



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	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 x 8
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-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f011ahh020ec00tr

HALT Mode

Executing the eZ8 CPU's HALT instruction places the device into HALT mode, which powers down the CPU but leaves all other peripherals active. In HALT mode, the operating characteristics are:

- Primary oscillator is enabled and continues to operate.
- System clock is enabled and continues to operate.
- eZ8 CPU is stopped.
- Program counter (PC) stops incrementing.
- Watchdog Timer's internal RC oscillator continues to operate.
- If enabled, the Watchdog Timer continues to operate.
- All other on-chip peripherals continue to operate, if enabled.

The eZ8 CPU can be brought out of HALT mode by any of the following operations:

- Interrupt
- Watchdog Timer time-out (interrupt or reset)
- Power-On Reset
- Voltage Brownout reset
- External $\overline{\text{RESET}}$ pin assertion

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

Peripheral-Level Power Control

In addition to the STOP and HALT modes, it is possible to disable each peripheral on each of the Z8 Encore! XP F082A Series devices. Disabling a given peripheral minimizes its power consumption.

Power Control Register Definitions

The following sections define the Power Control registers.

Power Control Register 0

Each bit of the following registers disables a peripheral block, either by gating its system clock input or by removing power from the block. The default state of the low-power

tions as a GPIO pin. If it is not present, the debug feature is disabled until/unless another reset event occurs. For more details, see [On-Chip Debugger](#) on page 173.

Crystal Oscillator Override

For systems using a crystal oscillator, PA0 and PA1 are used to connect the crystal. When the crystal oscillator is enabled (see [Oscillator Control Register Definitions](#) on page 190), the GPIO settings are overridden and PA0 and PA1 are disabled.

5 V Tolerance

All six I/O pins on the 8-pin devices are 5 V-tolerant, unless the programmable pull-ups are enabled. If the pull-ups are enabled and inputs higher than V_{DD} are applied to these parts, excessive current flows through those pull-up devices and can damage the chip.

- **Note:** *In the 20- and 28-pin versions of this device, any pin which shares functionality with an ADC, crystal or comparator port is not 5 V-tolerant, including PA[1:0], PB[5:0] and PC[2:0]. All other signal pins are 5 V-tolerant, and can safely handle inputs higher than V_{DD} except when the programmable pull-ups are enabled.*

External Clock Setup

For systems using an external TTL drive, PB3 is the clock source for 20- and 28-pin devices. In this case, configure PB3 for alternate function CLKIN. Write the Oscillator Control (OSCCTL) register (see [Oscillator Control Register Definitions](#) on page 190) such that the external oscillator is selected as the system clock. For 8-pin devices use PA1 instead of PB3.

Table 18. Port A–D Control Registers (PxCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FD1H, FD5H, FD9H, FDDH							

PCTL[7:0]—Port Control

The Port Control register provides access to all sub-registers that configure the GPIO Port operation.

Port A–D Data Direction Sub-Registers

The Port A–D Data Direction sub-register is accessed through the Port A–D Control register by writing 01H to the Port A–D Address register ([Table 19](#)).

Table 19. Port A–D Data Direction Sub-Registers (PxDD)

BITS	7	6	5	4	3	2	1	0
FIELD	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	If 01H in Port A–D Address Register, accessible through the Port A–D Control Register							

DD[7:0]—Data Direction

These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.

0 = Output. Data in the Port A–D Output Data register is driven onto the port pin.

1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated.

Port A–D Alternate Function Sub-Registers

The Port A–D Alternate Function sub-register ([Table 20](#)) is accessed through the Port A–D Control register by writing 02H to the Port A–D Address register. The Port A–D Alternate Function sub-registers enable the alternate function selection on pins. If disabled, pins functions as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the [Port A–D Alternate Function](#)

U0RXI—UART 0 Receiver Interrupt Request

0 = No interrupt request is pending for the UART 0 receiver.

1 = An interrupt request from the UART 0 receiver is awaiting service.

U0TXI—UART 0 Transmitter Interrupt Request

0 = No interrupt request is pending for the UART 0 transmitter.

1 = An interrupt request from the UART 0 transmitter is awaiting service.

ADCI—ADC Interrupt Request

0 = No interrupt request is pending for the analog-to-digital Converter.

