



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	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	6
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
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	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f041asb020sg

Reset, Stop Mode Recovery and Low Voltage Detection	22
Reset Types	22
Reset Sources	24
Power-On Reset	24
Voltage Brown-Out Reset	25
Watchdog Timer Reset	26
External Reset Input	26
External Reset Indicator	27
On-Chip Debugger Initiated Reset	27
Stop Mode Recovery	27
Stop Mode Recovery Using Watchdog Timer Time-Out	28
Stop Mode Recovery Using a GPIO Port Pin Transition	28
Stop Mode Recovery Using the External RESET Pin	29
Low Voltage Detection	29
Reset Register Definitions	29
Low-Power Modes	32
STOP Mode	32
HALT Mode	33
Peripheral-Level Power Control	33
Power Control Register Definitions	33
General-Purpose Input/Output	36
GPIO Port Availability By Device	36
Architecture	37
GPIO Alternate Functions	37
Direct LED Drive	38
Shared Reset Pin	38
Shared Debug Pin	39
Crystal Oscillator Override	39
5 V Tolerance	39
External Clock Setup	39
GPIO Interrupts	44
GPIO Control Register Definitions	44
Port A–D Address Registers	45
Port A–D Control Registers	46
Port A–D Data Direction Subregisters	46
Port A–D Alternate Function Subregisters	47
Port A–C Input Data Registers	52
Port A–D Output Data Register	52
LED Drive Enable Register	53
LED Drive Level High Register	53

Operating Modes	198
Crystal Oscillator Operation	198
Oscillator Operation with an External RC Network	201
Internal Precision Oscillator	203
Operation	203
eZ8 CPU Instruction Set	204
Assembly Language Programming Introduction	204
Assembly Language Syntax	205
eZ8 CPU Instruction Notation	206
eZ8 CPU Instruction Classes	207
eZ8 CPU Instruction Summary	212
Opcode Maps	222
Electrical Characteristics	226
Absolute Maximum Ratings	226
DC Characteristics	227
AC Characteristics	232
On-Chip Peripheral AC and DC Electrical Characteristics	233
General Purpose I/O Port Input Data Sample Timing	240
General Purpose I/O Port Output Timing	241
On-Chip Debugger Timing	242
UART Timing	243
Packaging	245
Ordering Information	246
Part Number Suffix Designations	255
Index	256
Customer Support	265

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
F85	Reserved	—	XX	
Oscillator Control				
F86	Oscillator Control	OSCCTL	A0	196
F87–F8F	Reserved	—	XX	
Comparator 0				
F90	Comparator 0 Control	CMP0	14	141
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	60
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	63
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	63
FC3	Interrupt Request 1	IRQ1	00	61
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	65
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	65
FC6	Interrupt Request 2	IRQ2	00	62
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	66
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	67
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	68
FCE	Shared Interrupt Select	IRQSS	00	68
FCF	Interrupt Control	IRQCTL	00	69
GPIO Port A				
FD0	Port A Address	PAADDR	00	44
FD1	Port A Control	PACTL	00	46
FD2	Port A Input Data	PAIN	XX	46
FD3	Port A Output Data	PAOUT	00	46
GPIO Port B				
FD4	Port B Address	PBADDR	00	44
FD5	Port B Control	PBCTL	00	46
FD6	Port B Input Data	PBIN	XX	46
FD7	Port B Output Data	PBOUT	00	46
GPIO Port C				
FD8	Port C Address	PCADDR	00	44

Notes:

1. XX = Undefined.
2. Refer to the [eZ8 CPU Core User Manual \(UM0128\)](#).

and as long as four. A reset pulse three clock cycles in duration might trigger a reset; a pulse four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8 Encore! XP F082A Series devices remain in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state on the system clock rising edge following $\overline{\text{RESET}}$ pin deassertion. Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the Reset Status (RSTSTAT) Register is set to 1.

