



Welcome to [E-XFL.COM](https://www.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	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	48KB (48K 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	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4822vs020ec00tr

Manual Objectives

This Product Specification provides detailed operating information for the Flash devices within Zilog's Z8 Encore! XP[®] 64K Series Flash Microcontrollers Microcontroller (MCU) products. Within this document, the Z8F642x, Z8F482x, Z8F322x, Z8F242x, and Z8F162x devices are referred to collectively as the Z8 Encore! XP[®] 64K Series Flash Microcontrollers unless specifically stated otherwise.

About This Manual

Zilog[®] recommends that you read and understand everything in this manual before setting up and using the product. However, we recognize that there are different styles of learning. Therefore, we have designed this Product Specification to be used either as a *how to* procedural manual or a reference guide to important data.

Intended Audience

This document is written for Zilog customers who are experienced at working with microcontrollers, integrated circuits, or printed circuit assemblies.

Manual Conventions

The following assumptions and conventions are adopted to provide clarity and ease of use:

Courier Typeface

Commands, code lines and fragments, bits, equations, hexadecimal addresses, and various executable items are distinguished from general text by the use of the `Courier` typeface. Where the use of the font is not indicated, as in the Index, the name of the entity is presented in upper case.

- Example: `FLAGS[1]` is `smrf`.

Hexadecimal Values

Hexadecimal values are designated by uppercase *H* suffix and appear in the `Courier` typeface.

- Example: R1 is set to `F8H`.

Brackets

The square brackets, `[]`, indicate a register or bus.

- Example: For the register `R1[7:0]`, R1 is an 8-bit register, `R1[7]` is the most significant bit, and `R1[0]` is the least significant bit.

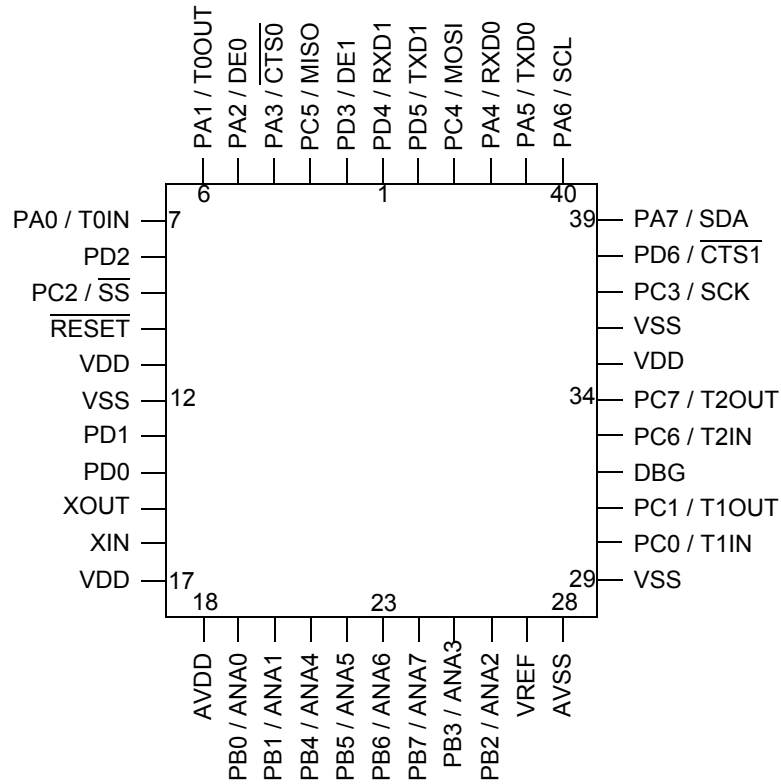


Figure 3. Z8 Encore! XP 64K Series Flash Microcontrollers in 44-Pin Plastic Leaded Chip Carrier (PLCC)

UART0 Transmit Data

U0TXD (F40H - Write Only)

D7 D6 D5 D4 D3 D2 D1 D0

UART0 transmitter data byte [7:0]