1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register ([Table 34](#)) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 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 1 register to determine if any interrupt requests are pending.

Table 34. Interrupt Request 1 Register (IRQ1)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VI	PA6CI	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
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
ADDR	FC3H							

PA7VI—Port A Pin 7 or LVD Interrupt Request

0 = No interrupt request is pending for GPIO Port A or LVD.

1 = An interrupt request from GPIO Port A or LVD.

PA6CI—Port A Pin 6 or Comparator Interrupt Request

0 = No interrupt request is pending for GPIO Port A or Comparator.

1 = An interrupt request from GPIO Port A or Comparator.

PAxI—Port A Pin x Interrupt Request

0 = No interrupt request is pending for GPIO Port A pin x.

1 = An interrupt request from GPIO Port A pin x is awaiting service.

where x indicates the specific GPIO Port pin number (0–5).

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register ([Table 35](#)) stores interrupt requests for both vectored and polled interrupts. When a request is presented 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 35. Interrupt Request 2 Register (IRQ2)

BITS	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
ADDR	FC6H							

Reserved—Must be 0.

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.

where *x* indicates the specific GPIO Port C pin number (0–3).

IRQ0 Enable High and Low Bit Registers

[Table 36](#) describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers ([Table 37](#) and [Table 38](#)) form a priority encoded enabling for interrupts in the Interrupt Request 0 register.

Table 36. IRQ0 Enable and Priority Encoding

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

where *x* indicates the register bits from 0–7.

Universal Asynchronous Receiver/Transmitter

The universal asynchronous receiver/transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer.
- Selectable even- and odd-parity generation and checking.
- Option of one or two STOP bits.
- Separate transmit and receive interrupts.
- Framing, parity, overrun and break detection.
- Separate transmit and receive enables.
- 16-bit baud rate generator (BRG).
- Selectable MULTIPROCESSOR (9-bit) mode with three configurable interrupt schemes.
- Baud rate generator (BRG) can be configured and used as a basic 16-bit timer.
- Driver enable (DE) output for external bus transceivers.

Architecture

The UART consists of three primary functional blocks: transmitter, receiver, and baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. [Figure 10](#) on page 98 displays the UART architecture.

(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 functions as a basic 16-bit timer with interrupt on time-out. Follow the steps below to configure the Baud Rate Generator as a timer with interrupt on time-out:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 register to 0.
2. Load the acceptable 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 infrared operation, see [Infrared Encoder/Decoder](#) on page 117.

UART Control 0 and Control 1 Registers

The UART Control 0 (UxCTL0) and Control 1 (UxCTL1) registers ([Table 61](#) and [Table 62](#)) configure the properties of the UART's transmit and receive operations. The UART Control registers must not be written while the UART is enabled.

Table 61. UART Control 0 Register (U0CTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
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
ADDR	F42H							

TEN—Transmit Enable

This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal

Analog-to-Digital Converter

The analog-to-digital converter (ADC) converts an analog input signal to its digital representation. The features of this sigma-delta ADC include:

- 11-bit resolution in DIFFERENTIAL mode.
- 10-bit resolution in SINGLE-ENDED mode.
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports.
- 9th analog input obtained from temperature sensor peripheral.
- 11 pairs of differential inputs also multiplexed with general-purpose I/O ports.
- Low-power operational amplifier (LPO).
- Interrupt on conversion complete.
- Bandgap generated internal voltage reference with two selectable levels.
- Manual in-circuit calibration is possible employing user code (offset calibration).
- Factory calibrated for in-circuit error compensation.

Architecture

Figure 19 displays the major functional blocks of the ADC. An analog multiplexer network selects the ADC input from the available analog pins, ANA0 through ANA7.

The input stage of the ADC allows both differential gain and buffering. The following input options are available:

- Unbuffered input (SINGLE-ENDED and DIFFERENTIAL modes).
- Buffered input with unity gain (SINGLE-ENDED and DIFFERENTIAL modes).
- LPO output with full pin access to the feedback path.

Table 73. ADC Data High Byte Register (ADCD_H)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCDH							
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	F72H							
X = Undefined.								

ADCDH—ADC Data High Byte

This byte contains the upper eight bits of the ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the most recent conversion output is held in this register. These bits are undefined after a Reset.

