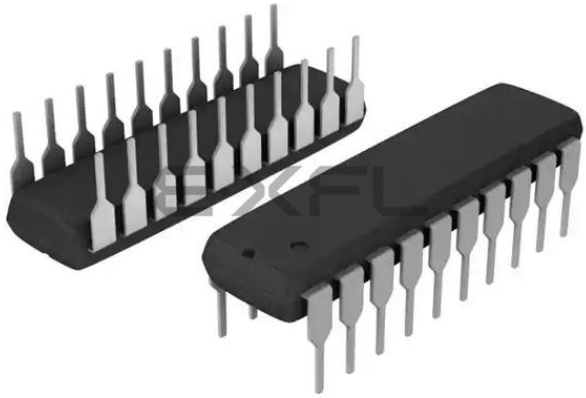

Welcome to E-XFL.COM

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	-
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	17
Program Memory Size	12KB (12K x 8)
Program Memory Type	FLASH
EEPROM Size	-
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	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1233ph020eg

List of Figures

The analog supply pins (AV_{DD} and AV_{SS}) are also not available on these parts and are replaced by PB6 and PB7.

At reset, by default, all pins of Port A, B and C are in Input state. The alternate functionality is also disabled, so the pins function as general purpose input ports until programmed otherwise. At power-up, the Port D0 pin defaults to the $\overline{\text{RESET}}$ Alternate function.

The pin configurations listed are preliminary and subject to change based on manufacturing limitations.

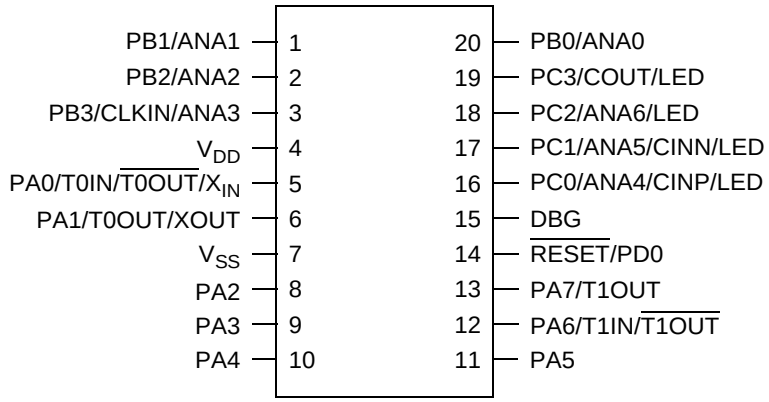

Figure 2. Z8F0830 Series in 20-Pin SOIC, SSOP, PDIP Package

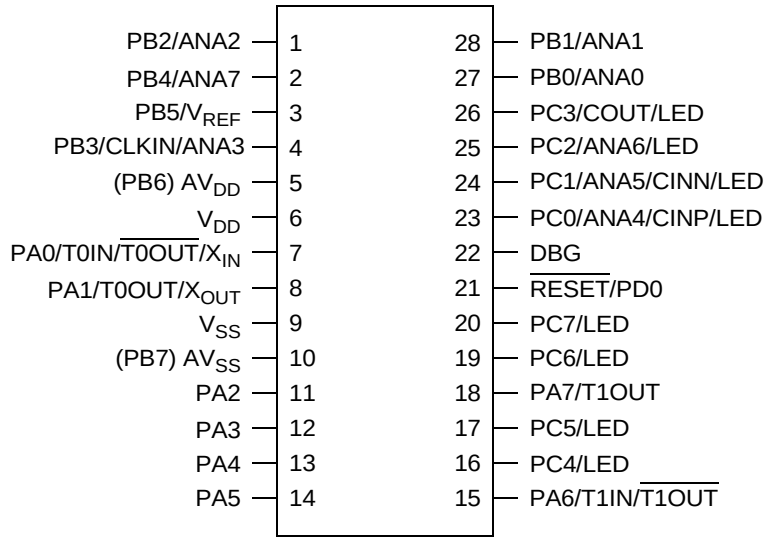

Figure 3. Z8F0830 Series in 28-Pin SOIC, SSOP, PDIP Package

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No.
Trim Bit Control				
FF6	Trim bit address	TRMADR	00	126
FF7	Trim data	TRMDR	XX	127
Flash Memory Controller				
FF8	Flash control	FCTL	00	119
FF8	Flash status	FSTAT	00	120
FF9	Flash page select	FPS	00	121
	Flash sector protect	FPROT	00	122
FFA	Flash programming frequency high byte	FFREQH	00	123
FFB	Flash programming frequency low byte	FFREQL	00	123
eZ8 CPU				
FFC	Flags	—	XX	Refer to the eZ8 CPU Core User Manual (UM0128)
FFD	Register pointer	RP	XX	
FFE	Stack pointer high byte	SPH	XX	
FFF	Stack pointer low byte	SPL	XX	

