pull-up Enable sub-register ([Table 24](#)) is accessed through the Port A–D Control register by writing 06H to the Port A–D Address register. Setting the bits in the Port A–D Pull-up Enable sub-registers enables a weak internal resistive pull-up on the specified Port pins.

Table 24. Port A–D Pull-Up Enable Sub-Registers (PxPUE)

BITS	7	6	5	4	3	2	1	0
FIELD	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0
RESET	00H (Ports A-C); 01H (Port D); 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 06H in Port A–D Address Register, accessible through the Port A–D Control Register							

PPUE[7:0]—Port Pull-up Enabled

0 = The weak pull-up on the Port pin is disabled.

1 = The weak pull-up on the Port pin is enabled.

Port A–D Alternate Function Set 1 Sub-Registers

The Port A–D Alternate Function Set1 sub-register ([Table 25](#)) is accessed through the Port A–D Control register by writing 07H to the Port A–D Address register. The Alternate Function Set 1 sub-registers selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register are defined in [GPIO Alternate Functions](#) on page 38.

► **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 25. Port A–D Alternate Function Set 1 Sub-Registers (PxAFS1)

BITS	7	6	5	4	3	2	1	0
FIELD	PAFS17	PAFS16	PAFS15	PAFS14	PAFS13	PAFS12	PAFS11	PAFS10
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
ADDR	If 07H in Port A–D Address Register, accessible through the Port A–D Control Register							

Table 31. LED Drive Level Low Register (LEDLVLL)

BITS	7	6	5	4	3	2	1	0
FIELD	LEDLVLL[7:0]							
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
ADDR	F84H							

LEDLVLL[7:0]—LED Level Low Bit
 {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin.
 00 = 3 mA
 01 = 7 mA
 10 = 13 mA
 11 = 20 mA

Follow the steps below for configuring a timer for COMPARE mode and initiating the count:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for COMPARE mode.
 - Set the prescale value.
 - Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate.
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the Compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control register to enable the timer and initiate counting.

In COMPARE mode, the system clock always provides the timer input. The Compare time can be calculated by the following equation:

$$\text{COMPARE Mode Time (s)} = \frac{(\text{Compare Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

GATED Mode

In GATED mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. When reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes (assuming the Timer Input signal remains asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

Follow the steps below for configuring a timer for GATED mode and initiating the count:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for GATED mode.
 - Set the prescale value.

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 in. 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 bit 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 a 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 (WDTCTL) Control register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTM, 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 the steps below to unlock the Watchdog Timer Reload Byte registers (WDTU, WDTM, 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 (WDTM) 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

configuration bits. In general, the address compare feature reduces the load on the CPU, because it does not require access to the UART when it receives data directed to other devices on the multi-node network. The following three MULTIPROCESSOR modes are available in hardware:

1. Interrupt on all address bytes.
2. Interrupt on matched address bytes and correctly framed data bytes.
3. Interrupt only on correctly framed data bytes.

These modes are selected with MPMD [1 : 0] in the UART Control 1 Register. For all multiprocessor modes, bit MPEN of the UART Control 1 Register must be set to 1.

The first scheme is enabled by writing 01b to MPMD[1:0]. In this mode, all incoming address bytes cause an interrupt, while data bytes never cause an interrupt. The interrupt service routine must manually check the address byte that caused triggered the interrupt. If it matches the UART address, the software clears MPMD[0]. Each new incoming byte interrupts the CPU. The software is responsible for determining the end of the frame. It checks for the end-of-frame by reading the MPRX bit of the UART Status 1 Register for each incoming byte. If MPRX=1, a new frame has begun. If the address of this new frame is different from the UART's address, MPMD[0] must be set to 1 causing the UART interrupts to go inactive until the next address byte. If the new frame's address matches the UART's, the data in the new frame is processed as well.

The second scheme requires the following: set MPMD[1:0] to 10B and write the UART's address into the UART Address Compare Register. This mode introduces additional hardware control, interrupting only on frames that match the UART's address. When an incoming address byte does not match the UART's address, it is ignored. All successive data bytes in this frame are also ignored. When a matching address byte occurs, an interrupt is issued and further interrupts now occur on each successive data byte. When the first data byte in the frame is read, the NEWFRM bit of the UART Status 1 Register is asserted. All successive data bytes have NEWFRM=0. When the next address byte occurs, the hardware compares it to the UART's address. If there is a match, the interrupts continues and the NEWFRM bit is set for the first byte of the new frame. If there is no match, the UART ignores all incoming bytes until the next address match.

The third scheme is enabled by setting MPMD[1:0] to 11b and by writing the UART's address into the UART Address Compare Register. This mode is identical to the second scheme, except that there are no interrupts on address bytes. The first data byte of each frame remains accompanied by a NEWFRM assertion.