ADC Data Low Byte Register

The ADC Data Low Byte (ADCD_L) register contains the lower bits of the ADC output as well as an overflow status bit. The output is a 13-bit two's complement value. During a single-shot conversion, this value is invalid. Access to the ADC Data Low Byte register is read-only. Reading the ADC Data High Byte register latches data in the ADC Low Bits register.

Table 74. ADC Data Low Byte Register (ADCD_L)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCDL					Reserved		OVF
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	F73H							
X = Undefined.								

ADCDL—ADC Data Low Bits

These bits are the least significant five bits of the 13-bits of the ADC output. These bits are undefined after a Reset.

Reserved—Must be undefined.

OVF—Overflow Status

0= A hardware overflow did not occur in the ADC for the current sample.

1= A hardware overflow did occur in the ADC for the current sample, therefore the current sample is invalid.

```

nop      ; wait for output to settle
clr irq0 ; clear any spurious interrupts pending
ei

```

Comparator Control Register Definitions

Comparator Control Register

The Comparator Control Register (CMP0) configures the comparator inputs and sets the value of the internal voltage reference.

Table 75. Comparator Control Register (CMP0)

BITS	7	6	5	4	3	2	1	0
FIELD	INPSEL	INNSEL	REFLVL				Reserved (20-/28-pin) REFLVL (8-pin)	
RESET	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F90H							

INPSEL—Signal Select for Positive Input

0 = GPIO pin used as positive comparator input

1 = temperature sensor used as positive comparator input

INNSEL—Signal Select for Negative Input

0 = internal reference disabled, GPIO pin used as negative comparator input

1 = internal reference enabled as negative comparator input

REFLVL—Internal Reference Voltage Level (this reference is independent of the ADC voltage reference). Note that the 8-pin devices contain two additional LSBs for increased resolution.

For 20-/28-pin devices:

0000 = 0.0 V

0001 = 0.2 V

0010 = 0.4 V

0011 = 0.6 V

0100 = 0.8 V

0101 = 1.0 V (Default)

0110 = 1.2 V

0111 = 1.4 V

1000 = 1.6 V

Flash Status Register

The Flash Status (FSTAT) register indicates the current state of the Flash Controller. This register can be read at any time. The read-only Flash Status register shares its Register File address with the Write-only Flash Control register.

Table 79. Flash Status Register (FSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved		FSTAT					
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
ADDR	FF8H							

Reserved—Must be 0.

FSTAT—Flash Controller Status

000000 = Flash Controller locked

000001 = First unlock command received (73H written)

000010 = Second unlock command received (8CH written)

000011 = Flash Controller unlocked

000100 = Sector protect register selected

001xxx = Program operation in progress

010xxx = Page erase operation in progress

100xxx = Mass erase operation in progress

Flash Page Select Register

The Flash Page Select (FPS) register shares address space with the Flash Sector Protect Register. Unless the Flash controller is unlocked and written with 5EH, writes to this address target the Flash Page Select Register.

The register is used to select one of the available Flash memory pages to be programmed or erased. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory having addresses with the most significant 7 bits given by FPS[6:0] are chosen for program/erase operation.

Table 100. Serialization Data Locations

Info Page Address	Memory Address	Usage
1C	FE1C	Serial Number Byte 3 (most significant)
1D	FE1D	Serial Number Byte 2
1E	FE1E	Serial Number Byte 1
1F	FE1F	Serial Number Byte 0 (least significant)

Randomized Lot Identifier**Table 101. Lot Identification Number (RAND_LOT)**

BITS	7	6	5	4	3	2	1	0
FIELD	RAND_LOT							
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	Interspersed throughout Information Page Memory							
Note: U = Unchanged by Reset. R/W = Read/Write.								

RAND_LOT—Randomized Lot ID

The randomized lot ID is a 32 byte binary value that changes for each production lot.