Note: XX = Undefined.

clock and reset signals, the required reset duration may be three or four clock periods. A reset pulse of three clock cycles in duration might trigger a reset and a reset pulse of four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted low, the Z8 Encore! F0830 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, 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 an Z8 Encore! F0830 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. See the Port A–D Control Registers section on page 41.

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 9 (see page 22) 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 OCD block is not reset, but the remainder 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

The device enters the STOP Mode when the STOP instruction is executed by the eZ8 CPU. See the Low-Power Modes chapter on page 30 for detailed STOP Mode information. During Stop Mode Recovery, the CPU is held in reset for about 66 IPO cycles if the crystal oscillator is disabled or about 5000 cycles if it is enabled.

Stop Mode Recovery does not affect the on-chip registers other than the Reset Status (RSTSTAT) Register 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 or IPO disabling is required, the Stop Mode Recovery code must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

Architecture

Figure 8 displays a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength is not displayed.

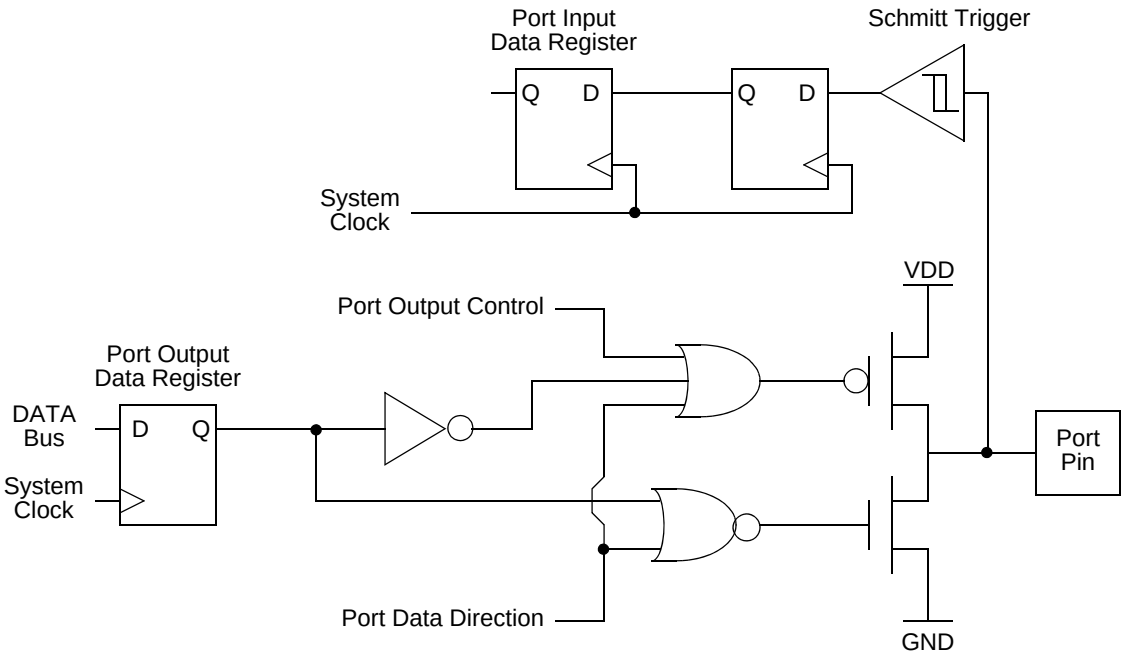

Figure 8. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins can be used for general purpose input/output and access to on-chip peripheral functions such as the timers and serial communication devices. The Port A–D Alternate Function subregisters configure these pins for either GPIO or Alternate function operation. When a pin is configured for Alternate function, control of the port pin direction (input/output) is passed from the Port A–D data direction registers to the Alternate function assigned to this pin. Table 16 on page 36 lists the alternate functions possible with each port pin. The alternate function associated at a pin is defined through Alternate Function subregisters AFS1 and AFS2.

