



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	17
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f041ash020sc00tr">https://www.e-xfl.com/product-detail/zilog/z8f041ash020sc00tr</a>

# Table of Contents

<b>Overview</b>	<b>1</b>
Features	1
Part Selection Guide	2
Block Diagram	4
CPU and Peripheral Overview	5
eZ8 CPU Features	5
10-Bit Analog-to-Digital Converter	5
Low-Power Operational Amplifier	6
Internal Precision Oscillator	6
Temperature Sensor	6
Analog Comparator	6
External Crystal Oscillator	6
Low Voltage Detector	6
On-Chip Debugger	6
Universal Asynchronous Receiver/Transmitter	7
Timers	7
General-Purpose Input/Output	7
Direct LED Drive	7
Flash Controller	7
Non-Volatile Data Storage	7
Interrupt Controller	8
Reset Controller	8
<b>Pin Description</b>	<b>9</b>
Available Packages	9
Pin Configurations	9
Signal Descriptions	11
Pin Characteristics	13
<b>Address Space</b>	<b>15</b>
Register File	15
Program Memory	15
Data Memory	17
Flash Information Area	17
<b>Register Map</b>	<b>19</b>

mation Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

**Table 6. Z8 Encore! XP F082A Series Flash Memory Information Area Map**

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog Option Bits/Calibration Data
FE40–FE53	Part Number 20-character ASCII alphanumeric code Left justified and filled with FFH
FE54–FE5F	Reserved
FE60–FE7F	Zilog Calibration Data
FE80–FFFF	Reserved

## 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

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 12. Power Control Register 0 (PWRCTL0)**

BITS	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
ADDR	F80H							

LPO—Low-Power Operational Amplifier Disable  
0 = LPO is enabled (this applies even in STOP mode).  
1 = LPO is disabled.

Reserved—Must be 0.

VBO—Voltage Brownout Detector Disable  
This bit and the VBO\_AO Flash option bit must both enable the VBO for the VBO to be active.  
0 = VBO Enabled  
1 = VBO Disabled

TEMP—Temperature Sensor Disable  
0 = Temperature Sensor Enabled  
1 = Temperature Sensor Disabled

ADC—Analog-to-Digital Converter Disable  
0 = Analog-to-Digital Converter Enabled  
1 = Analog-to-Digital Converter Disabled

COMP—Comparator Disable  
0 = Comparator is Enabled  
1 = Comparator is Disabled

Reserved—Must be 0.

► **Note:** *Asserting any power control bit disables the targeted block, regardless of any enable bits contained in the target block's control registers.*

LEDEN[7:0]—LED Drive Enable

These bits determine which Port C pins are connected to an internal current sink.

0 = Tristate the Port C pin.

1 = Enable controlled current sink on the Port C pin.

## LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin ([Table 30](#)).

These two bits select between four programmable drive levels. Each pin is individually programmable.

**Table 30. LED Drive Level High Register (LEDLVLH)**

BITS	7	6	5	4	3	2	1	0
FIELD	LEDLVLH[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	F83H							

LEDLVLH[7:0]—LED Level High 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

## LED Drive Level Low Register

The LED Drive Level registers contain two control bits for each Port C pin ([Table 31](#)).

These two bits select between four programmable drive levels. Each pin is individually programmable.

If the TPOL bit in the Timer Control register is set to 0, the Timer Output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the Reload value and is reset to 0001H.

Follow the steps below for configuring a timer for PWM SINGLE OUTPUT mode and initiating the PWM operation:

1. Write to the Timer Control register to:
  - Disable the timer.
  - Configure the timer for PWM SINGLE OUTPUT mode.
  - Set the prescale value.
  - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

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 PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

Watchdog Timer Reload Registers results in a one-second timeout at room temperature and 3.3 V supply voltage.

Timeouts 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 133](#) on page 230 for details.*

## Watchdog Timer Control Register Definitions

### 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, WDTL, 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 57. Watchdog Timer Control Register (WDTCTL)**

BITS	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
ADDR	FF0H							
X = Undefined.								

WDTUNLK—Watchdog Timer Unlock

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.

### Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTL, and WDTL) registers ([Table 58](#) through [Table 60](#)) form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value is {WDTU[7:0], WDTL[7:0], WDTL[7:0]}. Writing to these registers sets the appropriate Reload Value. Reading from these registers returns the current Watchdog Timer count value.



0 = No framing error occurred.

1 = A framing error occurred.