External Reset Indicator

During System Reset or when enabled by the GPIO logic (see [Table 20 on page 46](#)), the $\overline{\text{RESET}}$ pin functions as an open-drain (active Low) reset mode indicator in addition to the input functionality. This reset output feature allows a Z8 Encore! XP F082A Series device to reset other components to which it is connected, even if that reset is caused by internal sources such as POR, VBO or WDT events.

After an internal reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin Low. The $\overline{\text{RESET}}$ pin is held Low by the internal circuitry until the appropriate delay listed in [Table 8](#) has elapsed.

On-Chip Debugger Initiated Reset

A Power-On Reset can be initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The On-Chip Debugger block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the system reset. Following the system reset the POR bit in the Reset Status (RSTSTAT) Register is set.

Stop Mode Recovery

STOP Mode is entered by execution of a STOP instruction by the eZ8 CPU. See the [Low-Power Modes](#) chapter on page 32 for detailed STOP Mode information. During Stop Mode Recovery (SMR), the CPU is held in reset for 66 IPO cycles if the crystal oscillator is disabled or 5000 cycles if it is enabled. The SMR delay (see [Table 135](#) on page 233) T_{SMR} , also includes the time required to start up the IPO.

Stop Mode Recovery does not affect on-chip registers other than the Watchdog Timer Control Register (WDTCTL) and the Oscillator Control Register (OSCCTL). After any Stop Mode Recovery, the IPO is enabled and selected as the system clock. If another system clock source is required, the Stop Mode Recovery code must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vec-

Shared Debug Pin

On the 8-pin version of this device only, the Debug pin shares function with the PA0 GPIO pin. This pin performs as a general purpose input pin on power-up, but the debug logic monitors this pin during the reset sequence to determine if the unlock sequence occurs. If the unlock sequence is present, the debug function is unlocked and the pin no longer functions as a GPIO pin. If it is not present, the debug feature is disabled until/unless another reset event occurs. For more details, see the [On-Chip Debugger](#) chapter on page 180.

Crystal Oscillator Override

For systems using a crystal oscillator, PA0 and PA1 are used to connect the crystal. When the crystal oscillator is enabled, the GPIO settings are overridden and PA0 and PA1 are disabled. See the [Oscillator Control Register Definitions](#) section on page 196 for details.

5V 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 such that the external oscillator is selected as the system clock. See the [Oscillator Control Register Definitions](#) section on page 196 for details. For 8-pin devices, use PA1 instead of PB3.

Table 15. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port C ⁵	PC0	Reserved		AFS1[0]: 0
		ANA4/CINP	ADC or Comparator Input	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/CINN	ADC or Comparator Input	AFS1[1]: 1
	PC2	Reserved		AFS1[2]: 0
		ANA6/V _{REF} ⁴	ADC Analog Input or ADC Voltage Reference	AFS1[2]: 1
	PC3	COUT	Comparator Output	AFS1[3]: 0
		Reserved		AFS1[3]: 1
	PC4	Reserved		AFS1[4]: 0
				AFS1[4]: 1
	PC5	Reserved		AFS1[5]: 0
				AFS1[5]: 1
	PC6	Reserved		AFS1[6]: 0
				AFS1[6]: 1
	PC7	Reserved		AFS1[7]: 0
				AFS1[7]: 1
Port D ⁶	PD0	RESET	External Reset	N/A

Notes:

1. Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections automatically enables the associated alternate function. See the [Port A–D Alternate Function Subregisters \(PxAF\)](#) section on page 47 for details.
2. Whether PA0/PA6 takes on the timer input or timer output complement function depends on the timer configuration. See the [Timer Pin Signal Operation](#) section on page 84 for details.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the [Port A–D Alternate Function Subregisters \(PxAF\)](#) section on page 47 for details.
4. V_{REF} is available on PB5 in 28-pin products and on PC2 in 20-pin parts.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the [Port A–D Alternate Function Subregisters \(PxAF\)](#) section on page 47 for details.
6. Because there is only a single alternate function for the Port PD0 pin, the Alternate Function Set registers are not implemented for Port D. Enabling alternate function selections automatically enables the associated alternate function. See the [Port A–D Alternate Function Subregisters \(PxAF\)](#) section on page 47 for details.

Port A–D Control Registers

The Port A–D Control registers set the GPIO port operation. The value in the corresponding Port A–D Address Register determines which subregister is read from or written to by a Port A–D Control Register transaction; see Table 20.

Table 20. Port A–D Control Registers (PxCTL)

Bit	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
Address	FD1H, FD5H, FD9H, FDDH							

Bit	Description
[7:0]	Port Control
PCTLx	The Port Control Register provides access to all subregisters that configure the GPIO port operation.

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

Port A–D Data Direction Subregisters

The Port A–D Data Direction subregister is accessed through the Port A–D Control Register by writing 01H to the Port A–D Address Register; see Table 21.

Table 21. Port A–D Data Direction Subregisters (PxDD)

Bit	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
Address	If 01H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0]	Data Direction
DDx	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.

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

Table 42. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	7	6	5	4	3	2	1	0
Field	PA7VENH	PA6CENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC4H							

Bit	Description
[7] PA7VENH	Port A Bit[7] or LVD Interrupt Request Enable High Bit
[6] PA6CENH	Port A Bit[7] or Comparator Interrupt Request Enable High Bit
[5:0] PAxENH	Port A Bit[x] Interrupt Request Enable High Bit

See the [Shared Interrupt Select Register \(IRQSS\) Register](#) on page 68 for selection of either the LVD or the comparator as the interrupt source.

Table 43. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bit	7	6	5	4	3	2	1	0
Field	PA7VENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC5H							

Bit	Description
[7] PA7VENL	Port A Bit[7] or LVD Interrupt Request Enable Low Bit
[6] PA6CENL	Port A Bit[6] or Comparator Interrupt Request Enable Low Bit
[5:0] PAxENL	Port A Bit[x] Interrupt Request Enable Low Bit

IRQ2 Enable High and Low Bit Registers

Table 44 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers, shown in Tables 44 and 45, form a priority-encoded enabling for interrupts in the Interrupt Request 2 Register.

Table 46. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				C3ENL	C2ENL	C1ENL	C0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC8H							

Bit	Description
[7:4]	Reserved These bits are reserved and must be programmed to 0000.
[3] C3ENL	Port C3 Interrupt Request Enable Low Bit
[2] C2ENL	Port C2 Interrupt Request Enable Low Bit
[1] C1ENL	Port C1 Interrupt Request Enable Low Bit
[0] C0ENL	Port C0 Interrupt Request Enable Low Bit

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 47, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

Table 47. Interrupt Edge Select Register (IRQES)

Bit	7	6	5	4	3	2	1	0
Field	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FCDH							

Bit	Description
[7:0] IESx	Interrupt Edge Select x 0 = An interrupt request is generated on the falling edge of the PAX input. 1 = An interrupt request is generated on the rising edge of the PAX input.
Note: x indicates the specific GPIO port pin number (0–7).	

- 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 Compare value.
 4. Enable the timer interrupt, if appropriate and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt are generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL0 Register.
 5. Configure the associated GPIO port pin for the Timer Input alternate function.
 6. Write to the Timer Control Register to enable the timer.
 7. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

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

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

Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte Register is read, the contents of the Timer Low Byte Register are placed in a holding register. A subsequent read from the Timer Low Byte Register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte Register returns the actual value in the counter.

Timer Pin Signal Operation

The timer output function is a GPIO port pin alternate function. The Timer Output is toggled every time the counter is reloaded.

The timer input can be used as a selectable counting source. It shares the same pin as the complementary timer output. When selected by the GPIO Alternate Function registers, this pin functions as a timer input in all modes except for the DUAL PWM OUTPUT mode. For this mode, there is no timer input available.

Timer Control Register Definitions

This section defines the features of the following Timer Control registers.

Timer 0–1 Control Registers: see page 85

Timer 0–1 High and Low Byte Registers: see page 89

Timer Reload High and Low Byte Registers: see page 91

Timer 0–1 PWM High and Low Byte Registers: see page 92

Timer 0–1 Control Registers

The Timer Control registers are 8-bit read/write registers that control the operation of their associated counter/timers.

Time 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and Timer Control Register 1 (TxCTL1), shown in Table 50, determine the timer operating mode. These registers each include a programmable PWM deadband delay, two bits to configure timer interrupt definition and a status bit to identify if the most recent timer interrupt is caused by an input capture event.

Table 50. Timer 0–1 Control Register 0 (TxCTL0)

Bit	7	6	5	4	3	2	1	0
Field	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
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
Address	F06H, F0EH							

Bit	Description
[7] TMODEHI	Timer Mode High Bit This bit, along with the TMODE field in the TxCTL1 Register, determines the operating mode of the timer. This bit is the most significant bit of the Timer mode selection value. See the description of the <u>Timer 0–1 Control Register 1 (TxCTL1)</u> for details about the full timer mode decoding.

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Checks the UART Status 0 Register to determine the source of the interrupt - error, break, or received data.
2. Reads the data from the UART Receive Data Register if the interrupt was because of data available. If operating in MULTIPROCESSOR (9-bit) Mode, further actions may be required depending on the MULTIPROCESSOR Mode bits MPMD[1:0].
3. Clears the UART Receiver interrupt in the applicable Interrupt Request Register.
4. Executes 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 ($\overline{\text{CTS}}$) 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 $\overline{\text{CTS}}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this action is typically performed during Stop Bit transmission. If $\overline{\text{CTS}}$ deasserts in the middle of a character transmission, the current character is sent completely.

MULTIPROCESSOR (9-bit) Mode

The UART features 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 13. The character format is:

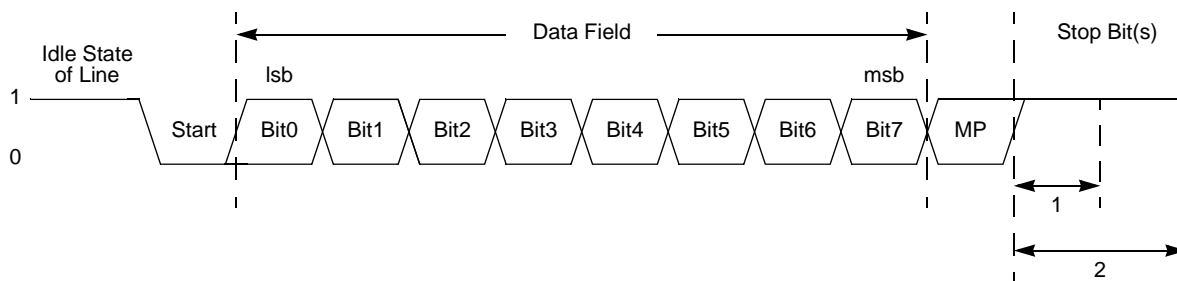


Figure 13. UART Asynchronous MULTIPROCESSOR Mode Data Format

The baud rate is set by the UART's Baud Rate Generator and supports IrDA standard baud rates from 9600 baud to 115.2 kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the infrared endec. The infrared endec data rate is calculated using the following equation:

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

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16 clocks wide. If the data to be transmitted is 1, the IR_TXD signal remains low for the full 16 clock period. If the data to be transmitted is 0, the transmitter first outputs a 7 clock low period, followed by a 3 clock high pulse. Finally, a 6 clock low pulse is output to complete the full 16 clock data period. Figure 17 displays IrDA data transmission. When the infrared endec is enabled, the UART's TXD signal is internal to the Z8 Encore! XP F082A Series products while the IR_TXD signal is output through the TXD pin.

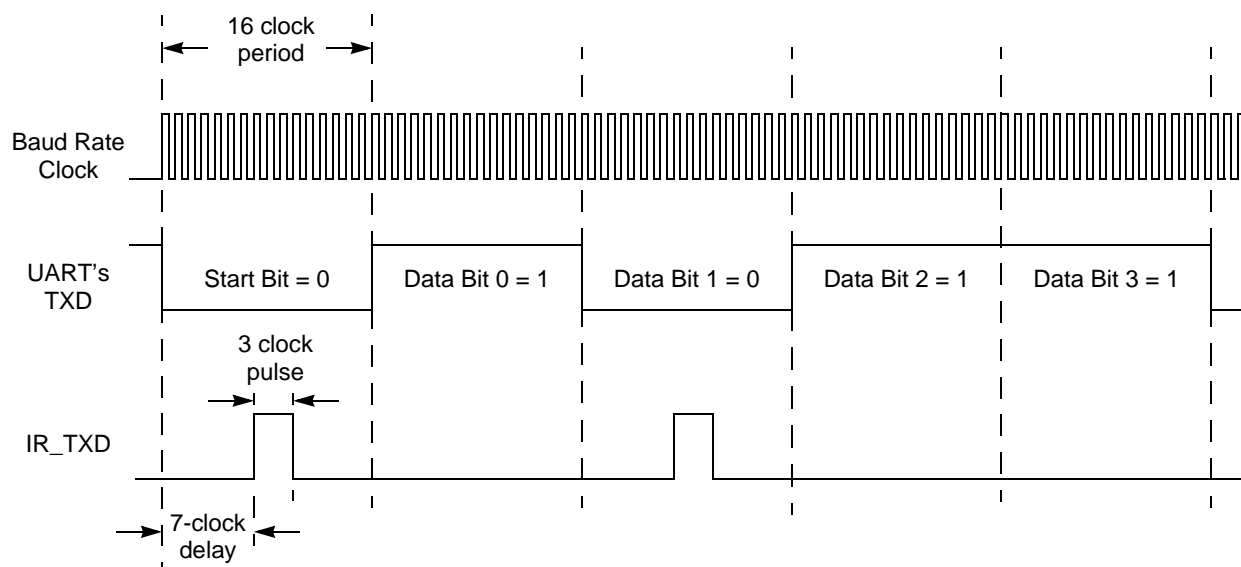


Figure 17. Infrared Data Transmission

Receiving IrDA Data

Data received from the infrared transceiver using the IR_RXD signal through the RXD pin is decoded by the infrared endec and passed to the UART. The UART's baud rate clock is used by the infrared endec to generate the demodulated signal (RXD) that drives the UART. Each UART/Infrared data bit is 16-clocks wide. Figure 18 displays data reception. When the infrared endec is enabled, the UART's RXD signal is internal to the Z8 Encore! XP F082A Series products while the IR_RXD signal is received through the RXD pin.

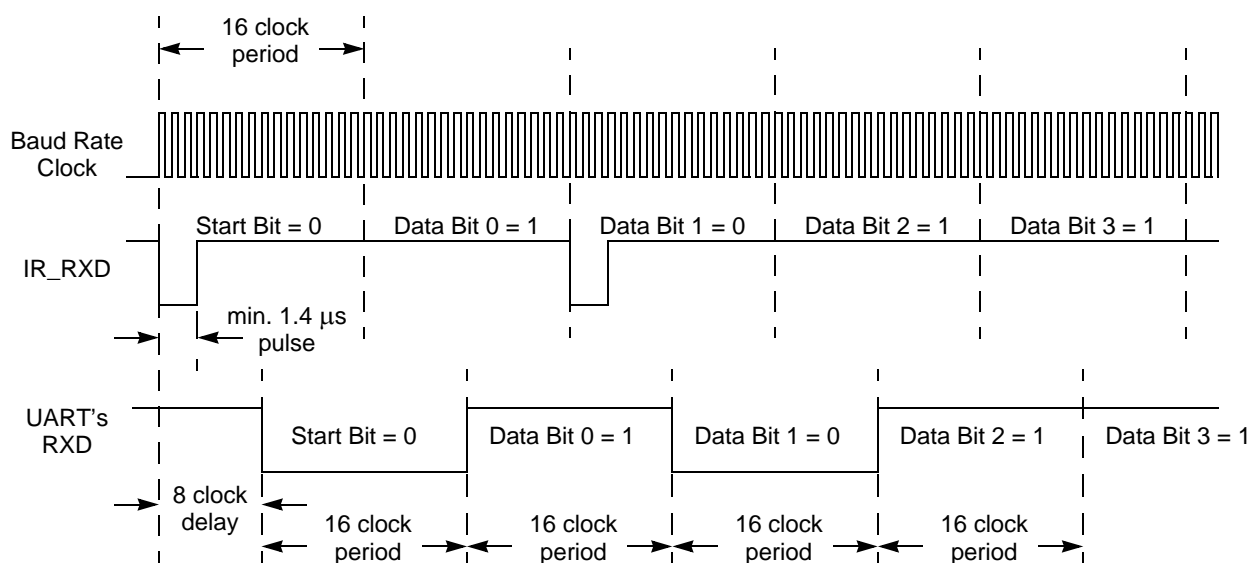


Figure 18. IrDA Data Reception

Infrared Data Reception

! Caution: The system clock frequency must be at least 1.0MHz to ensure proper reception of the 1.4μs minimum width pulses allowed by the IrDA standard.

Endec Receiver Synchronization

The IrDA receiver uses a local baud rate clock counter (0 to 15 clock periods) to generate an input stream for the UART and to create a sampling window for detection of incoming pulses. The generated UART input (UART RXD) is delayed by 8 baud rate clock periods with respect to the incoming IrDA data stream. When a falling edge in the input data stream is detected, the Endec counter is reset. When the count reaches a value of 8, the UART RXD value is updated to reflect the value of the decoded data. When the count reaches 12 baud clock periods, the sampling window for the next incoming pulse opens.

The window remains open until the count again reaches 8 (that is, 24 baud clock periods since the previous pulse was detected), giving the Endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal, allowing the Endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART Control registers as defined in the Universal Asynchronous Receiver/Transmitter section on page 99.

! **Caution:** To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.

Table 97. ADC Calibration Data Location (Continued)

Info Page Address	Memory Address	Compensation Usage	ADC Mode	Reference Type
12	FE12	Positive Gain High Byte	Differential Unbuffered	Internal 2.0 V
13	FE13	Positive Gain Low Byte	Differential Unbuffered	Internal 2.0 V
30	FE30	Negative Gain High Byte	Differential Unbuffered	Internal 2.0 V
31	FE31	Negative Gain Low Byte	Differential Unbuffered	Internal 2.0 V
72	FE72	Offset	Differential Unbuffered	Internal 1.0 V
14	FE14	Positive Gain High Byte	Differential Unbuffered	Internal 1.0 V
15	FE15	Positive Gain Low Byte	Differential Unbuffered	Internal 1.0 V
32	FE32	Negative Gain High Byte	Differential Unbuffered	Internal 1.0 V
33	FE33	Negative Gain Low Byte	Differential Unbuffered	Internal 1.0 V
75	FE75	Offset	Differential Unbuffered	External 2.0 V
16	FE16	Positive Gain High Byte	Differential Unbuffered	External 2.0 V
17	FE17	Positive Gain Low Byte	Differential Unbuffered	External 2.0 V
34	FE34	Negative Gain High Byte	Differential Unbuffered	External 2.0 V
35	FE35	Negative Gain Low Byte	Differential Unbuffered	External 2.0 V
78	FE78	Offset	Differential 1x Buffered	Internal 2.0 V
18	FE18	Positive Gain High Byte	Differential 1x Buffered	Internal 2.0 V
19	FE19	Positive Gain Low Byte	Differential 1x Buffered	Internal 2.0 V
36	FE36	Negative Gain High Byte	Differential 1x Buffered	Internal 2.0 V
37	FE37	Negative Gain Low Byte	Differential 1x Buffered	Internal 2.0 V
7B	FE7B	Offset	Differential 1x Buffered	External 2.0 V
1A	FE1A	Positive Gain High Byte	Differential 1x Buffered	External 2.0 V
1B	FE1B	Positive Gain Low Byte	Differential 1x Buffered	External 2.0 V
38	FE38	Negative Gain High Byte	Differential 1x Buffered	External 2.0 V
39	FE39	Negative Gain Low Byte	Differential 1x Buffered	External 2.0 V

Byte Write

To write a byte to the NVDS array, the user code must first push the address, then the data byte onto the stack. The user code issues a `CALL` instruction to the address of the byte-write routine (0x10B3). At the return from the sub-routine, the write status byte resides in working register R0. The bit fields of this status byte are defined in Table 106. The contents of the status byte are undefined for write operations to illegal addresses. Also, user code must pop the address and data bytes off the stack.

The write routine uses 13 bytes of stack space in addition to the two bytes of address and data pushed by the user. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS writes exhibit a nonuniform execution time. In general, a write takes 251 μ s (assuming a 20MHz system clock). Every 400 to 500 writes, however, a maintenance operation is necessary. In this rare occurrence, the write takes up to 61 ms to complete. Slower system clock speeds result in proportionally higher execution times.

NVDS byte writes to invalid addresses (those exceeding the NVDS array size) have no effect. Illegal write operations have a 2 μ s execution time.

Table 106. Write Status Byte

Bit	7	6	5	4	3	2	1	0
Field	Reserved				RCPY	PF	AWE	DWE
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:4]	Reserved These bits are reserved and must be programmed to 0000.
[3] RCPY	Recopy Subroutine Executed A recopy subroutine was executed. These operations take significantly longer than a normal write operation.
[2] PF	Power Failure Indicator A power failure or system reset occurred during the most recent attempted write to the NVDS array.
[1] AWE	Address Write Error An address byte failure occurred during the most recent attempted write to the NVDS array.
[0] DWE	Data Write Error A data byte failure occurred during the most recent attempted write to the NVDS array.