UART0 Receive Data

U0RXD (F40H - Read Only)

D7 D6 D5 D4 D3 D2 D1 D0

UART0 receiver data byte [7:0]

UART0 Status 0

U0STAT0 (F41H - Read Only)

D7 D6 D5 D4 D3 D2 D1 D0

CTS signal
Returns the level of the CTS signal

Transmitter Empty
0 = Data is currently transmitting
1 = Transmission is complete

Transmitter Data Register Empty
0 = Transmit Data Register is full
1 = Transmit Data register is empty

Break Detect
0 = No break occurred
1 = A break occurred

Framing Error
0 = No framing error occurred
1 = A framing error occurred

Overrun Error
0 = No overrun error occurred
1 = An overrun error occurred

Parity Error
0 = No parity error occurred
1 = A parity error occurred

Receive Data Available
0 = Receive Data Register is empty
1 = A byte is available in the Receive Data Register

UART0 Control 0

U0CTL0 (F42H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Loop Back Enable
0 = Normal operation
1 = Transmit data is looped back to the receiver

Stop Bit Select
0 = Transmitter sends 1 Stop bit
1 = Transmitter sends 2 Stop bits

Send Break
0 = No break is sent
1 = Output of the transmitter is zero

Parity Select
0 = Even parity
1 = Odd parity

Parity Enable
0 = Parity is disabled
1 = Parity is enabled

CTS Enable
0 = CTS signal has no effect on the transmitter
1 = UART recognizes CTS signal as a transmit enable control signal

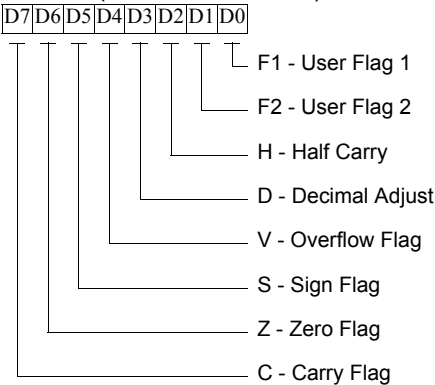
Receive Enable
0 = Receiver disabled
1 = Receiver enabled

Transmit Enable
0 = Transmitter disabled
1 = Transmitter enabled



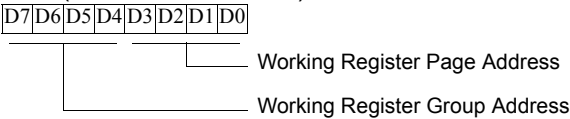
Flags

FLAGS (FFC - Read/Write)



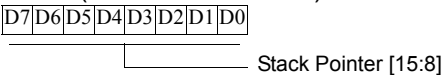
Register Pointer

RP (FFDH - Read/Write)



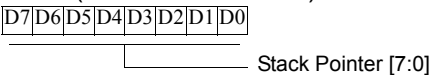
Stack Pointer High Byte

SPH (FFEH - Read/Write)



Stack Pointer Low Byte

SPL (FFFH - Read/Write)



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 33. IRQ2 Enable and Priority Encoding (Continued)

IRQ2ENH[x]	IRQ2ENL[x]	Priority	Description
1	1	Level 3	High

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

Table 34. IRQ2 Enable High Bit Register (IRQ2ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	T3ENH	U1RENH	U1TENH	DMAENH	C3ENH	C2ENH	C1ENH	C0ENH
RESET	0							
R/W	R/W							
ADDR	FC7H							

T3ENH—Timer 3 Interrupt Request Enable High Bit

U1RENH—UART 1 Receive Interrupt Request Enable High Bit

U1TENH—UART 1 Transmit Interrupt Request Enable High Bit

DMAENH—DMA Interrupt Request Enable High Bit

C3ENH—Port C3 Interrupt Request Enable High Bit

C2ENH—Port C2 Interrupt Request Enable High Bit

C1ENH—Port C1 Interrupt Request Enable High Bit

C0ENH—Port C0 Interrupt Request Enable High Bit

Table 35. IRQ2 Enable Low Bit Register (IRQ2ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	T3ENL	U1RENL	U1TENL	DMAENL	C3ENL	C2ENL	C1ENL	C0ENL
RESET	0							
R/W	R/W							
ADDR	FC8H							

