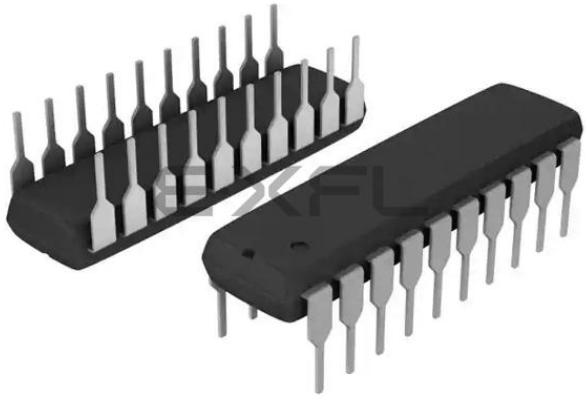


Welcome to E-XFL.COM

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, 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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°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/z8f011aph020ec

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 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.

Low-Power Modes

The Z8 Encore! XP F082A Series products contain power-saving features. The highest level of power reduction is provided by the STOP mode, in which nearly all device functions are powered down. The next lower level of power reduction is provided by the HALT mode, in which the CPU is powered down.

Further power savings can be implemented by disabling individual peripheral blocks while in Active mode (defined as being in neither STOP nor HALT mode).

STOP Mode

Executing the eZ8 CPU's STOP instruction places the device into STOP mode, powering down all peripherals except the Voltage Brownout detector, the Low-power Operational Amplifier and the Watchdog Timer. These three blocks may also be disabled for additional power savings. Specifically, the operating characteristics are:

- Primary crystal oscillator and internal precision oscillator are stopped; XIN and XOUT (if previously enabled) are disabled, and PA0/PA1 revert to the states programmed by the GPIO registers.
- System clock is stopped.
- eZ8 CPU is stopped.
- Program counter (PC) stops incrementing.
- Watchdog Timer's internal RC oscillator continues to operate if enabled by the Oscillator Control register.
- If enabled, the Watchdog Timer logic continues to operate.
- If enabled for operation in STOP mode by the associated Flash Option Bit, the Voltage Brownout protection circuit continues to operate.
- Low-power operational amplifier continues to operate if enabled by the Power Control register to do so.
- All other on-chip peripherals are idle.

To minimize current in STOP mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails (V_{CC} or GND). Additionally, any GPIOs configured as outputs must also be driven to one of the supply rails. The device can be brought out of STOP mode using Stop Mode Recovery. For more information on Stop Mode Recovery, see Reset, Stop Mode Recovery, and Low Voltage Detection on page 23.

Table 14. Port Alternate Function Mapping (Non 8-Pin Parts)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A	PA0	T0IN/$\overline{\text{T0OUT}}$*	Timer 0 Input/Timer 0 Output Complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 Output	
		Reserved		
	PA2	DE0	UART 0 Driver Enable	
		Reserved		
	PA3	$\overline{\text{CTS0}}$	UART 0 Clear to Send	
		Reserved		
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data	
		Reserved		
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data	
		Reserved		
	PA6	T1IN/$\overline{\text{T1OUT}}$*	Timer 1 Input/Timer 1 Output Complement	
		Reserved		
	PA7	T1OUT	Timer 1 Output	
		Reserved		

Note: Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in Port A–D Alternate Function Sub-Registers on page 47 automatically enables the associated alternate function.

* Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in Timer Pin Signal Operation on page 82.

Caution: *To avoid re-triggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, it is recommended that the service routine continues to read from the RSTSTAT register until the WDT bit is cleared as given in the following coding sample:*

```
CLEARWDT:
 LDX r0, RSTSTAT  ; read reset status register to clear wdt bit
 BTJNZ 5, r0, CLEARWDT   ; loop until bit is cleared
```

Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the Primary Oscillator Fail Trap, and the Watchdog Oscillator Fail Trap, the interrupt control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) register (Table 33) stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 0 register to determine if any interrupt requests are pending.

Table 33. Interrupt Request 0 Register (IRQ0)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved	T1I	T0I	U0RXI	U0TXI	Reserved	Reserved	ADCI
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	FC0H							

Reserved—Must be 0.

T1I—Timer 1 Interrupt Request
0 = No interrupt request is pending for Timer 1.
1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request
0 = No interrupt request is pending for Timer 0.
1 = An interrupt request from Timer 0 is awaiting service.

IRQ2 Enable High and Low Bit Registers

Table 42 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers (Table 43 and Table 44) form a priority encoded enabling for interrupts in the Interrupt Request 2 register.

Table 42. 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

where *x* indicates the register bits from 0–7.

