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	60
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6423ft020ec

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

ISO 9001:2000
FS 507510

Program Memory

The eZ8™ CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP 64K Series Flash Microcontrollers contains 16 KB to 64 KB of on-chip Flash in the Program Memory address space, depending upon the device. Reading from Program Memory addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 5 describes the Program Memory maps for the 64K Series products.

Table 5. Z8 Encore! XP 64K Series Flash Microcontrollers Program Memory Maps

Program Memory Address (Hex)	Function
Z8F162x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFF	Program Memory
Z8F242x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFF	Program Memory
Z8F322x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFF	Program Memory
Z8F482x Products	

Watchdog Timer Control
WDTCTL (FF0H - Read Only)
D7 D6 D5 D4 D3 D2 D1 D0

SM Configuration Indicator
Reserved
EXT
0 = Reset not generated by $\overline{\text{RESET}}$ pin
1 = Reset generated by $\overline{\text{RESET}}$ pin

WDT
0 = WDT timeout has not occurred
1 = WDT timeout occurred

STOP
0 = SMR has not occurred
1 = SMR has occurred

POR
0 = POR has not occurred
1 = POR has occurred

Watchdog Timer Reload Upper Byte
WDTU (FF1H - Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

WDT reload value [23:16]

Watchdog Timer Reload Middle Byte
WDTH (FF2 H- Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

WDT reload value [15:8]

Watchdog Timer Reload Low Byte
WDTL (FF3H - Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

WDT reload value [7:0]

Flash Control
FCTL (FF8H - Write Only)
D7 D6 D5 D4 D3 D2 D1 D0

Flash Command
73H = First unlock command
8CH = Second unlock command
95H = Page erase command
63H = Mass erase command
5EH = Flash Sector Protect reg select

Flash Status
FSTAT (FF8H - Read Only)
D7 D6 D5 D4 D3 D2 D1 D0

Flash Controller Status
00_0000 = Flash controller locked
00_0001 = First unlock received
00_0010 = Second unlock received
00_0011 = Flash controller unlocked
00_0100 = Flash Sector Protect register selected
00_1xxx = Programming in progress
01_0xxx = Page erase in progress
10_0xxx = Mass erase in progress

Reserved

Page Select
FPS (FF9H - Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

Page Select [6:0]
Identifies the Flash memory page for Page Erase operation.

Information Area Enable
0 = Information Area access is disabled
1 = Information Area access is enabled

Flash Sector Protect
FPROT (FF9H - Read/Write to 1's)
D7 D6 D5 D4 D3 D2 D1 D0

Flash Sector Protect [7:0]
0 = Sector can be programmed or erased from user code
1 = Sector is protected and cannot be programmed or erased from user code

Flash Frequency High Byte
FFREQH (FFAH - Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

Flash Frequency value [15:8]

Flash Frequency Low Byte
FFREQL (FFBH - Read/Write)
D7 D6 D5 D4 D3 D2 D1 D0

Flash Frequency value [7:0]

Stop Mode Recovery Using a GPIO Port Pin Transition HALT

Each of the GPIO Port pins may be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. The GPIO Stop Mode Recovery signals are filtered to reject pulses less than 10 ns (typical) in duration. In the Watchdog Timer Control register, the `STOP` bit is set to 1.

Caution: *In STOP mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin through the end of the Stop Mode Recovery delay. Thus, short pulses on the Port pin can initiate Stop Mode Recovery without being written to the Port Input Data register or without initiating an interrupt (if enabled for that pin).*

Table 19. Port A–H High Drive Enable Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
RESET	0							
R/W	R/W							
ADDR	If 04H in Port A-H Address Register, accessible through Port A-H Control Register							

PHDE[7:0]—Port High Drive Enabled
0 = The Port pin is configured for standard output current drive.
1 = The Port pin is configured for high output current drive.