External Driver Enable

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

MPRX—Multiprocessor Receive

Returns the value of the most recent multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

UART Transmit Data Register

Data bytes written to the UART Transmit Data (UxTXD) register (Table 65) are shifted out on the TXD_x pin. The Write-only UART Transmit Data register shares a Register File address with the read-only UART Receive Data register.

Table 65. UART Transmit Data Register (U0TXD)

BITS	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
ADDR	F40H							

TXD—Transmit Data

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 (Table 66). The read-only UART Receive Data register shares a Register File address with the Write-only UART Transmit Data register.

Table 66. UART Receive Data Register (U0RXD)

BITS	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
ADDR	F40H							
X = Undefined.								

RXD—Receive Data

UART receiver data byte from the RXD_x pin

Analog-to-Digital Converter

The analog-to-digital converter (ADC) converts an analog input signal to its digital representation. The features of this sigma-delta ADC include:

- 11-bit resolution in DIFFERENTIAL mode.
- 10-bit resolution in SINGLE-ENDED mode.
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports.
- 9th analog input obtained from temperature sensor peripheral.
- 11 pairs of differential inputs also multiplexed with general-purpose I/O ports.
- Low-power operational amplifier (LPO).
- Interrupt on conversion complete.
- Bandgap generated internal voltage reference with two selectable levels.
- Manual in-circuit calibration is possible employing user code (offset calibration).
- Factory calibrated for in-circuit error compensation.

Architecture

Figure 19 displays the major functional blocks of the ADC. An analog multiplexer network selects the ADC input from the available analog pins, ANA0 through ANA7.

The input stage of the ADC allows both differential gain and buffering. The following input options are available:

- Unbuffered input (SINGLE-ENDED and DIFFERENTIAL modes).
- Buffered input with unity gain (SINGLE-ENDED and DIFFERENTIAL modes).
- LPO output with full pin access to the feedback path.