Table 43. IRQ2 Enable High Bit Register (IRQ2ENH)

BITS	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
ADDR	FC7H							

Reserved—Must be 0.

C3ENH—Port C3 Interrupt Request Enable High Bit
C2ENH—Port C2 Interrupt Request Enable High Bit
C1ENH—Port C1 Interrupt Request Enable High Bit
C0ENH—Port C0 Interrupt Request Enable High Bit

Table 44. IRQ2 Enable Low Bit Register (IRQ2ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				C3ENL	C2ENL	C1ENL	C0ENL
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	FC8H							

- Set the prescale value.
- If using the Timer Output alternate function, set the initial output level (High or Low).

2. Write to the Timer High and Low Byte registers to set the starting count value (usually 0001H). This action only affects the first pass in CONTINUOUS mode. After the first timer Reload in CONTINUOUS mode, counting always begins at the reset value of 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. Enable the timer interrupt (if appropriate) and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin (if using the Timer Output function) for the Timer Output alternate function.
6. Write to the Timer Control register to enable the timer and initiate counting.

In CONTINUOUS mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{CONTINUOUS Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT mode equation to determine the first time-out period.

COUNTER Mode

In COUNTER mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO Port pin Timer Input alternate function. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the Timer Input signal. In COUNTER mode, the prescaler is disabled.

Caution: *The input frequency of the Timer Input signal must not exceed one-fourth the system clock frequency. Further, the high or low state of the input signal pulse must be no less than twice the system clock period. A shorter pulse may not be captured.*

Upon reaching the Reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. 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 Reload.

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.

Timer Control Register Definitions

Timer 0–1 Control Registers

Time 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and Timer Control Register 1 (TxCTL1) determine the timer operating mode (Table 48). It also includes a programmable PWM deadband delay, two bits to configure timer interrupt definition, and a status bit to identify if the most recent timer interrupt is caused by an input capture event.

Table 48. Timer 0–1 Control Register 0 (TxCTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
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
ADDR	F06H, F0EH							

TMODEHI—Timer Mode High Bit
This bit along with the TMODE field in TxCTL1 register determines the operating mode of the timer. This is the most significant bit of the Timer mode selection value. See the TxCTL1 register description for details of the full timer mode decoding.

TICONFIG—Timer Interrupt Configuration
This field configures timer interrupt definition.

0x = Timer Interrupt occurs on all defined Reload, Compare and Input Events
10 = Timer Interrupt only on defined Input Capture/Deassertion Events
11 = Timer Interrupt only on defined Reload/Compare Events

Reserved—Must be 0.

PWMD—PWM Delay value
This field is a programmable delay to control the number of system clock cycles delay before the Timer Output and the Timer Output Complement are forced to their active state.

000 = No delay
001 = 2 cycles delay
010 = 4 cycles delay
011 = 8 cycles delay
100 = 16 cycles delay
101 = 32 cycles delay

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. Follow the steps below 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.
   - Set or clear CTSE to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.
8. Execute an EI instruction to enable interrupts.

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 TXDx 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

Low Power Operational Amplifier

Overview

The LPO is a general-purpose low power operational amplifier. Each of the three ports of the amplifier is accessible from the package pins. The LPO contains only one pin configuration: ANA0 is the output/feedback node, ANA1 is the inverting input and ANA2 is the non-inverting input.

Operation

To use the LPO, it must be enabled in the Power Control Register 0 (PWRCTL0). The default state of the LPO is OFF. To use the LPO, the LPO bit must be cleared, turning it ON (Power Control Register 0 (PWRCTL0) on page 35). When making normal ADC measurements on ANA0 (measurements not involving the LPO output), the LPO bit must be OFF. Turning the LPO bit ON interferes with normal ADC measurements.

Warning: *The LPO bit enables the amplifier even in STOP mode. If the amplifier is not required in STOP mode, disable it. Failing to perform this results in STOP mode currents higher than necessary.*

As with other ADC measurements, any pins used for analog purposes must be configured as such in the GPIO registers (see Port A–D Alternate Function Sub-Registers on page 47).

LPO output measurements are made on ANA0, as selected by the ANAIN[3:0] bits of ADC Control Register 0. It is also possible to make single-ended measurements on ANA1 and ANA2 while the amplifier is enabled, which is often useful for determining offset conditions. Differential measurements between ANA0 and ANA2 may be useful for noise cancellation purposes.

If the LPO output is routed to the ADC, then the BUFFMODE[2:0] bits of ADC Control/Status Register 1 must also be configured for unity-gain buffered operation. Sampling the LPO in an unbuffered mode is not recommended.

When either input is overdriven, the amplifier output saturates at the positive or negative supply voltage. No instability results.