The crystal oscillator functionality is not controlled by the GPIO block. When the crystal oscillator is enabled in the oscillator control block, the GPIO functionality of PA0 and PA1 is overridden. In that case, pins PA0 and PA1 functions as input and output for the crystal oscillator.

LED Drive Level Low Register

The LED Drive Level Low Register, shown in Table 33, contains two control bits for each Port C pin. These two bits select one of four programmable current drive levels for each Port C pin. Each pin is individually programmable.

Table 33. LED Drive Level Low Register (LEDLVLL)

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLL[7:0]							
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	F84H							

Bit	Description
[7:0] LEDLVLL	**LED Level Low Bits** {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 35 stores the interrupt requests for both vectored and polled interrupts. When a request is sent to the Interrupt Controller, the corresponding bit in the IRQ0 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 0 Register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 0 Register (IRQ0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	T0I	Reserved				ADCI
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	FC0H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1I	**Timer 1 Interrupt Request** 0 = No interrupt request is pending for timer 1. 1 = An interrupt request from timer 1 is awaiting service.
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for timer 0. 1 = An interrupt request from timer 0 is awaiting service.
[4:1]	**Reserved** These registers are reserved and must be programmed to 0000.
[0] 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.

ADC Timing

Each ADC measurement consists of three phases:

1. Input sampling (programmable, minimum of 1.0µs)
2. Sample-and-hold amplifier settling (programmable, minimum of 0.5µs)
3. Conversion is 13 ADCLK cycles

Figures 12 and 13 display the timing of an ADC conversion.

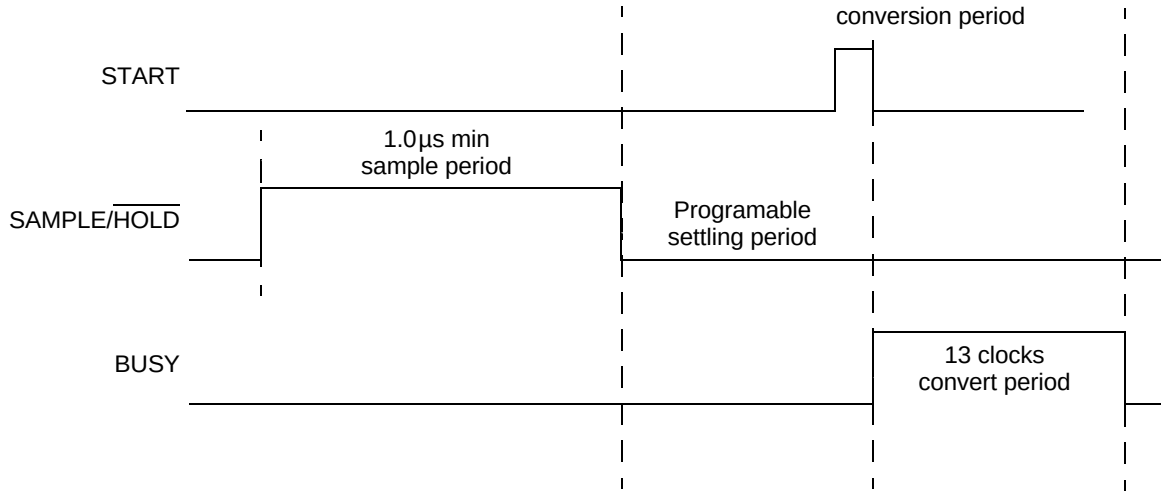

Figure 12. ADC Timing Diagram

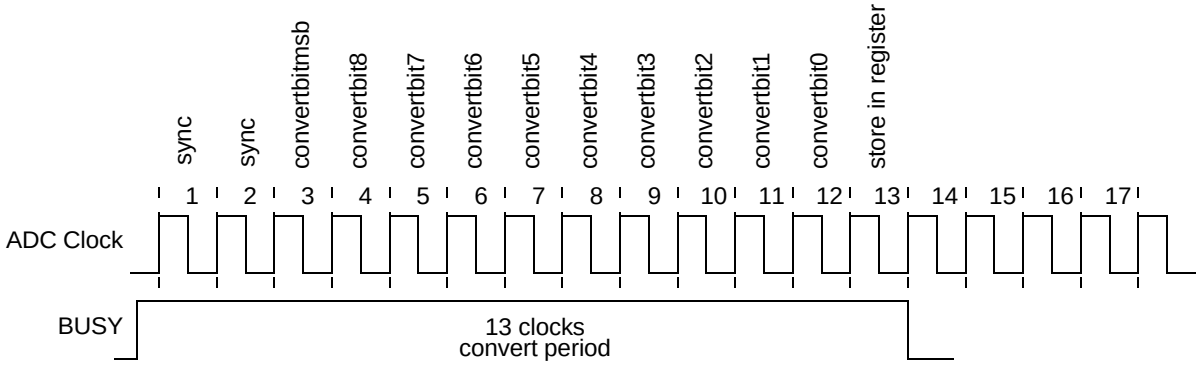

Figure 13. ADC Convert Timing

Sample Settling Time Register

The Sample Settling Time Register, shown in Table 66, is used to program a delay after the SAMPLE/HOLD signal is asserted and before the START signal is asserted; an ADC conversion then begins. The number of clock cycles required for settling will vary from system to system depending on the system clock period used. The system designer should program this register to contain the number of clocks required to meet a 0.5µs minimum settling time.

Table 66. Sample Settling Time (ADCSST)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				SST			
RESET	0				1	1	1	1
R/W	R				R/W			
Address	F74H							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] SST	0h–Fh = Sample settling time in number of system clock periods to meet 0.5 µs minimum.

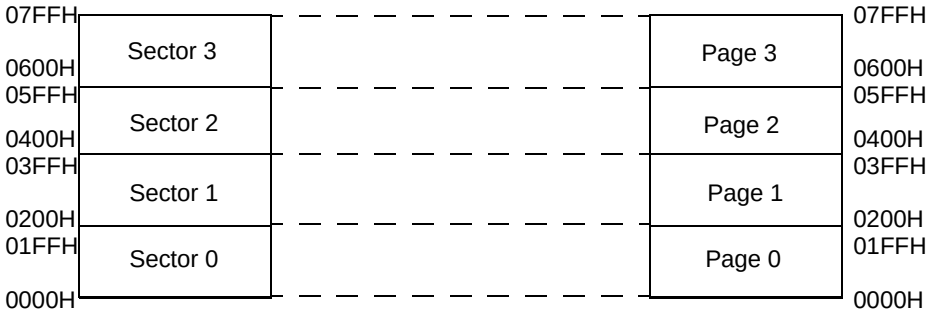

Figure 15. 2K Flash with NVDS

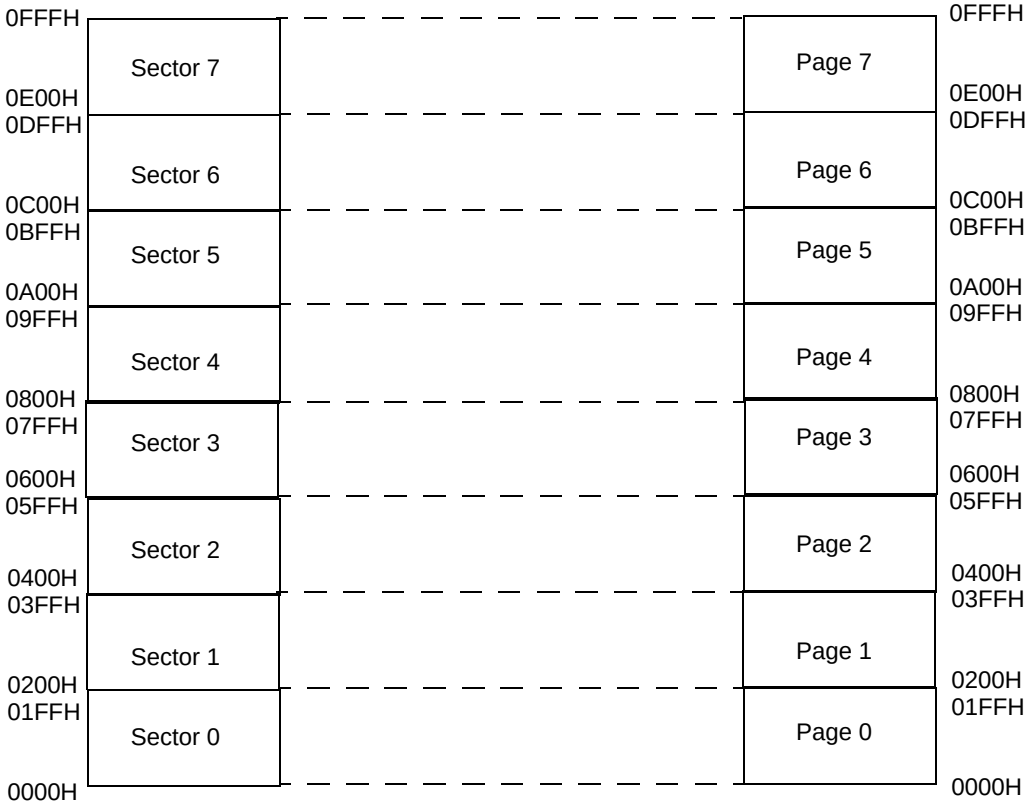

Figure 16. 4K Flash with NVDS

bits can only be set to 1. Thus, sectors can be protected, but not unprotected, via register write operations. Writing a value other than 5EH to the Flash Control Register deselects the Flash Sector Protect Register and reenables access to the Page Select Register. Observe the following procedure to setup the Flash Sector Protect Register from user code:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write 5EH to the Flash Control Register to select the Flash Sector Protect Register.
3. Read and/or write the Flash Sector Protect Register which is now at Register File address FF9H.
4. Write 00H to the Flash Control Register to return the Flash Controller to its reset state.

The Sector Protect Register is initialized to 0 on reset, putting each sector into an unprotected state. When a bit in the Sector Protect Register is written to 1, the corresponding sector can no longer be written or erased. After setting a bit in the Sector Protect Register, the bit cannot be cleared by the user.

Byte Programming

Flash memory is enabled for byte programming after unlocking the Flash Controller and successfully enabling either mass erase or page erase. When the Flash Controller is unlocked and mass erase is successfully enabled, all of the program memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and page erase is successfully enabled, only the locations of the selected page are available for byte programming. An erased Flash byte contains all 1's (FFH). The programming operation can only be used to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the page erase or mass erase commands.

Byte programming can be accomplished using the On-Chip Debugger's write memory command or eZ8 CPU execution of the LDC or LDCI instructions. Refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com, for the description of the LDC and LDCI instructions. While the Flash Controller programs the Flash memory, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. To exit programming mode and lock the Flash, write any value to the Flash Control Register, except the mass erase or page erase commands.

Caution: The byte at each address within Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs.

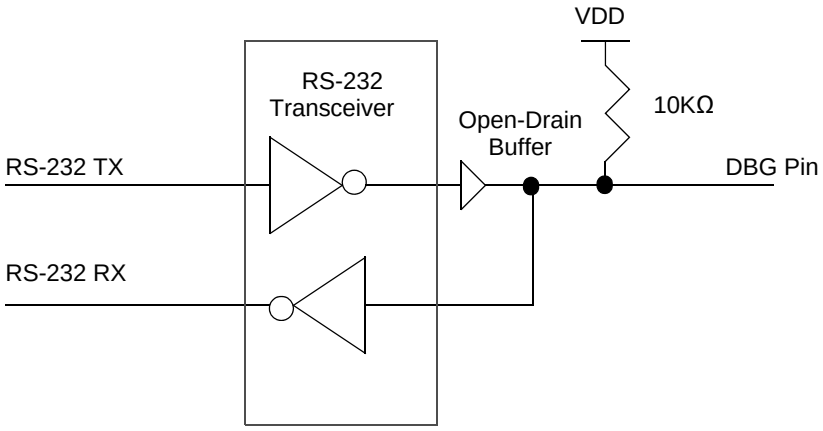

Figure 22. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #2 of 2

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 the device is in STOP Mode
- All enabled on-chip peripherals operate, unless the device is in STOP Mode
- Automatically exits HALT Mode
- Constantly refreshes the Watchdog Timer, if enabled

Entering DEBUG Mode

- The device enters DEBUG Mode after the eZ8 CPU executes a Breakpoint (BRK) instruction
- If the DBG pin is held low during the most recent clock cycle of system reset, the device enters DEBUG Mode on exiting system reset

Exiting DEBUG Mode

The device exits DEBUG Mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset

```
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 0). If the device is not in DEBUG Mode or if the Flash 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 0). If the device is not in DEBUG Mode or if the flash 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 from 1 to 65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG Mode, this command returns FFH for the data.

```
DBG ← 0DH
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 Program Memory CRC (0EH). The read program memory CRC command, computes and returns the cyclic redundancy check (CRC) of program memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG Mode, this command returns FFFFH for the CRC value. Unlike the other OCD read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads program memory, calculates the CRC value and returns the result. The delay is a function of program mem-

Table 98. Oscillator Configuration and Selection

Clock Source	Characteristics	Required Setup
Internal precision RC oscillator	• 32.8 kHz or 5.53 MHz • ± 4% accuracy when trimmed • No external components required	• Unlock and write to the Oscillator Control Register (OSCCTL) to enable and select oscillator at either 5.53 MHz or 32.8 kHz
External crystal/resonator	• 32 kHz to 20 MHz • Very high accuracy (dependent on crystal or resonator used) • Requires external components	• Configure Flash option bits for correct external OSCILLATOR Mode • Unlock and write OSCCTL to enable crystal oscillator, wait for it to stabilize and select as system clock (if the XTLDIS option bit has been de-asserted, no waiting is required)
External RC oscillator	• 32 kHz to 4 MHz • Accuracy dependent on external components	• Configure Flash option bits for correct external OSCILLATOR Mode • Unlock and write OSCCTL to enable crystal oscillator and select as system clock
External clock drive	• 0 to 20 MHz • Accuracy dependent on external clock source	• Write GPIO registers to configure PB3 pin for external clock function • Unlock and write OSCCTL to select external system clock • Apply external clock signal to GPIO
Internal Watchdog Timer Oscillator	• 10 kHz nominal • ± 40% accuracy; no external components required • Low power consumption	• Enable WDT if not enabled and wait until WDT oscillator is operating. • Unlock and write to the Oscillator Control Register (OSCCTL) to enable and select oscillator

Caution: Unintentional accesses to the Oscillator Control Register can actually stop the chip by switching to a nonfunctioning oscillator. To prevent this condition, the oscillator control block employs a register unlocking/locking scheme.

OSC Control Register Unlocking/Locking

To write the Oscillator Control Register, unlock it by making two writes to the OSCCTL Register with the values E7H followed by 18H. A third write to the OSCCTL Register changes the value of the actual register and returns the register to a Locked state. Any other sequence of Oscillator Control Register writes have no effect. The values written to unlock the register must be ordered correctly, but are not necessarily consecutive. It is possible to write to or read from other registers within the unlocking/locking operation.

Table 113. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	dst ← dst AND src	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	dst[bit] ← 0	r		E2	–	*	*	0	–	–	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	–	*	*	0	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	–	*	*	0	–	–	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
CALL dst	SP ← SP –2 @SP ← PC PC ← dst	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	C ← ~C			EF	*	–	–	–	–	–-	1	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.

Figures 29 and 30 provide information about each of the eZ8 CPU instructions.

Upper Nibble (Hex)	Lower Nibble (Hex) 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1.1 **BRK**	2.2 **SRP** IM	2.3 **ADD** r1,r2	2.4 **ADD** r1,Ir2	3.3 **ADD** R2,R1	3.4 **ADD** IR2,R1	3.3 **ADD** R1,IM	3.4 **ADD** IR1,IM	4.3 **ADDX** ER2,ER1	4.3 **ADDX** IM,ER1	2.3 **DJNZ** r1,X	2.2 **JR** cc,X	2.2 **LD** r1,IM	3.2 **JP** cc,DA	1.2 **INC** r1	1.2 **NOP**
1	2.2 **RLC** R1	2.3 **RLC** IR1	2.3 **ADC** r1,r2	2.4 **ADC** r1,Ir2	3.3 **ADC** R2,R1	3.4 **ADC** IR2,R1	3.3 **ADC** R1,IM	3.4 **ADC** IR1,IM	4.3 **ADCX** ER2,ER1	4.3 **ADCX** IM,ER1						See 2nd Op Code Map
2	2.2 **INC** R1	2.3 **INC** IR1	2.3 **SUB** r1,r2	2.4 **SUB** r1,Ir2	3.3 **SUB** R2,R1	3.4 **SUB** IR2,R1	3.3 **SUB** R1,IM	3.4 **SUB** IR1,IM	4.3 **SUBX** ER2,ER1	4.3 **SUBX** IM,ER1						
3	2.2 **DEC** R1	2.3 **DEC** IR1	2.3 **SBC** r1,r2	2.4 **SBC** r1,Ir2	3.3 **SBC** R2,R1	3.4 **SBC** IR2,R1	3.3 **SBC** R1,IM	3.4 **SBC** IR1,IM	4.3 **SBCX** ER2,ER1	4.3 **SBCX** IM,ER1						
4	2.2 **DA** R1	2.3 **DA** IR1	2.3 **OR** r1,r2	2.4 **OR** r1,Ir2	3.3 **OR** R2,R1	3.4 **OR** IR2,R1	3.3 **OR** R1,IM	3.4 **OR** IR1,IM	4.3 **ORX** ER2,ER1	4.3 **ORX** IM,ER1						
5	2.2 **POP** R1	2.3 **POP** IR1	2.3 **AND** r1,r2	2.4 **AND** r1,Ir2	3.3 **AND** R2,R1	3.4 **AND** IR2,R1	3.3 **AND** R1,IM	3.4 **AND** IR1,IM	4.3 **ANDX** ER2,ER1	4.3 **ANDX** IM,ER1						1.2 **WDT**
6	2.2 **COM** R1	2.3 **COM** IR1	2.3 **TCM** r1,r2	2.4 **TCM** r1,Ir2	3.3 **TCM** R2,R1	3.4 **TCM** IR2,R1	3.3 **TCM** R1,IM	3.4 **TCM** IR1,IM	4.3 **TCMX** ER2,ER1	4.3 **TCMX** IM,ER1						1.2 **STOP**
7	2.2 **PUSH** R2	2.3 **PUSH** IR2	2.3 **TM** r1,r2	2.4 **TM** r1,Ir2	3.3 **TM** R2,R1	3.4 **TM** IR2,R1	3.3 **TM** R1,IM	3.4 **TM** IR1,IM	4.3 **TMX** ER2,ER1	4.3 **TMX** IM,ER1						1.2 **HALT**
8	2.5 **DECW** RR1	2.6 **DECW** IRR1	2.5 **LDE** r1,Irr2	2.9 **LDEI** Ir1,Irr2	3.2 **LDX** r1,ER2	3.3 **LDX** Ir1,ER2	3.4 **LDX** IRR2,R1	3.5 **LDX** IRR2,IR1	3.4 **LDX** r1,rr2,X	3.4 **LDX** rr1,r2,X						1.2 **DI**
9	2.2 **RL** R1	2.3 **RL** IR1	2.5 **LDE** r2,Irr1	2.9 **LDEI** Ir2,Irr1	3.2 **LDX** r2,ER1	3.3 **LDX** Ir2,ER1	3.4 **LDX** R2,IRR1	3.5 **LDX** IR2,IRR1	3.3 **LEA** r1,r2,X	3.5 **LEA** rr1,rr2,X						1.2 **EI**
A	2.5 **INCW** RR1	2.6 **INCW** IRR1	2.3 **CP** r1,r2	2.4 **CP** r1,Ir2	3.3 **CP** R2,R1	3.4 **CP** IR2,R1	3.3 **CP** R1,IM	3.4 **CP** IR1,IM	4.3 **CPX** ER2,ER1	4.3 **CPX** IM,ER1						1.4 **RET**
B	2.2 **CLR** R1	2.3 **CLR** IR1	2.3 **XOR** r1,r2	2.4 **XOR** r1,Ir2	3.3 **XOR** R2,R1	3.4 **XOR** IR2,R1	3.3 **XOR** R1,IM	3.4 **XOR** IR1,IM	4.3 **XORX** ER2,ER1	4.3 **XORX** IM,ER1						1.5 **IRET**
C	2.2 **RRC** R1	2.3 **RRC** IR1	2.5 **LDC** r1,Irr2	2.9 **LDCI** Ir1,Irr2	2.3 **JP** IRR1	2.9 **LDC** Ir1,Irr2		3.4 **LD** r1,r2,X	3.2 **PUSHX** ER2							1.2 **RCF**
D	2.2 **SRA** R1	2.3 **SRA** IR1	2.5 **LDC** r2,Irr1	2.9 **LDCI** Ir2,Irr1	2.6 **CALL** IRR1	2.2 **BSWAP** R1	3.3 **CALL** DA	3.4 **LD** r2,r1,X	3.2 **POPX** ER1							1.2 **SCF**
E	2.2 **RR** R1	2.3 **RR** IR1	2.2 **BIT** p,b,r1	2.3 **LD** r1,Ir2	3.2 **LD** R2,R1	3.3 **LD** IR2,R1	3.2 **LD** R1,IM	3.3 **LD** IR1,IM	4.2 **LDX** ER2,ER1	4.2 **LDX** IM,ER1						1.2 **CCF**
F	2.2 **SWAP** R1	2.3 **SWAP** IR1	2.6 **TRAP** Vector	2.3 **LD** Ir1,r2	2.8 **MULT** RR1	3.3 **LD** R2,IR1	3.3 **BTJ** p,b,r1,X	3.4 **BTJ** p,b,Ir1,X								

Figure 29. First Op Code Map

Figure 31 displays the typical current consumption while operating at 25 ºC, 3.3 V, versus the system clock frequency in HALT Mode.

Figure 31. I_{CC} Versus System Clock Frequency (HALT Mode)

General Purpose I/O Port Input Data Sample Timing

Figure 33 displays timing of the GPIO port input sampling. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The port value is available to the eZ8 CPU on the second rising clock edge following the change of the port value.

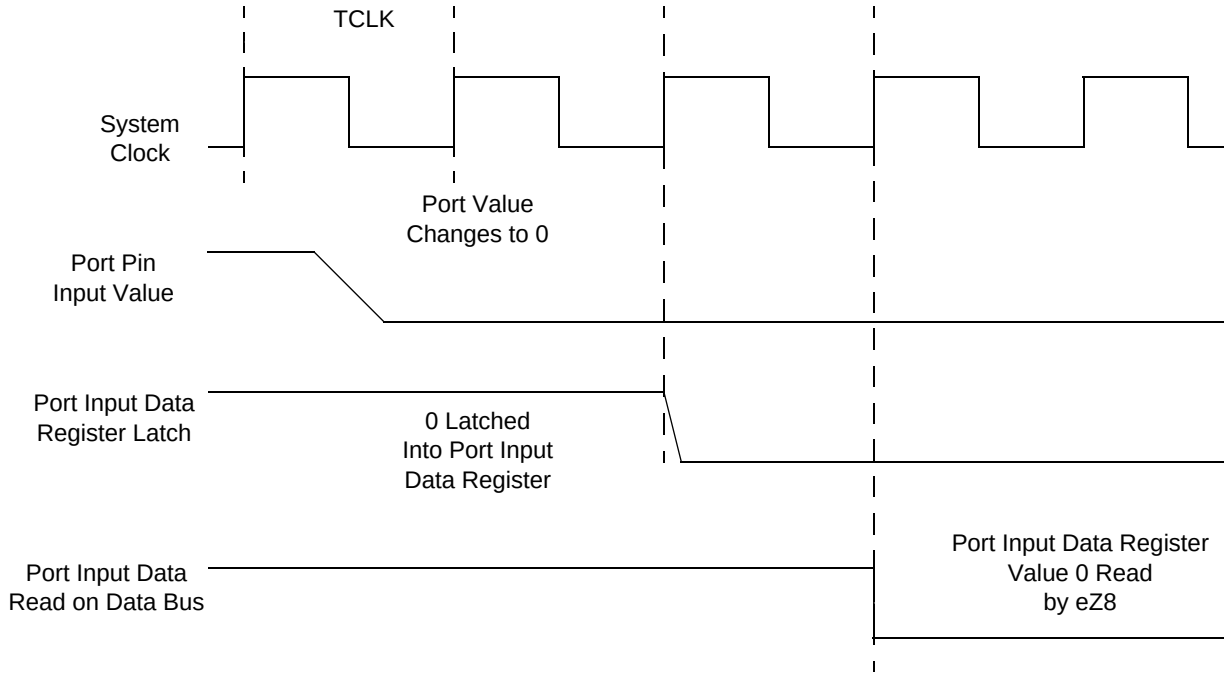

Figure 33. Port Input Sample Timing

Table 124. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_{S_PORT}	Port Input Transition to X_{IN} Rise Setup Time (not pictured)	5	–
T_{H_PORT}	X_{IN} Rise to Port Input Transition Hold Time (not pictured)	0	–
T_{SMR}	GPIO port pin pulse width to ensure Stop Mode Recovery (for GPIO port pins enabled as SMR sources)	1µs	

Hex Address: FDF

Table 183. Port D Output Data Register (PDOUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
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	FDFH							

Hex Addresses: FE0–FEF

This address range is reserved.

Watchdog Timer

For more information about the Watchdog Timer registers, see the Watchdog Timer Control Register Definitions section on page 95.

Hex Address: FF0

The Watchdog Timer Control Register address is shared with the read-only Reset Status Register.

Table 184. Watchdog Timer Control Register (WDTCTL)

Bit	7	6	5	4	3	2	1	0
Field	WDTUNLK							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
Address	FF0H							

Table 185. Reset Status Register (RSTSTAT)

Bit	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	Reserved			
RESET	See Table 12 on page 29			0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF0H							

P

R