T3ENL—Timer 3 Interrupt Request Enable Low Bit

U1RENL—UART 1 Receive Interrupt Request Enable Low Bit

U1TENL—UART 1 Transmit Interrupt Request Enable Low Bit

DMAENL—DMA Interrupt Request Enable Low Bit

C3ENL—Port C3 Interrupt Request Enable Low Bit

C2ENL—Port C2 Interrupt Request Enable Low Bit

The timer continues counting up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt and continues counting.

Follow the steps below for configuring a timer for CAPTURE mode and initiating the count:

1. Write to the Timer Control 1 register to:
 - Disable the timer
 - Configure the timer for CAPTURE mode.
 - Set the prescale value.
 - Set the Capture edge (rising or falling) for the Timer Input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. This allows the software to determine if interrupts were generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain 0000H after the interrupt, then the interrupt was generated by a Reload.
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 Input alternate function.
7. Write to the Timer Control 1 register to enable the timer and initiate counting.

In CAPTURE mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

COMPARE Mode

In COMPARE mode, the timer counts up to the 16-bit maximum Compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Compare value, the timer generates an interrupt and counting continues (the timer value is not reset to 0001H). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon Compare.

If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting.

Table 39. Timer 0-3 High Byte Register (TxH)

BITS	7	6	5	4	3	2	1	0
FIELD	TH							
RESET	0							
R/W	R/W							
ADDR	F00H, F08H, F10H, F18H							

Table 40. Timer 0-3 Low Byte Register (TxL)

BITS	7	6	5	4	3	2	1	0
FIELD	TL							
RESET	0							1
R/W	R/W							
ADDR	F01H, F09H, F11H, F19H							

TH and TL—Timer High and Low Bytes

These 2 bytes, {TMRH[7:0], TMRL[7:0]}, contain the current 16-bit timer count value.

Timer Reload High and Low Byte Registers

The Timer 0-3 Reload High and Low Byte (TxRH and TxRL) registers (see [Table 41](#) and [Table 42](#) on page 92) store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte register are stored in a temporary holding register. When a write to the Timer Reload Low Byte register occurs, the temporary holding register value is written to the Timer High Byte register. This operation allows simultaneous updates of the 16-bit Timer Reload value.

In COMPARE mode, the Timer Reload High and Low Byte registers store the 16-bit Compare value.

Table 41. Timer 0-3 Reload High Byte Register (TxRH)

BITS	7	6	5	4	3	2	1	0
FIELD	TRH							
RESET	1							
R/W	R/W							
ADDR	F02H, F0AH, F12H, F1AH							

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 (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 (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 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 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:

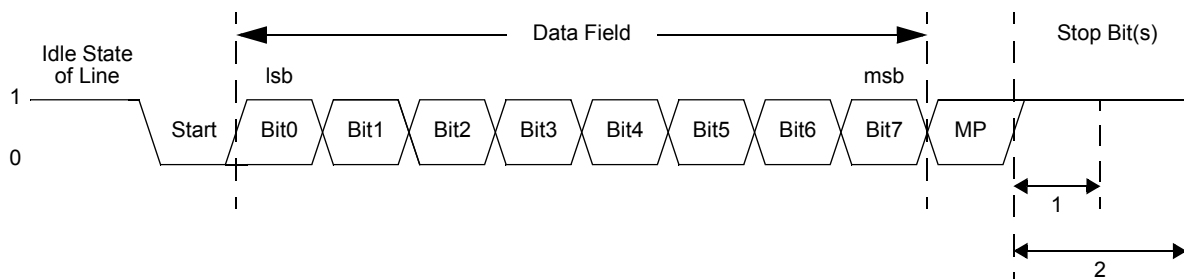


Figure 16. UART Asynchronous MULTIPROCESSOR Mode Data Format

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

(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 Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

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

$$\text{Interrupt Interval(s)} = \text{System Clock Period (s)} \times \text{BRG[15:0]}$$

UART Control Register Definitions

The UART control registers support the UART and the associated Infrared Encoder/Decoders. For more information on the infrared operation, see [Infrared Encoder/Decoder](#) on page 125.

