# E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### 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	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0830qh020eg

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# **Part Selection Guide**

Table 1 lists the basic features available for each device within the Z8 Encore! F0830 Series product line. See the <u>Ordering Information</u> chapter on page 200 for details.

Part	Flash	RAM	NVDS	
Number	(KB)	(B)	(64B)	ADC
Z8F1232	12	256	No	Yes
Z8F1233	12	256	No	No
Z8F0830	8	256	Yes	Yes
Z8F0831	8	256	Yes	No
Z8F0430	4	256	Yes	Yes
Z8F0431	4	256	Yes	No
Z8F0230	2	256	Yes	Yes
Z8F0231	2	256	Yes	No
Z8F0130	1	256	Yes	Yes
Z8F0131	1	256	Yes	No

Table 1. Z8 Encore! F0830 Series Family Part Selection Guide

# **Data Memory**

The Z8 Encore! F0830 Series does not use the eZ8 CPU's 64KB data memory address space.

# **Flash Information Area**

Table 7 maps the Z8 Encore! F0830 Series Flash information area. The 128-byte information area is accessed, by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash information area is mapped into program memory and overlays these 128 bytes at addresses FE00H to FE7FH. When information area access is enabled, all reads from these program memory addresses return information area data rather than program memory data. Access to the Flash information area is read-only.

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog option bits
FE40–FE53	Part Number 20-character ASCII alphanumeric code Left-justified and filled with FH
FE54–FE5F	Reserved
FE60–FE7F	Reserved
FE80–FFFF	Reserved

#### Table 7. Z8 Encore! F0830 Series Flash Memory Information Area Map

# LED Drive Enable Register

The LED Drive Enable Register, shown in Table 31, activates the controlled current drive. The Alternate Function Register has no control over the LED function; therefore, setting the Alternate Function Register to select the LED function is not required. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Bit	7	6	5	4	3	2	1	0
Field				LEDE	N[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	F82H							

Table 31	I FD	Drive	Fnable	(I EDEN)
Table JT.		DIIVE		

Bit	Description
[7:0]	LED Drive Enable
LEDEN	These bits determine which Port C pins are connected to an internal current sink. 0 = Tristate the Port C pin.
	1= Connect controlled current sink to the Port C pin.

# LED Drive Level High Register

The LED Drive Level High Register, shown in Table 32, 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.

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLH[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				F8	3H			

### Bit Description

[7:0] LED Level High Bits

LEDLVLH {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA.

01= 7 mA.

10= 13mA.

11= 20mA.

### **Interrupt Request 1 Register**

The Interrupt Request 1 (IRQ1) Register, shown in Table 36, stores interrupt requests for both vectored and polled interrupts. When a request is sent 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.

Bit	7	6	5	4	3	2	1	0
Field	PA7I	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
Address	FC3H							
Bit	Descriptio	n						

#### Table 36. Interrupt Request 1 Register (IRQ1)

Bit	Description
[7] PA7I	<b>Port A7</b> 0 = No interrupt request is pending for GPIO Port A. 1 = An interrupt request from GPIO Port A.
[6] PA6CI	Port A6 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.
[5] PAxl	<ul> <li>Port A Pin x Interrupt Request</li> <li>0 = No interrupt request is pending for GPIO Port A pin x.</li> <li>1 = An interrupt request from GPIO Port A pin x is awaiting service.</li> </ul>
Note: x	indicates the specific GPIO port pin number (5–0).

### **Interrupt Request 2 Register**

The Interrupt Request 2 (IRQ2) Register, shown in Table 37, stores interrupt requests for both vectored and polled interrupts. When a request is sent 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.

Bit	7	6	5	4	3	2	1	0
Field		Rese	erved		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
Address				FC	6H			
Bit	Descriptio	n						
[7:4]	Reserved							
	These regis	sters are res	erved and n	nust be prog	rammed to	0000.		
[3]	Port C Pin	x Interrupt	Request					
PCxI	0 = No inter	rrupt reques	t is pending	for GPIO Po	ort C pin <i>x</i> .			
	1 = An inter	1 = An interrupt request from GPIO Port C pin x is awaiting service.						
Note: x inc	ndicates the specific GPIO port pin number (3–0).							

#### Table 37. Interrupt Request 2 Register (IRQ2)

### **IRQ0 Enable High and Low Bit Registers**

Table 38 lists the priority control values for IRQ0. The IRQ0 Enable High and Low Bit registers, shown in Tables 39 and 40, form a priority-encoded enabling service for interrupts in the Interrupt Request 0 Register. Priority is generated by setting the bits in each register.

IRQ0I	ENH[ <i>x</i> ]	IRQ0ENL[x]	Priority	Description
	0	0	Disabled	Disabled
	0	1	Level 1	Low
	1	0	Level 2	Nominal
	1	1	Level 3	High
Note:	x indicate			

Table 38. IRQ0 Enable and Priority Encoding

# **IRQ1 Enable High and Low Bit Registers**

Table 41 describes the priority control for IRQ1. The IRQ1 Enable High and Low Bit registers, shown in Tables 42 and 43, form a priority-encoded enabling service for interrupts in the Interrupt Request 1 Register. Priority is generated by setting the bits in each register.

IRQ1ENH[x]	IRQ1ENL[x]	Priority	Description		
0	0	Disabled	Disabled		
0	1	Level 1	Low		
1	0	Level 2	Nominal		
1	1	Level 3	High		
Note: x indicates register bits in the address range $7-0$ .					

Table 41. IRQ1 Enable and Priority Encoding

#### Table 42. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENH	PA6CENH	PA5ENH	PA4ENH	<b>PA3ENH</b>	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] PA7ENH	Port A Bit[7] Interrupt Request Enable High Bit
[6] PA6CENH	Port A Bit[7] or Comparator Interrupt Request Enable High Bit
[5:0] PA <i>x</i> ENH	<b>Port A Bit</b> [ <i>x</i> ] Interrupt Request Enable High Bit See the interrupt port select register for selection of either Port A or Port D as the interrupt
Note: x indic	cates register bits in the address range 5–0.

reload. For the timer output to make a state change at a ONE-SHOT time-out (rather than a single cycle pulse), first set the TPOL bit in the Timer Control Register to the start value before enabling ONE-SHOT Mode. After starting the timer, set TPOL to the opposite bit value.

Observe the following steps for configuring a timer for ONE-SHOT Mode and for initiating the count:

- 1. Write to the Timer Control Register to:
  - Disable the timer
  - Configure the timer for ONE-SHOT Mode
  - Set the prescale value
  - Set the initial output level (High or Low) if using the timer output Alternate function
- 2. Write to the Timer High and Low Byte registers to set the starting count value.
- 3. Write to the Timer Reload High and Low Byte registers to set the reload value.
- 4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 5. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
- 6. Write to the Timer Control Register to enable the timer and initiate counting.

In ONE-SHOT Mode, the system clock always provides the timer input. The timer period is calculated with the following equation:

One-Shot Mode Time-Out Period (s) =  $\frac{(\text{Reload Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$ 

#### **CONTINUOUS Mode**

In CONTINUOUS Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and the counting resumes. Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps for configuring a timer for CONTINUOUS Mode and for initiating the count:

1. Write to the Timer Control Register to:

- Disable the timer
- Configure the timer for CONTINUOUS Mode
- Set the prescale value
- If using the timer output Alternate function, set the initial output level (High or Low)
- 2. Write to the Timer High and Low Byte registers to set the starting count value (usually 0001H). This action only affects the first pass in CONTINUOUS Mode. After the first timer reload in CONTINUOUS Mode, counting always begins at the reset value of 0001H.
- 3. Write to the Timer Reload High and Low Byte registers to set the reload value.
- 4. Enable the timer interrupt (if appropriate) and set the timer interrupt priority by writing to the relevant interrupt registers.
- 5. Configure the associated GPIO port pin (if using the timer output function) for the timer output alternate function.
- 6. Write to the Timer Control Register to enable the timer and initiate counting.

In CONTINUOUS Mode, the system clock always provides the timer input. The timer period is calculated with the following equation:

Continuous Mode Time-Out Period (s) =  $\frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$ 

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT Mode equation to determine the first time-out period.

### **COUNTER Mode**

In COUNTER Mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO port pin: timer input alternate function. The TPOL bit in the Timer Control Register determines whether the count occurs on the rising edge or the falling edge of the timer input signal. In COUNTER Mode, the prescaler is disabled.

**Caution:** The input frequency of the timer input signal must not exceed one-fourth the system clock frequency.

Upon reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Additionally, if the timer output alternate function

tion and reload events. The user can configure the timer interrupt to be generated only at the input deassertion event or the reload event by setting the TICONFIG field of the TxCTL1 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. Assert the timer input signal to initiate the counting.

### **CAPTURE/COMPARE Mode**

In CAPTURE/COMPARE Mode, the timer begins counting on the first external timer input transition. The acceptable transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent acceptable transition (after the first) of the timer input signal, captures the current count value. The capture value is written to the timer PWM High and Low Byte registers. When the capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H and the counting resumes. The INPCAP bit in the TxCTL1 Register is set to indicate that the timer interrupt is caused by an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in the TxCTL1 Register is cleared to indicate that the timer interrupt has not been caused by an input capture event.

Observe the following steps for configuring a timer for CAPTURE/COMPARE Mode and for initiating the count:

- 1. Write to the Timer Control Register to:
  - Disable the timer
  - Configure the timer for CAPTURE/COMPARE 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 compare value.
- 4. Enable the timer interrupt 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. The user can configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.
- 5. Configure the associated GPIO port pin for the timer input alternate function.

## Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers, shown in Tables 54 and 55, control PWM operations. These registers also store the capture values for the CAPTURE and CAPTURE/COMPARE modes.

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
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	F04H, F0CH							

#### Table 54. Timer 0–1 PWM High Byte Register (TxPWMH)

#### Table 55. Timer 0–1 PWM Low Byte Register (TxPWML)

Bit	7	6	5	4	3	2	1	0
Field	PWML							
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	F05H, F0DH							

#### Bit Description

o the current
VM output
vhen operat-
-
,

Bit	Description (Continued)
[6] TPOL (cont'd)	<ul> <li>PWM DUAL OUTPUT Mode</li> <li>0 = Timer output is forced Low (0) and timer output complement is forced High (1), when the timer is disabled. When enabled and the PWM count matches, the timer output is forced High (1) and forced Low (0) when enabled and reloaded. When enabled and the PWM count matches, the timer output complement is forced Low (0) and forced High (1) when enabled and reloaded.</li> <li>1 = Timer output is forced High (1) and timer output complement is forced Low (0) when the timer is disabled. When enabled and the PWM count matches, the timer output is forced Low (0) when the timer is disabled. When enabled and the PWM count matches, the timer output is forced Low (0) and forced High (1) when enabled and reloaded. When enabled and the PWM count matches, the timer output is forced Low (0) and forced High (1) when enabled and reloaded. When enabled and the PWM count matches, the timer output complement is forced High (1) and forced Low (0) when enabled and reloaded. The PWMD field in the TxCTL0 register determines an optional added delay on the assertion (Low to High) transition of both timer output and timer output complement for deadband generation.</li> </ul>
	<ul> <li>CAPTURE RESTART Mode</li> <li>0 = Count is captured on the rising edge of the timer input signal.</li> <li>1 = Count is captured on the falling edge of the timer input signal.</li> <li>COMPARATOR COUNTER Mode</li> <li>When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled, the timer output signal is complemented on timer reload.</li> </ul>
	<b>Caution:</b> When the timer output alternate function TxOUT on a GPIO port pin is enabled, TxOUT will change to whatever state the TPOL bit is in. The timer does not need to be enabled for that to happen. Additionally, the port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL bit when the timer is enabled and running does not immediately change the polarity TxOUT.
[5:3] PRES	Prescale Value The timer input clock is divided by 2 <sup>PRES</sup> , where PRES can be set from 0 to 7. The prescaler is reset each time the timer is disabled. This reset ensures proper clock division each time the timer is restarted. 000 = Divide by 1. 001 = Divide by 2. 010 = Divide by 4. 011 = Divide by 8. 100 = Divide by 16. 101 = Divide by 32. 110 = Divide by 64. 111 = Divide by 128.





Figure 11. Analog-to-Digital Converter Block Diagram

# Operation

The ADC converts the analog input,  $ANA_X$ , to a 10-bit digital representation. The equation for calculating the digital value is represented by:

ADCOutput =  $1024 \times (ANA_x \div V_{REF})$ 

Assuming zero gain and offset errors, any voltage outside the ADC input limits of  $AV_{SS}$  and  $V_{REF}$  returns all 0s or 1s, respectively. A new conversion can be initiated by a software to the ADC Control Register's start bit.

Initiating a new conversion, stops any conversion currently in progress and begins a new conversion. To avoid disrupting a conversion already in progress, the START bit can be read to determine ADC operation status (busy or available).

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 <u>eZ8 CPU</u> <u>Core User Manual (UM0128)</u>, which is available for download on <u>www.zilog.com</u>, 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.

# **Power Failure Protection**

NVDS routines employ error-checking mechanisms to ensure that any power failure will only endanger the most recently written byte. Bytes previously written to the array are not perturbed. For this protection to function, the VBO must be enabled (see the <u>Low-Power</u> <u>Modes</u> chapter on page 30) and configured for a threshold voltage of 2.4V or greater (see *the* <u>Trim Bit Address Space</u> *section on page 129*).

A system reset (such as a pin reset or Watchdog Timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array are unperturbed.

# **Optimizing NVDS Memory Usage for Execution Speed**

As indicated in Table 93, the NVDS read time varies drastically; this discrepancy being a trade-off for minimizing the frequency of writes that require post-write page erases. The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N as well as the number of writes since the most recent page erase. Neglecting the effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb to consider is that every write since the most recent page erase causes read times of unwritten addresses to increase by  $0.8\mu s$  up to a maximum of  $258\mu s$ .

Operation	Minimum Latency (µs)	Maximum Latency (µs)
Read	71	258
Write	126	136
Illegal Read	6	6
Illegal Write	7	7

Table 93. NVDS Read Time	Table	93.	NVDS	Read	Time
--------------------------	-------	-----	------	------	------

• **Note:** For every 200 writes, a maintenance operation is necessary. In this rare occurrence, the write takes up to 58 ms to complete.

If NVDS read performance is critical to your software architecture, you can optimize your code for speed by using either of the two methods listed below.

1. Periodically refresh all addresses that are used; this is the more useful method. The optimal use of NVDS, in terms of speed, is to rotate the writes evenly among all addresses planned for use, thereby bringing all reads closer to the minimum read time.

This example indicates that the source data is added to the destination data; 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

Tables 105 through 112 contain 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 instructions 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*.

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
СР	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
СРХ	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment

Abbreviation	Description	Abbreviation	Description
b	Bit position	IRR	Indirect Register Pair
CC	Condition code	р	Polarity (0 or 1)
Х	8-bit signed index or displace- ment	r	4-bit Working Register
DA	Destination address	R	8-bit register
ER	Extended Addressing Register	r1, R1, Ir1, Irr1, IR1, rr1, RR1, IRR1, ER1	Destination address
IM	Immediate data value	r2, R2, Ir2, Irr2, IR2, rr2, RR2, IRR2, ER2	Source address
lr	Indirect Working Register	RA	Relative
IR	Indirect Register	rr	Working Register Pair
Irr	Indirect Working Register Pair	RR	Register Pair

### Table 114. Op Code Map Abbreviations

# **Electrical Characteristics**

The data in this chapter represents all known data prior to qualification and characterization of the F0830 Series of products, and is therefore subject to change. Additional electrical characteristics may be found in the individual chapters of this document.

# **Absolute Maximum Ratings**

Stresses greater than those listed in Table 115 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, tie unused inputs to one of the supply voltages ( $V_{DD}$  or  $V_{SS}$ ).

Parameter	Minimum	Maximum	Units	Notes
Ambient temperature under bias	0	+105	°C	
Storage temperature	-65	+150	°C	
Voltage on any pin with respect to V <sub>SS</sub>	-0.3	+5.5	V	
Voltage on $V_{DD}$ pin with respect to $V_{SS}$	-0.3	+3.6	V	
Maximum current on input and/or inactive output pin	-5	+5	μA	
Maximum output current from active output pin	-25	+25	mA	
20-pin Packages Maximum Ratings at 0°C to 70°C				
Total power dissipation		430	mW	
Maximum current into V <sub>DD</sub> or out of V <sub>SS</sub>		120	mA	
28-pin Packages Maximum Ratings at 0°C to 70°C				
Total power dissipation		450	mW	
Maximum current into V <sub>DD</sub> or out of V <sub>SS</sub>		125	mA	

#### **Table 115. Absolute Maximum Ratings**

### 195

### 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

		Dela	y (ns)
Parameter	Abbreviation	Minimum	Maximum
T <sub>S_PORT</sub>	Port Input Transition to X <sub>IN</sub> Rise Setup Time (not pictured)	5	-
T <sub>H_PORT</sub>	X <sub>IN</sub> Rise to Port Input Transition Hold Time (not pictured)	0	-
T <sub>SMR</sub>	GPIO port pin pulse width to ensure Stop Mode Recovery (for GPIO port pins enabled as SMR sources)	1µs	

### Hex Addresses: F87–F8F

This address range is reserved.

# Comparator 0

For more information about the Comparator Register, see the <u>Comparator Control Register Definitions</u> section on page 107.

### Hex Address: F90

Bit	7	6	5	4	3	2	1	0		
Field	Reserved	INNSEL		REFLVL				Reserved		
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		
Address	F90H									

#### Table 156. Comparator Control Register (CMP0)

### Hex Addresses: F91–FBF

This address range is reserved.

# **Interrupt Controller**

For more information about the Interrupt Control registers, see the <u>Interrupt Control Reg-</u> <u>ister Definitions</u> section on page 57.

### Hex Address: FC0

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	TOI	Reserved	Reserved	Reserved	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							

#### Table 157. Interrupt Request 0 Register (IRQ0)

### Hex Address: FD3

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	FD3H								

#### Table 172. Port A Output Data Register (PAOUT)

### Hex Address: FD4

#### Table 173. Port B GPIO Address Register (PBADDR)

Bit	7	6	5	4	3	2	1	0	
Field	PADDR[7:0]								
RESET	00H								
R/W	R/W	R/W R/W R/W R/W R/W R/W R/W							
Address	FD4H								

### Hex Address: FD5

### Table 174. Port B Control Registers (PBCTL)

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	FD5H								

### Hex Address: FD6

#### Table 175. Port B Input Data Registers (PBIN)

Bit	7	6	5	4	3	2	1	0	
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0	
RESET	Х	Х	Х	Х	Х	Х	Х	Х	
R/W	R	R	R	R	R	R	R	R	
Address	FD6H								