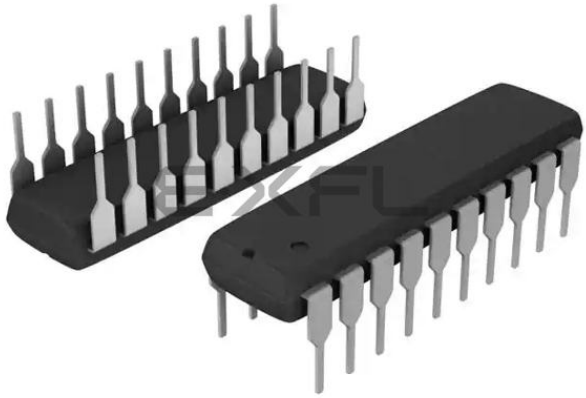


Welcome to E-XFL.COM

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	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0411ph020sg

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
GPIO Port C				
FD8	Port C Address	PCADDR	00	32
FD9	Port C Control	PCCTL	00	33
FDA	Port C Input Data	PCIN	XX	38
FDB	Port C Output Data	PCOUT	00	39
FDC-FEF	Reserved	—	XX	
Watchdog Timer (WDT)				
FF0	Watchdog Timer Control	WDTCTL	XXX00000b	73
FF1	Watchdog Timer Reload Upper Byte	WDTU	FF	75
FF2	Watchdog Timer Reload High Byte	WDTH	FF	75
FF3	Watchdog Timer Reload Low Byte	WDTL	FF	75
FF4-FF7	Reserved	—	XX	
Flash Memory Controller				
FF8	Flash Control	FCTL	00	150
FF8	Flash Status	FSTAT	00	151
FF9	Page Select	FPS	00	152
FF9 (if enabled)	Flash Sector Protect	FPROT	00	153
FFA	Flash Programming Frequency High Byte	FFREQH	00	153
FFB	Flash Programming Frequency Low Byte	FFREQL	00	153
Read-Only Memory				
FF8	Reserved	—	XX	
FF9	Page Select	RPS	00	152
FFA-FFB	Reserved	—	XX	
eZ8 CPU				
FFC	Flags	—	XX	Refer to the eZ8 CPU Core User Manual (UM0128)
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	

Note: XX = undefined.

Reset and Stop Mode Recovery

The Reset Controller within the Z8 Encore! XP® F0822 Series controls Reset and Stop Mode Recovery operation. In typical operation, the following events cause a Reset to occur:

- Power-On Reset (POR)
- Voltage Brown-Out
- WDT time-out (when configured through the WDT_RES option bit to initiate a Reset)
- External $\overline{\text{RESET}}$ pin assertion
- On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)

When the Z8 Encore! XP® F0822 Series device is in STOP Mode, a Stop Mode Recovery is initiated by any of the following events:

- WDT time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source
- DBG pin driven Low

Reset Types

Z8 Encore! XP® F0822 Series provides two types of reset operation (System Reset and Stop Mode Recovery). The type of reset is a function of both the current operating mode of the Z8 Encore! XP® F0822 Series device and the source of the Reset. Table 8 lists the types of Resets and their operating characteristics.

Table 8. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Control Registers**	**eZ8 CPU**	**Reset Latency (Delay)**
System Reset	Reset (as applicable)	Reset	66 WDT Oscillator cycles + 16 System Clock cycles
Stop Mode Recovery	Unaffected, except for the WDT_CTL Register	Reset	66 WDT Oscillator cycles + 16 System Clock cycles

System Reset

During a System Reset, a Z8 Encore! XP® F0822 Series device is held in Reset for 66 cycles of the WDT oscillator followed by 16 cycles of the system clock. At the beginning

the POR status bit in the Watchdog Timer Control Register (WDTCTL) is set to 1. Figure 7 displays the VBO operation. See the Electrical Characteristics chapter on page 176 for the VBO and POR threshold voltages (V_{VBO} and V_{POR}).

The VBO circuit can be either enabled or disabled during STOP Mode. Operation during STOP Mode is set by the VBO_AO option bit. For information about configuring VBO_AO, see the Option Bits chapter on page 155.

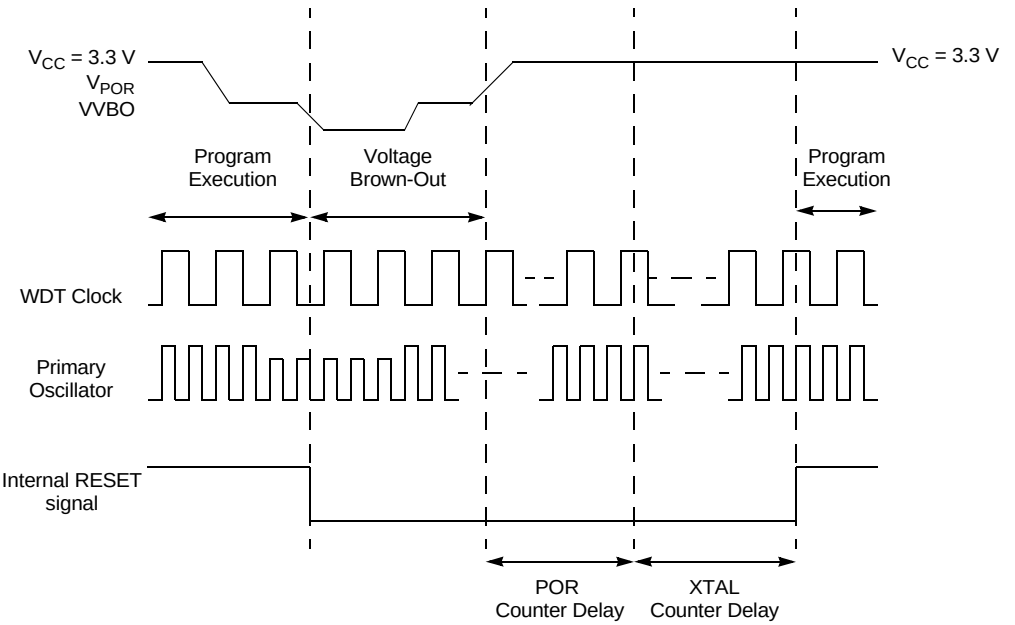


Figure 7. Voltage Brown-Out Reset Operation

Watchdog Timer Reset

If the device is in NORMAL or HALT Mode, WDT initiates a System Reset at time-out, if the WDT_RES option bit is set to 1. This setting is the default (unprogrammed) setting of the WDT_RES option bit. The WDT status bit in the WDT Control Register is set to signify that the reset was initiated by the WDT.

External Pin Reset

The $\overline{\text{RESET}}$ pin contains a Schmitt-triggered input, an internal pull-up, an analog filter, and a digital filter to reject noise. After the $\overline{\text{RESET}}$ pin is asserted for at least 4 system

Port A–C Output Control Subregisters

The Port A–C Output Control Subregister, shown in Table 18, is accessed through the Port A–C Control Register by writing 03H to the Port A–C Address Register. Setting the bits in the Port A–C 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 18. Port A–C Output Control Subregisters

Bit	7	6	5	4	3	2	1	0
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	0							
R/W	R/W							
Address	See footnote.							

Note: If 03H is written to the Port A–C Address Register, then it is accessible via the Port A–C 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 register bits in the range [7:0].

Table 47. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10 kHz Typical WDT Oscillator Frequency)	
		Typical	Description
000004	4	400 µs	Minimum time-out delay
FFFFFF	16,777,215	1677.5 s	Maximum time-out delay

Watchdog Timer Refresh

When first enabled, the WDT is loaded with the value in the WDT Reload registers. The WDT then counts down to 000000H unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the downcounter to be reloaded with the WDT reload value stored in the WDT Reload registers. Counting resumes following the reload operation.

