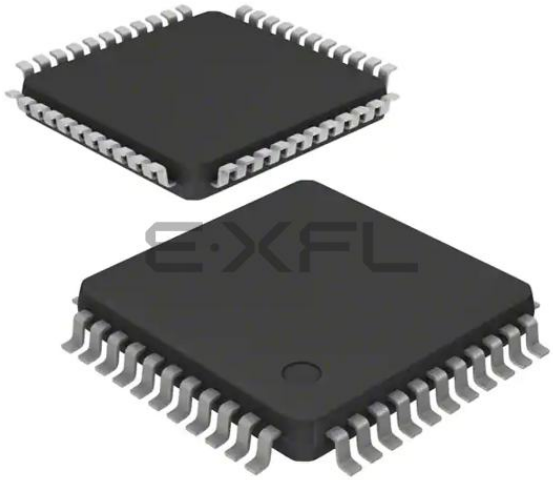


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	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/z8f3221an020sc

Table 4. Pin Characteristics of the Z8 Encore! XP 64K Series Flash Microcontrollers

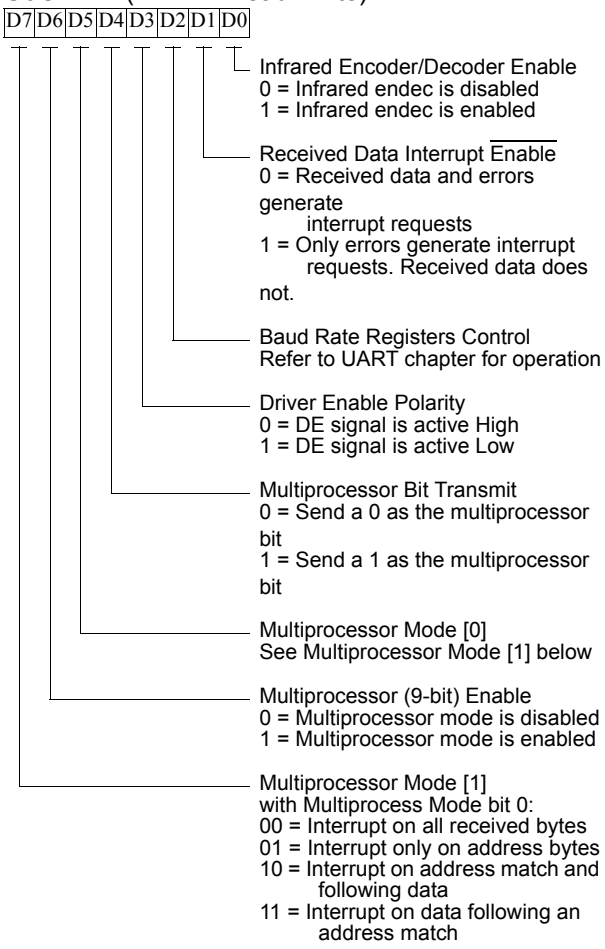
Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tri-State Output	Internal Pull-up or Pull-down	Schmitt-Trigger Input	Open Drain Output
AVSS	N/A	N/A	N/A	N/A	No	No	N/A
AVDD	N/A	N/A	N/A	N/A	No	No	N/A
DBG	I/O	I	N/A	Yes	No	Yes	Yes
VSS	N/A	N/A	N/A	N/A	No	No	N/A
PA[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PB[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PC[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PD[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PE7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PF[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PG[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PH[3:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
RESET	I	I	Low	N/A	Pull-up	Yes	N/A
VDD	N/A	N/A	N/A	N/A	No	No	N/A
XIN	I	I	N/A	N/A	No	No	N/A
XOUT	O	O	N/A	Yes, in STOP mode	No	No	No

Note: *x* represents integer 0, 1,... to indicate multiple pins with symbol mnemonics that differ only by the integer.

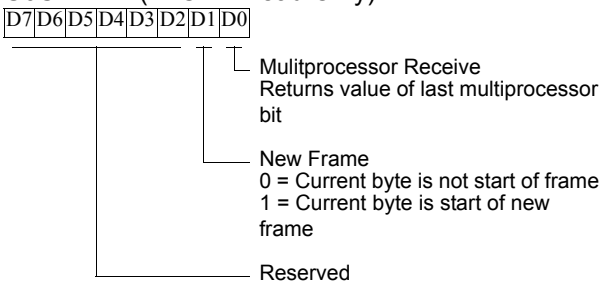
Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FCD	Interrupt Edge Select	IRQES	00	78
FCE	Interrupt Port Select	IRQPS	00	78
FCF	Interrupt Control	IRQCTL	00	79
GPIO Port A				
FD0	Port A Address	PAADDR	00	61
FD1	Port A Control	PACTL	00	62
FD2	Port A Input Data	PAIN	XX	66
FD3	Port A Output Data	PAOUT	00	66
GPIO Port B				
FD4	Port B Address	PBADDR	00	61
FD5	Port B Control	PBCTL	00	62
FD6	Port B Input Data	PBIN	XX	66
FD7	Port B Output Data	PBOUT	00	66
GPIO Port C				
FD8	Port C Address	PCADDR	00	61
FD9	Port C Control	PCCTL	00	62
FDA	Port C Input Data	PCIN	XX	66
FDB	Port C Output Data	PCOUT	00	66
GPIO Port D				
FDC	Port D Address	PDADDR	00	61
FDD	Port D Control	PDCTL	00	62
FDE	Port D Input Data	PDIN	XX	66
FDF	Port D Output Data	PDOUT	00	66
GPIO Port E				
FE0	Port E Address	PEADDR	00	61
FE1	Port E Control	PECTL	00	62
FE2	Port E Input Data	PEIN	XX	66
FE3	Port E Output Data	PEOUT	00	66
GPIO Port F				
FE4	Port F Address	PFADDR	00	61
FE5	Port F Control	PFCTL	00	62
FE6	Port F Input Data	PFIN	XX	66
FE7	Port F Output Data	PFOUT	00	66
GPIO Port G				
FE8	Port G Address	PGADDR	00	61
FE9	Port G Control	PGCTL	00	62
FEA	Port G Input Data	PGIN	XX	66
FEB	Port G Output Data	PGOUT	00	66
GPIO Port H				
FEC	Port H Address	PHADDR	00	61
FED	Port H Control	PHCTL	00	62
FEE	Port H Input Data	PHIN	XX	66

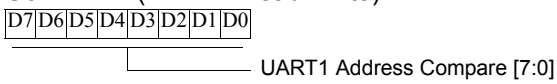
UART1 Control 1
U0CTL1 (F4BH - Read/Write)



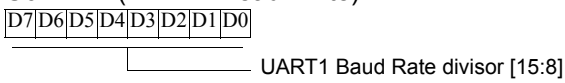
UART1 Status 1
U0STAT1 (F4CH - Read Only)



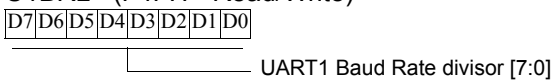
UART1 Address Compare
U0ADDR (F4DH - Read/Write)



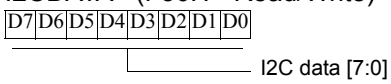
UART1 Baud Rate Generator High Byte
U0BRH (F4EH - Read/Write)



UART1 Baud Rate Generator Low Byte
U1BRL (F4FH - Read/Write)



I^2C Data
I2CDATA (F50H - Read/Write)



HALT Mode

Execution of the eZ8 CPU's HALT instruction places the device into HALT mode. In HALT mode, the operating characteristics are:

- Primary crystal 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.
- The Watchdog Timer continues to operate, if enabled.
- 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 Brownout Reset
- External $\overline{\text{RESET}}$ pin assertion

To minimize current in HALT mode, all GPIO pins which are configured as inputs must be driven to one of the supply rails (V_{CC} or GND).

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

1. Write to the Timer Control 1 register to:
   - Disable the timer
   - Configure the timer for PWM 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 desired, 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 1 register to enable the timer and initiate counting.

The PWM period is given 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 must be used 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 given 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 given by:

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

CAPTURE Mode

In CAPTURE mode, the current timer count value is recorded when the desired external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control 1 register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the timer continues counting.

– Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.

5. Check the TDRE bit in the UART Status 0 register to determine if the Transmit Data register is empty (indicated by a 1). If empty, continue to step 6. If the Transmit Data register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data register becomes available to receive new data.
6. Write the UART Control 1 register to select the outgoing address bit.
7. Set the MULTIPROCESSOR Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
8. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.
9. If desired and MULTIPROCESSOR mode is enabled, make any changes to the MULTIPROCESSOR Bit Transmitter (MPBT) value.
10. To transmit additional bytes, return to step 5.

Transmitting Data using the Interrupt-Driven Method

The UART transmitter interrupt indicates the availability of the Transmit Data register to accept new data for transmission. Follow the steps below to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the desired baud rate.
2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the desired priority.
5. If MULTIPROCESSOR mode is desired, write to the UART Control 1 register to enable MULTIPROCESSOR (9-bit) mode functions.
6. Set the MULTIPROCESSOR Mode Select (MPEN) to Enable MULTIPROCESSOR mode.
7. Write to the UART Control 0 register to:
   – Set the transmit enable bit (TEN) to enable the UART for data transmission.
   – Enable parity, if desired and if MULTIPROCESSOR mode is not enabled, and select either even or odd parity.
   – Set or clear the CTSE bit to enable or disable control from the remote receiver via the $\overline{\text{CTS}}$ pin.

PE—Parity Error
This bit indicates that a parity error has occurred. Reading the UART Receive Data register clears this bit.
0 = No parity error occurred.
1 = A parity error occurred.

OE—Overrun Error
This bit indicates that an overrun error has occurred. An overrun occurs when new data is received and the UART Receive Data register has not been read. If the RDA bit is reset to 0, then reading the UART Receive Data register clears this bit.
0 = No overrun error occurred.
1 = An overrun error occurred.

FE—Framing Error
This bit indicates that a framing error (no Stop bit following data reception) was detected. Reading the UART Receive Data register clears this bit.
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 zeros then 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.

UART Status 1 Register

This register contains multiprocessor control and status bits.

Table 64. SPI Control Register (SPICTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN
RESET	0							
R/W	R/W							
ADDR	F61H							

IRQE—Interrupt Request Enable
0 = SPI interrupts are disabled. No interrupt requests are sent to the Interrupt Controller.
1 = SPI interrupts are enabled. Interrupt requests are sent to the Interrupt Controller.

STR—Start an SPI Interrupt Request
0 = No effect.
1 = Setting this bit to 1 also sets the IRQ bit in the SPI Status register to 1. Setting this bit forces the SPI to send an interrupt request to the Interrupt Control. This bit can be used by software for a function similar to transmit buffer empty in a UART. Writing a 1 to the IRQ bit in the SPI Status register clears this bit to 0.

BIRQ—BRG Timer Interrupt Request
If the SPI is enabled, this bit has no effect. If the SPI is disabled:
0 = The Baud Rate Generator timer function is disabled.
1 = The Baud Rate Generator timer function and time-out interrupt are enabled.

PHASE—Phase Select
Sets the phase relationship of the data to the clock. For more information on operation of the PHASE bit, see SPI Clock Phase and Polarity Control on page 132.

CLKPOL—Clock Polarity
0 = SCK idles Low (0).
1 = SCK idle High (1).

WOR—Wire-OR (OPEN-DRAIN) Mode Enabled
0 = SPI signal pins not configured for open-drain.
1 = All four SPI signal pins (SCK, $\overline{SS}$, MISO, MOSI) configured for open-drain function. This setting is typically used for multi-master and/or multi-slave configurations.

MMEN—SPI Master Mode Enable
0 = SPI configured in Slave mode.
1 = SPI configured in Master mode.

SPIEN—SPI Enable
0 = SPI disabled.
1 = SPI enabled.

Follow the steps below for the data transfer for a read operation to a 10-bit addressed slave:

1. Software writes 11110B followed by the two address bits and a 0 (write) to the I2C Data register.
2. Software asserts the START and TXI bits of the I2C Control register.
3. The I2C Controller sends the Start condition.
4. The I2C Controller loads the I2C Shift register with the contents of the I2C Data register.
5. After the first bit has been shifted out, a Transmit interrupt is asserted.
6. Software responds by writing the lower eight bits of address to the I2C Data register.
7. The I2C Controller completes shifting of the two address bits and a 0 (write).
8. If the I2C slave acknowledges the first address byte 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 with step 9.

   If the slave does not acknowledge the first address byte, 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 following steps).
9. The I2C Controller loads the I2C Shift register with the contents of the I2C Data register (second address byte).
10. The I2C Controller shifts out the second address byte. After the first bit is shifted, the I2C Controller generates a Transmit interrupt.
11. Software responds by setting the START bit of the I2C Control register to generate a repeated START and by clearing the TXI bit.
12. Software responds by writing 11110B followed by the 2-bit slave address and a 1 (read) to the I2C Data register.
13. If only one byte is to be read, software sets the NAK bit of the I2C Control register.
14. After the I2C Controller shifts out the 2nd address byte, 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 with step 15.

    If the slave does not acknowledge the second address byte, 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 following steps).

5. Re-write the page written in step 2 to the Page Select register.
6. Write the Page Erase command 95H to the Flash Control register.

Mass Erase

The Flash memory cannot be Mass Erased by user code.

Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Programming algorithms by controlling the Flash programming signals directly.

Flash Controller Bypass is recommended for gang programming applications and large volume customers who do not require in-circuit programming of the Flash memory.

For more information on bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore!* available for download at www.zilog.com.

Flash Controller Behavior in Debug Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored.
- The Flash Sector Protect register is ignored for programming and erase operations.
- Programming operations are not limited to the page selected in the Page Select register.
- Bits in the Flash Sector Protect register can be written to one or zero.
- The second write of the Page Select register to unlock the Flash Controller is not necessary.
- The Page Select register can be written when the Flash Controller is unlocked.
- The Mass Erase command is enabled through the Flash Control register.

Caution: *For security reasons, Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the Flash controller must go through the unlock sequence again to select another page.*

Table 106. DC Characteristics (Continued)

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

[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.

I²C Timing

Figure 55 and Table 119 provide timing information for I²C pins.

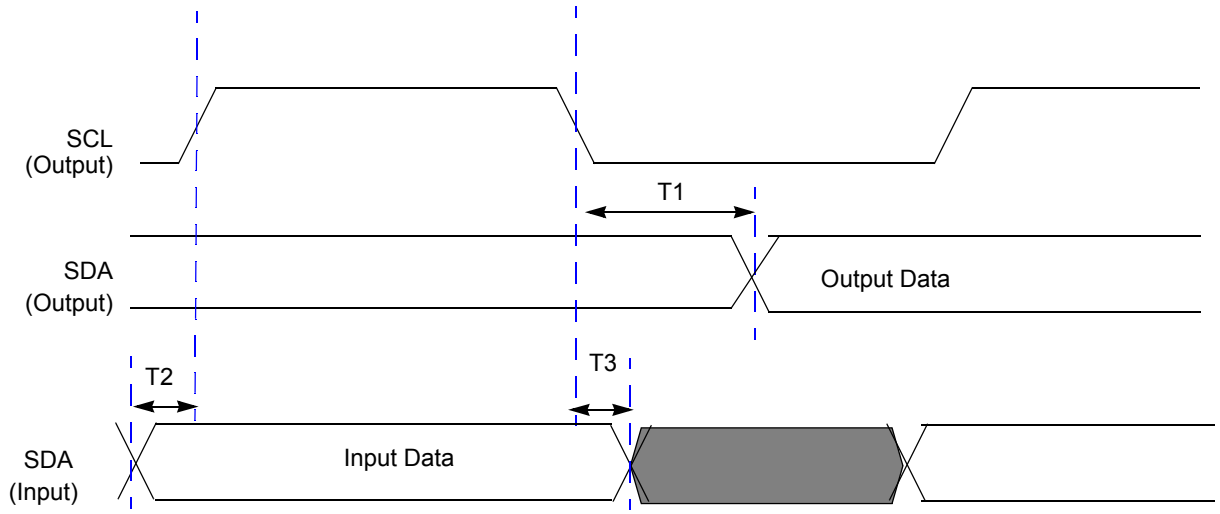


Figure 55. I²C Timing

Table 119. I²C Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
I²C			
T_1	SCL Fall to SDA output delay		SCL period/4
T_2	SDA Input to SCL rising edge Setup Time	0	
T_3	SDA Input to SCL falling edge Hold Time	0	

```
; value 01H, is the source. The value 01H is written into the
; Register at address 234H.
```

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as ‘destination, source’. After assembly, the object code usually has the operands in the order ‘source, destination’, but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed if you prefer manual program coding or intend to implement your own assembler.

Example 1: If the contents of Registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2: In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0 - R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

Refer to the device-specific Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status Flags, and address modes are represented by a notational shorthand that is described in Table 122.

Table 122. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B).
cc	Condition Code	—	Refer to Condition Codes overview in the eZ8 CPU User Manual.
DA	Direct Address	Addrs	Addrs. represents a number in the range of 0000H to FFFFH
ER	Extended Addressing Register	Reg	Reg. represents a number in the range of 000H to FFFH
IM	Immediate Data	#Data	Data is a number between 00H to FFH
Ir	Indirect Working Register	@Rn	n = 0 –15
IR	Indirect Register	@Reg	Reg. represents a number in the range of 00H to FFH
Irr	Indirect Working Register Pair	@RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14
IRR	Indirect Register Pair	@Reg	Reg. represents an even number in the range 00H to FEH
p	Polarity	p	Polarity is a single bit binary value of either 0B or 1B.
r	Working Register	Rn	n = 0 – 15
R	Register	Reg	Reg. represents a number in the range of 00H to FFH
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 123 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 123. 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. For example,

dst ← dst + src

indicates the source data is added to the destination data and the result is stored in the destination location.

Condition Codes

The C, Z, S and V Flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the Flag settings are encoded in a 4-bit field called the condition code (cc), which forms Bits 7:4 of the conditional jump instructions. The condition codes are summarized in Table 124. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the Flag test operation decides if the conditional jump is executed.

Table 124. Condition Codes

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	–
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1

Table 125 through Table 132 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instruction can be considered as a subset of more than one category. Within these tables, the source operand is identified as 'src', the destination operand is 'dst' and a condition code is 'cc'.

Table 125. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using Extended Addressing
ADD	dst, src	Add
ADDX	dst, src	Add using Extended Addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
CPX	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
MULT	dst	Multiply
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using Extended Addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using Extended Addressing

Table 126. Bit Manipulation Instructions

Mnemonic	Operands	Instruction
BCLR	bit, dst	Bit Clear
BIT	p, bit, dst	Bit Set or Clear
BSET	bit, dst	Bit Set

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
Z8F482x with 48 KB Flash, 10-Bit Analog-to-Digital Converter										
Standard Temperature: 0 °C to 70 °C										
Z8F4821PM020SC	48 KB	4 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F4821AN020SC	48 KB	4 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F4821VN020SC	48 KB	4 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F4822AR020SC	48 KB	4 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F4822VS020SC	48 KB	4 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F4823FT020SC	48 KB	4 KB	60	24	4	12	1	1	2	QFP 80-pin package
Extended Temperature: –40 °C to +105 °C										
Z8F4821PM020EC	48 KB	4 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F4821AN020EC	48 KB	4 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F4821VN020EC	48 KB	4 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F4822AR020EC	48 KB	4 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F4822VS020EC	48 KB	4 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F4823FT020EC	48 KB	4 KB	60	24	4	12	1	1	2	QFP 80-pin package
Automotive/Industrial Temperature: –40 °C to +125 °C										
Z8F4821PM020AC	48 KB	4 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F4821AN020AC	48 KB	4 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F4821VN020AC	48 KB	4 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F4822AR020AC	48 KB	4 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F4822VS020AC	48 KB	4 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F4823FT020AC	48 KB	4 KB	60	24	4	12	1	1	2	QFP 80-pin package

P

Q

R