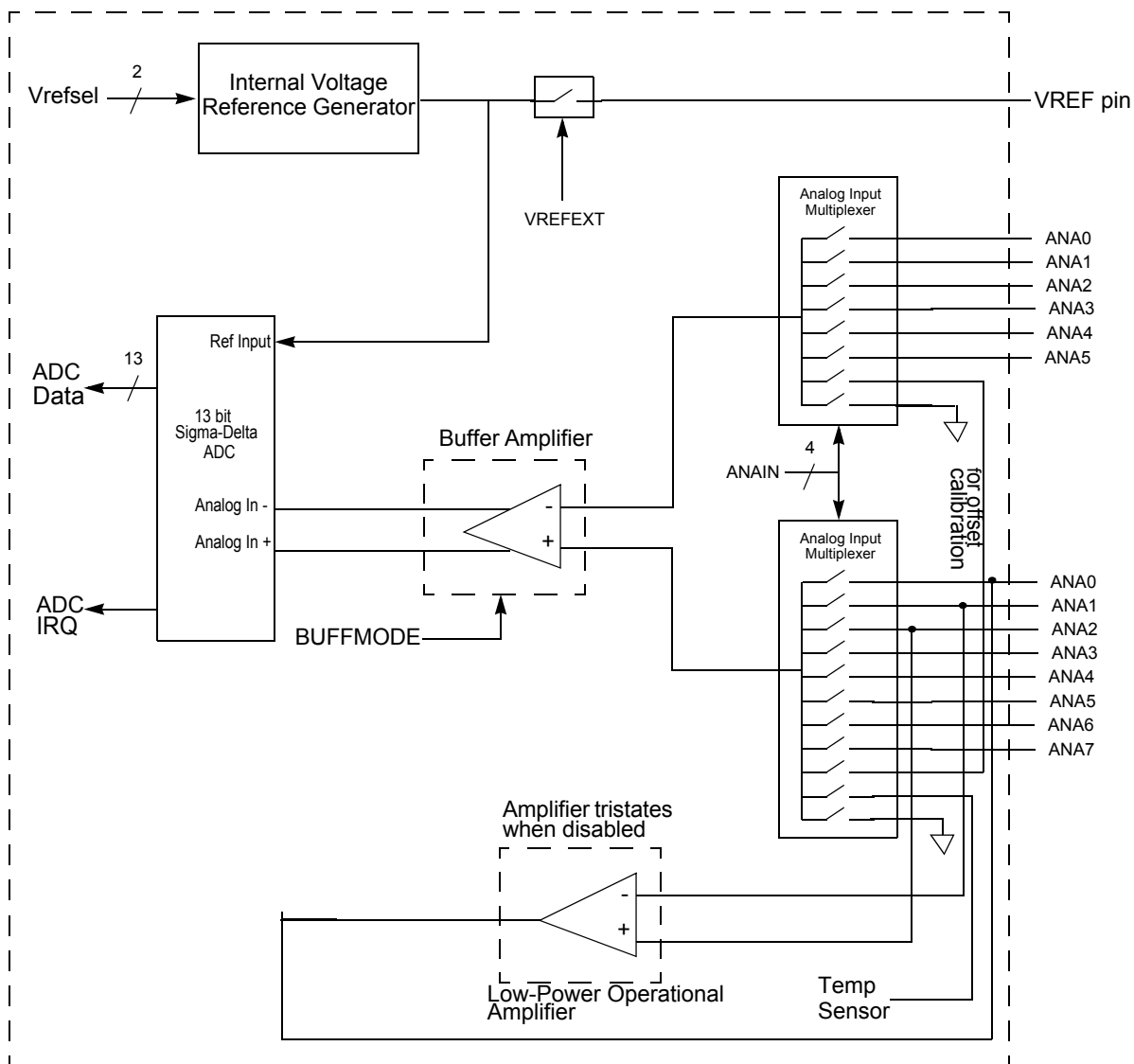


Figure 19. Analog-to-Digital Converter Block Diagram

Operation

Data Format

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

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 `REFEXT` 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 to 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. Alternatively, you can perform your own calibration, storing the values into Flash themselves. Thirdly, the user code can perform a manual offset calibration during DIFFERENTIAL mode operation.

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.

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 fields 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 μ s (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 μ s 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

Table 104. NVDS Read Time (Continued)

Operation	Minimum Latency	Maximum Latency
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, there are some things you can do to optimize your code for speed, listed in order from most helpful to least helpful:

- 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.
- Use as few unique addresses as possible: this helps to optimize the impact of refreshing as well as minimize the requirement for it.

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP[®] F082A Series with 4 KB Flash											
Standard Temperature: 0 °C to 70 °C											
Z8F041APB020SC	4 KB	1 KB	128 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F041AQB020SC	4 KB	1 KB	128 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F041ASB020SC	4 KB	1 KB	128 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F041ASH020SC	4 KB	1 KB	128 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F041AHH020SC	4 KB	1 KB	128 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F041APH020SC	4 KB	1 KB	128 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F041ASJ020SC	4 KB	1 KB	128 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F041AHJ020SC	4 KB	1 KB	128 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F041APJ020SC	4 KB	1 KB	128 B	25	19	2	0	1	1	0	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C											
Z8F041APB020EC	4 KB	1 KB	128 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F041AQB020EC	4 KB	1 KB	128 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F041ASB020EC	4 KB	1 KB	128 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F041ASH020EC	4 KB	1 KB	128 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F041AHH020EC	4 KB	1 KB	128 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F041APH020EC	4 KB	1 KB	128 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F041ASJ020EC	4 KB	1 KB	128 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F041AHJ020EC	4 KB	1 KB	128 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F041APJ020EC	4 KB	1 KB	128 B	25	19	2	0	1	1	0	PDIP 28-pin package
Replace C with G for Lead-Free Packaging											

DJNZ 206
EI 204
HALT 204
INC 203
INCW 203
IRET 206
JP 206
LD 205
LDC 205
LDCI 204, 205
LDE 205
LDEI 204
LDX 205
LEA 205
logical 205
MULT 203
NOP 204
OR 205
ORX 206
POP 205
POPX 205
program control 206
PUSH 205
PUSHX 205
RCF 204
RET 206
RL 206
RLC 206
rotate and shift 206
RR 206
RRC 206
SBC 203
SCF 204, 205
SRA 207
SRL 207
SRP 205
STOP 205
SUB 203
SUBX 203
SWAP 207
TCM 204
TCMX 204
TM 204
TMX 204

TRAP 206
Watchdog Timer refresh 205
XOR 206
XORX 206
instructions, eZ8 classes of 202
interrupt control register 67
interrupt controller 55
 architecture 55
 interrupt assertion types 58
 interrupt vectors and priority 58
 operation 57
 register definitions 60
 software interrupt assertion 59
interrupt edge select register 66
interrupt request 0 register 60
interrupt request 1 register 61
interrupt request 2 register 62
interrupt return 206
interrupt vector listing 55
interrupts
 UART 105
IR 201
Ir 201
IrDA
 architecture 117
 block diagram 117
 control register definitions 120
 operation 117
 receiving data 119
 transmitting data 118
IRET 206
IRQ0 enable high and low bit registers 62
IRQ1 enable high and low bit registers 63
IRQ2 enable high and low bit registers 65
IRR 201
Irr 201

J

JP 206
jump, conditional, relative, and relative conditional 206