



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 - Microcontrollers</u>"

Details	
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f012ahh020sg

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2012 Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore! and Z8 Encore! XP are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

Z8 Encore! XP[®] F082A Series Product Specification

xii

Figure 25.	Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2
Figure 26.	OCD Data Format
Figure 27.	Recommended 20MHz Crystal Oscillator Configuration 199
Figure 28.	Connecting the On-Chip Oscillator to an External RC Network 201
Figure 29.	Typical RC Oscillator Frequency as a Function of the External Capacitance with a $45 \mathrm{k}\Omega$ Resistor
Figure 30.	Opcode Map Cell Description
Figure 31.	First Opcode Map
Figure 32.	Second Opcode Map after 1FH
Figure 33.	Typical Active Mode IDD Versus System Clock Frequency
Figure 34.	Port Input Sample Timing
Figure 35.	GPIO Port Output Timing
Figure 36.	On-Chip Debugger Timing
Figure 37.	UART Timing With CTS
Figure 38.	UART Timing Without CTS

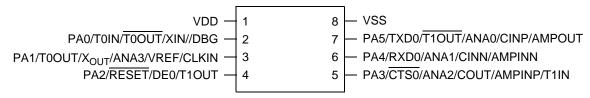


Figure 2. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 8-Pin SOIC, QFN/MLF-S, or PDIP Package

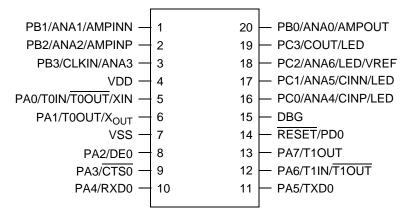


Figure 3. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 20-Pin SOIC, SSOP or PDIP Package

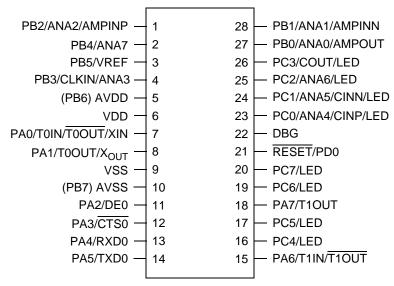


Figure 4. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 28-Pin SOIC, SSOP or PDIP Package

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 (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! XPTM 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 8 KB of on-chip Flash memory in the Program Memory address space, depending on the device. Reading from Program Memory

Table 16. Port Alternate Function Mapping (8-Pin Parts)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Select Register AFS1	Alternate Function Select Register AFS2
Port A	PA0	TOIN	Timer 0 Input	AFS1[0]: 0	AFS2[0]: 0
		Reserved		AFS1[0]: 0	AFS2[0]: 1
		Reserved		AFS1[0]: 1	AFS2[0]: 0
		T0OUT	Timer 0 Output Complement	AFS1[0]: 1	AFS2[0]: 1
	PA1	T0OUT	Timer 0 Output	AFS1[1]: 0	AFS2[1]: 0
		Reserved		AFS1[1]: 0	AFS2[1]: 1
		CLKIN	External Clock Input	AFS1[1]: 1	AFS2[1]: 0
		Analog Functions ¹	ADC Analog Input/V _{REF}	AFS1[1]: 1	AFS2[1]: 1
	PA2	DE0	UART 0 Driver Enable	AFS1[2]: 0	AFS2[2]: 0
		RESET	External Reset	AFS1[2]: 0	AFS2[2]: 1
		T1OUT	Timer 1 Output	AFS1[2]: 1	AFS2[2]: 0
		Reserved		AFS1[2]: 1	AFS2[2]: 1
	PA3	CTS0	UART 0 Clear to Send	AFS1[3]: 0	AFS2[3]: 0
		COUT	Comparator Output	AFS1[3]: 0	AFS2[3]: 1
		T1IN	Timer 1 Input	AFS1[3]: 1	AFS2[3]: 0
		Analog Functions ²	ADC Analog Input/LPO Input (P)	AFS1[3]: 1	AFS2[3]: 1
	PA4	RXD0	UART 0 Receive Data	AFS1[4]: 0	AFS2[4]: 0
		Reserved		AFS1[4]: 0	AFS2[4]: 1
		Reserved		AFS1[4]: 1	AFS2[4]: 0
		Analog Functions ²	ADC/Comparator Input (N)/LPO Input (N)	AFS1[4]: 1	AFS2[4]: 1
	PA5	TXD0	UART 0 Transmit Data	AFS1[5]: 0	AFS2[5]: 0
		T10UT	Timer 1 Output Complement	AFS1[5]: 0	AFS2[5]: 1
		Reserved		AFS1[5]: 1	AFS2[5]: 0
		Analog Functions ²	ADC/Comparator Input (P) LPO Output	AFS1[5]: 1	AFS2[5]: 1

Notes:

- 1. Analog functions include ADC inputs, ADC reference, comparator inputs and LPO ports.
- 2. The alternate function selection must be enabled; see the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.