BRKD—Break Detect

This bit indicates that a break occurred. If the data bits, parity/multiprocessor bit, and Stop bit(s) are all 0s this bit is set to 1. Reading the UART Receive Data register clears this bit.

0 = No break occurred.

1 = A break occurred.

TDRE—Transmitter Data Register Empty

This bit indicates that the UART Transmit Data register is empty and ready for additional data. Writing to the UART Transmit Data register resets this bit.

0 = Do not write to the UART Transmit Data register.

1 = The UART Transmit Data register is ready to receive an additional byte to be transmitted.

TXE—Transmitter Empty

This bit indicates that the transmit shift register is empty and character transmission is finished.

0 = Data is currently transmitting.

1 = Transmission is complete.

CTS— $\overline{\text{CTS}}$  signal

When this bit is read it returns the level of the  $\overline{\text{CTS}}$  signal. This signal is active Low.

## UART Status 1 Register

This register contains multiprocessor control and status bits.

**Table 64. UART Status 1 Register (U0STAT1)**

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved						NEWFRM	MPRX
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R
ADDR	F44H							

Reserved—Must be 0.

NEWFRM—Status bit denoting the start of a new frame. Reading the UART Receive Data register resets this bit to 0.

0 = The current byte is not the first data byte of a new frame.

1 = The current byte is the first data byte of a new frame.

#5 MSB	#5 LSB
--------	--------

6. Add the gain correction factor to the original offset corrected value.

#5 MSB	#5 LSB
--------	--------

+

#1 MSB	#1 LSB
--------	--------

=

#6 MSB	#6 LSB
--------	--------

7. Shift the result to the right, using the sign bit determined in [Step 1](#). This allows for the detection of computational overflow.

S->	#6 MSB	#6 LSB
-----	--------	--------

## Output Data

The following is the output format of the corrected ADC value.

MSB	LSB
s v b a 9 8 7 6	5 4 3 2 1 0 - -

The overflow bit in the corrected output indicates that the computed value was greater than the maximum logical value (+1023) or less than the minimum logical value (-1024). Unlike the hardware overflow bit, this is not a simple binary Flag. For a normal sample (non-overflow), the sign and the overflow bit matches. If the sign bit and overflow bit do not match, a computational overflow has occurred.

## Input Buffer Stage

Many applications require the measurement of an input voltage source with a high output impedance. This ADC provides a buffered input for such situations. The drawback of the buffered input is a limitation of the input range. When using unity gain buffered mode, the input signal must be prevented from coming too close to either  $V_{SS}$  or  $V_{DD}$ . See [Table 135](#) on page 231 for details.

This condition applies only to the input voltage level (with respect to ground) of each differential input signal. The actual differential input voltage magnitude may be less than 300 mV.

The input range of the unbuffered ADC swings from  $V_{SS}$  to  $V_{DD}$ . Input signals smaller than 300 mV must use the unbuffered input mode. If these signals do not contain low output impedances, they might require off-chip buffering.

Signals outside the allowable input range can be used without instability or device damage. Any ADC readings made outside the input range are subject to greater inaccuracy than specified.

# Flash Memory

The products in the Z8 Encore! XP<sup>®</sup> F082A Series feature a non-volatile Flash memory of 8 KB (8192), 4 KB (4096), 2 KB (2048 bytes), or 1 KB (1024) with read/write/erase capability. The Flash Memory can be programmed and erased in-circuit by user code or through the On-Chip Debugger. The features include:

- User controlled read and write protect capability
- Sector-based write protection scheme
- Additional protection schemes against accidental program and erasure

## Architecture

The Flash memory array is arranged in pages with 512 bytes per page. The 512 byte page is the minimum Flash block size that can be erased. Each page is divided into 8 rows of 64 bytes.

For program or data protection, the Flash memory is also divided into sectors. In the Z8 Encore! XP F082A Series, these sectors are either 1024 bytes (in the 8 KB devices) or 512 bytes (all other memory sizes) in size. Page and sector sizes are not generally equal.

The first 2 bytes of the Flash Program memory are used as Flash Option Bits. For more information about their operation, see [Flash Option Bits](#) on page 153.

[Table 76](#) describes the Flash memory configuration for each device in the Z8 Encore! XP F082A Series. [Figure 21](#) displays the Flash memory arrangement.

