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, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0421hh020eg

Revision History

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

Date	Revision Level	Description	Page
Oct 2011	18	Added LDWX information to Load Instructions table, eZ8 CPU Instruction Summary table and to Second Op Code Map after 1FH figure; revised Flash Sector Protect Register description; revised Packaging chapter.	206, 212, 220, 152, 221
May 2008	17	Removed *Flash Microcontrollers* from the title throughout the document.	All
Feb 2008	16	Updated the flag status for BCLR, BIT, and BSET in eZ8 CPU Instruction Summary table.	208
Dec 2007	15	Updated Zilog logo/text, Foreword section. Updated Z8 Encore! 8K Series to Z8 Encore! XP® F0822 Series Flash Microcontrollers throughout the document.	All

Pin Characteristics

Table 4 provides detailed information about the characteristics for each pin available on Z8 Encore! XP® F0822 Series products. The data in Table 4 is sorted alphabetically by pin symbol mnemonic.

Table 4. Pin Characteristics

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tri-State Output	Internal Pull-Up or Pull-Down	Schmitt-Trigger Input	Open Drain Output
AV_{DD}	N/A	N/A	N/A	N/A	No	No	N/A
AV_{SS}	N/A	N/A	N/A	N/A	No	No	N/A
DBG	I/O	I	N/A	Yes	No	Yes	Yes
PA[7:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable
PB[4:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable
PC[5:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable
RESET	I	I	Low	N/A	Pull-up	Yes	N/A
V_{DD}	N/A	N/A	N/A	N/A	No	No	N/A
V_{REF}	Analog	N/A	N/A	N/A	No	No	N/A
V_{SS}	N/A	N/A	N/A	N/A	No	No	N/A
X_{IN}	I	I	N/A	N/A	No	No	N/A
X_{OUT}	O	O	N/A	No	No	No	No

Address Space

The eZ8 CPU accesses three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, Peripheral, and GPIO Port Control registers
- The Program Memory contains addresses for all memory locations having executable code and/or data
- The Data Memory contains addresses for all memory locations that hold data only

These three address spaces are covered briefly in the following sections. For more information about the eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download at www.zilog.com.

Register File

The Register File address space in the Z8 Encore! XP® F0822 Series is 4KB (4096 bytes). It is composed of two sections: Control registers and General-Purpose registers. When instructions are executed, registers are read from when defined as sources and written to when defined as destinations. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 1KB Register File address space is reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256-byte Control Register section is reserved (unavailable). Reading from the reserved Register File addresses returns an undefined value. Writing to reserved Register File addresses is not recommended by Zilog because it can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. Z8 Encore! XP® F0822 Series contains 1KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

Table 42. Timer 0–1 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							

Bit	Description
[7] TRH, TRL	**Timer Reload Register High and Low** These two bytes form the 16-bit reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to 0001H. In COMPARE Mode, these two bytes form the 16-bit Compare value.

Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers, shown in Tables 43 and 44, are used for Pulse-Width Modulator (PWM) operations. These registers also store the Capture values for the CAPTURE and CAPTURE/COMPARE modes.

Table 43. Timer 0–1 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							

Table 44. Timer 0–1 PWM Low Byte Register (TxPWML)

Bit	7	6	5	4	3	2	1	0
Field	PWML							
RESET	0							
R/W	R/W							
Address	F05H, F0DH							