Writing a 1 to the IRQE bit in the Interrupt Control Register

Interrupts are globally disabled by any of the following actions:

- Execution of a Disable Interrupt (DI) instruction
- eZ8 CPU acknowledgement of an interrupt service request from the interrupt controller
- Writing a 0 to the IRQE bit in the Interrupt Control Register
- Reset
- Execution of a Trap instruction
- Illegal Instruction Trap
- Primary Oscillator Fail Trap
- Watchdog Oscillator Fail Trap

Interrupt Vectors and Priority

The interrupt controller supports three levels of interrupt priority. Level 3 is the highest priority, Level 2 is the second highest priority and Level 1 is the lowest priority. If all of the interrupts are enabled with identical interrupt priority (all as Level 2 interrupts, for example), the interrupt priority is assigned from highest to lowest as specified in Table 34 on page 56. Level 3 interrupts are always assigned higher priority than Level 2 interrupts which, in turn, always are assigned higher priority than Level 1 interrupts. Within each interrupt priority level (Level 1, Level 2, or Level 3), priority is assigned as specified in Table 34, above. Reset, Watchdog Timer interrupt (if enabled), Primary Oscillator Fail Trap, Watchdog Oscillator Fail Trap and Illegal Instruction Trap always have highest (level 3) priority.

Interrupt Assertion

Interrupt sources assert their interrupt requests for only a single system clock period (single pulse). When the interrupt request is acknowledged by the eZ8 CPU, the corresponding bit in the Interrupt Request Register is cleared until the next interrupt occurs. Writing a 0 to the corresponding bit in the Interrupt Request Register likewise clears the interrupt request.

Caution: Zilog recommends not using a coding style that clears bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 1, which follows.

Interrupt Control Register

The Interrupt Control (IRQCTL) Register, shown in Table 49, contains the master enable bit for all interrupts.

Table 49. Interrupt Control Register (IRQCTL)

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

Bit	Description
[7] IRQE	Interrupt Request Enable This bit is set to 1 by executing an EI (Enable Interrupts) or IRET (Interrupt Return) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, Reset or by a direct register write of a 0 to this bit. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	Reserved These bits are reserved and must be programmed to 0000000.

- Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate
- 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 Compare value.
- 4. Enable the timer interrupt, if appropriate 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 COMPARE Mode, the system clock always provides the timer input. The Compare time can be calculated by the following equation:

$$\begin{array}{ll} \text{COMPARE Mode Time (s) } = & \frac{\text{(Compare Value - Start Value)} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}} \\ \end{array}$$

GATED Mode

In GATED Mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control Register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

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. When 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 (assuming the Timer Input signal remains asserted). Also, 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 reset.

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

- 1. Write to the Timer Control Register to:
 - Disable the timer

- 6. Read data from the UART Receive Data Register. If operating in MULTIPROCES-SOR (9-bit) Mode, further actions may be required depending on the MULTIPRO-CESSOR Mode bits MPMD[1:0].
- 7. Return to Step 4 to receive additional data.

Receiving Data using the Interrupt-Driven Method

The UART Receiver interrupt indicates the availability of new data (and error conditions). Observe the following steps to configure the UART receiver for interrupt-driven operation:

- 1. Write to the UART Baud Rate High and Low Byte registers to set the acceptable baud
- 2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
- 3. Execute a DI instruction to disable interrupts.
- 4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the acceptable priority.
- 5. Clear the UART Receiver interrupt in the applicable Interrupt Request Register.
- 6. Write to the UART Control 1 Register to enable Multiprocessor (9-bit) mode functions, if appropriate.
 - Set the Multiprocessor Mode Select (MPEN) to Enable MULTIPROCESSOR
 - Set the Multiprocessor Mode Bits, MPMD[1:0], to select the acceptable address matching scheme.
 - Configure the UART to interrupt on received data and errors or errors only (interrupt on errors only is unlikely to be useful for Z8 Encore! devices without a DMA block)
- 7. Write the device address to the Address Compare Register (automatic MULTIPRO-CESSOR Modes only).
- 8. Write to the UART Control 0 Register to:
 - Set the receive enable bit (REN) to enable the UART for data reception
 - Enable parity, if appropriate and if multiprocessor mode is not enabled and select either even or odd parity
- 9. Execute an EI instruction to enable interrupts.