UART Transmit Data Register

Data bytes written to the UART Transmit Data register ([Table 52](#)) are shifted out on the TXDx pin. The Write-only UART Transmit Data register shares a Register File address with the Read-only UART Receive Data register.

Table 52. UART Transmit Data Register (UxTXD)

BITS	7	6	5	4	3	2	1	0
FIELD	TXD							
RESET	X							
R/W	W							
ADDR	F40H and F48H							

Serial Peripheral Interface

Overview

The Serial Peripheral Interface is a synchronous interface allowing several SPI-type devices to be interconnected. SPI-compatible devices include EEPROMs, Analog-to-Digital Converters, and ISDN devices. Features of the SPI include:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface
- Data transfers rates up to a maximum of one-half the system clock frequency
- Error detection
- Dedicated Baud Rate Generator

Architecture

The SPI may be configured as either a Master (in single or multi-master systems) or a Slave as displayed in [Figure 22](#) through [Figure 24](#).

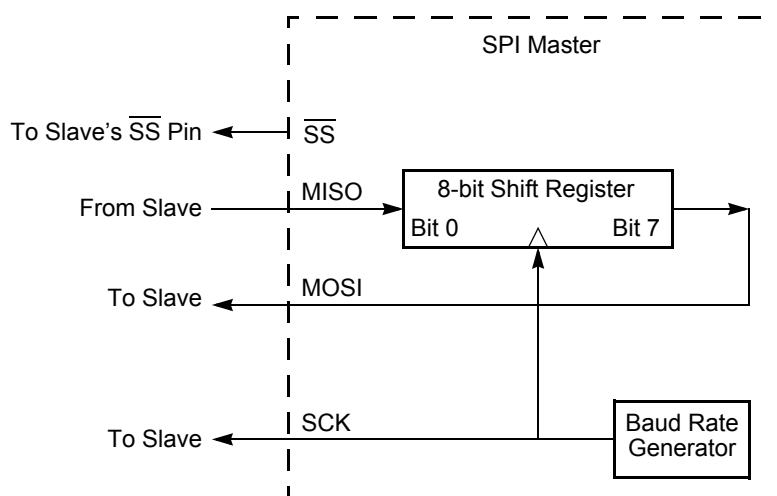


Figure 22. SPI Configured as a Master in a Single Master, Single Slave System

The Master and Slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles (see NUMBITS field in the [SPI Mode Register](#) on page 140). In both Master and Slave SPI devices, data is shifted on one edge of the SCK and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI phase and polarity control.

Slave Select

The active Low Slave Select (\overline{SS}) input signal selects a Slave SPI device. \overline{SS} must be Low prior to all data communication to and from the Slave device. \overline{SS} must stay Low for the full duration of each character transferred. The \overline{SS} signal may stay Low during the transfer of multiple characters or may deassert between each character.

When the SPI is configured as the only Master in an SPI system, the \overline{SS} pin can be set as either an input or an output. For communication between the Z8F642x family Z8R642x family device's SPI Master and external Slave devices, the \overline{SS} signal, as an output, can assert the \overline{SS} input pin on one of the Slave devices. Other GPIO output pins can also be employed to select external SPI Slave devices.

When the SPI is configured as one Master in a multi-master SPI system, the \overline{SS} pin must be set as an input. The \overline{SS} input signal on the Master must be High. If the \overline{SS} signal goes Low (indicating another Master is driving the SPI bus), a Collision error Flag is set in the SPI Status register.

