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	-
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	17
Program Memory Size	12KB (12K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 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	20-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1233qh020eg

Table of Contents

CPU and Peripheral Overview

The eZ8 CPU, Zilog's latest 8-bit CPU, meets the continuing demand for faster and more 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 CPU code

- Expanded internal register file allows access up to 4 KB

- New instructions improve execution efficiency for code developed using high-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), which is available for download on www.zilog.com.

General Purpose Input/Output

The Z8 Encore! F0830 Series features up to 25 port pins (Ports A–D) for general-purpose input/output (GPIO). The number of GPIO pins available is a function of package. Each pin is individually programmable.

Flash Controller

The Flash Controller programs and erases the Flash memory. It also supports protection against accidental programming and erasure.

The analog supply pins (AV_{DD} and AV_{SS}) are also not available on these parts and are replaced by PB6 and PB7.

At reset, by default, all pins of Port A, B and C are in Input state. The alternate functionality is also disabled, so the pins function as general purpose input ports until programmed otherwise. At power-up, the Port D0 pin defaults to the $\overline{RESET}$ Alternate function.

The pin configurations listed are preliminary and subject to change based on manufacturing limitations.

```
PB1/ANA1        — 1    20 —  PB0/ANA0
PB2/ANA2        — 2    19 —  PC3/COUT/LED
PB3/CLKIN/ANA3  — 3    18 —  PC2/ANA6/LED
V_DD            — 4    17 —  PC1/ANA5/CINN/LED
PA0/T0IN/T0OUT/X_IN — 5 16 —  PC0/ANA4/CINP/LED
PA1/T0OUT/XOUT  — 6    15 —  DBG
V_SS            — 7    14 —  RESET/PD0
PA2             — 8    13 —  PA7/T1OUT
PA3             — 9    12 —  PA6/T1IN/T1OUT
PA4             — 10   11 —  PA5
```
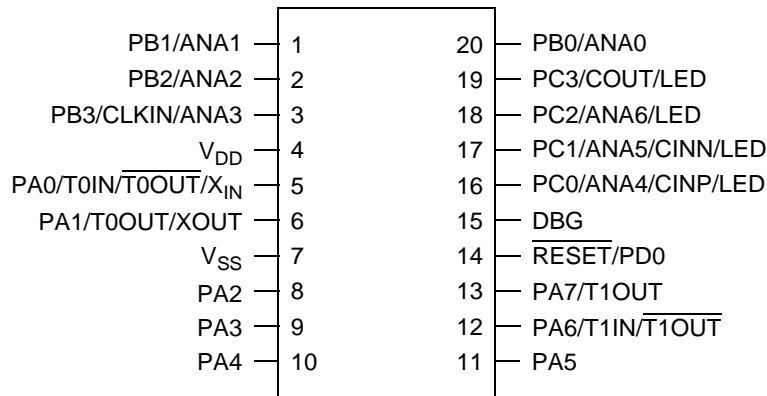
Figure 2. Z8F0830 Series in 20-Pin SOIC, SSOP, PDIP Package

```
PB2/ANA2         — 1    28 —  PB1/ANA1
PB4/ANA7         — 2    27 —  PB0/ANA0
PB5/V_REF        — 3    26 —  PC3/COUT/LED
PB3/CLKIN/ANA3   — 4    25 —  PC2/ANA6/LED
(PB6) AV_DD      — 5    24 —  PC1/ANA5/CINN/LED
V_DD             — 6    23 —  PC0/ANA4/CINP/LED
PA0/T0IN/T0OUT/X_IN — 7 22 —  DBG
PA1/T0OUT/X_OUT  — 8    21 —  RESET/PD0
V_SS             — 9    20 —  PC7/LED
(PB7) AV_SS      — 10   19 —  PC6/LED
PA2              — 11   18 —  PA7/T1OUT
PA3              — 12   17 —  PC5/LED
PA4              — 13   16 —  PC4/LED
PA5              — 14   15 —  PA6/T1IN/T1OUT
```
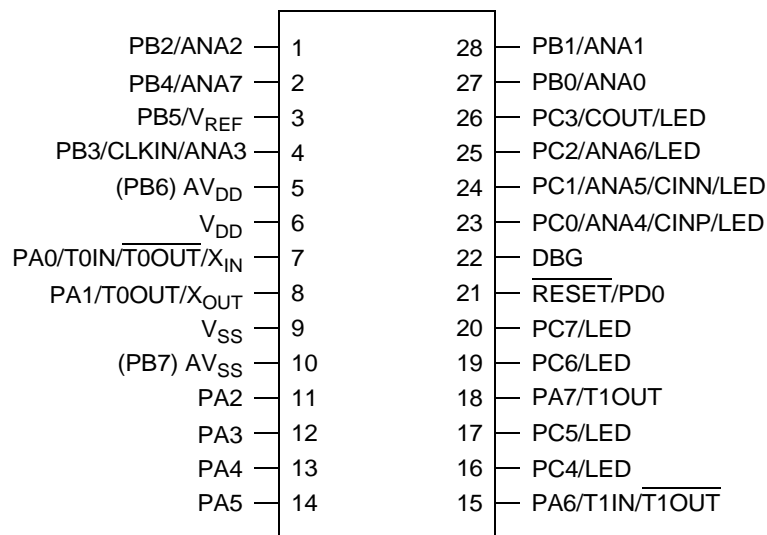
Figure 3. Z8F0830 Series in 28-Pin SOIC, SSOP, PDIP Package

