



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	Active
Core Processor	eZ8
Core Size	8-Bit
Speed	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	6
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K 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	<a href="https://www.e-xfl.com/product-detail/zilog/z8f0813qb005eg">https://www.e-xfl.com/product-detail/zilog/z8f0813qb005eg</a>

# Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Chapter/Section	Description	Page No.
Sep	15	LED Drive Enable Register	Clarified statement surrounding the Alternate Function Register as it relates to the LED function; revised Flash Sector Protect Register description; revised Packaging chapter.	<u>51</u> , <u>144</u> , <u>210</u>
Mar 2008	14	n/a	Changed branding to <i>Z8 Encore! XP F0823 Series</i> where appropriate.	All
Dec 2007	13	Pin Description, General-Purpose Input/Output, Interrupt Controller, Watchdog Timer, Electrical Characteristics, and Ordering Information	Updated title from <i>Z8 Encore! 8K and 4K Series</i> to <i>Z8 Encore! XP Z8F0823 Series</i> . Updated Figure 3, Table 15, Table 35, Tables 59 through 61, Table 119 and Part Number Suffix Designations section.	<u>8</u> , <u>36</u> , <u>60</u> , <u>95</u> , <u>199</u> , and <u>220</u>
Aug 2007	12	Part Selection Guide, External Clock Setup, and Program Memory	Updated Table 1, Table 16, and Program Memory section.	<u>2</u> , <u>35</u> , and <u>13</u>
Jun 2007	11	n/a	Updated to combine Z8 Encore! 8K and Z8 Encore! 4K Series.	All
Dec 2006	10	Ordering Information	Updated Ordering Information chapter.	<u>211</u>

OCD Status Register .....	168
Oscillator Control .....	169
Operation .....	169
System Clock Selection .....	169
Clock Failure Detection and Recovery .....	170
Oscillator Control Register Definitions .....	171
Internal Precision Oscillator .....	173
Operation .....	173
eZ8 CPU Instruction Set .....	174
Assembly Language Programming Introduction .....	174
Assembly Language Syntax .....	175
eZ8 CPU Instruction Notation .....	176
eZ8 CPU Instruction Classes .....	178
eZ8 CPU Instruction Summary .....	182
Opcode Maps .....	192
Electrical Characteristics .....	196
Absolute Maximum Ratings .....	196
DC Characteristics .....	197
AC Characteristics .....	200
On-Chip Peripheral AC and DC Electrical Characteristics .....	201
General Purpose I/O Port Input Data Sample Timing .....	204
General Purpose I/O Port Output Timing .....	206
On-Chip Debugger Timing .....	207
UART Timing .....	208
Packaging .....	210
Ordering Information .....	211
Part Number Suffix Designations .....	220
Index .....	221
Customer Support .....	230

## **CPU and Peripheral Overview**

The eZ8 CPU, Zilog's latest 8-bit central processing unit (CPU), meets the continuing demand for faster and code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8 instruction set. The eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required program memory
- Software stack allows much greater depth in subroutine calls and interrupts than hardware stacks
- Compatible with existing Z8 code
- Expanded internal Register File allows access of up to 4 KB
- New instructions improve execution efficiency for code developed using higher-level programming languages, including C
- Pipelined instruction fetch and execution
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT, and SRL
- New instructions support 12-bit linear addressing of the Register File
- Up to 10 MIPS operation
- C-Compiler friendly
- 2 to 9 clock cycles per instruction

For more information about the eZ8 CPU, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#) available for download at [www.zilog.com](http://www.zilog.com).

## **General-Purpose I/O**

F0823 Series features 6 to 24 port pins (Ports A–C) for general-purpose I/O (GPIO). The number of GPIO pins available is a function of package. Each pin is individually programmable. 5V-tolerant input pins are available on all I/Os on 8-pin devices, most I/Os on other package types.

## **Flash Controller**

The Flash Controller programs and erases Flash memory. The Flash Controller supports protection against accidental program and erasure, as well as factory serialization and read protection.