**Table 76. Z8 Encore! XP F082A Series Flash Memory Configurations**

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (Bytes)
Z8F08xA	8 (8192)	16	0000H–1FFFH	1024
Z8F04xA	4 (4096)	8	0000H–0FFFH	512
Z8F02xA	2 (2048)	4	0000H–07FFH	512
Z8F01xA	1 (1024)	2	0000H–03FFH	512

## Flash Control Register Definitions

### Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FCTL) register before programming or erasing the Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control register unlocks the Flash Controller. When the Flash Controller is unlocked, the Flash memory can be enabled for Mass Erase or Page Erase by writing the appropriate enable command to the FCTL. Page Erase applies only to the active page selected in Flash Page Select register. Mass Erase is enabled only through the On-Chip Debugger. Writing an invalid value or an invalid sequence returns the Flash Controller to its locked state. The Write-only Flash Control Register shares its Register File address with the read-only Flash Status Register.

**Table 78. Flash Control Register (FCTL)**

<b>BITS</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>FIELD</b>	FCMD							
<b>RESET</b>	0	0	0	0	0	0	0	0
<b>R/W</b>	W	W	W	W	W	W	W	W
<b>ADDR</b>	FF8H							

FCMD—Flash Command

73H = First unlock command.

8CH = Second unlock command.

95H = Page Erase command (must be third command in sequence to initiate Page Erase).

63H = Mass Erase command (must be third command in sequence to initiate Mass Erase).

5EH = Enable Flash Sector Protect Register Access

### Randomized Lot Identification Bits

As an optional feature, Zilog is able to provide a factory-programmed random lot identifier. With this feature, all devices in a given production lot are programmed with the same random number. This random number is uniquely regenerated for each successive production lot and is not likely to be repeated.

The randomized lot identifier is a 32 byte binary value, stored in the Flash information page (see [Reading the Flash Information Page](#) on page 155 and [Randomized Lot Identifier](#) on page 166 for more details) and is unaffected by mass erasure of the device's Flash memory.

### Reading the Flash Information Page

The following code example shows how to read data from the Flash information area.

```
; get value at info address 60 (FE60h)

ldx FPS, #%80 ; enable access to flash info page

ld R0, #%FE

ld R1, #%60

ldc R2, @RR0 ; R2 now contains the calibration value
```

## Flash Option Bit Control Register Definitions

### Trim Bit Address Register

The Trim Bit Address (TRMADR) register contains the target address for an access to the trim option bits ([Table 84](#)).

**Table 84. Trim Bit Address Register (TRMADR)**

BITS	7	6	5	4	3	2	1	0
FIELD	TRMADR - Trim Bit Address (00H to 1FH)							
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	FF6H							

read operations to illegal addresses. Also, the user code must pop the address byte off the stack.

The read routine uses 9 bytes of stack space in addition to the one byte of address pushed by the user. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS reads exhibit a non-uniform execution time. A read operation takes between 44  $\mu$ s and 489  $\mu$ s (assuming a 20 MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from invalid addresses (those exceeding the NVDS array size) return 0xff. Illegal read operations have a 2  $\mu$ s execution time.

The status byte returned by the NVDS read routine is zero for successful read, as determined by a CRC check. If the status byte is non-zero, there was a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have a CRC error.

## Power Failure Protection

The NVDS routines employ error checking mechanisms to ensure a power failure endangers only the most recently written byte. Bytes previously written to the array are not perturbed.

A system reset (such as a pin reset or Watchdog Timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array are unperturbed.

## Optimizing NVDS Memory Usage for Execution Speed

The NVDS read time varies drastically, this discrepancy being a trade-off for minimizing the frequency of writes that require post-write page erases (see [Table 104](#)). The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N, as well as the number of writes since the most recent page erase. Neglecting effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb is that every write since the most recent page erase causes read times of unwritten addresses to increase by 1  $\mu$ s, up to a maximum of (511-NVDS\_SIZE)  $\mu$ s.

**Table 104. NVDS Read Time**

Operation	Minimum Latency	Maximum Latency
Read (16 byte array)	875	9961
Read (64 byte array)	876	8952

Write Memory, Read Memory, Write Register, Read Register, Read Memory CRC, Step Instruction, Stuff Instruction, and Execute Instruction commands.

```
DBG ← 03H
DBG → RuntimeCounter[15:8]
DBG → RuntimeCounter[7:0]
```

- **Write OCD Control Register (04H)**—The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Flash Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of returning the device to normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

- **Read OCD Control Register (05H)**—The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

- **Write Program Counter (06H)**—The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

- **Read Program Counter (07H)**—The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

- **Write Register (08H)**—The Write Register command writes data to the Register File. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in DEBUG mode, the address and data values are discarded. If the Flash Read Protect Option bit is enabled, only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0, Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```