Address Space

The eZ8 CPU can access the following three distinct address spaces:

- The register file addresses access for the general purpose registers and the eZ8 CPU, peripheral and general purpose I/O port control registers

- The program memory addresses access for all of the memory locations having executable code and/or data

- The data memory addresses access for all of the memory locations containing only the data

The following sections describe these three address spaces. 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! MCU is 4 KB (4096 bytes). The register file consists of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as *source* are read and registers defined as *destinations* are written. 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 4 KB register file address space are reserved for controlling the eZ8 CPU, on-chip peripherals and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256B Control Register section are reserved (unavailable). Reading from a reserved register file address returns an undefined value. Writing to reserved register file addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the register file address space. The Z8 Encore! F0830 Series devices contain up to 256B 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 has no effect.

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No.
Interrupt Controller (cont'd)				
FCE	Shared interrupt select	IRQSS	00	66
FCF	Interrupt control	IRQCTL	00	67
GPIO Port A				
FD0	Port A address	PAADDR	00	39
FD1	Port A control	PACTL	00	41
FD2	Port A input data	PAIN	XX	41
FD3	Port A output data	PAOUT	00	41
GPIO Port B				
FD4	Port B address	PBADDR	00	39
FD5	Port B control	PBCTL	00	41
FD6	Port B input data	PBIN	XX	41
FD7	Port B output data	PBOUT	00	41
GPIO Port C				
FD8	Port C address	PCADDR	00	39
FD9	Port C control	PCCTL	00	41
FDA	Port C input data	PCIN	XX	41
FDB	Port C output data	PCOUT	00	41
GPIO Port D				
FDC	Port D address	PDADDR	00	39
FDD	Port D control	PDCTL	00	41
FDE	Reserved	—	XX	
FDF	Port D output data	PDOUT	00	41
FE0–FEF	Reserved	—	XX	
Watchdog Timer (WDT)				
FF0	Reset status	RSTSTAT	XX	95
	Watchdog Timer control	WDTCTL	XX	95
FF1	Watchdog Timer reload upper byte	WDTU	FF	96
FF2	Watchdog Timer reload high byte	WDTH	FF	96
FF3	Watchdog Timer reload low byte	WDTL	FF	97
FF4–FF5	Reserved	—	XX	

Note: XX = Undefined.

General Purpose Input/Output

The Z8 Encore! F0830 Series products support a maximum of 25 port pins (Ports A–D) for General Purpose Input/Output (GPIO) operations. Each port contains control and data registers. The GPIO control registers determine data direction, open-drain, output drive current, programmable pull-ups, Stop Mode Recovery functionality and alternate pin functions. Each port pin is individually programmable. In addition, the Port C pins are capable of direct LED drive at programmable drive strengths.

GPIO Port Availability by Device

Table 15 lists the port pins available with each device and package type.

Table 15. Port Availability by Device and Package Type

Devices	Package	10-Bit ADC	Port A	Port B	Port C	Port D	Total I/O
Z8F1232, Z8F0830, Z8F0430, Z8F0230, Z8F0130	20-pin	Yes	[7:0]	[3:0]	[3:0]	[0]	17
Z8F1233, Z8F0831 Z8F0431, Z8F0231 Z8F0131	20-pin	No	[7:0]	[3:0]	[3:0]	[0]	17
Z8F1232, Z8F0830, Z8F0430, Z8F0230, Z8F0130	28-pin	Yes	[7:0]	[5:0]	[7:0]	[0]	23
Z8F1233, Z8F0831 Z8F0431, Z8F0231 Z8F0131	28-pin	No	[7:0]	[7:0]	[7:0]	[0]	25

Note: 20-pin and 28-pin and 10-bit ADC Enabled or Disabled can be selected via the option bits.