## **On-Chip Debugger**

F0823 Series products feature an integrated On-Chip Debugger. The OCD provides a rich-set of debugging capabilities, such as reading and writing registers, programming Flash memory, setting breakpoints and executing code. A single-pin interface provides communication to the OCD.

## **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 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

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 Brown-Out 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 F0823 Series devices. Disabling a given peripheral minimizes its power consumption.

## **Power Control Register Definitions**

The following sections describe 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.

## Port A–C Input Data Registers

Reading from the Port A–C Input Data registers (Table 30) returns the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

**Table 30. Port A–C Input Data Registers (PxIN)**

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	FD2H, FD6H, FDAH							

Bit	Description
[7:0] PxIN	<b>Port Input Data</b> Sampled data from the corresponding port pin input. 0 = Input data is logical 0 (Low). 1 = Input data is logical 1 (High).

Note: x indicates the specific GPIO port pin number (7–0).

Table 40. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	Reserved		ADCENH
RESET	0	0	0	0	0	0		0
R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Address	FC1H							

Bit	Description
[7]	<b>Reserved</b> This bit is reserved and must be programmed to 0.
[6] T1ENH	<b>Timer 1 Interrupt Request Enable High Bit</b>
[5] T0ENH	<b>Timer 0 Interrupt Request Enable High Bit</b>
[4] U0RENH	<b>UART 0 Receive Interrupt Request Enable High Bit</b>
[3] U0TENH	<b>UART 0 Transmit Interrupt Request Enable High Bit</b>
[2:1]	<b>Reserved</b> These bits are reserved and must be programmed to 00.
[0] ADCENH	<b>ADC Interrupt Request Enable High Bit</b>

Table 41. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENL	T0ENL	U0RENL	U0TENL	Reserved		ADCENL
RESET	0	0	0	0	0	0		0
R/W	R	R/W	R/W	R/W	R/W	R		R/W
Address	FC2H							

Bit	Description
[7]	<b>Reserved</b> This bit is reserved and must be programmed to 0 when read.
[6] T1ENL	<b>Timer 1 Interrupt Request Enable Low Bit</b>

Observe the following steps to configure a timer for PWM DUAL OUTPUT Mode and initiating the PWM operation:

1. Write to the Timer Control Register to:
  - Disable the timer
  - Configure the timer for PWM DUAL OUTPUT Mode. Setting the mode also involves writing to the TMODEHI bit in the TxCTL1 Register
  - 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 write 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 PWM Control Register to set the PWM dead band delay value. The dead-band delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).
5. 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.
6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
8. 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, the ONE-SHOT Mode equation determines 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:

**Table 61. Watchdog Timer Reload Upper Byte Register (WDTU)**

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	00H							
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF1H							
Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value.								

Bit	Description
[7:0] WDTU	<b>WDT Reload Upper Byte</b> Most significant byte (MSB), Bits[23:16], of the 24-bit WDT reload value.

**Table 62. Watchdog Timer Reload High Byte Register (WDTH)**

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
RESET	04H							
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF2H							
Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value.								

Bit	Description
[7:0] WDTH	<b>WDT Reload High Byte</b> Middle byte, Bits[15:8], of the 24-bit WDT reload value.

**Table 63. Watchdog Timer Reload Low Byte Register (WDTL)**

Bit	7	6	5	4	3	2	1	0
Field	WDTL							
RESET	00H							
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF3H							
Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value.								

$$\left(\frac{1}{\text{Baud Rate (Hz)}}\right) \leq \text{DE to Start Bit Setup Time (s)} \leq \left(\frac{2}{\text{Baud Rate (Hz)}}\right)$$

### Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Data Register Empty bit (TDRE) is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The TDRE interrupt occurs after the Transmit shift register has shifted the first bit of data out. The Transmit Data Register can now be written with the next character to send. This action provides 7 bit periods of latency to load the Transmit Data Register before the Transmit shift register completes shifting the current character. Writing to the UART Transmit Data Register clears the TDRE bit to 0.

### Receiver Interrupts

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

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

---

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

---

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

### UART Overrun Errors

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

After the valid data has been read, the UART Status 0 Register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data Register contains a data byte. However, because the overrun error

## UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (Table 68 and Table 69) configure the properties of the UART's transmit and receive operations. The UART Control registers must not be written while the UART is enabled.