SPI Clock Phase and Polarity Control

The SPI supports four combinations of serial clock phase and polarity using two bits in the SPI Control register. The clock polarity bit, CLKPOL, selects an active high or active Low clock and has no effect on the transfer format. [Table 62](#) lists the SPI Clock Phase and Polarity Operation parameters. The clock phase bit, PHASE, selects one of two fundamentally different transfer formats. For proper data transmission, the clock phase and polarity must be identical for the SPI Master and the SPI Slave. The Master always places data on the MOSI line a half-cycle before the receive clock edge (SCK signal), in order for the Slave to latch the data.

Table 62. SPI Clock Phase (PHASE) and Clock Polarity (CLKPOL) Operation

PHASE	CLKPOL	SCK Transmit Edge	SCK Receive Edge	SCK Idle State
0	0	Falling	Rising	Low
0	1	Rising	Falling	High
1	0	Rising	Falling	Low
1	1	Falling	Rising	High

0101 = ADC Analog Inputs 0-5 updated.
 0110 = ADC Analog Inputs 0-6 updated.
 0111 = ADC Analog Inputs 0-7 updated.
 1000 = ADC Analog Inputs 0-8 updated.
 1001 = ADC Analog Inputs 0-9 updated.
 1010 = ADC Analog Inputs 0-10 updated.
 1011 = ADC Analog Inputs 0-11 updated.
 1100-1111 = Reserved.

DMA Status Register

The DMA Status register (Table 85 on page 173) indicates the DMA channel that generated the interrupt and the ADC Analog Input that is currently undergoing conversion. Reads from this register reset the Interrupt Request Indicator bits (IRQA, IRQ1, and IRQ0) to 0. Therefore, software interrupt service routines that read this register must process all three interrupt sources from the DMA.

Table 85. DMA_ADC Status Register (DMAA_STAT)

BITS	7	6	5	4	3	2	1	0
FIELD	CADC[3:0]				Reserved	IRQA	IRQ1	IRQ0
RESET	0							
R/W	R							
ADDR	FBFH							

CADC[3:0]—Current ADC Analog Input

This field identifies the Analog Input that the ADC is currently converting.

Reserved

This bit is reserved and must be 0.

IRQA—DMA_ADC Interrupt Request Indicator

This bit is automatically reset to 0 each time a read from this register occurs.

0 = DMA_ADC is not the source of the interrupt from the DMA Controller.

1 = DMA_ADC completed transfer of data from the last ADC Analog Input and generated an interrupt.

IRQ1—DMA1 Interrupt Request Indicator

This bit is automatically reset to 0 each time a read from this register occurs.

0 = DMA1 is not the source of the interrupt from the DMA Controller.

1 = DMA1 completed transfer of data to/from the End Address and generated an interrupt.

IRQ0—DMA0 Interrupt Request Indicator

This bit is automatically reset to 0 each time a read from this register occurs.

Table 90. Flash Memory Sector Addresses

Sector Number	Flash Sector Address Ranges				
	Z8F162x	Z8F242x	Z8F322x	Z8F482x	Z8F642x
0	0000H-07FFH	0000H-0FFFH	0000H-0FFFH	0000H-1FFFH	0000H-1FFFH
1	0800H-0FFFH	1000H-1FFFH	1000H-1FFFH	2000H-3FFFH	2000H-3FFFH
2	1000H-17FFH	2000H-2FFFH	2000H-2FFFH	4000H-5FFFH	4000H-5FFFH
3	1800H-1FFFH	3000H-3FFFH	3000H-3FFFH	6000H-7FFFH	6000H-7FFFH
4	2000H-27FFH	4000H-4FFFH	4000H-4FFFH	8000H-9FFFH	8000H-9FFFH
5	2800H-2FFFH	5000H-5FFFH	5000H-5FFFH	A000H-BFFFH	A000H-BFFFH
6	3000H-37FFH	N/A	6000H-6FFFH	N/A	C000H-DFFFH
7	3800H-3FFFH	N/A	7000H-7FFFH	N/A	E000H-FFFFH