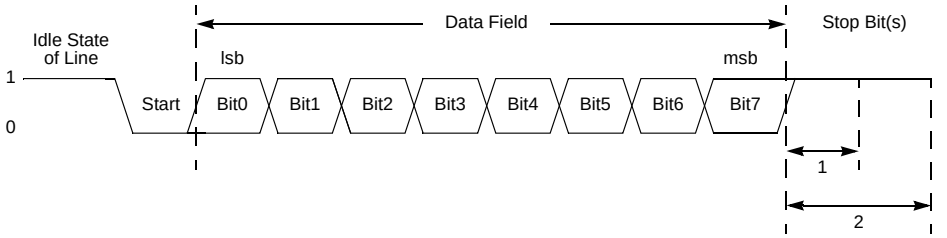


Figure 12. UART Asynchronous Data Format without Parity

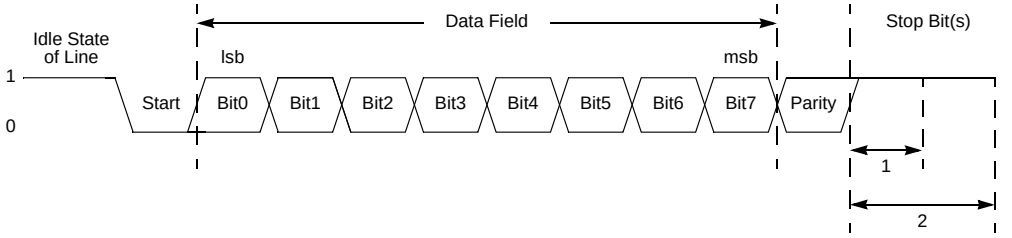


Figure 13. UART Asynchronous Data Format with Parity

Transmitting Data using Polled Method

Observe the following procedure to transmit data using polled method of operation:

1. Write to the UART Baud Rate High Byte and Low Byte registers to set the required baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. If MULTIPROCESSOR Mode is required, write to the UART Control 1 Register to enable multiprocessor (9-bit) mode functions.
   - Set the Multiprocessor Mode Select (MPEN) to enable MULTIPROCESSOR Mode.
4. Write to the UART Control 0 Register to:
   - Set the transmit enable bit (TEN) to enable the UART for data transmission
   - If parity is required, and MULTIPROCESSOR Mode is not enabled, set the parity enable bit (PEN) and select either even or odd parity (PSEL).

Receiving IrDA Data

Data received from the infrared transceiver through the **IR_RXD** signal through the RXD pin is decoded by the infrared endec and passed to the UART. The UART's baud rate clock is used by the infrared endec to generate the demodulated signal (RXD) that drives the UART. Each UART/Infrared data bit is 16-clocks wide. Figure 19 displays data reception. When the infrared endec is enabled, the UART's RXD signal is internal to the Z8 Encore! XP® F0822 Series products while the IR_RXD signal is received through the RXD pin.

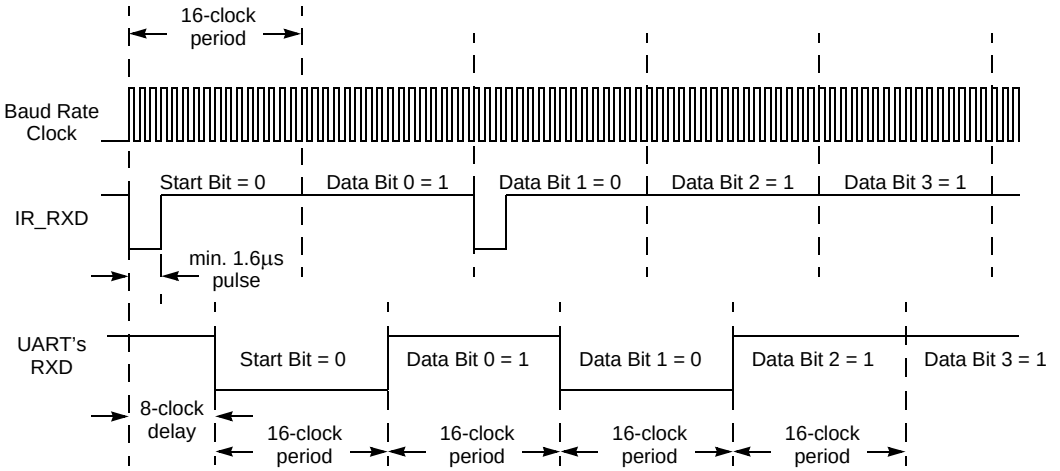


Figure 19. Infrared Data Reception

Caution: The system clock frequency must be at least 1.0 MHz to ensure proper reception of the 1.6 µs minimum width pulses allowed by the IrDA standard.