Table 102. Randomized Lot ID Locations

Info Page Address	Memory Address	Usage
3C	FE3C	Randomized Lot ID Byte 31 (most significant)
3D	FE3D	Randomized Lot ID Byte 30
3E	FE3E	Randomized Lot ID Byte 29
3F	FE3F	Randomized Lot ID Byte 28
58	FE58	Randomized Lot ID Byte 27
59	FE59	Randomized Lot ID Byte 26
5A	FE5A	Randomized Lot ID Byte 25
5B	FE5B	Randomized Lot ID Byte 24

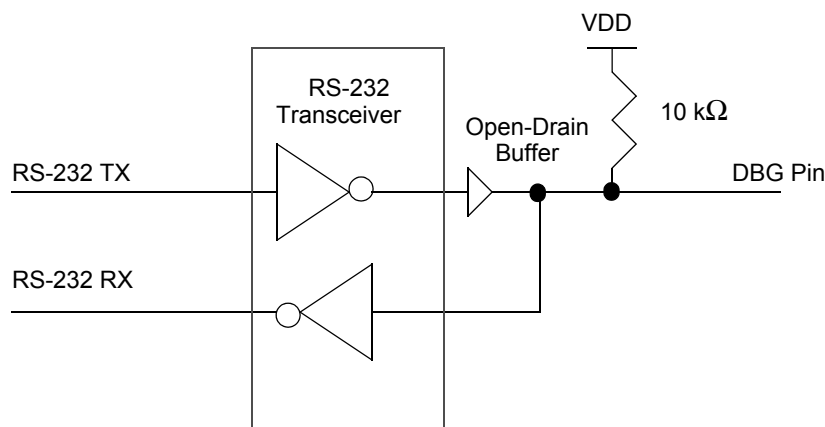


Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

DEBUG Mode

The operating characteristics of the devices in DEBUG mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions.
- The system clock operates unless in STOP mode.
- All enabled on-chip peripherals operate unless in STOP mode.
- Automatically exits HALT mode.
- Constantly refreshes the Watchdog Timer, if enabled.

Entering DEBUG Mode

The operating characteristics of the devices entering DEBUG mode are:

- The device enters DEBUG mode after the eZ8 CPU executes a BRK (Breakpoint) instruction.
- If the DBG pin is held Low during the final clock cycle of system reset, the part enters DEBUG mode immediately (20-/28-pin products only).

► **Note:** *Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an Autobaud sequence (see [OCD Auto-Baud Detector/Generator](#) on page 176).*



Caution: *It is possible to disable the clock failure detection circuitry as well as all functioning clock sources. In this case, the Z8 Encore! XP F082A Series device ceases functioning and can only be recovered by Power-On-Reset.*

Oscillator Control Register Definitions

Oscillator Control Register

The Oscillator Control Register (OSCCTL) enables/disables the various oscillator circuits, enables/disables the failure detection/recovery circuitry and selects the primary oscillator, which becomes the system clock.

The Oscillator Control Register must be unlocked before writing. Writing the two step sequence E7H followed by 18H to the Oscillator Control Register unlocks it. The register is locked at successful completion of a register write to the OSCCTL.

Table 109. Oscillator Control Register (OSCCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	INTEN	XTLEN	WDTEN	SOFEN	WDFEN	SCKSEL		
RESET	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F86H							

INTEN—Internal Precision Oscillator Enable

1 = Internal precision oscillator is enabled

0 = Internal precision oscillator is disabled

XTLEN—Crystal Oscillator Enable; this setting overrides the GPIO register control for PA0 and PA1

1 = Crystal oscillator is enabled

0 = Crystal oscillator is disabled

WDTEN—Watchdog Timer Oscillator Enable

1 = Watchdog Timer oscillator is enabled

0 = Watchdog Timer oscillator is disabled

SOFEN—System Clock Oscillator Failure Detection Enable

1 = Failure detection and recovery of system clock oscillator is enabled

0 = Failure detection and recovery of system clock oscillator is disabled

Assembly Language Syntax

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

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

Table 112. Assembly Language Syntax Example 1

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

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

Table 113. Assembly Language Syntax Example 2

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

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

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status flags, and address modes are represented by a notational shorthand that is described in [Table 114](#).

Table 115. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow. For example,

$$\text{dst} \leftarrow \text{dst} + \text{src}$$

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

eZ8 CPU Instruction Classes

eZ8 CPU instructions can be divided functionally into the following groups:

- Arithmetic
- Bit Manipulation
- Block Transfer
- CPU Control
- Load
- Logical
- Program Control
- Rotate and Shift

Table 116 through Table 123 lists the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instruction can be considered as a subset of more than one category. Within these tables, the source operand is identified as ‘src’, the destination operand is ‘dst’ and a condition code is ‘cc’.