Port A–D Output Control Subregisters

The Port A–D Output Control Subregister, shown in Table 23, is accessed through the Port A–D Control Register by writing 03H to the Port A–D Address Register. Setting the bits in the Port A–D Output Control subregisters to 1 configures the specified port pins for open-drain operation. These subregisters affect the pins directly and, as a result, alternate functions are also affected.

Table 23. Port A–D Output Control Subregisters (PxOC)

Bit	7	6	5	4	3	2	1	0
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
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	If 03H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] POCx	**Port Output Control** These bits function independently of the Alternate function bit and always disable the drains, if set to 1. 0 = The drains are enabled for any OUTPUT Mode (unless overridden by the Alternate function). 1 = The drain of the associated pin is disabled (OPEN-DRAIN mode).

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

LED Drive Level Low Register

The LED Drive Level Low Register, shown in Table 33, contains two control bits for each Port C pin. These two bits select one of four programmable current drive levels for each Port C pin. Each pin is individually programmable.

Table 33. LED Drive Level Low Register (LEDLVLL)

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLL[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F84H							

Bit	Description
[7:0] LEDLVLL	**LED Level Low Bits** {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA.

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) Register, shown in Table 37, stores interrupt requests for both vectored and polled interrupts. When a request is sent to the Interrupt Controller, the corresponding bit in the IRQ2 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 2 Register to determine if any interrupt requests are pending.

Table 37. Interrupt Request 2 Register (IRQ2)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				PC3I	PC2I	PC1I	PC0I
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	FC6H							

Bit	Description
[7:4]	**Reserved** These registers are reserved and must be programmed to 0000.
[3] PCxI	**Port C Pin x Interrupt Request** 0 = No interrupt request is pending for GPIO Port C pin x. 1 = An interrupt request from GPIO Port C pin x is awaiting service.

Note: x indicates the specific GPIO port pin number (3–0).

IRQ0 Enable High and Low Bit Registers

Table 38 lists the priority control values for IRQ0. The IRQ0 Enable High and Low Bit registers, shown in Tables 39 and 40, form a priority-encoded enabling service for interrupts in the Interrupt Request 0 Register. Priority is generated by setting the bits in each register.

Table 38. IRQ0 Enable and Priority Encoding

IRQ0ENH[x]	IRQ0ENL[x]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: x indicates the register bits in the range 7–0.

Interrupt Control Register Definitions

Table 43. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
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							

Bit	Description
[7] PA7ENL	**Port A Bit[7] Interrupt Request Enable Low Bit**
[6] PA6CENL	**Port A Bit[7] or Comparator Interrupt Request Enable Low Bit**
[5:0] PAxENL	**Port A Bit[x] Interrupt Request Enable Low Bit** See the interrupt port select register for selection of either Port A or Port D as the interrupt source.

Note: x indicates register bits in the address range 5–0.

IRQ2 Enable High and Low Bit Registers

Table 44 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers, shown in Tables 45 and 46, form a priority-encoded enabling service for interrupts in the Interrupt Request 2 Register. Priority is generated by setting the bits in each register.

Table 44. IRQ2 Enable and Priority Encoding

IRQ2ENH[x]	IRQ2ENL[x]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: x indicates register bits in the address range 7–0.

ADC Timing

Each ADC measurement consists of three phases:

1. Input sampling (programmable, minimum of 1.0µs)

2. Sample-and-hold amplifier settling (programmable, minimum of 0.5µs)

3. Conversion is 13 ADCLK cycles

Figures 12 and 13 display the timing of an ADC conversion.
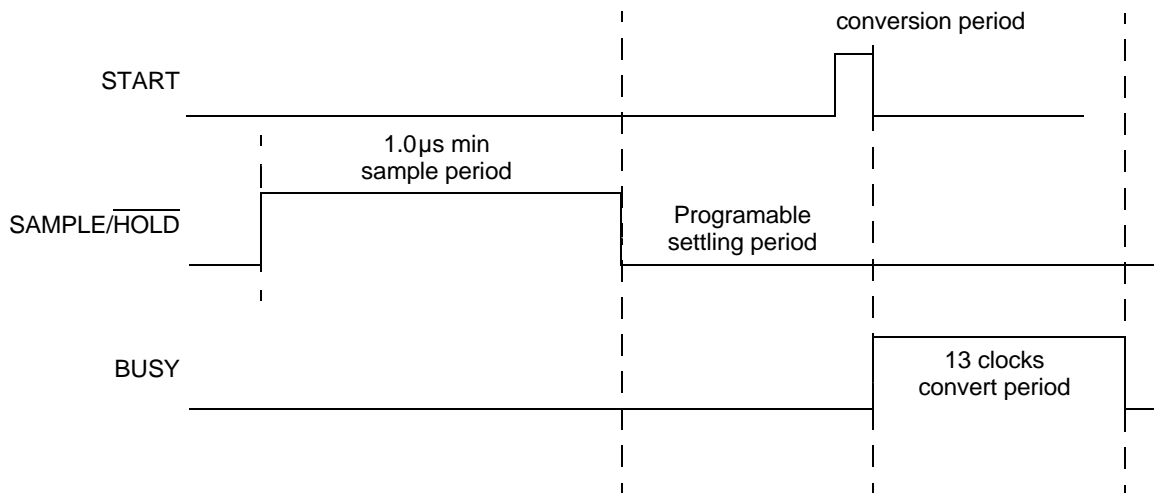
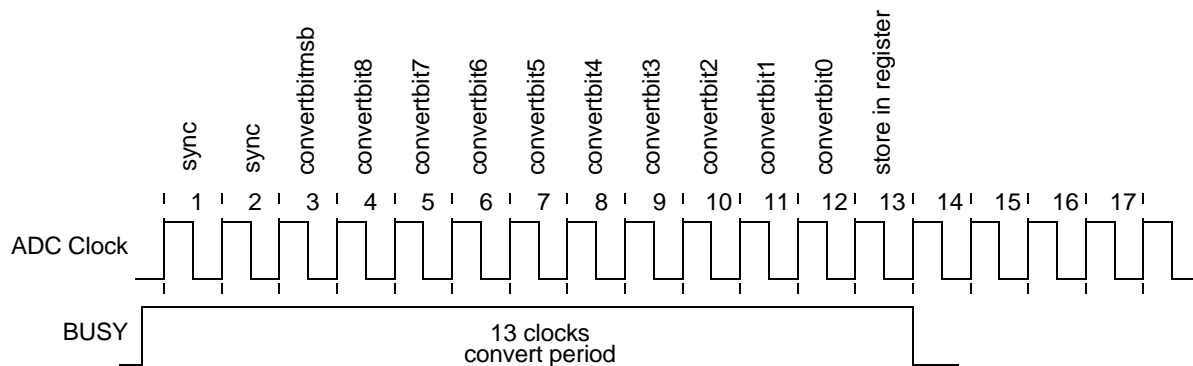**Figure 12. ADC Timing Diagram**

Figure 13. ADC Convert Timing

Flash information area is mapped into program memory and overlays the 128 bytes in the address range FE00H to FE7FH. When the information area access is enabled, all reads from these program memory addresses return the information area data rather than the program memory data. Access to the Flash information area is read-only.

The trim bits are handled differently than the other Zilog Flash option bits. The trim bits are the hybrid of the user option bits and the standard Zilog option bits. These trim bits must be user-accessible for reading at all times using external registers regardless of the state of bit 7 in the Flash Page Select Register. Writes to the trim space change the value of the Option Bit Holding Register but do not affect the Flash bits, which remain as read-only.

Table 70. Z8F083 Flash Memory Area Map

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog option bits
FE40–FE53	Part number 20-character ASCII alphanumeric code Left justified and filled with FH
FE54–FE5F	Reserved
FE60–FE7F	Reserved

Operation

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for byte programming, page erase and mass erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

The flowchart in Figure 19 display basic Flash Controller operation. The following subsections provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase and Mass Erase) displayed in Figure 19.

Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers, shown in Tables 76 and 77, combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:

$$\text{FFREQ[15:0]} = \{\text{FFREQH[7:0],FFREQL[7:0]}\} = \frac{\text{System Clock Frequency}}{1000}$$

! **Caution:** Flash programming and erasure is not supported for system clock frequencies below 10 kHz or above 20 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device.

Table 76. Flash Frequency High Byte Register (FFREQH)

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
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	FFAH							

Bit	Description
[7:0] FFREQH	**Flash Frequency High Byte** High byte of the 16-bit Flash frequency value.

Table 77. 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] FFREQL	**Flash Frequency High Byte** Low byte of the 16-bit Flash frequency value.