Table 69. UART Address Compare Register (U0ADDR)

Bit	7	6	5	4	3	2	1	0
Field				COMP	_ADDR			
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		F45H						

Bit	Description
[7:0]	Compare Address
COMP_ADDR	This 8-bit value is compared to incoming address bytes.

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High (UxBRH) and Low Byte (UxBRL) registers, shown in Tables 70 and 71, combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART.

Table 70. UART Baud Rate High Byte Register (U0BRH)

Bit	7	6	5	4	3	2	1	0
Field				BF	RH			
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		F46H						

Bit	Description
[7:0] BRH	UART Baud Rate High Byte

Table 71. UART Baud Rate Low Byte Register (U0BRL)

Bit	7	6	5	4	3	2	1	0
Field				ВГ	₹L			
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		F47H						

Bit	Description
[7:0] BRL	UART Baud Rate Low Byte

Bit	Description (Continued)
[2:1]	Reserved
	These bits are reserved and must be undefined.
[0]	Overflow Status
OVF	0 = A hardware overflow did not occur in the ADC for the current sample.
	1= A hardware overflow did occur in the ADC for the current sample, therefore the current sample is invalid.

Figure 22 displays a basic Flash Controller flow. The following subsections provide details about the various operations displayed in Figure 22.

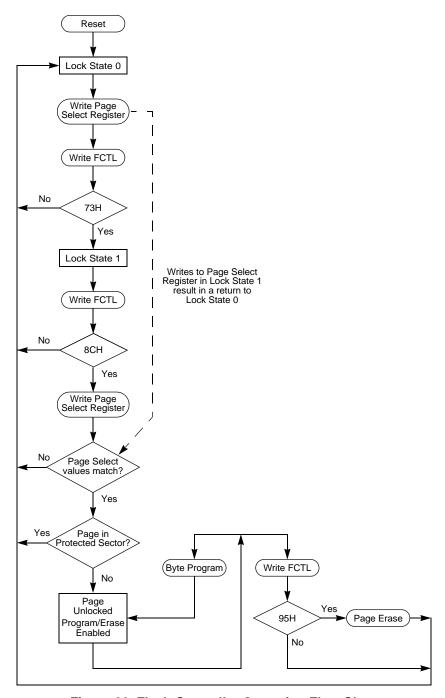


Figure 22. Flash Controller Operation Flow Chart

ter with 5EH. After the Flash Sector Protect Register is selected, it can be accessed at the Page Select Register address. When user code writes the Flash Sector Protect Register, 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 is no longer written or erased by the CPU. External Flash programming through the OCD or via the Flash Controller Bypass mode are unaffected. After a bit of the Sector Protect Register has been set, it cannot be cleared except by powering down the device.

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 completed, all Program Memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and Page Erase is successfully completed, 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 Core User Manual (UM0128)</u>, available for download on <u>www.zilog.com</u>, for a 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.

Table 109. Debug Command Enable/Disable (Continued)

Debug Command	Command Byte	Enabled when Not in DEBUG Mode?	Disabled by Flash Read Protect Option Bit
Write Program Counter	06H	_	Disabled.
Read Program Counter	07H	_	Disabled.
Write Register	08H	-	Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control Register.
Read Register	09H	_	Disabled.
Write Program Memory	0AH	_	Disabled.
Read Program Memory	0BH	_	Disabled.
Write Data Memory	0CH	_	Yes.
Read Data Memory	0DH	_	_
Read Program Memory CRC	0EH	_	-
Reserved	0FH	_	_
Step Instruction	10H	_	Disabled.
Stuff Instruction	11H	_	Disabled.
Execute Instruction	12H	_	Disabled.
Reserved	13H–FFH	_	_

In the list of OCD commands that follows, data and commands sent from the host to the On-Chip Debugger are identified by DBG \leftarrow Command/Data. Data sent from the On-Chip Debugger back to the host is identified by DBG \rightarrow Data.