Table 116. Arithmetic Instructions

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

Table 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
COM dst	dst ← ~dst	R		60	–	*	*	0	–	–	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	–	–	2	3
		r	lr	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	–	–	3	3
		r	lr	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	–	–	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	–	–	4	3
		ER	IM	A9							4	3
DA dst	dst ← DA(dst)	R		40	*	*	*	X	–	–	2	2
		IR		41							2	3
DEC dst	dst ← dst - 1	R		30	–	*	*	*	–	–	2	2
		IR		31							2	3
DECW dst	dst ← dst - 1	RR		80	–	*	*	*	–	–	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	–	–	–	–	–	–	1	2
DJNZ dst, RA	dst ← dst - 1 if dst ≠ 0 PC ← PC + X	r		0A-FA	–	–	–	–	–	–	2	3
EI	IRQCTL[7] ← 1			9F	–	–	–	–	–	–	1	2
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

Figure 38 and Table 143 provide timing information for UART pins for the case where CTS is not used for flow control. DE asserts after the transmit data register has been written. DE remains asserted for multiple characters as long as the transmit data register is written with the next character before the current character has completed.

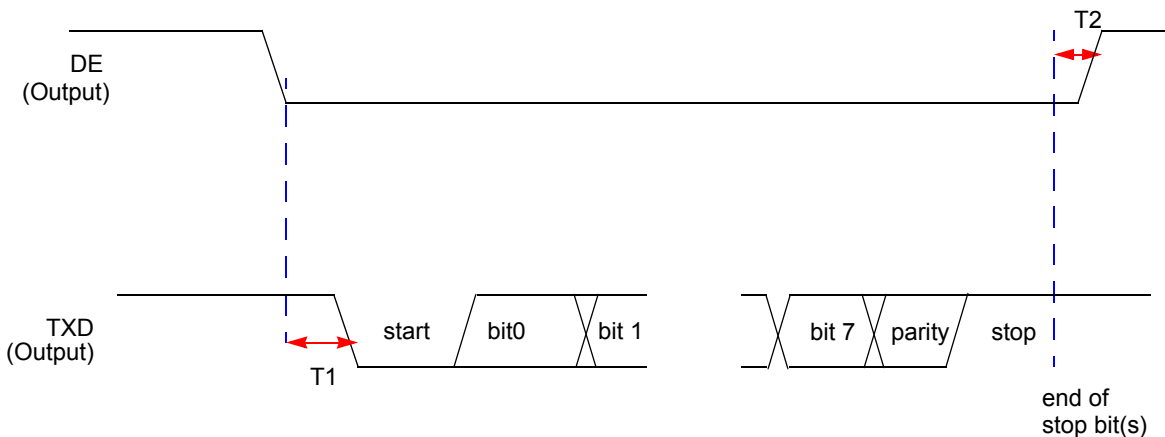


Figure 38. UART Timing Without CTS

Table 143. UART Timing Without CTS

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
UART			
T ₁	DE assertion to TXD falling edge (start bit) delay	1 * XIN period	1 bit time
T ₂	End of Stop Bit(s) to DE deassertion delay (Tx data register is empty)	± 5	

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP[®] F082A Series with 1 KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0 °C to 70 °C											
Z8F012APB020SC	1 KB	256 B	16 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F012AQB020SC	1 KB	256 B	16 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F012ASB020SC	1 KB	256 B	16 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F012ASH020SC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F012AHH020SC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F012APH020SC	1 KB	256 B	16 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F012ASJ020SC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F012AHJ020SC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F012APJ020SC	1 KB	256 B	16 B	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C											
Z8F012APB020EC	1 KB	256 B	16 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F012AQB020EC	1 KB	256 B	16 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F012ASB020EC	1 KB	256 B	16 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F012ASH020EC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F012AHH020EC	1 KB	256 B	16 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F012APH020EC	1 KB	256 B	16 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F012ASJ020EC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F012AHJ020EC	1 KB	256 B	16 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F012APJ020EC	1 KB	256 B	16 B	23	20	2	8	1	1	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging											