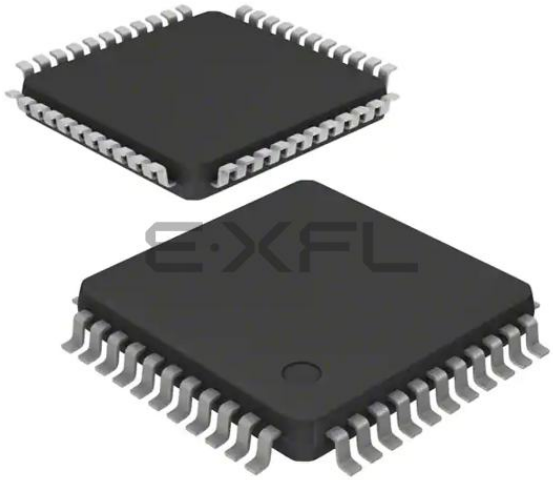

Welcome to E-XFL.COM

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	Active
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f3221an020sg

Revision History

Each instance in the Revision History table reflects a change to this document from its previous revision. For more details, refer to the corresponding pages or appropriate links listed in the table below.

Date	Revision Level	Description	Page
Jan 2013	24	Restored 40-pin PDIP package to Signal and Pin Descriptions and Packaging chapters.	7, 286
Feb 2012	23	Corrected formatting of I_{DDS} section, Table 107; corrected language in the General Purpose RAM section of Appendix A;	202, 248
Sep 2011	22	Revised Flash Sector Protect Register description; revised Packaging chapter.	178, 286
Mar 2008	21	Changed title to Z8 Encore! XP F64xx Series.	All
Feb 2008	20	Changed Z8 Encore! XP 64K Series Flash Microcontrollers to Z8 Encore! XP F64xx Series Flash Microcontrollers. Deleted three sentences that mentioned Z8R642. Removed the 40 PDIP package. Added ZENETSC0100ZACG to the end of the Ordering Information table. Changed the flag status to unaffected for BIT, BSET, and BCLR in the eZ8 CPU Instruction Summary table.	287, 234
Dec 2007	19	Updated Zilog logo, Disclaimer section, and implemented style guide. Updated Table 113. Changed Z8 Encore! 64K Series to Z8 Encore! XP 64K Series Flash Microcontrollers throughout the document.	All
Dec 2006	18	Updated Flash Memory Electrical Characteristics and Timing table and Ordering Information chapter.	213, 287
Nov 2006	17	Updated Part Number Suffix Designations section.	292

Port A–H Address Registers

The Port A–H Address registers, shown in Table 14, select the GPIO port functionality accessible through the Port A–H Control registers. The Port A–H Address and Control registers combine to provide access to all GPIO port control.

Table 14. Port A–H GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

Bit	Description
[7:0] PADDR	**Port Address** This port address selects one of the subregisters accessible through the Port A–H Control Registers. 00H = No function. Provides some protection against accidental port reconfiguration. 01H = Data Direction. 02H = Alternate Function. 03H = Output Control (Open-Drain). 04H = High Drive Enable. 05H = Stop Mode Recovery Source Enable. 06H–FFH = No function.

To avoid missing interrupts, use the coding style in Example 2 to clear bits in the Interrupt Request 0 Register:

Example 2. A good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

Software Interrupt Assertion

Program code can generate interrupts directly. Writing a 1 to the appropriate bit in the Interrupt Request Register triggers an interrupt (assuming that interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the Interrupt Request Register is automatically cleared to 0.

! Caution: Zilog recommends not using a coding style to generate software interrupts by setting bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 3, which follows.

Example 3. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 4 to set bits in the Interrupt Request registers:

Example 4. A good coding style that avoids lost interrupt requests:

```
ORX IRQ0, MASK
```

Interrupt Control Register Definitions

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

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 24, 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.

6. Configure the associated GPIO port pin for the timer input alternate function.
7. Write to the Timer Control 1 Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

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

COMPARE Mode

In COMPARE Mode, the timer counts up to the 16-bit maximum compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the compare value, the timer generates an interrupt and counting continues (the timer value is not reset to 0001H). Also, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) upon compare.

If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting.

Observe the following procedure for configuring a timer for COMPARE Mode and initiating the count:

1. Write to the Timer Control 1 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. If appropriate, enable the timer interrupt 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 1 Register to enable the timer and initiate counting.