SPROT7-SPROT0—Sector Protection
Each bit corresponds to a 512 byte Flash sector. For the Z8F08xx devices, the upper 3 bits must be zero. For the Z8F04xx devices all bits are used. For the Z8F02xx devices, the upper 4 bits are unused. For the Z8F01xx devices, the upper 6 bits are unused.

Flash Frequency High and Low Byte Registers

The Flash Frequency High (FFREQH) and Low Byte (FFREQL) registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:

$$FFREQ[15:0] = \{FFREQH[7:0],FFREQL[7:0]\} = \frac{\text{System Clock Frequency}}{1000}$$

Caution: *The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device. Also, Flash programming and erasure is not supported for system clock frequencies below 20 kHz or above 20 MHz.*

Table 82. Flash Frequency High Byte Register (FFREQH)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQH							
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	FFAH							

FFREQH—Flash Frequency High Byte
High byte of the 16-bit Flash Frequency value.

Table 83. Flash Frequency Low Byte Register (FFREQL)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQL							
RESET	0							
R/W	R/W							
ADDR	FFBH							

FFREQL—Flash Frequency Low Byte
Low byte of the 16-bit Flash Frequency value.

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

Caution: *It is possible to disable the clock failure detection circuitry as well as all functioning clock sources. In this case, the Z8 Encore! XP F082A Series device ceases functioning and can only be recovered by Power-On-Reset.*

Oscillator Control Register Definitions

Oscillator Control Register

The Oscillator Control Register (OSCCTL) enables/disables the various oscillator circuits, enables/disables the failure detection/recovery circuitry and selects the primary oscillator, which becomes the system clock.

The Oscillator Control Register must be unlocked before writing. Writing the two step sequence E7H followed by 18H to the Oscillator Control Register unlocks it. The register is locked at successful completion of a register write to the OSCCTL.

Table 109. Oscillator Control Register (OSCCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	INTEN	XTLEN	WDTEN	SOFEN	WDFEN	SCKSEL		
RESET	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F86H							

INTEN—Internal Precision Oscillator Enable
1 = Internal precision oscillator is enabled
0 = Internal precision oscillator is disabled

XTLEN—Crystal Oscillator Enable; this setting overrides the GPIO register control for PA0 and PA1
1 = Crystal oscillator is enabled
0 = Crystal oscillator is disabled

WDTEN—Watchdog Timer Oscillator Enable
1 = Watchdog Timer oscillator is enabled
0 = Watchdog Timer oscillator is disabled

SOFEN—System Clock Oscillator Failure Detection Enable
1 = Failure detection and recovery of system clock oscillator is enabled
0 = Failure detection and recovery of system clock oscillator is disabled

Table 132. Flash Memory Electrical Characteristics and Timing

	V_{DD} = 2.7 V to 3.6 V T_A = -40 °C to +105 °C (unless otherwise stated)				
Parameter	**Minimum**	**Typical**	**Maximum**	**Units**	**Notes**
Flash Byte Read Time	100	–	–	ns	
Flash Byte Program Time	20	–	40	µs	
Flash Page Erase Time	10	–	–	ms	
Flash Mass Erase Time	200	–	–	ms	
Writes to Single Address Before Next Erase	–	–	2		
Flash Row Program Time	–	–	8	ms	Cumulative program time for single row cannot exceed limit before next erase. This parameter is only an issue when bypassing the Flash Controller.
Data Retention	100	–	–	years	25 °C
Endurance	10,000	–	–	cycles	Program/erase cycles

Table 133. Watchdog Timer Electrical Characteristics and Timing

		V_{DD} = 2.7 V to 3.6 V T_A = -40 °C to +105 °C (unless otherwise stated)				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
F_{WDT}	WDT Oscillator Frequency		10		kHz	
F_{WDT}	WDT Oscillator Error			±50	%	
T_{WDTCAL}	WDT Calibrated Timeout	0.98	1	1.02	s	V_{DD} = 3.3 V; T_A = 30 °C
		0.70	1	1.30	s	V_{DD} = 2.7 V to 3.6 V T_A = 0 °C to 70 °C
		0.50	1	1.50	s	V_{DD} = 2.7 V to 3.6 V T_A = -40 °C to +105 °C

Table 136. Low Power Operational Amplifier Electrical Characteristics

		V_{DD} = 2.7 V to 3.6 V T_A = -40 °C to +105 °C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
Av	Open loop voltage gain		80		dB	
GBW	Gain/Bandwidth product		500		kHz	
PM	Phase Margin		50		deg	Assuming 13 pF 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		Vdd - 1	V	
V_{OUT}	Output Voltage Range	0.3		Vdd - 1	V	I_{OUT} = 45 µA

Table 137. Comparator Electrical Characteristics

		V_{DD} = 2.7 V to 3.6 V T_A = -40 °C to +105 °C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
V_{OS}	Input DC Offset		5		mV	
V_{CREF}	Programmable Internal Reference Voltage		±5		%	20-/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	

General Purpose I/O Port Output Timing

Figure 35 and Table 140 provide timing information for GPIO Port pins.

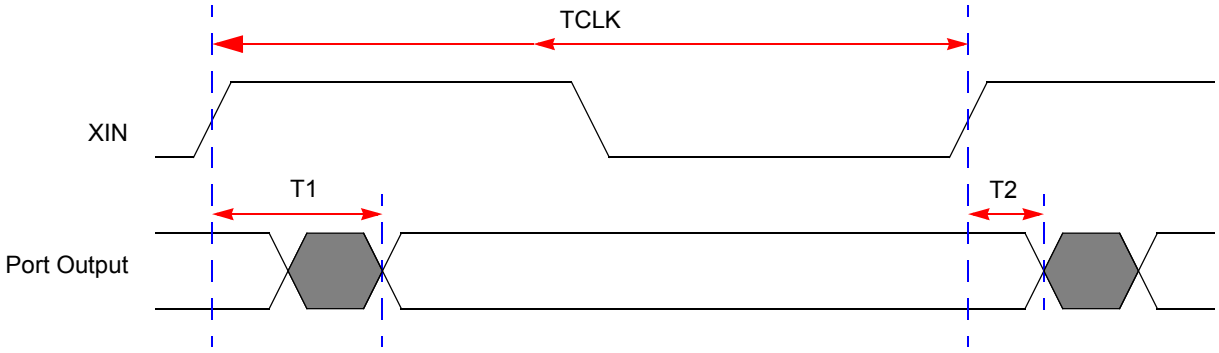
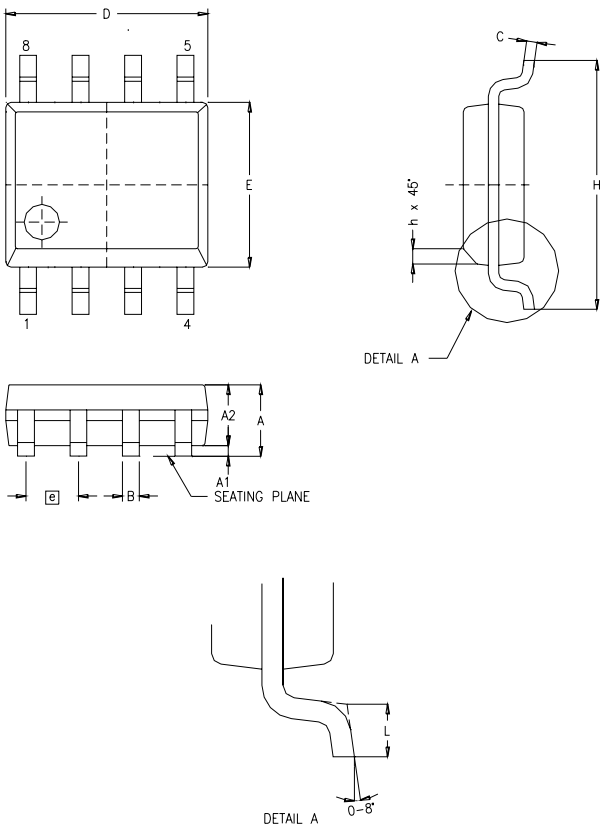


Figure 35. GPIO Port Output Timing

Table 140. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
GPIO Port pins			
T_1	XIN Rise to Port Output Valid Delay	–	15
T_2	XIN Rise to Port Output Hold Time	2	–

Figure 40 displays the 8-pin Small Outline Integrated Circuit package (SOIC) available for the Z8 Encore! XP® F082A Series devices.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	1.55	1.73	0.061	0.068
A1	0.10	0.25	0.004	0.010
A2	1.40	1.55	0.055	0.061
B	0.36	0.48	0.014	0.019
C	0.18	0.25	0.007	0.010
D	4.80	4.98	0.189	0.196
E	3.81	3.99	0.150	0.157
e	1.27 BSC		.050 BSC	
H	5.84	6.15	0.230	0.242
h	0.25	0.40	0.010	0.016
L	0.46	0.81	0.018	0.032

CONTROLLING DIMENSIONS : MM
LEADS ARE COPLANAR WITHIN .004 INCH.

Figure 40. 8-Pin Small Outline Integrated Circuit Package (SOIC)