Table 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
AND dst, src	$\text{dst} \leftarrow \text{dst AND src}$	r	r	52	–	*	*	0	–	–	2	3
		r	lr	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	$\text{dst} \leftarrow \text{dst AND src}$	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	$\text{dst}[\text{bit}] \leftarrow 0$	r		E2	–	–	–	–	–	–	2	2
BIT p, bit, dst	$\text{dst}[\text{bit}] \leftarrow p$	r		E2	–	–	–	–	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	$\text{dst}[\text{bit}] \leftarrow 1$	r		E2	–	–	–	–	–	–	2	2
BSWAP dst	$\text{dst}[7:0] \leftarrow \text{dst}[0:7]$	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if $\text{src}[\text{bit}] = p$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJNZ bit, src, dst	if $\text{src}[\text{bit}] = 1$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJZ bit, src, dst	if $\text{src}[\text{bit}] = 0$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
CALL dst	$\text{SP} \leftarrow \text{SP} - 2$ $@\text{SP} \leftarrow \text{PC}$ $\text{PC} \leftarrow \text{dst}$	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	$C \leftarrow \sim C$			EF	*	–	–	–	–	–	1	2
CLR dst	$\text{dst} \leftarrow 00H$	R		B0	–	–	–	–	–	–	2	2
		IR		B1							2	3
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

## General Purpose I/O Port Output Timing

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

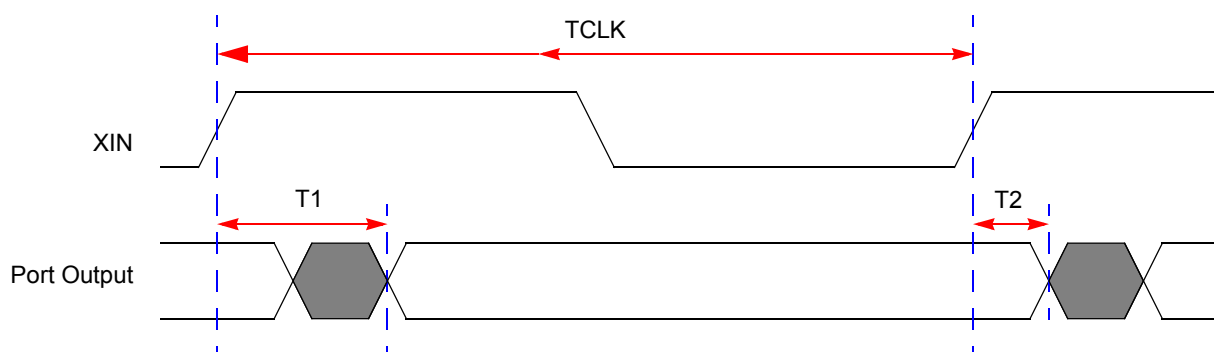


Figure 35. GPIO Port Output Timing

Table 140. GPIO Port Output Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
GPIO Port pins			
T <sub>1</sub>	XIN Rise to Port Output Valid Delay	–	15
T <sub>2</sub>	XIN Rise to Port Output Hold Time	2	–

Figure 41 displays the 8-pin Quad Flat No-Lead package (QFN)/MLF-S available for the Z8 Encore! XP F082A Series devices. This package has a footprint identical to that of the 8-pin SOIC, but with a lower profile.

