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	Ubsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	· .
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f022ash020sg

Email: info@E-XFL.COM

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

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Chapter/Section	Description	Page No.
Dec 2012	27	Port Alternate Function Map- ping (Non 8-Pin Parts), Port Alternate Function Mapping (8- Pin Parts)	Added missing Port D data to Table 15; cor- rected active Low status (set overlines) for PA0 (TOOUT), PA2 (RESET) and PA5 (T1OUT) in Table 16.	<u>40, 43</u>
Sep 2011	26	LED Drive Enable Register	Clarified statement surrounding the Alternate Function Register as it relates to the LED function; revised Flash Sector Protect Regis- ter description; revised Packaging chapter.	<u>53,</u> <u>157,</u> <u>245</u>
Sep 2008	25	Overview, Address Space, Register Map, General-Pur- pose Input/Output, Available Packages, Ordering Informa- tion	Added references to F042A Series back in Table 1, Table 5, Table 7 and Table 14.	<u>2, 8,</u> <u>16, 18,</u> <u>36,</u> <u>246</u>
May 2008	24	Overview, Address Space, Register Map, General-Pur- pose Input/Output, Available Packages, Ordering Informa- tion	Changed title to Z8 Encore! XP F082A Series and removed references to F042A Series in Table 1, Table 5, Table 7 and Table 14.	<u>2, 8,</u> <u>16, 18,</u> <u>36,</u> <u>246</u>
Dec 2007	23	Pin Description, General-Pur- pose Input/Output, Watchdog Timer	Updated Figure 3, Table 15, Tables 60 through 62.	<u>9, 40,</u> <u>97</u>
Jul 2007	22	Electrical Characteristics	Updated Tables 16 and 132; power con- sumption data.	<u>43,</u> 229
Jun 2007	21	n/a	Revision number update.	All

iii

15

Address Space

The eZ8 CPU can access the following three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral and general-purpose I/O port control registers.
- The Program Memory contains addresses for all memory locations having executable code and/or data.
- The Data Memory contains addresses for all memory locations that contain data only.

These three address spaces are covered briefly in the following subsections. For more information about eZ8 CPU and its address space, refer to the <u>eZ8 CPU Core User Manual</u> (<u>UM0128</u>), which is available for download on <u>www.zilog.com</u>.

Register File

The Register File address space in the Z8 Encore! MCU is 4 KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4 KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B control register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8 Encore! XP^{TM} F082A Series devices contain 256 B to 1KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

Program Memory

The eZ8 CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP F082A Series devices contain 1 KB to 8KB of on-chip Flash memory in the Program Memory address space, depending on the device. Reading from Program Memory

40

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A ^{1,2}	PA0	T0IN/T0OUT	Timer 0 Input/Timer 0 Output Complement	N/A
		Reserved		-
	PA1	TOOUT	Timer 0 Output	-
		Reserved		-
	PA2	DE0	UART 0 Driver Enable	-
		Reserved		-
	PA3	CTS0	UART 0 Clear to Send	-
		Reserved		-
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data	-
		Reserved		-
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data	-
		Reserved		-
	PA6	T1IN/T1OUT	Timer 1 Input/Timer 1 Output Complement	-
		Reserved		-
	PA7	T1OUT	Timer 1 Output	-
		Reserved		-

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

Notes:

- 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 <u>Port A–D Alternate Function Subregisters (PxAF)</u> 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 <u>Timer Pin Signal Operation</u> section on page 84 for details.
- 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 <u>A–D Alternate Function Subregisters (PxAF)</u> section on page 47 for details.
- 4. V_{REF} is available on PB5 in 28-pin products and on PC2 in 20-pin parts.
- 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 <u>A–D Alternate Function Subregisters (PxAF)</u> section on page 47 for details.
- 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 <u>Port A–D Alternate Function Subregisters (PxAF)</u> section on page 47 for details.

Port A–D Address Registers

The Port A–D Address registers select the GPIO port functionality accessible through the Port A–D Control registers. The Port A–D Address and Control registers combine to provide access to all GPIO port controls; see Tables 18 and 19.

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	FD0H, FD4H, FD8H, FDCH									

Table 18. Port A–D GPIO Address Registers (PxADDR)

Bit	Description
[7:0]	Port Address
PADDRx	The Port Address selects one of the subregisters accessible through the Port Control Register.
Note: x inc	dicates the specific GPIO port pin number (7–0).

Table 19. Port A–D GPIO Address Registers by Bit Description

PADDR[7:0]	Port Control Subregister accessible using the Port A–D Control Registers
00H	No function. Provides some protection against accidental port reconfiguration.
01H	Data Direction.
02H	Alternate Function.
03H	Output Control (Open-Drain).
04H	High Drive Enable.
05H	Stop Mode Recovery Source Enable.
06H	Pull-up Enable.
07H	Alternate Function Set 1.
08H	Alternate Function Set 2.
09H–FFH	No function.

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	LEDEN[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		Drive	Enable	
Table ST.	LEV	Dire	Ellable	(LEDEN)

Bit Description

[7:0] LED Drive Enable

LEDENx These bits determine which Port C pins are connected to an internal current sink.

0 = Tristate the Port C pin.

1 = Enable controlled current sink on the Port C pin.

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

LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin, as shown in Table 32. These two bits select between four programmable drive levels. 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	F83H									
Bit	Description									
[7:0] LEDLVLHx	LED Level High Bit {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA									

Tabla '	22 I EI	סעויים ר		Linh	Dogistor	٨
i abie .	32. LEI	JUIIVe	Levei	піціі	Redister	

01 = 7mA 10 = 13mA

11 = 20 mA

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

GPIO Mode Interrupt Controller

The interrupt controller on the Z8 Encore! XP F082A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 possible interrupt sources with 18 unique interrupt vectors:
 - Twelve GPIO port pin interrupt sources (two interrupt vectors are shared)
 - Eight on-chip peripheral interrupt sources (two interrupt vectors are shared)
- Flexible GPIO interrupts:
 - Eight selectable rising and falling edge GPIO interrupts
 - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer and LVD can be configured to generate an interrupt
- Supports vectored and polled interrupts

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information about interrupt servicing by the eZ8 CPU, refer to the <u>eZ8 CPU Core User Manual (UM0128)</u>, which is available for download on <u>www.zilog.com</u>.

Interrupt Vector Listing

Table 34 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

Note: Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

PWM SINGLE OUTPUT Mode

In PWM SINGLE OUTPUT Mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. 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 counting resumes.

If the TPOL bit in the Timer Control Register is set to 1, the Timer Output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the reload value and is reset to 0001H.

If the TPOL bit in the Timer Control Register is set to 0, the Timer Output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the reload value and is reset to 0001H.

Observe the following steps for configuring a timer for PWM SINGLE OUTPUT Mode and initiating the PWM operation:

- 1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for PWM SINGLE OUTPUT Mode
 - Set the prescale value
 - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function
- 2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
- 3. Write to the PWM High and Low Byte registers to set the PWM value.
- 4. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
- 5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 6. Configure the associated GPIO port pin for the Timer Output alternate function.
- 7. Write to the Timer Control Register to enable the timer and initiate counting.

into the Watchdog Timer Reload registers results in a one-second time-out at room temperature and 3.3V supply voltage. Time-outs other than one second may be obtained by scaling the calibration values up or down as required.

Note: The Watchdog Timer accuracy still degrades as temperature and supply voltage vary. See <u>Table 137</u> on page 235 for details.

Watchdog Timer Control Register Definitions

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

Watchdog Timer Control Register (WDTCTL): see page 96

Watchdog Timer Reload Upper Byte Register (WDTU): see page 97

Watchdog Timer Reload High Byte Register (WDTH): see page 97

Watchdog Timer Reload Low Byte Register (WDTL): see page 98

Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) Register is a write-only control register. Writing the 55H, AAH unlock sequence to the WDTCTL Register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL Register address produce no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the reload registers. This register address is shared with the read-only Reset Status Register.

Bit	7	6	5	4	3	2	1	0		
Field	WDTUNLK									
RESET	Х	Х	Х	Х	Х	Х	Х	Х		
R/W	W	W	W	W	W	W	W	W		
Address	s FF0H									
Note: X = Undefined.										

ы	Description
[7:0]	Watchdog Timer Unlock
WDTUNLK	The software must write the correct unlocking sequence to this register before it is allowed
	to modify the contents of the Watchdog Timer reload registers.

Description

Dit

Acceptable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Acceptable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	N/A	N/A	N/A	625.0	N/A	N/A	N/A
250.0	1	223.72	-10.51	250.0	N/A	N/A	N/A
115.2	2	111.9	-2.90	115.2	1	115.2	0.00
57.6	4	55.9	-2.90	57.6	2	57.6	0.00
38.4	6	37.3	-2.90	38.4	3	38.4	0.00
19.2	12	18.6	-2.90	19.2	6	19.2	0.00
9.60	23	9.73	1.32	9.60	12	9.60	0.00
4.80	47	4.76	-0.83	4.80	24	4.80	0.00
2.40	93	2.41	0.23	2.40	48	2.40	0.00
1.20	186	1.20	0.23	1.20	96	1.20	0.00
0.60	373	0.60	-0.04	0.60	192	0.60	0.00
0.30	746	0.30	-0.04	0.30	384	0.30	0.00

Table 72. UART Baud Rates (Continued)

150

Table 79. Flash Code Protection Using the Flash Option Bits

FWP	Flash Code Protection Description
0	Programming and erasing disabled for all of Flash Program Mem-
	ory. In user code programming, Page Erase and Mass Erase are all
	disabled. Mass Erase is available through the On-Chip Debugger.
1	Programming, Page Erase and Mass Erase are enabled for all of
	Flash Program Memory.

Flash Code Protection Using the Flash Controller

At Reset, the Flash Controller locks to prevent accidental program or erasure of the Flash memory. To program or erase the Flash memory, first write the Page Select Register with the target page. Unlock the Flash Controller by making two consecutive writes to the Flash Control Register with the values 73H and 8CH, sequentially. The Page Select Register must be rewritten with the target page. If the two Page Select writes do not match, the controller reverts to a locked state. If the two writes match, the selected page becomes active. See Figure 22 on page 148 for details.

After unlocking a specific page, you can enable either Page Program or Erase. Writing the value 95H causes a Page Erase only if the active page resides in a sector that is not protected. Any other value written to the Flash Control Register locks the Flash Controller. Mass Erase is not allowed in the user code but only in through the Debug Port.

After unlocking a specific page, you can also write to any byte on that page. After a byte is written, the page remains unlocked, allowing for subsequent writes to other bytes on the same page. Further writes to the Flash Control Register cause the active page to revert to a locked state.

Sector-Based Flash Protection

The final protection mechanism is implemented on a per-sector basis. The Flash memories of Z8 Encore! XP devices are divided into maximum number of 8 sectors. A sector is 1/8 of the total Flash memory size unless this value is smaller than the page size – in which case, the sector and page sizes are equal. On Z8 Encore! F082A Series devices, the sector size is varied according to the Flash memory configuration shown in <u>Table 78</u> on page 146.

The Flash Sector Protect Register can be configured to prevent sectors from being programmed or erased. After a sector is protected, it cannot be unprotected by user code. The Flash Sector Protect Register is cleared after reset, and any previously-written protection values are lost. User code must write this register in their initialization routine if they prefer to enable sector protection.

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Regis-

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.

Bit	7	6	5	4	3	2	1	0		
Field	INFO_EN		PAGE							
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	FF9H									

Table 82. Flash Page Select Register (FPS)

Bit Description

[7] Information Area Enable

INFO_EN 0 = Information Area us not selected.

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

[6:0] Page Select

PAGE This 7-bit field identifies the Flash memory page for Page Erase and page unlocking. Program Memory Address[15:9] = PAGE[6:0]. For the Z8F08xx devices, the upper 3 bits must be zero. For the Z8F04xx devices, the upper 4 bits must be zero. For Z8F02xx devices, the upper 5 bits must always be 0. For the Z8F01xx devices, the upper 6 bits must always be 0.

Trim Bit Address 0002H

Table 92. Trim Option Bits at 0002H (TIPO)

Bit	7	6	5	4	3	2	1	0			
Field	IPO_TRIM										
RESET	U										
R/W	R/W										
Address	Information Page Memory 0022H										
Note: U = Unchanged by Reset. R/W = Read/Write.											

Bit	Description
[7:0]	Internal Precision Oscillator Trim Byte
IPO_TRIM	Contains trimming bits for the Internal Precision Oscillator.

Trim Bit Address 0003H

Note: The LVD is available on 8-pin devices only.

Table 93. Trim Option Bits at Address 0003H (TLVD)

Bit	7	6	5	4	3	2	1	0		
Field	Reserved			LVD_TRIM						
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		
Address	Information Page Memory 0023H									
Note: U =	Note: U = Unchanged by Reset, R/W = Read/Write.									

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description
[7:5]	Reserved These bits are reserved and must be programmed to 111.
[4:0] LVD_TRIM	Low Voltage Detect Trimm This trimming affects the low voltage detection threshold. Each LSB represents a 50mV change in the threshold level. Alternatively, the low voltage threshold may be computed from the options bit value by the following equation:
	$LVD_LVL = 3.6 V - LVD_TRIM \times 0.05 V$
	These values are tabulated in Table 94.

ADC Calibration Data

Table 96. ADC Calibration Bits

Bit	7	6	5	4	3	2	1	0		
Field	ADC_CAL									
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		
Address	s Information Page Memory 0060H–007DH									
Noto: II -	Linchanged k		/ - Pood/Mrit	0						

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit Description

[7:0] Analog-to-Digital Converter Calibration Values
ADC_CAL Contains factory-calibrated values for ADC gain and offset compensation. Each of the ten supported modes has one byte of offset calibration and two bytes of gain calibration. These values are read by the software to compensate ADC measurements as described in the Software Compensation Procedure Using Factory Calibration Data section on page 129. The location of each calibration byte is provided in Table 97.

Info Page	Memory			
Address	Address	Compensation Usage	ADC Mode	Reference Type
60	FE60	Offset	Single-Ended Unbuffered	Internal 2.0 V
08	FE08	Gain High Byte	Single-Ended Unbuffered	Internal 2.0 V
09	FE09	Gain Low Byte	Single-Ended Unbuffered	Internal 2.0 V
63	FE63	Offset	Single-Ended Unbuffered	Internal 1.0 V
0A	FE0A	Gain High Byte	Single-Ended Unbuffered	Internal 1.0 V
0B	FE0B	Gain Low Byte	Single-Ended Unbuffered	Internal 1.0 V
66	FE66	Offset	Single-Ended Unbuffered	External 2.0 V
0C	FE0C	Gain High Byte	Single-Ended Unbuffered	External 2.0 V
0D	FE0D	Gain Low Byte	Single-Ended Unbuffered	External 2.0 V
69	FE69	Offset	Single-Ended 1x Buffered	Internal 2.0 V
0E	FE0E	Gain High Byte	Single-Ended 1x Buffered	Internal 2.0 V
0F	FE0F	Gain Low Byte	Single-Ended 1x Buffered	Internal 2.0 V
6C	FE6C	Offset	Single-Ended 1x Buffered	External 2.0 V
10	FE10	Gain High Byte	Single-Ended 1x Buffered	External 2.0 V
11	FE11	Gain Low Byte	Single-Ended 1x Buffered	External 2.0 V
6F	FE6F	Offset	Differential Unbuffered	Internal 2.0 V

Table 97. ADC Calibration Data Location

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 $251 \mu s$ (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 $2\mu s$ execution time.

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

Table 106. Write Status Byte

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

Bit	Description (Continued)
[3] WDFEN	Watchdog Timer Oscillator Failure Detection Enable1 = Failure detection of Watchdog Timer oscillator is enabled.0 = Failure detection of Watchdog Timer oscillator is disabled.
[2:0] SCKSEL	System Clock Oscillator Select 000 = Internal precision oscillator functions as system clock at 5.53MHz. 001 = Internal precision oscillator functions as system clock at 32kHz. 010 = Crystal oscillator or external RC oscillator functions as system clock. 011 = Watchdog Timer oscillator functions as system. 100 = External clock signal on PB3 functions as system clock. 101 = Reserved. 110 = Reserved. 111 = Reserved.

Oscillator Operation with an External RC Network

Figure 28 displays a recommended configuration for connection with an external resistorcapacitor (RC) network.



Figure 28. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of $45 \text{ k}\Omega$ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 k Ω . The typical oscillator frequency can be estimated from the values of the resistor (*R* in k Ω) and capacitor (*C* in pF) elements using the following equation:

Oscillator Frequency (kHz) = $\frac{1 \times 10^6}{(0.4 \times R \times C) + (4 \times C)}$

Figure 29 displays the typical $(3.3 \text{ V} \text{ and } 25^{\circ}\text{C})$ oscillator frequency as a function of the capacitor (C, in pF) employed in the RC network assuming a $45 \text{ K}\Omega$ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board must be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20pF are recommended.

Mnemonic	Operands	Instruction
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap

Table 126. Program Control Instructions

Table 127. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

217

Assombly		Add Mo	lress ode	Opcode(s)			Fla	ags			Fetch	Instr.
Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Ζ	S	۷	D	Н	S	S
LDX dst, src	dst ← src	r	ER	84	-	_	_	-	_	_	3	2
		lr	ER	85	-						3	3
		R	IRR	86	-						3	4
		IR	IRR	87	-						3	5
		r	X(rr)	88	-						3	4
		X(rr)	r	89	-						3	4
		ER	r	94	-						3	2
		ER	lr	95	-						3	3
		IRR	R	96	-						3	4
		IRR	IR	97	-						3	5
		ER	ER	E8	-						4	2
		ER	IM	E9	-						4	2
LEA dst, X(src)	$dst \gets src + X$	r	X(r)	98	_	_	_	-	_	_	3	3
		rr	X(rr)	99	-						3	5
MULT dst	dst[15:0] ← dst[15:8] * dst[7:0]	RR		F4	_	-	_	-	_	-	2	8
NOP	No operation			0F	_	_	-	-	_	_	1	2
OR dst, src	$dst \gets dst \ OR \ src$	r	r	42	-	*	*	0	-	-	2	3
		r	lr	43	-						2	4
		R	R	44	-						3	3
		R	IR	45	-						3	4
		R	IM	46	-						3	3
		IR	IM	47	-						3	4

Table 128. eZ8 CPU Instruction Summary (Continued)

Note: Flags Notation: * = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 =Set to 1.

Assembly		Address Mode		_ Opcode(s)			Fla	ags	Fetch Cvcle	Instr. Cvcle		
Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Ζ	S	۷	D	Н	S	S
ORX dst, src	$dst \gets dst \ OR \ src$	ER	ER	48	-	*	*	0	-	-	4	3
		ER	IM	49	-						4	3
POP dst	$dst \gets @SP$	R		50	_	_	_	_	_	-	2	2
	$SP \leftarrow SP + 1$	IR		51							2	3
POPX dst	dst $\leftarrow @SP$ SP \leftarrow SP + 1	ER		D8	-	-	-	-	-	-	3	2
PUSH src	$SP \leftarrow SP - 1$ @ $SP \leftarrow src$	R		70	-	_	_	_	_	_	2	2
		IR		71	-						2	3
		IM		IF70	-						3	2
PUSHX src	$SP \leftarrow SP - 1$ @SP ← src	ER		C8	-	_	_	-	-	_	3	2
RCF	C ← 0			CF	0	-	-	-	-	-	1	2
RET	$\begin{array}{l} PC \leftarrow @SP \\ SP \leftarrow SP + 2 \end{array}$			AF	-	_	_	-	-	_	1	4
RL dst	C D7D6D5D4D3D2D1D0	R		90	*	*	*	*	-	-	2	2
		IR		91							2	3
RLC dst	C T D7D6D5D4D3D2D1D0 dst	R		10	*	*	*	*	_	_	2	2
		IR		11	-						2	3
RR dst]	R		E0	*	*	*	*	_	_	2	2
	► D7 D6 D5 D4 D3 D2 D1 D0 C	IR		E1	-						2	3

Table 128. eZ8 CPU Instruction Summary (Continued)

Note: Flags Notation:

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

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at <u>http://support.zilog.com</u>.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at <u>http://</u><u>zilog.com/kb</u> or consider participating in the Zilog Forum at <u>http://zilog.com/forum</u>.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at <u>http://www.zilog.com</u>.