Read OCD Revision (00H). The Read OCD Revision command determines the version of the On-Chip Debugger. If OCD commands are added, removed, or changed, this revision number changes.

```
DBG \leftarrow 00H

DBG \rightarrow OCDRev[15:8] (Major revision number)

DBG \rightarrow OCDRev[7:0] (Minor revision number)
```

Read OCD Status Register (02H). The Read OCD Status Register command reads the OCDSTAT Register.

```
DBG \leftarrow 02H
DBG \rightarrow OCDSTAT[7:0]
```

Read Runtime Counter (03H). The Runtime Counter counts system clock cycles in between Breakpoints. The 16-bit Runtime Counter counts up from 0000H and stops at the maximum count of FFFFH. The Runtime Counter is overwritten during the Write Memory,

Bit	Description (Continued)
[5] DBGACK	Debug Acknowledge This bit enables the debug acknowledge feature. If this bit is set to 1, the OCD sends a Debug Acknowledge character (FFH) to the host when a Breakpoint occurs. 0 = Debug Acknowledge is disabled. 1 = Debug Acknowledge is enabled.
[4:1]	Reserved These bits are reserved and must be programmed to 0000.
[0] RST	Reset Setting this bit to 1 resets the Z8F04xA family device. The device goes through a normal Power-On Reset sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 at the end of reset. 0 = No effect. 1 = Reset the Flash Read Protect Option Bit device.

OCD Status Register

The OCD Status Register reports status information about the current state of the debugger and the system.

Table 111. OCD Status Register (OCDSTAT)

Bit	7	6	5	4	3	2	1	0
Field	DBG	HALT	FRPENB			Reserved		
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Description
[7] DBG	Debug Status 0 = NORMAL Mode. 1 = DEBUG Mode.
[6] HALT	HALT Mode 0 = Not in HALT Mode. 1 = In HALT Mode.
[5] FRPENB	Flash Read Protect Option Bit Enable 0 = FRP bit enabled, that allows disabling of many OCD commands. 1 = FRP bit has no effect.
[4:0]	Reserved These bits are reserved and must be programmed to 00000.

Table 126. Program Control Instructions

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 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

Table 128. eZ8 CPU Instruction Summary (Continued)

Assembly			lress ode	_ Opcode(s)			Fla	ags			Fetch Cycle	Instr. Cycle
Mnemonic	Symbolic Operation	dst	src	(Hex)		Z	S	٧	D	Н	S	S
JR dst	$PC \leftarrow PC + X$	DA		8B	-	_	_	-	-	-	2	2
JR cc, dst	if cc is true PC ← PC + X	DA		0B-FB	-	_	-	-	_	-	2	2
LD dst, rc	dst ← src	r	IM	0C-FC	-	-	-	-	-	-	2	2
		r	X(r)	C7	-						3	3
		X(r)	r	D7	-						3	4
		r	lr	E3	-						2	3
		R	R	E4	-						3	2
		R	IR	E5	-						3	4
		R	IM	E6	-						3	2
		IR	IM	E7	-						3	3
		lr	r	F3	-						2	3
		IR	R	F5	-						3	3
LDC dst, src	dst ← src	r	Irr	C2	-	_	_	_	_	_	2	5
		lr	Irr	C5	-						2	9
		Irr	r	D2	-						2	5
LDCI dst, src	dst ← src	lr	Irr	C3	-	-	_	-	_	-	2	9
	$r \leftarrow r + 1$ $rr \leftarrow rr + 1$	Irr	lr	D3	-						2	9
LDE dst, src	dst ← src	r	Irr	82	-	-	_	-	-	-	2	5
		Irr	r	92	-						2	5
LDEI dst, src	dst ← src	lr	Irr	83	-	_	_	-	-	-	2	9
	$r \leftarrow r + 1$ $rr \leftarrow rr + 1$	Irr	lr	93	-						2	9
LDWX dst, src	dst ← src	ER	ER	1FE8	_	_	_	_	_	_	5	4

Note: Flags Notation:

^{* =} Value is a function of the result of the operation.

⁻ = Unaffected.

X = Undefined.

^{0 =} Reset to 0.

^{1 =} Set to 1.

General Purpose I/O Port Output Timing

Figure 35 and Table 144 provide timing information for GPIO port pins.

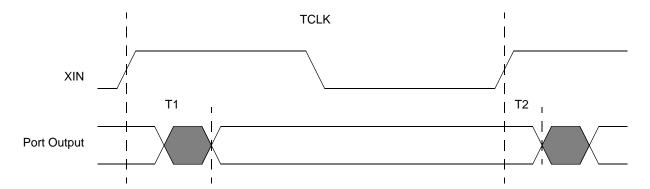


Figure 35. GPIO Port Output Timing

Table 144. GPIO Port Output Timing

		Dela	y (ns)
Parameter	Abbreviation	Minimum	Maximum
GPIO port pi	ns		
T ₁	X _{IN} Rise to Port Output Valid Delay	_	15
T ₂	X _{IN} Rise to Port Output Hold Time	2	_