Port A–H Stop Mode Recovery Source Enable Sub-Registers

The Port A–H Stop Mode Recovery Source Enable sub-register (Table 20) is accessed through the Port A–H Control register by writing `05H` to the Port A–H Address register. Setting the bits in the Port A–H Stop Mode Recovery Source Enable sub-registers to 1 configures the specified Port pins as a Stop Mode Recovery source. During STOP Mode, any logic transition on a Port pin enabled as a Stop Mode Recovery source initiates Stop Mode Recovery.

Table 20. Port A–H Stop Mode Recovery Source Enable Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	PSMRE7	PSMRE6	PSMRE5	PSMRE4	PSMRE3	PSMRE2	PSMRE1	PSMRE0
RESET	0							
R/W	R/W							
ADDR	If 05H in Port A–H Address Register, accessible through Port A–H Control Register							

PSMRE[7:0]—Port Stop Mode Recovery Source Enabled
0 = The Port pin is not configured as a Stop Mode Recovery source. Transitions on this pin during STOP mode do not initiate Stop Mode Recovery.

1 = The Port pin is configured as a Stop Mode Recovery source. Any logic transition on this pin during STOP mode initiates Stop Mode Recovery.

Table 29. IRQ0 Enable Low Bit Register (IRQ0ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	T2ENL	T1ENL	T0ENL	U0RENL	U0TENL	I2CENL	SPIENL	ADCENL
RESET	0							
R/W	R/W							
ADDR	FC2H							

T2ENL—Timer 2 Interrupt Request Enable Low Bit
T1ENL—Timer 1 Interrupt Request Enable Low Bit
T0ENL—Timer 0 Interrupt Request Enable Low Bit
U0RENL—UART 0 Receive Interrupt Request Enable Low Bit
U0TENL—UART 0 Transmit Interrupt Request Enable Low Bit
I2CENL—I^2C Interrupt Request Enable Low Bit
SPIENL—SPI Interrupt Request Enable Low Bit
ADCENL—ADC Interrupt Request Enable Low Bit

IRQ1 Enable High and Low Bit Registers

The IRQ1 Enable High and Low Bit registers (see Table 31 and Table 32 on page 76) form a priority encoded enabling for interrupts in the Interrupt Request 1 register. Priority is generated by setting bits in each register. Table 30 describes the priority control for IRQ1.

Table 30. IRQ1 Enable and Priority Encoding

IRQ1ENH[*x*]	IRQ1ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: where *x* indicates the register bits from 0 through 7.

3. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
4. Execute the IRET instruction to return from the interrupt-service routine and await more data.

Clear To Send ($\overline{CTS}$) Operation

The CTS pin, if enabled by the CTSE bit of the UART Control 0 register, performs flow control on the outgoing transmit datastream. The Clear To Send ($\overline{CTS}$) input pin is sampled one system clock before beginning any new character transmission. To delay transmission of the next data character, an external receiver must deassert $\overline{CTS}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this would typically be done during Stop Bit transmission. If $\overline{CTS}$ deasserts in the middle of a character transmission, the current character is sent completely.

MULTIPROCESSOR (9-bit) Mode

The UART has a MULTIPROCESSOR (9-bit) mode that uses an extra (9th) bit for selective communication when a number of processors share a common UART bus. In MULTIPROCESSOR mode (also referred to as 9-Bit mode), the multiprocessor bit (MP) is transmitted immediately following the 8-bits of data and immediately preceding the Stop bit(s) as displayed in Figure 16. The character format is:

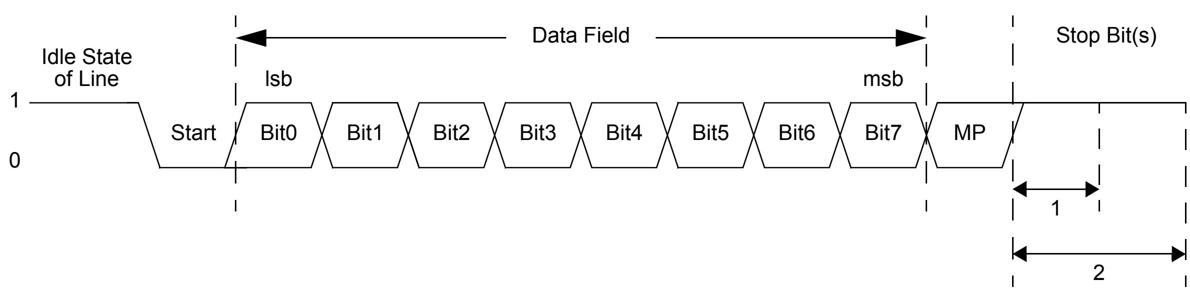

Figure 16. UART Asynchronous MULTIPROCESSOR Mode Data Format

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

MULTIPROCESSOR (9-bit) Mode Receive Interrupts

When MULTIPROCESSOR mode is enabled, the UART only processes frames addressed to it. The determination of whether a frame of data is addressed to the UART can be made in hardware, software or some combination of the two, depending on the multiprocessor

Table 55. UART Status 1 Register (U*x*STAT1)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved						NEWFRM	MPRX
RESET	0							
R/W	R				R/W		R	
ADDR	F44H and F4CH							

Reserved—Must be 0.

NEWFRM—Status bit denoting the start of a new frame. Reading the UART Receive Data register resets this bit to 0.
0 = The current byte is not the first data byte of a new frame.
1 = The current byte is the first data byte of a new frame.

MPRX—Multiprocessor Receive
Returns the value of the last multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (see Table 56 and Table 57 on page 118) configure the properties of the UART's transmit and receive operations. The UART Control registers must not been written while the UART is enabled.

Table 56. UART Control 0 Register (U*x*CTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
RESET	0							
R/W	R/W							
ADDR	F42H and F4AH							

TEN—Transmit Enable
This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled.
0 = Transmitter disabled.
1 = Transmitter enabled.

TXRXSTATE	State Description
1_1101	10-bit addressing: Bit 3 of 2nd address byte 7-bit addressing: Bit 3 of address byte
1_1110	10-bit addressing: Bit 2 of 2nd address byte 7-bit addressing: Bit 2 of address byte
1_1111	10-bit addressing: Bit 1 of 2nd address byte 7-bit addressing: Bit 1 of address byte

I²C Diagnostic Control Register

The I²C Diagnostic register (Table 76) provides control over diagnostic modes. This register is a read/write register used for I²C diagnostics.

Table 76. I²C Diagnostic Control Register (I2CDIAG)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved							DIAG
RESET	0							
R/W	R							R/W
ADDR	F56H							

DIAG = Diagnostic Control Bit - Selects 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.

Direct Memory Access Controller

Overview

The 64K Series Direct Memory Access (DMA) Controller provides three independent Direct Memory Access channels. Two of the channels (DMA0 and DMA1) transfer data between the on-chip peripherals and the Register File. The third channel (DMA_ADC) controls the ADC operation and transfers SINGLE-SHOT mode ADC output data to the Register File.

Operation

DMA0 and DMA1 Operation

DMA0 and DMA1, referred to collectively as DMA*x*, transfer data either from the on-chip peripheral control registers to the Register File, or from the Register File to the on-chip peripheral control registers. The sequence of operations in a DMA*x* data transfer is:

1. DMA*x* trigger source requests a DMA data transfer.
2. DMA*x* requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA*x* transfers either a single byte or a two-byte word (depending upon configuration) and then returns system bus control back to the eZ8 CPU.
4. If Current Address equals End Address:
   - DMA*x* reloads the original Start Address
   - If configured to generate an interrupt, DMA*x* sends an interrupt request to the Interrupt Controller
   - If configured for single-pass operation, DMA*x* resets the DEN bit in the DMA*x* Control register to 0 and the DMA is disabled.