In COMPARE Mode, the system clock always provides the timer input. The compare time is calculated using the following equation:

In MULTIPROCESSOR (9-Bit) Mode, the parity bit location (9th bit) becomes the MULTIPROCESSOR control bit. The UART Control 1 and Status 1 registers provide MULTIPROCESSOR (9-Bit) Mode control and status information. If an automatic address matching scheme is enabled, the UART Address Compare Register holds the network address of the device.

MULTIPROCESSOR (9-bit) Mode Receive Interrupts

When MULTIPROCESSOR Mode is enabled, the UART only processes frames addressed to it. The determination of whether a frame of data is addressed to the UART can be made in hardware, software or some combination of the two, depending on the multiprocessor configuration bits. In general, the address compare feature reduces the load on the CPU, since it does not need to access the UART when it receives data directed to other devices on the multinode network. The following three MULTIPROCESSOR modes are available in hardware:

- Interrupt on all address bytes
- Interrupt on matched address bytes and correctly framed data bytes
- 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]. At this point, each new incoming byte interrupts the CPU. The software is then responsible for determining the end of the frame. It checks for 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, then set MPMD[0] 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 is enabled by setting MPMD[1:0] to 10b and writing the UART's address into the UART Address Compare Register. This mode introduces more 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. The first data byte in the frame contains the NEWFRM=1 in the UART Status 1 Register. When the next address byte occurs, the hardware compares it to the UART's address. If there is a match, the interrupts continue sand the NEWFRM bit is set for the first byte of the new frame. If there is no match, then the UART ignores all incoming bytes until the next address match.

SPI Diagnostic State Register

The SPI Diagnostic State Register, shown in Table 68, provides observability of internal state. This register is a read-only register that is used for SPI diagnostics.

Table 68. SPI Diagnostic State Register (SPIDST)

Bit	7	6	5	4	3	2	1	0
Field	SCKEN	TCKEN	SPISTATE					
RESET	0							
R/W	R							
Address	F64H							

Bit	Description
[7] SCKEN	**Shift Clock Enable** 0 = The internal Shift Clock Enable signal is deasserted. 1 = The internal Shift Clock Enable signal is asserted (shift register is updates on next system clock).
[6] TCKEN	**Transmit Clock Enable** 0 = The internal Transmit Clock Enable signal is deasserted. 1 = The internal Transmit Clock Enable signal is asserted. When this is asserted the serial data out is updated on the next system clock (MOSI or MISO).
[5:0] SPISTATE	**SPI State Machine** Defines the current state of the internal SPI State Machine.

SPI Baud Rate High and Low Byte Registers

The SPI Baud Rate High and Low Byte registers, shown in Tables 69 and 70, combine to form a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator.

When configured as a general purpose timer, the SPI BRG interrupt interval is calculated using the following equation:

SPI BRG Interrupt Interval (s) = System Clock Period (s) × BRG[15:0]

DMA*x* Control Register

The DMA*x* Control Register, shown in Table 78, enables and selects the mode of operation for DMA*x*.

Table 78. DMA*x* Control Register (DMA*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	DEN	DLE	DDIR	IRQEN	WSEL	RSS		
RESET	0							
R/W	R/W							
Address	FB0H, FB8H							

Bit	Description
[7] DEN	**DMA*x* Enable** 0 = DMA*x* is disabled and data transfer requests are disregarded. 1 = DMA*x* is enabled and initiates a data transfer upon receipt of a request from the trigger source.
[6] DLE	**DMA*x* Loop Enable** 0 = DMA*x* reloads the original Start Address and is then disabled after the End Address data is transferred. 1 = DMA*x*, after the End Address data is transferred, reloads the original Start Address and continues operating.
[5] DDIR	**DMA*x* Data Transfer Direction** 0 = Register File → on-chip peripheral control register. 1 = On-chip peripheral control → Register File.
[4] IRQEN	**DMA*x* Interrupt Enable** 0 = DMA*x* does not generate any interrupts. 1 = DMA*x* generates an interrupt when the End Address data is transferred.

Electrical Characteristics

The data in this chapter represents all known data prior to qualification and characterization of the Z8 Encore! XP F64xx Series of products, and is therefore subject to change. Additional electrical characteristics may be found in the individual chapters of this document.