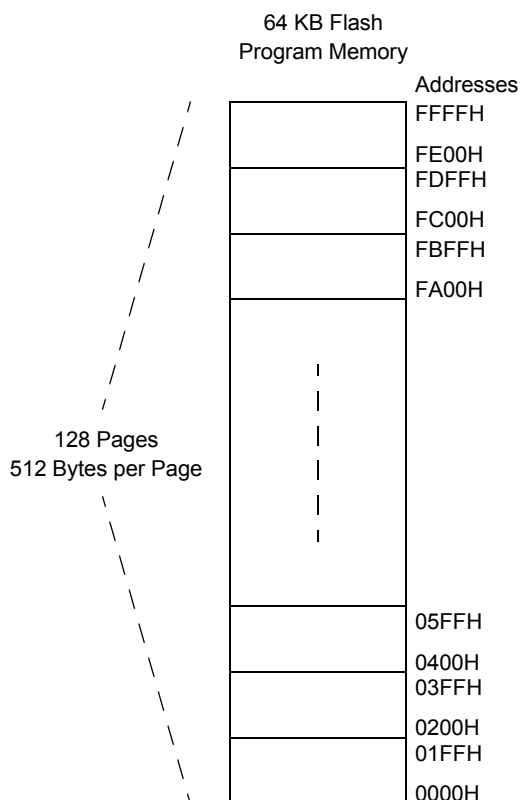


Figure 35. Flash Memory Arrangement

Reserved

These bits are reserved and must be 0.

FSTAT—Flash Controller Status

00_0000 = Flash Controller locked

00_0001 = First unlock command received

00_0010 = Second unlock command received

00_0011 = Flash Controller unlocked

00_0100 = Flash Sector Protect register selected

00_1xxx = Program operation in progress

01_0xxx = Page erase operation in progress

10_0xxx = Mass erase operation in progress

Page Select Register

The Page Select (FPS) register (Table 94) selects one of the 128 available Flash memory pages to be erased or programmed. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory locations with the 7 most significant bits of the address given by the PAGE field are erased to FFH.

The Page Select register shares its Register File address with the Flash Sector Protect Register. The Page Select register cannot be accessed when the Flash Sector Protect register is enabled.

Table 94. Page Select Register (FPS)

BITS	7	6	5	4	3	2	1	0
FIELD	INFO_EN	PAGE						
RESET	0							
R/W	R/W							
ADDR	FF9H							

INFO_EN—Information Area Enable

0 = Information Area is not selected.

1 = Information Area is selected. The Information area is mapped into the Flash Memory address space at addresses FE00H through FFFFH.

PAGE—Page Select

This 7-bit field selects the Flash memory page for Programming and Page Erase operations. Flash Memory Address[15:9] = PAGE[6:0].

- Asserting the $\overline{\text{RESET}}$ 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 39](#)).

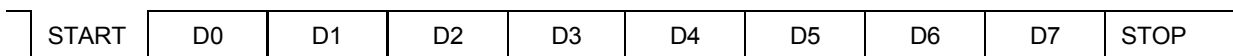


Figure 39. OCD Data Format

OCD Auto-Baud Detector/Generator

To run over a range of baud rates (bits per second) with various system clock frequencies, the On-Chip Debugger has an Auto-Baud 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 Auto-Baud Detector measures this period and sets the OCD Baud Rate Generator accordingly.

The Auto-Baud 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 100](#) lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 100. OCD Baud-Rate Limits

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (kbits/s)	Minimum Baud Rate (kbits/s)
20.0	2500	39.1
1.0	125.0	1.96
0.032768 (32 kHz)	4.096	0.064

If the OCD receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. The Auto-Baud Detector/Generator can then be reconfigured by sending 80H.