When Z8 Encore! XP® F0822 Series device is operating in DEBUG Mode (using the OCD), the WDT is continuously refreshed to prevent spurious WDT time-outs.

Watchdog Timer Time-Out Response

The WDT times out when the counter reaches 000000H. A WDT time-out generates either an Interrupt or a Reset. The WDT_RES option bit determines the time-out response of the WDT. For information regarding programming of the WDT_RES option bit, see the Option Bits chapter on page 155.

WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the WDT issues an interrupt request to the interrupt controller and sets the WDT status bit in the WDT Control Register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the WDT interrupt vector and executing the code from the vector address. After time-out and interrupt generation, the WDT counter rolls over to its maximum value of FFFFFH and continues counting. The WDT counter is not automatically returned to its reload value.

WDT Reset in STOP Mode

If enabled in STOP Mode and configured to generate a Reset when a time-out occurs and the device is in STOP Mode, the WDT initiates a Stop Mode Recovery. Both the WDT status bit and the stop bit in the WDT Control Register is set to 1 following the WDT time-out in STOP Mode. For more information, see the Reset and Stop Mode Recovery chapter on page 21. Default operation is for the WDT and its RC oscillator to be enabled during STOP Mode.

UART Data and Error Handling Procedure

Figure 16 displays the recommended procedure for UART receiver ISRs.

Baud Rate Generator Interrupts

If the BRG interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the BRG to function as an additional counter if the UART functionality is not employed.

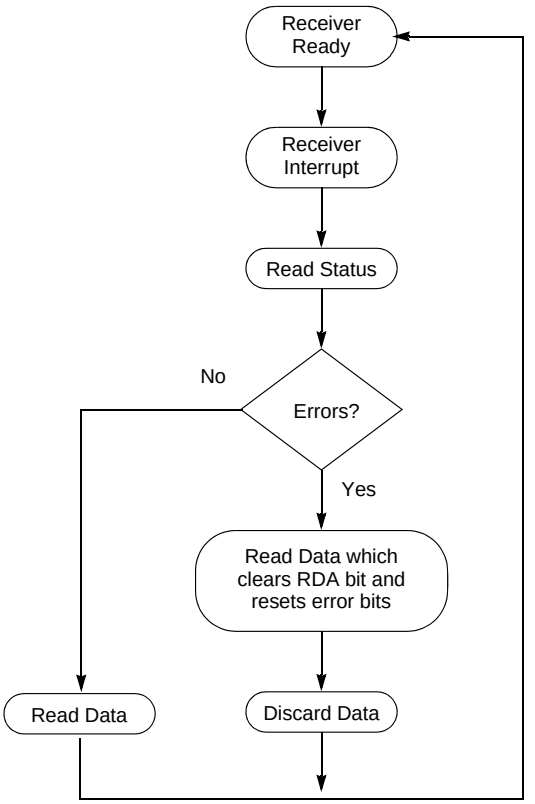


Figure 16. UART Receiver Interrupt Service Routine Flow

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the BRG is the system clock. The UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

When the UART is disabled, the BRG functions as a basic 16-bit timer with interrupt upon time-out. Observe the following procedure to configure the BRG as a timer with interrupt upon time-out:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 Register to 0.
2. Load the appropriate 16-bit count value into the UART Baud Rate High and Low Byte registers.
3. Enable the BRG timer function and associated interrupt by setting the BKGCTL bit in the UART Control 1 Register to 1.

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]]

UART Control Register Definitions

The UART Control registers support the UART and the associated Infrared Encoder/Decoders. See the Infrared Encoder/Decoder chapter on page 97 for more information about the infrared operation.

UART Transmit Data Register

Data bytes written to the UART Transmit Data Register, shown in Table 53, are shifted out on the TXD*x* pin. The write-only UART Transmit Data Register shares a Register File address with the read-only UART Receive Data Register.

UART Status 0 Register

The UART Status 0 and Status 1 registers, shown in Tables 55 and 56, identify the current UART operating configuration and status.

Table 55. UART Status 0 Register (U0STAT0)

Bit	7	6	5	4	3	2	1	0
Field	RDA	PE	OE	FE	BRKD	TDRE	TXE	CTS
RESET	0					1		X
R/W	R							
Address	F41H							

Bit	Description
[7] RDA	**Receive Data Available** This bit indicates that the UART Receive Data Register has received data. Reading the UART Receive Data Register clears this bit. 0 = The UART Receive Data Register is empty. 1 = There is a byte in the UART Receive Data Register.
[6] 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 has occurred. 1 = A parity error has occurred.
[5] 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.
[4] 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.
[3] 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.
[2] TDRE	

Start and Stop Conditions

The Master (I^2C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I^2C Controller generates a start condition by pulling the SDA signal Low while SCL is High. To complete a transaction, the I^2C Controller generates a Stop condition by creating a Low-to-High transition of the SDA signal while the SCL signal is High. The start and stop bits in the I^2C Control Register control the sending of start and stop conditions. A Master is also allowed to end one transaction and begin a new one by issuing a restart. This restart issuance is accomplished by setting the start bit at the end of a transaction rather than setting the stop bit.

Note: The start condition is not sent until the start bit is set and data has been written to the I^2C Data Register.

Master Write and Read Transactions

The following sections provide Zilog's recommended procedure for performing I^2C write and read transactions from the I^2C Controller (Master) to slave I^2C devices. In general, software should rely on the TDRE, RDRF and NCKI bits of the status register (these bits generate interrupts) to initiate software actions. When using interrupts or DMA, the TXI bit is set to start each transaction and cleared at the end of each transaction to eliminate a *trailing* transmit interrupt.

Caution: Caution should be used in using the ACK status bit within a transaction because it is difficult for software to tell when it is updated by hardware.

When writing data to a slave, the I^2C pauses at the beginning of the Acknowledge cycle if the data register has not been written with the next value to be sent (TDRE bit in the I^2C Status Register equal to 1). In this scenario where software is not keeping up with the I^2C bus (TDRE asserted longer than one byte time), the Acknowledge clock cycle for byte *n* is delayed until the data register is written with byte *n+1*, and appears to be grouped with the data clock cycles for byte *n+1*. If either the start or stop bit is set, the I^2C does not pause prior to the Acknowledge cycle because no additional data is sent.

When a Not Acknowledge condition is received during a write (either during the address or data phases), the I^2C Controller generates the Not Acknowledge interrupt (NCKI = 1) and pause until either the stop or start bit is set. Unless the Not Acknowledge was received on the last byte, the data register will already have been written with the next address or data byte to send. In this case the FLUSH bit of the control register should be set at the same time the stop or start bit is set to remove the stale transmit data and enable subsequent transmit interrupts.

6. The I2C Controller sends the start condition to the I2C Slave.
7. The I2C Controller loads the I2C Shift Register with the contents of the I2C Data Register.
8. After one bit of an address is shifted out by the SDA signal, the transmit interrupt is asserted.
9. Software responds by writing the second byte of the address into the contents of the I2C Data Register.
10. The I2C Controller shifts the rest of the first byte of address and write bit out the SDA signal.
11. If the I2C Slave sends an acknowledge by pulling the SDA signal Low during the next High period of SCL and the I2C Controller sets the ACK bit in the I2C Status Register, continue to Step 12.

    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 the remaining steps in this sequence.
12. The I2C Controller loads the I2C Shift Register with the contents of the I2C Data Register (2nd byte of address).
13. The I2C Controller shifts the second address byte out the SDA signal. After the first bit has been sent, the transmit interrupt is asserted.
14. Software responds by setting the stop bit in the I2C Control Register. The TXI bit can be cleared at the same time.
15. Software polls the stop bit of the I2C Control Register. Hardware deasserts the stop bit when the transaction is completed (stop condition has been sent).
16. Software checks the ACK bit of the I2C Status Register. If the slave acknowledged, the ACK bit is equal to 1. If the slave does not acknowledge, the ACK bit is equal to 0. The NCKI interrupt do not occur because the stop bit was set.

Write Transaction with a 10-Bit Address

Figure 29 displays the data transfer format for a 10-bit addressed slave. 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 1st Seven Bits	W = 0	A	Slave Address 2nd Byte	A	Data	A	Data	A/A	P/S

Figure 29. 10-Bit Addressed Slave Data Transfer Format

I²C Diagnostic Control Register

The I²C Diagnostic Register, shown in Table 77, provides control over diagnostic modes. This register is a read/write register used for I²C diagnostics.

Table 77. I²C Diagnostic Control Register (I2CDIAG)

Bit	7	6	5	4	3	2	1	0
Field	Reserved							DIAG
RESET	0							
R/W	R							R/W
Address	F56H							

Bit	Description
[7:1]	**Reserved** These bits are reserved and must be programmed to 0000000.
[0] DIAG	**Diagnostic Control Bit** Selects the read-back value of the Baud Rate Reload registers. 0 = Normal Mode. Reading the Baud Rate High and Low Byte registers returns the baud rate reload value. 1 = Diagnostic Mode. Reading the Baud Rate High and Low Byte registers returns the baud rate counter value.

Exiting Debug Mode

The device exits DEBUG Mode following any of the following operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset
- Asserting the RESET pin Low to initiate a Reset
- Driving the DBG pin Low while the device is in STOP Mode initiates a System Reset

OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 start bit, 8 data bits (least-significant bit first), and 1 stop bit; see Figure 37.

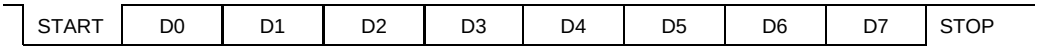


Figure 37. OCD Data Format

OCD Autobaud Detector/Generator

To run over a range of baud rates (bits per second) with various system clock frequencies, the OCD contains an Autobaud Detector/Generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the host is the character 80H. The character 80H has eight continuous bits Low (one start bit plus 7 data bits). The Autobaud Detector measures this period and sets the OCD Baud Rate Generator accordingly.

The Autobaud Detector/Generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. For optimal operation, the maximum recommended baud rate is the system clock frequency divided by 8. The theoretical maximum baud rate is the system clock frequency divided by 4. This theoretical maximum is possible for low-noise designs with clean signals. Table 92 lists minimum and recommended maximum baud rates for sample crystal frequencies.

! **Caution:** The OCDCNTR Register is used by many of the OCD commands. It counts the number of bytes for the register and memory read/write commands. It holds the residual value when generating the CRC. Therefore, if the OCDCNTR is being used to generate a BRK, its value should be written as a last step before leaving DEBUG Mode.

Because this register is overwritten by various OCD commands, it should only be used to generate temporary breakpoints, such as stepping over CALL instructions or running to a specific instruction and stopping.

On-Chip Debugger Commands

The host communicates to the OCD by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG Mode, all OCD commands become available unless the user code and control registers are protected by programming the Read Protect option bit (RP). This Read Protect option bit prevents the code in memory from being read out of the Z8 Encore! XP® F0822 Series products. When this option is enabled, several of the OCD commands are disabled. Table 93 contains a summary of the OCD commands. Each OCD command is described further in the bulleted list. It also lists the commands that operate when the device is not in DEBUG Mode (normal operation) and those commands that are disabled by programming the Read Protect option bit.

Table 93. On-Chip Debugger Commands

Debug Command	Command Byte	Enabled When Not in DEBUG Mode?	Disabled by Read Protect Option Bit
Read OCD Revision	00H	Yes	–
Write OCD Counter Register	01H	–	–
Read OCD Status Register	02H	Yes	–
Read OCD Counter Register	03H	–	–
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit
Read OCD Control Register	05H	Yes	–
Write Program Counter	06H	–	Disabled
Read Program Counter	07H	–	Disabled
Write Register	08H	–	Only writes of the peripheral control registers at address F00H–FFH are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control Register.

Figure 42 displays the maximum active mode current consumption across the full operating temperature range of the device and plotted opposite the system clock frequency. All GPIO pins are configured as outputs and driven High.

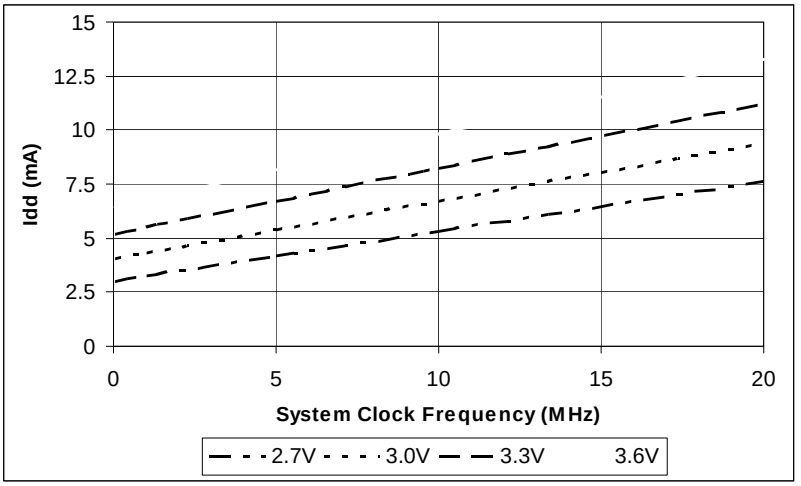


Figure 42. Maximum Active Mode I_{DD} vs. System Clock Frequency

Table 119. Arithmetic Instructions (Continued)

Mnemonic	Operands	Instruction
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 120. 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 121. 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 127. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
LD dst, rc	dst ← src	r	IM	0C-FC	-	-	-	-	-	-	2	2
		r	X(r)	C7							3	3
		X(r)	r	D7							3	4
		r	Ir	E3							2	3
		R	R	E4							3	2
		R	IR	E5							3	4
		R	IM	E6							3	2
		IR	IM	E7							3	3
		Ir	r	F3							2	3
		IR	R	F5							3	3
LDC dst, src	dst ← src	r	Irr	C2	-	-	-	-	-	-	2	5
		Ir	Irr	C5							2	9
		Irr	r	D2							2	5
LDCI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	C3	-	-	-	-	-	-	2	9
		Irr	Ir	D3							2	9
LDE dst, src	dst ← src	r	Irr	82	-	-	-	-	-	-	2	5
		Irr	r	92							2	5
LDEI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	83	-	-	-	-	-	-	2	9
		Irr	Ir	93							2	9
LDWX dst, src	dst ← src	ER	ER	1F E8	-	-	-	-	-	-	5	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.

Flags Register

The Flags Register contains the status information regarding the most recent arithmetic, logical, bit manipulation or rotate and shift operation. The Flags Register contains six bits of status information that are set or cleared by CPU operations. Four of the bits (C, V, Z and S) can be tested with conditional jump instructions. Two flags (H and D) cannot be tested and are used for Binary-Coded Decimal (BCD) arithmetic.

The two remaining bits, User Flags (F1 and F2), are available as general-purpose status bits. User Flags are unaffected by arithmetic operations and must be set or cleared by instructions. The User Flags cannot be used with conditional Jumps. They are undefined at initial power-up and are unaffected by Reset. Figure 56 illustrates the flags and their bit positions in the Flags Register.

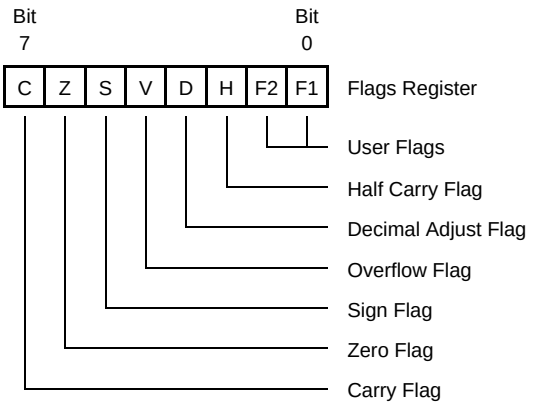


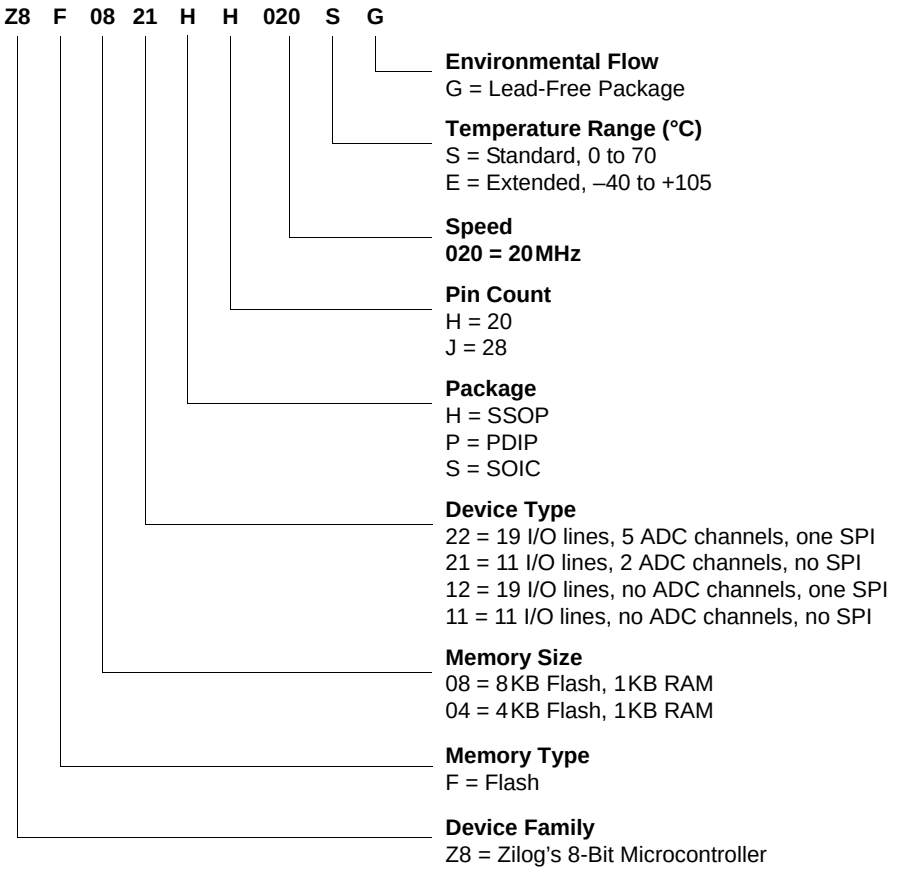
Figure 56. Flags Register

Interrupts, the Software Trap (TRAP) instruction, and Illegal Instruction Traps all write the value of the Flags Register to the stack. Executing an Interrupt Return (IRET) instruction restores the value saved on the stack into the Flags Register.

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: F41

Table 148. UART Status 0 Register (U0STAT0)

Bit	7	6	5	4	3	2	1	0
Field	RDA	PE	OE	FE	BRKD	TDRE	TXE	CTS
RESET	0					1		X
R/W	R							
Address	F41H							

Hex Address: F42

Table 149. UART Control 0 Register (U0CTL0)

Bit	7	6	5	4	3	2	1	0
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
RESET	0							
R/W	R/W							
Address	F42H							

Hex Address: F43

Table 150. UART Control 1 Register (U0CTL1)

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							

Hex Address: F44

Table 151. UART Status 1 Register (U0STAT1)

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

D

E

F