**Table 68. UART Control 0 Register (U0CTL0)**

Bit	7	6	5	4	3	2	1	0
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
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	F42H							

Bit	Description
[7] TEN	<b>Transmit Enable</b> This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled. 0 = Transmitter disabled. 1 = Transmitter enabled.
[6] REN	<b>Receive Enable</b> This bit enables or disables the receiver. 0 = Receiver disabled. 1 = Receiver enabled.
[5] CTSE	<b>CTSE—CTS Enable</b> 0 = The $\overline{\text{CTS}}$ signal has no effect on the transmitter. 1 = The UART recognizes the CTS signal as an enable control from the transmitter.
[4] PEN	<b>Parity Enable</b> This bit enables or disables parity. Even or odd is determined by the PSEL bit. 0 = Parity is disabled. 1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit .
[3] PSEL	<b>Parity Select</b> 0 = Even parity is transmitted and expected on all received data. 1 = Odd parity is transmitted and expected on all received data.
[2] SBRK	<b>Send Break</b> This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has finished sending data before setting this bit. 0 = No break is sent. 1 = Forces a break condition by setting the output of the transmitter to zero.



The window remains open until the count again reaches 8 (that is, 24 baud clock periods since the previous pulse was detected), giving the endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal, allowing the endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

## Infrared Encoder/Decoder Control Register Definitions

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

---

**!** **Caution:** To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the endec before enabling the GPIO port alternate function for the corresponding pin.

---

# Oscillator Control

Z8 Encore! XP F0823 Series devices uses three possible clocking schemes, each user-selectable. These three schemes are:

- On-chip precision trimmed RC oscillator
- External clock drive
- On-chip low power Watchdog Timer oscillator

In addition, F0823 Series devices contain clock failure detection and recovery circuitry, which allow continued operation despite a failure of the primary oscillator.

## Operation

This chapter discusses the logic used to select the system clock and handle primary oscillator failures. A description of the specific operation of each oscillator is outlined elsewhere in this document.

## System Clock Selection

The oscillator control block selects from the available clocks. Table 104 details each clock source and its usage.

**Table 104. Oscillator Configuration and Selection**

Clock Source	Characteristics	Required Setup
Internal Precision RC Oscillator	<ul style="list-style-type: none"><li>• 32.8kHz or 5.53MHz</li><li>• <math>\pm 4\%</math> accuracy when trimmed</li><li>• No external components required</li></ul>	<ul style="list-style-type: none"><li>• Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 5.53MHz or 32.8kHz</li></ul>
External Clock Drive	<ul style="list-style-type: none"><li>• 0 to 20MHz</li><li>• Accuracy dependent on external clock source</li></ul>	<ul style="list-style-type: none"><li>• Write GPIO registers to configure PB3 pin for external clock function</li><li>• Unlock and write OSCCTL to select external system clock</li><li>• Apply external clock signal to GPIO</li></ul>
Internal Watchdog Timer Oscillator	<ul style="list-style-type: none"><li>• 10kHz nominal</li><li>• <math>\pm 40\%</math> accuracy; no external components required</li><li>• Very Low power consumption</li></ul>	<ul style="list-style-type: none"><li>• Enable WDT if not enabled and wait until WDT Oscillator is operating.</li><li>• Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator</li></ul>

---

**! Caution:** Unintentional accesses to the Oscillator Control Register can actually stop the chip by switching to a non-functioning oscillator. To prevent this condition, the oscillator control block employs a register unlocking/locking scheme.

---

### OSC Control Register Unlocking/Locking

To write to the Oscillator Control Register, unlock it by making two writes to the OSCCTL Register with the values `E7H` followed by `18H`. A third write to the OSCCTL Register changes the value of the actual register and returns the register to a locked state. Any other sequence of Oscillator Control Register writes has no effect. The values written to unlock the register must be ordered correctly, but are not necessarily consecutive. It is possible to write to or read from other registers within the unlocking/locking operation.

When selecting a new clock source, the primary oscillator failure detection circuitry and the Watchdog Timer oscillator failure circuitry must be disabled. If `POFEN` and `WOFEN` are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The Failure detection circuitry can be enabled anytime after a successful write of `OSCSEL` in the Oscillator Control Register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it is appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

## Clock Failure Detection and Recovery

Should an oscillator or timer fail, there are methods of recovery, as this section describes.

### Primary Oscillator Failure

Z8 Encore! XP F0823 Series devices can generate non-maskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer oscillator to drive the system clock. The Watchdog Timer oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer is the primary oscillator. It is also unavailable if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function outlined in the [Watchdog Timer](#) section on page 91.

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below  $1\text{ kHz} \pm 50\%$ . If an external signal is selected as the system oscillator, it is possible that a very slow but non-failing clock can generate a failure condition. Under these

**Table 115. Logical Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using Extended Addressing
COM	dst	Complement
OR	dst, src	Logical OR
ORX	dst, src	Logical OR using Extended Addressing
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using Extended Addressing

**Table 116. Program Control Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap

**Table 117. Rotate and Shift Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry

Table 118. 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		
TCM dst, src	(NOT dst) AND src	r	r	62	–	*	*	0	–	–	2	3
		r	lr	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	–	*	*	0	–	–	4	3
		ER	IM	69							4	3
TM dst, src	dst AND src	r	r	72	–	*	*	0	–	–	2	3
		r	lr	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	–	*	*	0	–	–	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	–	–	–	–	–	–	2	6
WDT				5F	–	–	–	–	–	–	1	2

Note: Flags Notation:

\* = Value is a function of the result of the operation.

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

**Table 126. Flash Memory Electrical Characteristics and Timing**

$V_{DD} = 2.7V \text{ to } 3.6V$ $T_A = -40^{\circ}C \text{ to } +105^{\circ}C$ (unless otherwise stated)					
Parameter	Minimum	Typical	Maximum	Units	Notes
Flash Byte Read Time	100	–	–	ns	
Flash Byte Program Time	20	–	40	$\mu s$	
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 127. Watchdog Timer Electrical Characteristics and Timing**

V <sub>DD</sub> = 2.7V to 3.6V T <sub>A</sub> = −40°C to +105°C (unless otherwise stated)						
Symbol	Parameter	Minimum	Typical	Maximum	Units	Conditions
F <sub>WDT</sub>	WDT Oscillator Frequency		10		kHz	
F <sub>WDT</sub>	WDT Oscillator Error			±50	%	
T <sub>WDT-CAL</sub>	WDT Calibrated Timeout	0.98	1	1.02	s	V <sub>DD</sub> = 3.3 V; T <sub>A</sub> = 30°C
		0.70	1	1.30	s	V <sub>DD</sub> = 2.7V to 3.6V T <sub>A</sub> = 0°C to 70°C
		0.50	1	1.50	s	V <sub>DD</sub> = 2.7V to 3.6V T <sub>A</sub> = −40°C to +105°C

# Packaging

Zilog's F0823 Series of MCUs includes the Z8F0113, Z8F0123, Z8F0213, Z8F0223, Z8F0413, Z8F0423, Z8F0813 and Z8F0823 devices, which are available in the following packages:

- 8-pin Plastic Dual Inline Package (PDIP)
- 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S<sup>1</sup>
- 20-pin Plastic Dual-Inline Package (PDIP)
- 20-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Small Shrink Outline Package (SSOP)
- 28-pin Plastic Dual-Inline Package (PDIP)
- 28-pin Small Outline Integrated Circuit Package (SOIC)
- 28-pin Small Shrink Outline Package (SSOP)

Current diagrams for each of these packages are published in Zilog's Packaging Product Specification (PS0072), which is available free for download from the Zilog website.