   If Current Address does not equal End Address, the Current Address increments by 1 (single-byte transfer) or 2 (two-byte word transfer).

finish the interrupt service routine it may be in and return the BRK instruction. When the CPU returns to the BRK instruction it was previously looping on, it automatically sets the DBGMODE bit and enter DEBUG mode.

Software detects that the majority of the OCD commands are still disabled when the eZ8™ CPU is looping on a BRK instruction. The eZ8 CPU must be stopped and the part must be in DEBUG mode before these commands can be issued.

Breakpoints in Flash Memory

The BRK instruction is opcode `00H`, which corresponds to the fully programmed state of a byte in Flash memory. To implement a Breakpoint, write `00H` to the desired address, overwriting the current instruction. To remove a Breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

On-Chip Debugger Commands

The host communicates to the On-Chip Debugger 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). The Read Protect Option Bit prevents the code in memory from being read out of the 64K Series products. When this option is enabled, several of the OCD commands are disabled. Table 101 contains a summary of the On-Chip Debugger commands. Each OCD command is described in detail in the bulleted list following Table 101.
Table 101 indicates those 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 101. 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	-
Read OCD Status Register	02H	Yes	-
Read Runtime Counter	03H	-	-
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit
Read OCD Control Register	05H	Yes	-

```
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Program Memory (0BH)**—The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns `FFH` for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

- **Write Data Memory (0CH)**—The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Data Memory (0DH)**—The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the device is not in DEBUG mode, this command returns `FFH` for the data.

```
DBG ← 0DH
DBG ← Data Memory Address[15:8]
```

On-Chip Oscillator

Overview

The products in the 64K Series feature an on-chip oscillator for use with external crystals with frequencies from 32 kHz to 20 MHz. In addition, the oscillator can support external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with oscillation frequencies up to 20 MHz. This oscillator generates the primary system clock for the internal eZ8™ CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32 kHz–20 MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected.

When configured for use with crystal oscillators or external clock drivers, the frequency of the signal on the X_{IN} input pin determines the frequency of the system clock (that is, no internal clock divider). In RC operation, the system clock is driven by a clock divider (divide by 2) to ensure 50% duty cycle.

Operating Modes

The 64K Series products support four different oscillator modes:

- On-chip oscillator configured for use with external RC networks (<4 MHz).
- Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).
- Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).
- Maximum power for use with high frequency crystals or ceramic resonators (8.0 MHz to 20.0 MHz).

The oscillator mode is selected through user-programmable Option Bits. For more information, see Option Bits on page 195.

Crystal Oscillator Operation

Figure 40 on page 212 displays a recommended configuration for connection with an external fundamental-mode, parallel-resonant crystal operating at 20 MHz. Recommended 20 MHz crystal specifications are provided in Table 104 on page 212. Resistor R1 is optional and limits total power dissipation by the crystal. The printed circuit board layout

Figure 45 displays the maximum HALT mode current consumption across the full operating temperature range of the device and versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

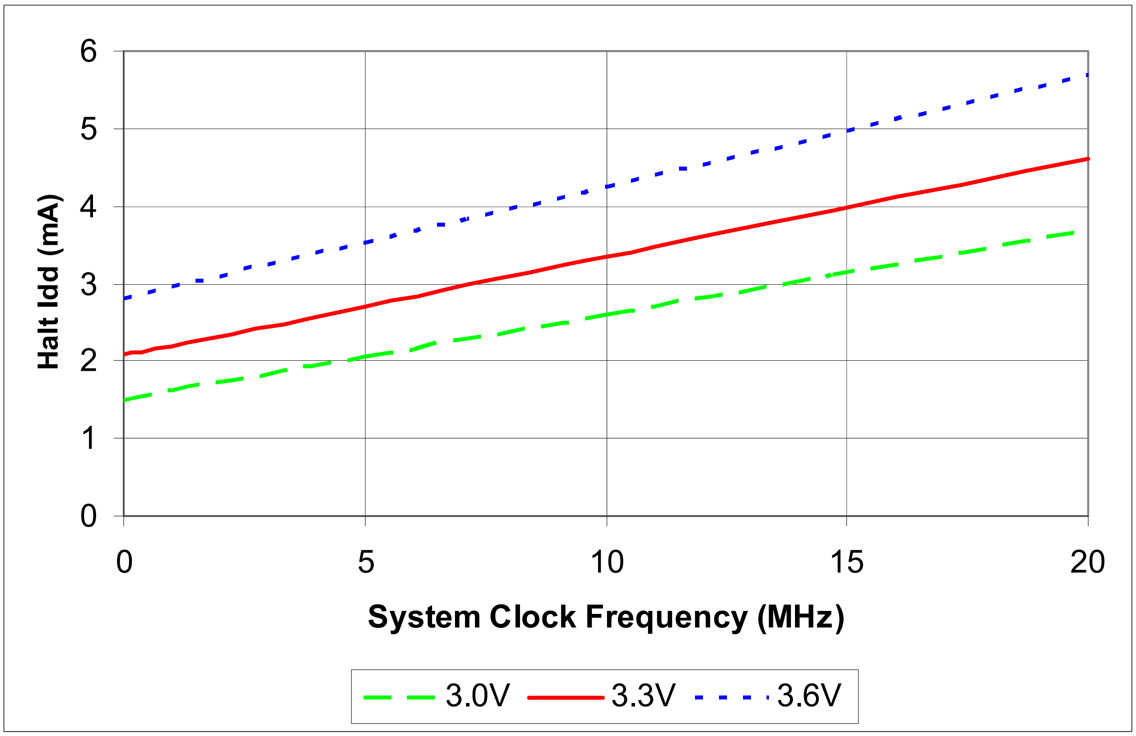

Figure 46. Maximum HALT Mode Icc Versus System Clock Frequency

eZ8™ CPU Instruction Set

Assembly Language Programming Introduction

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

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

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

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

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

Assembly Language Source Program Example

```
JP START          ; Everything after the semicolon is a comment.

START:            ; A label called "START". The first instruction (JP START) in this
                  ; example causes program execution to jump to the point within the
                  ; program where the START label occurs.

LD R4, R7         ; A Load (LD) instruction with two operands. The first operand,
                  ; Working Register R4, is the destination. The second operand,
                  ; Working Register R7, is the source. The contents of R7 is
                  ; written into R4.

LD 234H, #%01     ; Another Load (LD) instruction with two operands.
                  ; The first operand, Extended Mode Register Address 234H,
                  ; identifies the destination. The second operand, Immediate Data
```

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.

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
Z8F162x with 16 KB Flash, 10-Bit Analog-to-Digital Converter										
Standard Temperature: 0 °C to 70 °C										
Z8F1621PM020SC	16 KB	2 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F1621AN020SC	16 KB	2 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F1621VN020SC	16 KB	2 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F1622AR020SC	16 KB	2 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F1622VS020SC	16 KB	2 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Extended Temperature: –40 °C to +105 °C										
Z8F1621PM020EC	16 KB	2 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F1621AN020EC	16 KB	2 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F1621VN020EC	16 KB	2 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F1622AR020EC	16 KB	2 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F1622VS020EC	16 KB	2 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Automotive/Industrial Temperature: –40 °C to +125 °C										
Z8F1621PM020AC	16 KB	2 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F1621AN020AC	16 KB	2 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F1621VN020AC	16 KB	2 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F1622AR020AC	16 KB	2 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F1622VS020AC	16 KB	2 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F64200100KITG										Development Kit
ZUSBSC00100ZACG										USB Smart Cable Accessory Kit
ZUSBOPTSC01ZACG										Opto-Isolated USB Smart Cable Accessory Kit

Note: Replace C with G for lead-free packaging.