Absolute Maximum Ratings

Stresses greater than those listed in Table 106 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, unused inputs must be tied to one of the supply voltages (V_{DD} or V_{SS}).

Table 106. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units	Notes
Ambient temperature under bias	–40	+125	C	
Storage temperature	–65	+150	C	
Voltage on any pin with respect to V_{SS}	–0.3	+5.5	V	1
Voltage on V_{DD} pin with respect to V_{SS}	–0.3	+3.6	V	
Maximum current on input and/or inactive output pin	–5	+5	µA	
Maximum output current from active output pin	–25	+25	mA	
80-pin QFP maximum ratings at –40°C to 70°C				
Total power dissipation		550	mW	
Maximum current into V_{DD} or out of V_{SS}		150	mA	
80-pin QFP maximum ratings at 70°C to 125°C				
Total power dissipation		200	mW	
Maximum current into V_{DD} or out of V_{SS}		56	mA	
68-pin PLCC maximum ratings at –40°C to 70°C				
Total power dissipation		1000	mW	
Maximum current into V_{DD} or out of V_{SS}		275	mA	
68-pin PLCC maximum ratings at 70°C to 125°C				
Total power dissipation		500	mW	
Maximum current into V_{DD} or out of V_{SS}		140	mA	

Table 107. DC Characteristics (Continued)

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
I_{DDS}	Stop Mode Supply Current; GPIO pins configured as outputs (see Figure 47 on page 209 and Figure 48 on page 210)	–	520		µA	VBO and WDT enabled
				700		V_{DD} = 3.6V
				650		V_{DD} = 3.3V
		–	10		µA	VBO disabled, WDT enabled, T_A = 0 to 70ºC
				25		V_{DD} = 3.6V
				20		V_{DD} = 3.3V
		–	–		µA	VBO disabled, WDT enabled, T_A = –40 to +105ºC
				80		V_{DD} = 3.6V
				70		V_{DD} = 3.3V
		–	–		µA	VBO disabled, WDT enabled, T_A = –40 to +125ºC
				250		V_{DD} = 3.6V
				150		V_{DD} = 3.3V

Notes:
1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.

General-Purpose I/O Port Output Timing

Figure 51 and Table 116 provide timing information for GPIO port pins.

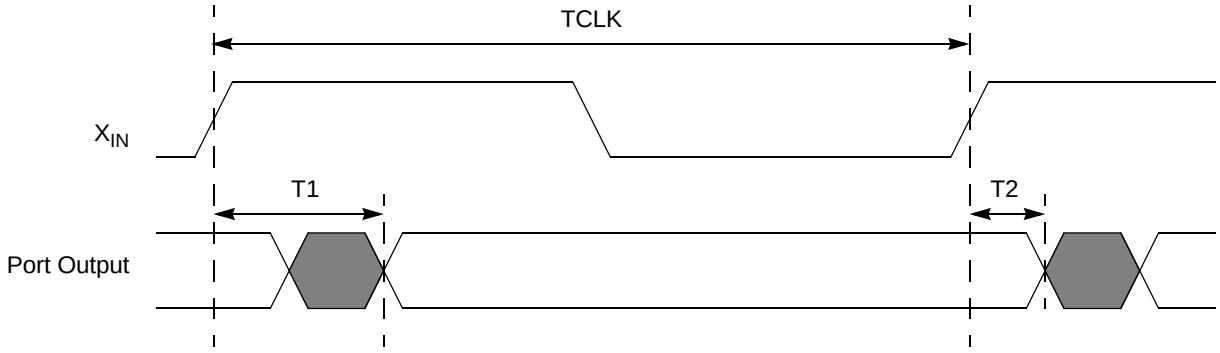

Figure 51. GPIO Port Output Timing

Table 116. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
GPIO port pins			
T_1	X_{IN} Rise to Port Output Valid Delay	–	20
T_2	X_{IN} Rise to Port Output Hold Time	2	–

eZ8 CPU Instruction Set

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

Assembly Language Programming Introduction

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

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

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

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

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

Table 136. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	03							2	4
		R	R	04							3	3
		R	IR	05							3	4
		R	IM	06							3	3
		IR	IM	07							3	4
ADDX dst, src	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	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	dst ← 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	dst[bit] ← 0	r		E2	–	–	–	–	–	–	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	–	–	–	–	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	–	–	–	–	–	–	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Hex Address: F11