Endec Receiver Synchronization

The IrDA receiver uses a local baud rate clock counter (0 to 15 clock periods) to generate an input stream for the UART and to create a sampling window for detection of incoming pulses. The generated UART input (UART RXD) is delayed by 8 baud rate clock periods with respect to the incoming IrDA data stream. When a falling edge in the input data stream is detected, the endec counter is reset. When the count reaches a value of 8, the UART RXD value is updated to reflect the value of the decoded data. When the count reaches 12 baud clock periods, the sampling window for the next incoming pulse opens. The window remains open until the count again reaches 8 (or, in other words, 24 baud

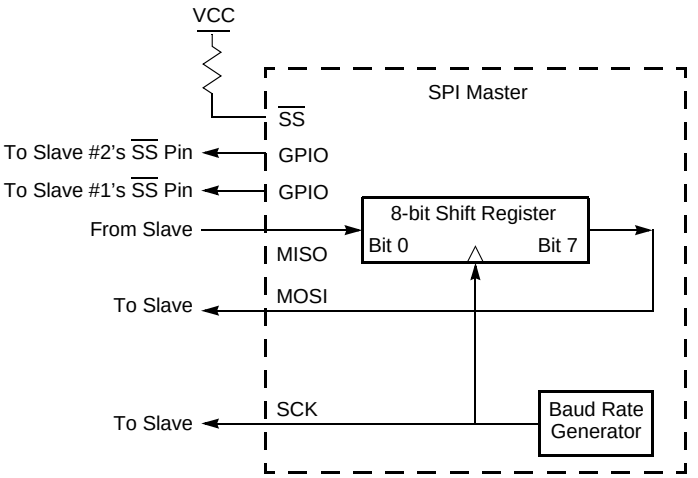


Figure 21. SPI Configured as a Master in a Single Master, Multiple Slave System

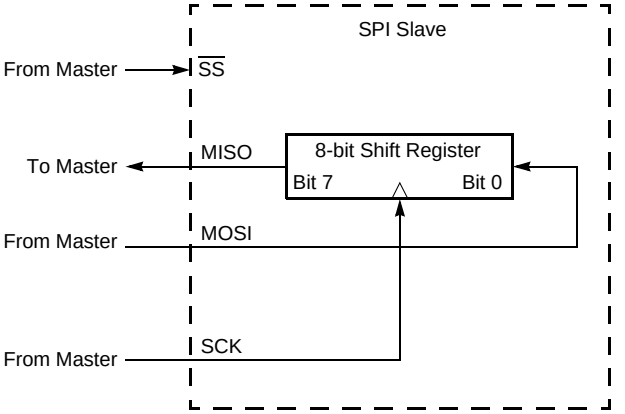


Figure 22. SPI Configured as a Slave

16. If the I²C Slave sends an acknowledge by pulling the SDA signal Low during the next High period of SCL, the I²C Controller sets the ACK bit in the I²C Status Register. Continue to Step 17.

    If the slave does not acknowledge the second address byte or one of the data bytes, the I²C Controller sets the NCKI bit and clears the ACK bit in the I²C Status Register. Software responds to the Not Acknowledge interrupt by setting the stop and flush bits and clearing the TXI bit. The I²C Controller sends the stop condition on the bus and clears the stop and NCKI bits. The transaction is complete; ignore the remainder of this sequence.
17. The I²C Controller shifts the data out by the SDA signal. After the first bit is sent, the transmit interrupt is asserted.
18. If more bytes remain to be sent, return to Step 14.
19. If the last byte is currently being sent, software sets the stop bit of the I²C Control Register (or start bit to initiate a new transaction). In the stop case, software simultaneously clears the TXI bit of the I²C Control Register.
20. The I²C Controller completes transmission of the last data byte on the SDA signal.
21. The slave can either Acknowledge or Not Acknowledge the last byte. Because either the stop or start bit is already set, the NCKI interrupt does not occur.
22. The I²C Controller sends the stop (or restart) condition to the I²C bus and clears the stop (or start) bit.

Read Transaction with a 7-Bit Address

Figure 30 displays the data transfer format for a read operation to a 7-bit addressed slave. The shaded regions indicate data transferred from the I²C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I²C Controller.

S	Slave Address	R = 1	A	Data	A	Data	A	P/S

Figure 30. Receive Data Transfer Format for a 7-Bit Addressed Slave

Observe the following procedure for a read operation to a 7-bit addressed slave:

1. Software writes the I²C Data Register with a 7-bit Slave address plus the read bit (=1).
2. Software asserts the start bit of the I²C Control Register.
3. If this transfer is a single byte transfer, Software asserts the NAK bit of the I²C Control Register so that after the first byte of data has been read by the I²C Controller, a Not Acknowledge is sent to the I²C Slave.
4. The I²C Controller sends the start condition.

I²C Control Register

The I²C Control Register, shown in Table 73, enables I²C operation.

Table 73. I²C Control Register (I2CCTL)

Bit	7	6	5	4	3	2	1	0
Field	IEN	START	STOP	BIRQ	TXI	NAK	FLUSH	FILTEN
RESET	0							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	W	R/W
Address	F52H							

Bit	Description
[7] IEN	**I²C Enable** 1 = The I²C transmitter and receiver are enabled. 0 = The I²C transmitter and receiver are disabled.
[6] START	**Send Start Condition** This bit sends the start condition. After it is asserted, it is cleared by the I²C Controller after it sends the START condition or if the IEN bit is deasserted. If this bit is 1, it cannot be cleared to 0 by writing to the register. After this bit is set, the Start condition is sent if there is data in the I²C Data or I²C Shift Register. If there is no data in one of these registers, the I²C Controller waits until the data register is written. If this bit is set while the I²C Controller is shifting out data, it generates a start condition after the byte shifts and the acknowledge phase completes. If the stop bit is also set, it also waits until the stop condition is sent before sending the start condition.
[5] STOP	**Send Stop Condition** This bit causes the I²C Controller to issue a stop condition after the byte in the I²C Shift Register has completed transmission or after a byte is received in a receive operation. After it is set, this bit is reset by the I²C Controller after a stop condition is sent or by deasserting the IEN bit. If this bit is 1, it cannot be cleared to 0 by writing to the register.
[4] BIRQ	**Baud Rate Generator Interrupt Request** This bit allows the I²C Controller to be used as an additional timer when the I²C Controller is disabled. This bit is ignored when the I²C Controller is enabled. 1 = An interrupt occurs every time the BRG counts down to 1. 0 = No BRG interrupt occurs.
[3] TXI	**Enable TDRE interrupts** This bit enables the transmit interrupt when the I²C Data Register is empty (TDRE = 1). 1 = transmit interrupt (and DMA transmit request) is enabled. 0 = transmit interrupt (and DMA transmit request) is disabled.
[2] NAK	**Send NAK** This bit sends a Not Acknowledge condition after the next byte of data is read from the I²C Slave. After it is asserted, it is deasserted after a Not Acknowledge is sent or the IEN bit is deasserted. If this bit is 1, it cannot be cleared to 0 by writing to the register.

Bit	Description (Continued)
[1] FLUSH	**Flush Data** Setting this bit to 1 clears the I2C Data Register and sets the TDRE bit to 1. This bit allows flushing of the I2C Data Register when a Not Acknowledge interrupt is received after the data has been sent to the I2C Data Register. Reading this bit always returns 0.
[0] FILTEN	**I2C Signal Filter Enable** This bit enables low-pass digital filters on the SDA and SCL input signals. These filters reject any input pulse with periods less than a full system clock cycle. The filters introduce a 3-system clock cycle latency on the inputs. 1 = Low-pass filters are enabled. 0 = Low-pass filters are disabled.

I2C Baud Rate High and Low Byte Registers

The I2C Baud Rate High and Low Byte registers, shown in Tables 74 and 75, combine to form a 16-bit reload value, BRG[15:0], for the I2C Baud Rate Generator. When configured as a general purpose timer, the interrupt interval is calculated using the following equation:

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

Table 74. I2C Baud Rate High Byte Register (I2CBRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	FFH							
R/W	R/W							
Address	F53H							

Bit	Description
[7:0] BRH	**I2C Baud Rate High Byte** Most significant byte, BRG[15:8], of the I2C Baud Rate Generator's reload value. **Note:** If the DIAG bit in the I2C Diagnostic Control Register is set to 1, a read of the I2CBRH Register returns the current value of the I2C Baud Rate Counter[15:8].

Table 88. Flash Frequency High Byte Register (FFREQH)

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
RESET	0							
R/W	R/W							
Address	FFAH							

Table 89. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Bit	Description
[7:0] FFREQH, FFREQL	**Flash Frequency High and Low Bytes** These 2 bytes, {FFREQH[7:0], FFREQL[7:0]}, contain the 16-bit Flash Frequency value.

Bit	Description (Continued)
[1]	**Reserved** This bit is reserved and must always be 1.
[0] FWP	**Flash Write Protect*** These two option bits combine to provide three levels of Program memory protection. 0 = Programming, Page Erase, and Mass Erase using User Code is disabled. Mass Erase is available through the OCD. 1 = Programming and Page Erase are enabled for all of Flash program memory.

Note: *Applies only to the Flash versions of the F0822 Series of devices.

Flash Memory Address 0001H

Table 91. Options Bits at Flash Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U							
R/W	R/W							
Address	Program Memory 0001H							

Note: U = Unchanged by Reset; R/W = Read/Write.

Bit	Description
[7:0]	**Reserved** These option bits are reserved and must always be 1. This setting is the default for unprogrammed (erased) Flash.

General Purpose I/O Port Input Data Sample Timing

Figure 48 displays timing of the GPIO Port input sampling. Table 106 lists the GPIO port input timing.

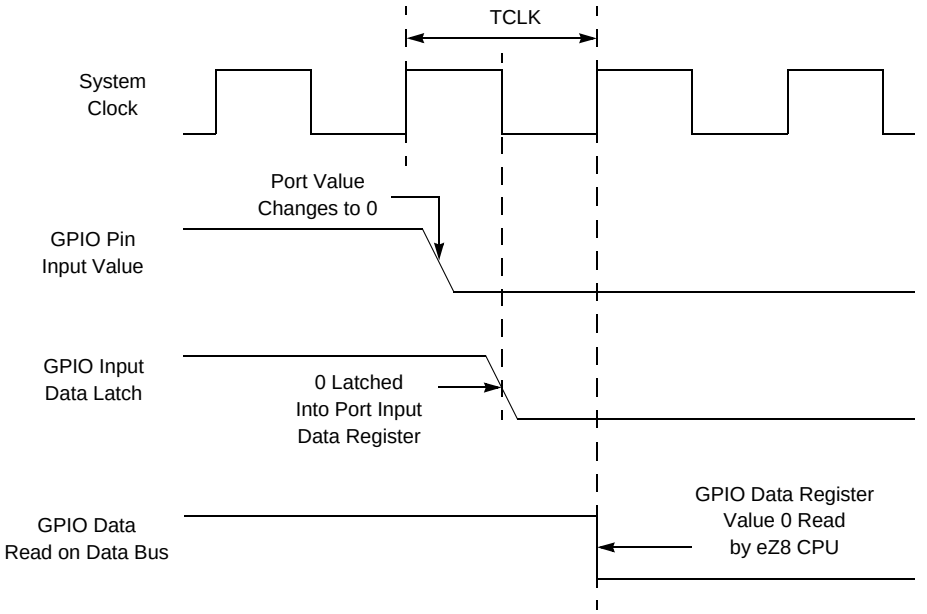


Figure 48. Port Input Sample Timing

Table 106. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_{S_PORT}	Port Input Transition to X_{IN} Fall Setup Time (not pictured)	5	–
T_{H_PORT}	X_{IN} Fall to Port Input Transition Hold Time (not pictured)	5	–
T_{SMR}	GPIO Port Pin Pulse Width to Insure Stop Mode Recovery (for GPIO port pins enabled as SMR sources)	1µs	

Table 116. Notational Shorthand (Continued)

Notation	Description	Operand	Range
RA	Relative Address	X	X represents an index in the range of +127 to –128, which is an offset relative to the address of the next instruction
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14
RR	Register Pair	Reg	*Reg* represents an even number in the range of 00H to FEH
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

Table 117 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 117. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow, as shown in the following example.

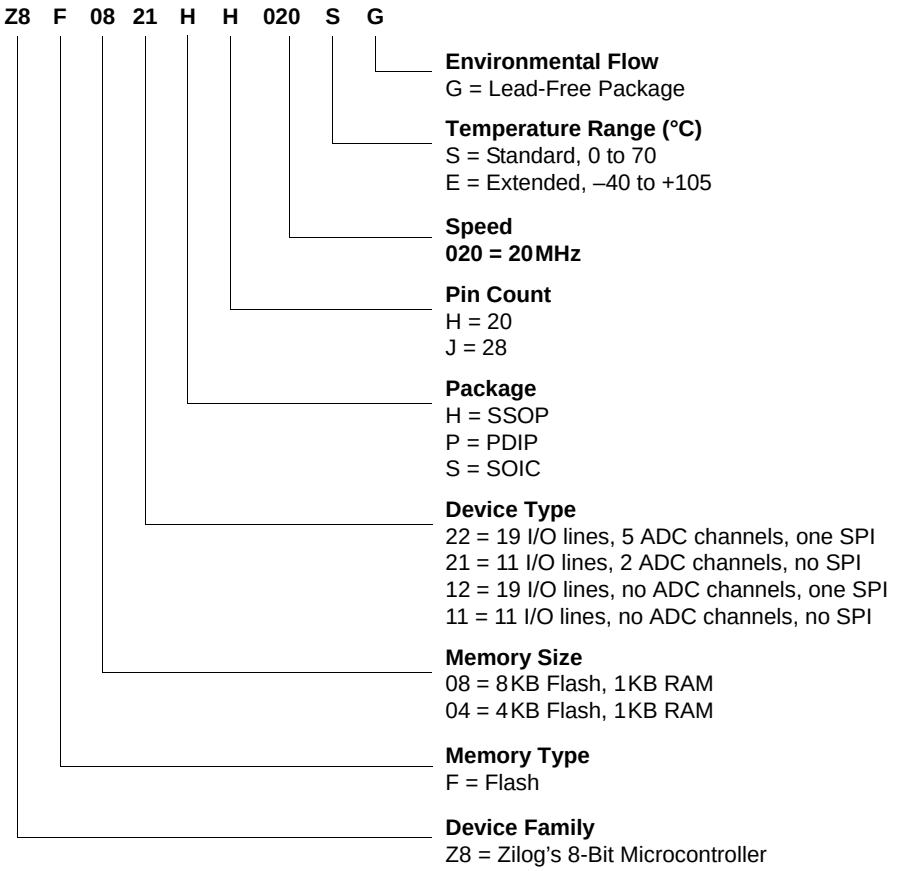
dst ← dst + src

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

Part Number Suffix Designations

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

Example. Part number Z8F0821HH020SG is an 8-bit Flash Motor Controller with 8KB of Program Memory, equipped with 11 I/O lines and 2 ADC channels in a 20-pin SSOP package, operating within a 0ºC to +70ºC temperature range and built using lead-free solder.



Hex Address: F45

Table 152. UART Address Compare Register (U0ADDR)

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

Hex Address: F46

Table 153. UART Baud Rate High Byte Register (U0BRH)

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

Hex Address: F47

Table 154. UART Baud Rate Low Byte Register (U0BRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Address	F47H							

Hex Addresses: F48–F4F

This address range is reserved.

R

S

T

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.