On-Chip Debugger

The Z8 Encore! devices contain an integrated On-Chip Debugger (OCD) that provides the following advanced debugging features:

- Reading and writing of the register file

- Reading and writing of program and data memory

- Setting of breakpoints and watchpoints

- Executing eZ8 CPU instructions

Architecture

The On-Chip Debugger consists of four primary functional blocks: transmitter, receiver, autobaud detector/generator and debug controller. Figure 20 displays the architecture of the On-Chip Debugger.

Figure 20. On-Chip Debugger Block Diagram

ory size and is approximately equal to the system clock period multiplied by the number of bytes in program memory.

```
DBG  ←  0EH
DBG  →  CRC[15:8]
DBG  →  CRC[7:0]
```

Step Instruction (10H). The step instruction command, steps one assembly instruction at the current program counter (PC) location. If the device is not in DEBUG Mode or the Flash read protect option bit is enabled, the OCD ignores this command.

```
DBG  ←  10H
```

Stuff Instruction (11H). The stuff instruction command, steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0–4 bytes of the instruction are read from program memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a breakpoint. If the device is not in DEBUG Mode or the Flash read protect option bit is enabled, the OCD ignores this command.

```
DBG  ←  11H
DBG  ←  opcode[7:0]
```

Execute Instruction (12H). The execute instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not in DEBUG Mode or the Flash read protect option bit is enabled, this command reads and discards one byte.

```
DBG  ←  12H
DBG  ←  1-5 byte opcode
```

On-Chip Debugger Control Register Definitions

This section describes the features of the On-Chip Debugger Control and Status registers.

OCD Control Register

The OCD Control Register controls the state of the On-Chip Debugger. This register is used to enter or exit DEBUG Mode and to enable the BRK instruction. It can also reset the Z8 Encore! F0830 Series device.

A reset and stop function can be achieved by writing 81H to this register. A *reset and go* function can be achieved by writing 41H to this register. If the device is in DEBUG Mode, a run function can be implemented by writing 40H to this register.

Oscillator Control

The Z8 Encore! F0830 Series device uses five possible clocking schemes. Each one of these is user-selectable.

- On-chip precision trimmed RC oscillator

- On-chip oscillator using off-chip crystal or resonator

- On-chip oscillator using external RC network

- External clock drive

- On-chip low precision Watchdog Timer Oscillator

In addition, Z8 Encore! F0830 Series devices contain clock failure detection and recovery circuitry, allowing 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 further in this document.

System Clock Selection

The oscillator control block selects from the available clocks. *Table 98* describes each clock source and its usage.

Table 105. Arithmetic Instructions (Continued)

Mnemonic	Operands	Instruction
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 106. Bit Manipulation Instructions

Mnemonic	Operands	Instruction
BCLR	bit, dst	Bit Clear
BIT	p, bit, dst	Bit Set or Clear
BSET	bit, dst	Bit Set
BSWAP	dst	Bit Swap
CCF	—	Complement Carry Flag
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
TCM	dst, src	Test Complement Under Mask
TCMX	dst, src	Test Complement Under Mask using Extended Addressing
TM	dst, src	Test Under Mask
TMX	dst, src	Test Under Mask using Extended Addressing

Table 107. Block Transfer Instructions

Mnemonic	Operands	Instruction
LDCI	dst, src	Load Constant to/from Program Memory and Auto-Increment Addresses
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment Addresses

Table 113. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	src	Op Code(s) (Hex)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
DJNZ dst, RA	dst ← dst − 1 if dst ≠ 0 PC ← PC + X	r		0A–FA	–	–	–	–	–	–	2	3
EI	IRQCTL[7] ← 1			9F	–	–	–	–	–	–	1	2
HALT	HALT Mode			7F	–	–	–	–	–	–	1	2
INC dst	dst ← dst + 1	R		20	–	*	*	–	–	–	2	2
		IR		21							2	3
		r		0E–FE							1	2
INCW dst	dst ← dst + 1	RR		A0	–	*	*	*	–	–	2	5
		IRR		A1							2	6
IRET	FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1			BF	*	*	*	*	*	*	1	5
JP dst	PC ← dst	DA		8D	–	–	–	–	–	–	3	2
		IRR		C4							2	3
JP cc, dst	if cc is true PC ← dst	DA		0D–FD	–	–	–	–	–	–	3	2
JR dst	PC ← PC + X	DA		8B	–	–	–	–	–	–	2	2
JR cc, dst	if cc is true PC ← PC + X	DA		0B–FB	–	–	–	–	–	–	2	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.

Lower Nibble (Hex)

Figure 30. Second Op Code Map after 1FH