Table 101. On-Chip Debugger Commands (Continued)

Debug Command	Command Byte	Enabled when NOT in DEBUG mode?	Disabled by Read Protect Option Bit
Write Program Counter	06H	-	Disabled
Read Program Counter	07H	-	Disabled
Write Register	08H	-	Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control register.
Read Register	09H	-	Disabled
Write Program Memory	0AH	-	Disabled
Read Program Memory	0BH	-	Disabled
Write Data Memory	0CH	-	Disabled
Read Data Memory	0DH	-	Disabled
Read Program Memory CRC	0EH	-	-
Reserved	0FH	-	-
Step Instruction	10H	-	Disabled
Stuff Instruction	11H	-	Disabled
Execute Instruction	12H	-	Disabled
Reserved	13H - FFH	-	-

In the following list of OCD Commands, data and commands sent from the host to the On-Chip Debugger are identified by 'DBG ← Command/Data'. Data sent from the On-Chip Debugger back to the host is identified by 'DBG → Data'.

- Read OCD Revision (00H)**—The Read OCD Revision command determines the version of the On-Chip Debugger. If OCD commands are added, removed, or changed, this revision number changes.
 - DBG ← 00H
 - DBG → OCDREV[15:8] (Major revision number)
 - DBG → OCDREV[7:0] (Minor revision number)
- Read OCD Status Register (02H)**—The Read OCD Status Register command reads the OCDSTAT register.
 - DBG ← 02H
 - DBG → OCDSTAT[7:0]

AC Characteristics

The section provides information on the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs. [Table 113](#) lists the 64K Series AC characteristics and timing.

Table 113. AC Characteristics

Symbol	Parameter	V _{DD} = 3.0–3.6V T _A = –40 °C to 125 °C		Units	Conditions
		Minimum	Maximum		
F _{sysclk}	System Clock Frequency	–	20.0	MHz	Read-only from Flash memory.
		0.032768	20.0	MHz	Program or erasure of the Flash memory.
F _{XTAL}	Crystal Oscillator Frequency	0.032768	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
T _{XIN}	Crystal Oscillator Clock Period	50	–	ns	T _{CLK} = 1/F _{sysclk}
T _{XINH}	System Clock High Time	20		ns	
T _{XINL}	System Clock Low Time	20		ns	
T _{XINR}	System Clock Rise Time	–	3	ns	T _{CLK} = 50 ns. Slower rise times can be tolerated with longer clock periods.
T _{XINF}	System Clock Fall Time	–	3	ns	T _{CLK} = 50 ns. Slower fall times can be tolerated with longer clock periods.

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
```

Index

Symbols

244
% 244
@ 244

Numerics

10-bit ADC 4
40-lead plastic dual-inline package 265
44-lead low-profile quad flat package 266
44-lead plastic lead chip carrier package 267
64-lead low-profile quad flat package 267
68-lead plastic lead chip carrier package 268
80-lead quad flat package 269

A

absolute maximum ratings 215
AC characteristics 231
ADC 246
 architecture 175
 automatic power-down 176
 block diagram 176
 continuous conversion 177
 control register 179
 control register definitions 179
 data high byte register 180
 data low bits register 180
 DMA control 178
 electrical characteristics and timing 229
 operation 176
 single-shot conversion 177
ADCCTL register 179
ADCDH register 180
ADC DL register 180
ADCX 246
ADD 246
add - extended addressing 246
add with carry 246
add with carry - extended addressing 246

additional symbols 244
address space 19
ADDX 246
analog signals 15
analog-to-digital converter (ADC) 175
AND 248
ANDX 248
arithmetic instructions 246
assembly language programming 241
assembly language syntax 242

B

B 244
b 243
baud rate generator, UART 113
BCLR 246
binary number suffix 244
BIT 246
bit 243
 clear 246
 manipulation instructions 246
 set 246
 set or clear 246
 swap 247
 test and jump 249
 test and jump if non-zero 249
 test and jump if zero 249
bit jump and test if non-zero 249
bit swap 249
block diagram 3
block transfer instructions 247
BRK 249
BSET 246
BSWAP 247, 249
BTJ 249
BTJNZ 249
BTJZ 249

C

CALL procedure 249
capture mode 95
capture/compare mode 95