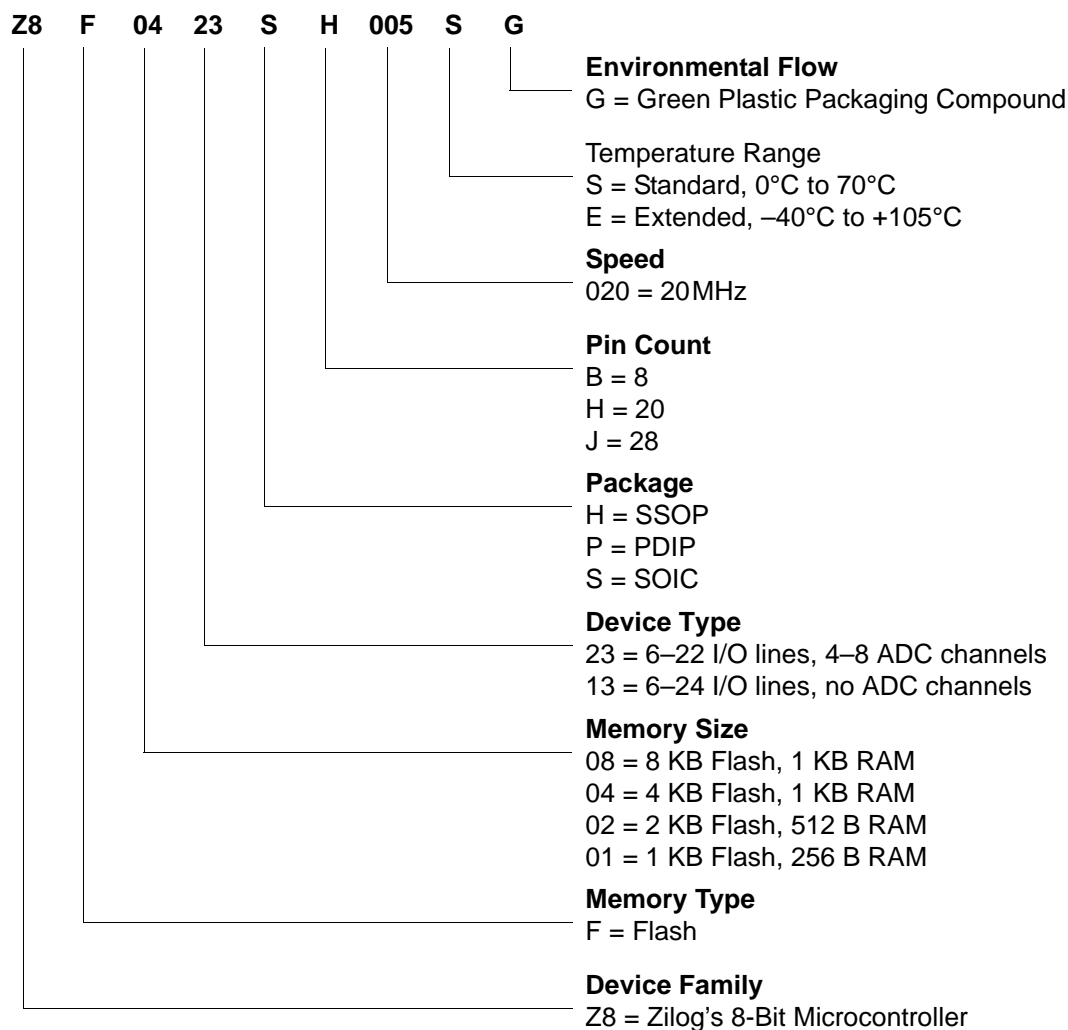
---

1. The footprint of the QFN)/MLF-S package is identical to that of the 8-pin SOIC package, but with a lower profile.

## Part Number Suffix Designations

Zilog part numbers consist of a number of components, as indicated in the following example.

**Example.** Part number Z8F0423SH005SG is an 8-bit 20MHz Flash MCU with 4KB of Program Memory and equipped with 6–22 I/O lines and 4–8 ADC channels in a 20-pin SOIC package, operating within a 0°C to +70°C temperature range and built using lead-free solder.





## ***Customer Support***

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at <http://support.zilog.com>.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at <http://zilog.com/kb> or consider participating in the Zilog Forum at <http://zilog.com/forum>.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at <http://www.zilog.com>.