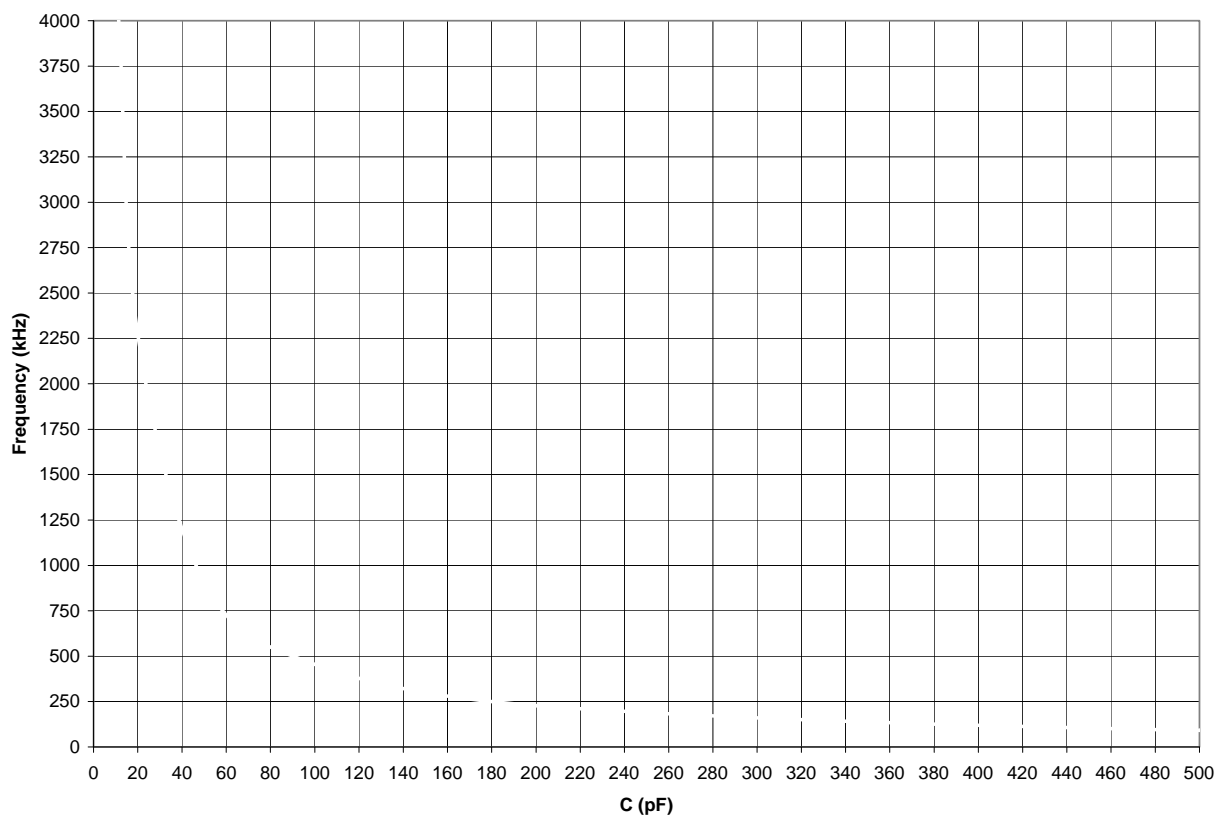


Figure 29. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45k Ω Resistor

! Caution: When using the external RC oscillator mode, the oscillator can stop oscillating if the power supply drops below 2.7 V, but before the power supply drops to the Voltage Brown-Out threshold. The oscillator resumes oscillation when the supply voltage exceeds 2.7 V.

Table 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycle s	Instr. Cycle s
		dst	src		C	Z	S	V	D	H		
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
HALT	Halt Mode			7F	–	–	–	–	–	–	1	2
INC dst	dst ← dst + 1	R		20	–	*	*	–	–	–	2	2
		IR		21							2	3
		r		0E-FE							1	2
INCW dst	dst ← dst + 1	RR		A0	–	*	*	*	–	–	2	5
		IRR		A1							2	6
IRET	FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1			BF	*	*	*	*	*	*	1	5
JP dst	PC ← dst	DA		8D	–	–	–	–	–	–	3	2
		IRR		C4							2	3
JP cc, dst	if cc is true PC ← dst	DA		0D-FD	–	–	–	–	–	–	3	2

Note: Flags Notation:

* = Value is a function of the result of the operation.

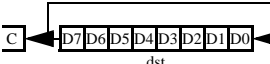
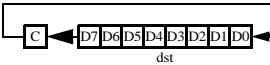

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycle s	Instr. Cycle s
		dst	src		C	Z	S	V	D	H		
ORX dst, src	dst ← dst OR src	ER	ER	48	–	*	*	0	–	–	4	3
		ER	IM	49							4	3
POP dst	dst ← @SP SP ← SP + 1	R		50	–	–	–	–	–	–	2	2
		IR		51							2	3
POPX dst	dst ← @SP SP ← SP + 1	ER		D8	–	–	–	–	–	–	3	2
PUSH src	SP ← SP – 1 @SP ← src	R		70	–	–	–	–	–	–	2	2
		IR		71							2	3
		IM		IF70							3	2
PUSHX src	SP ← SP – 1 @SP ← src	ER		C8	–	–	–	–	–	–	3	2
RCF	C ← 0			CF	0	–	–	–	–	–	1	2
RET	PC ← @SP SP ← SP + 2			AF	–	–	–	–	–	–	1	4
RL dst		R		90	*	*	*	*	–	–	2	2
		IR		91							2	3
RLC dst		R		10	*	*	*	*	–	–	2	2
		IR		11							2	3
RR dst		R		E0	*	*	*	*	–	–	2	2
		IR		E1							2	3

Note: Flags Notation:

* = Value is a function of the result of the operation.

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.