Table 155. Timer 0–3 Low Byte Register (TxL)

Bit	7	6	5	4	3	2	1	0
Field	TL							
RESET	0							1
R/W	R/W							
Address	F01H, F09H, F11H, F19H							

Hex Address: F12

Table 156. Timer 0–3 Reload High Byte Register (TxRH)

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1							
R/W	R/W							
Address	F02H, F0AH, F12H, F1AH							

Hex Address: F13

Table 157. Timer 0–3 Reload Low Byte Register (TxRL)

Bit	7	6	5	4	3	2	1	0
Field	TRL							
RESET	1							
R/W	R/W							
Address	F03H, F0BH, F13H, F1BH							

Hex Address: F14

Table 158. Timer 0–3 PWM High Byte Register (TxPWMH)

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	0							
R/W	R/W							
Address	F04H, F0CH, F14H, F1CH							

Hex Address: F43

Table 174. UART Control 1 Register (U*x*CTL1)

Bit	7	6	5	4	3	2	1	0
Field	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
RESET	0							
R/W	R/W							
Address	F43H and F4BH							

Hex Address: F44

Table 175. UART Status 1 Register (U*x*STAT1)

Bit	7	6	5	4	3	2	1	0
Field	Reserved						NEWFRM	MPRX
RESET	0							
R/W	R				R/W		R	
Address	F44H and F4CH							

Hex Address: F45

Table 176. UART Address Compare Register (U*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
RESET	0							
R/W	R/W							
Address	F45H and F4DH							

Hex Address: F46

Table 177. UART Baud Rate High Byte Register (U*x*BRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	1							
R/W	R/W							
Address	F46H and F4EH							

Hex Address: FC5

Table 222. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bit	7	6	5	4	3	2	1	0
Field	PAD7ENL	PAD6ENL	PAD5ENL	PAD4ENL	PAD3ENL	PAD2ENL	PAD1ENL	PAD0ENL
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	FC5H							

Hex Address: FC6

Table 223. Interrupt Request 2 Register (IRQ2)

Bit	7	6	5	4	3	2	1	0
Field	T3I	U1RXI	U1TXI	DMAI	PC3I	PC2I	PC1I	PC0I
RESET	0							
R/W	R/W							
Address	FC6H							

Hex Address: FC7

Table 224. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	7	6	5	4	3	2	1	0
Field	T3ENH	U1RENH	U1TENH	DMAENH	C3ENH	C2ENH	C1ENH	C0ENH
RESET	0							
R/W	R/W							
Address	FC7H							

Hex Address: FC8

Table 225. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	7	6	5	4	3	2	1	0
Field	T3ENL	U1RENL	U1TENL	DMAENL	C3ENL	C2ENL	C1ENL	C0ENL
RESET	0							
R/W	R/W							
Address	FC8H							

Table 271. Z8 Encore! XP F64xx Series Ordering Matrix

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	I^2C	SPI	UARTs with IrDA	Description
Z8F242x with 24KB Flash, 10-Bit Analog-to-Digital Converter										
Standard Temperature: 0°C to 70°C										
Z8F2421PM020SG	24KB	2KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F2421AN020SG	24KB	2KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F2421VN020SG	24KB	2KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F2422AR020SG	24KB	2KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F2422VS020SG	24KB	2KB	46	24	4	12	1	1	2	PLCC 68-pin package
Extended Temperature: –40°C to 105°C										
Z8F2421PM020EG	24KB	2KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F2421AN020EG	24KB	2KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F2421VN020EG	24KB	2KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F2422AR020EG	24KB	2KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F2422VS020EG	24KB	2KB	46	24	4	12	1	1	2	PLCC 68-pin package
Automotive/Industrial Temperature: –40°C to 125°C										
Z8F2421PM020AG	24KB	2KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F2421AN020AG	24KB	2KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F2421VN020AG	24KB	2KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F2422AR020AG	24KB	2KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F2422VS020AG	24KB	2KB	46	24	4	12	1	1	2	PLCC 68-pin package

Part Number Suffix Designations

Zilog part numbers consist of a number of component. In the following example, part number Z8F6421AN020SG is an 8-bit Flash MCU with 4KB of program memory in a 44-pin LQFP package, operating with a maximum 20MHz external clock frequency over a 0ºC to +70ºC temperature range and built using environmentally friendly (lead-free) solder.