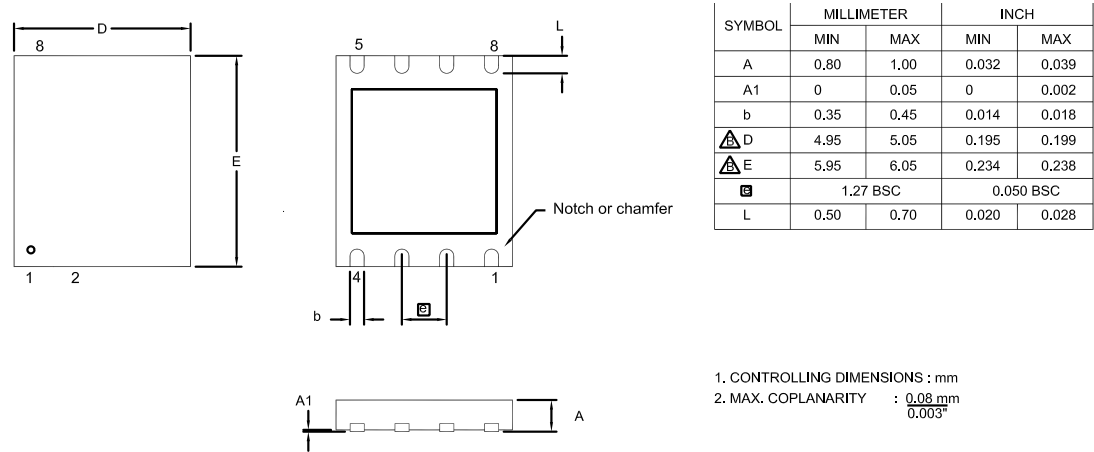


Figure 41. 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S

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
<b>Z8 Encore! XP<sup>®</sup> F082A Series with 2 KB Flash</b>											
<b>Standard Temperature: 0 °C to 70 °C</b>											
Z8F021APB020SC	2 KB	512 B	64 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F021AQB020SC	2 KB	512 B	64 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F021ASB020SC	2 KB	512 B	64 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F021ASH020SC	2 KB	512 B	64 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F021AHH020SC	2 KB	512 B	64 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F021APH020SC	2 KB	512 B	64 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F021ASJ020SC	2 KB	512 B	64 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F021AHJ020SC	2 KB	512 B	64 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F021APJ020SC	2 KB	512 B	64 B	25	19	2	0	1	1	0	PDIP 28-pin package
<b>Extended Temperature: -40 °C to 105 °C</b>											
Z8F021APB020EC	2 KB	512 B	64 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F021AQB020EC	2 KB	512 B	64 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F021ASB020EC	2 KB	512 B	64 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F021ASH020EC	2 KB	512 B	64 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F021AHH020EC	2 KB	512 B	64 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F021APH020EC	2 KB	512 B	64 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F021ASJ020EC	2 KB	512 B	64 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F021AHJ020EC	2 KB	512 B	64 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F021APJ020EC	2 KB	512 B	64 B	25	19	2	0	1	1	0	PDIP 28-pin package
Replace C with G for Lead-Free Packaging											

## register 201

- ADC control (ADCCTL) 130, 132
- ADC data high byte (ADCDH) 132
- ADC data low bits (ADC DL) 133
- flash control (FCTL) 149, 155, 156
- flash high and low byte (FFREQH and FREEQL) 152
- flash page select (FPS) 150, 151
- flash status (FSTAT) 150
- GPIO port A-H address (PxADDR) 46
- GPIO port A-H alternate function sub-registers 48
- GPIO port A-H control address (PxCTL) 47
- GPIO port A-H data direction sub-registers 47
- OCD control 184
- OCD status 185
- UARTx baud rate high byte (UxBRH) 114
- UARTx baud rate low byte (UxBRL) 114
- UARTx Control 0 (UxCTL0) 108, 114
- UARTx control 1 (UxCTL1) 109
- UARTx receive data (UxRXD) 113
- UARTx status 0 (UxSTAT0) 111
- UARTx status 1 (UxSTAT1) 112
- UARTx transmit data (UxTXD) 113
- Watchdog Timer control (WDTCTL) 31, 94, 136, 190
- Watchdog Timer reload high byte (WDTH) 95
- Watchdog Timer reload low byte (WDTL) 95
- Watchdog Timer reload upper byte (WD-TU) 95

## register file 15

## register pair 201

## register pointer 202

## reset

- and stop mode characteristics 24
- and Stop Mode Recovery 23
- carry flag 204
- sources 25

## RET 206

## return 206

## RL 206

## RLC 206

## rotate and shift instructions 206

## rotate left 206

## rotate left through carry 206

## rotate right 206

## rotate right through carry 206

## RP 202

## RR 201, 206

## rr 201

## RRC 206

## S

## SBC 203

## SCF 204, 205

## second opcode map after 1FH 219

## set carry flag 204, 205

## set register pointer 205

## shift right arithmetic 207

## shift right logical 207

## signal descriptions 11

## single-shot conversion (ADC) 123

## software trap 206

## source operand 202

## SP 202

## SRA 207

## src 202

## SRL 207

## SRP 205

## stack pointer 202

## STOP 205

## STOP mode 33

## stop mode 205

## Stop Mode Recovery

### sources 28

### using a GPIO port pin transition 29

### using Watchdog Timer time-out 29

## stop mode recovery

### sources 30

### using a GPIO port pin transition 30

## SUB 203

## subtract 203

## subtract - extended addressing 203

## subtract with carry 203