LEA 210	b 206
load 210	cc 206
load constant 209	DA 206
load constant to/from program memory 210	ER 206
load constant with auto-increment addresses 210	IM 206
load effective address 210	IR 206
load external data 210	Ir 206
load external data to/from data memory and auto-	IRR 206
increment addresses 209	Irr 206
load external to/from data memory and auto-incre-	p 206
ment addresses 210	R 206
load using extended addressing 210	r 206
logical AND 210	RA 206
logical AND/extended addressing 210	RR 206
logical exclusive OR 210	rr 206
logical exclusive OR/extended addressing 210	vector 207
logical instructions 210	X 207
logical OR 210	notational shorthand 206
	notational shorthand 200
logical OR/extended addressing 210 low power modes 32	
low power modes 32	0
	_
M	OCD
•••	architecture 180
master interrupt enable 57	auto-baud detector/generator 183
memory	baud rate limits 184
data 17	block diagram 180
program 15	breakpoints 185
mode	commands 186
CAPTURE 87, 88	control register 191
CAPTURE/COMPARE 88	data format 183
CONTINUOUS 87	DBG pin to RS-232 Interface 181
COUNTER 87	debug mode 182
GATED 88	debugger break 211
ONE-SHOT 87	interface 181
PWM 87, 88	serial errors 184
modes 87	status register 192
MULT 208	timing 242
multiply 208	OCD commands
multiprocessor mode, UART 105	execute instruction (12H) 190
	read data memory (0DH) 190
NI.	read OCD control register (05H) 188
N	read OCD revision (00H) 187
NOP (no operation) 209	read OCD status register (02H) 187
notation	read program counter (07H) 188

UARTx control I (UxCTL1) 112 UARTx receive data (UxRXD) 116 UARTx status 0 (UxSTAT0) 114 SP 207 UARTx status 1 (UxSTAT1) 115 UARTx transmit data (UxTXD) 116 Watchdog Timer control (WDTCTL) 30, 96, 141, 196 Watchdog Timer reload high byte (WDTH) 97 Watchdog Timer reload low byte (WDTU) 97 Watchdog Timer reload upper byte (WDTU) 97 Watchdog Timer reload 22 stop mode 210 Stop Mode Recovery 22 sources 27 using a GPIO port pin transition 28 using Watchdog Timer time-out 28 stop mode recovery sources 29 using a GPIO port pin transition 29 SUB 208 subtract with carry 208 subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 SUBX 208 SWAP 211 swap nibbles 211 symbols, additional 207 TCM 209 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209
UARTx status 1 (UxSTAT1) 115 UARTx transmit data (UXTXD) 116 Watchdog Timer control (WDTCTL) 30, 96, 141, 196 Watchdog Timer reload high byte (WDTH) 97 Watchdog Timer reload low byte (WDTL) 98 Watchdog Timer reload upper byte (WDTU) 98 Watchdog Timer reload upper byte (WDTU) 97 Watchdog Timer reload upper byte (WDTU) 98 Watchdog Timer reload upper byte (WDTU) 97 register file 15 register pair 206 register pointer 207 resest and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RLC 211 rotate and shift instuctions 211 rotate left 211 rotate left 211 rotate left through carry 211 rotate right through carry 211 RRP 207 RRR 206, 211 TT 206 RRC 211 SBC 208 SBC 208 SRA 211 src 207 SRL 211 SRP 207 SRP 210 SATC 211 Stop Mode Recovery sources 27 using a GPIO port pin transition 28 stop mode recovery sources 29 using a GPIO port pin transition 29 subtract 208 subtract 208 subtract 208 subtract with carry 208 subtract with carry 208 subtract with carry 208 SWAP 211 swap nibbles 211 symbols, additional 207 TT TCMX 209 test complement under mask 209 test under mask 209 test under mask 209
UARTx transmit data (UxTXD) 116 Watchdog Timer control (WDTCTL) 30, 96, 141, 196 Watchdog Timer reload high byte (WDTH) 97 Watchdog Timer reload low byte (WDTU) 97 Watchdog Timer reload upper byte (WDTU) 97 register file 15 register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RLC 211 RLC 211 RLC 211 RLC 211 rotate left through carry 211 rotate right through carry 211 rr 206 RRC 211 SBC 208 SBC 208 SRC 211 SRP 210 STOP mode 32 stop mode 210 Stop Mode Recovery sources 27 using a GPIO port pin transition 28 using a GPIO port pin transition 29 SUB 208 subtract 208 subtract 208 subtract 208 subtract with carry 208 subtract with carry - extended addressing 208 SWAP 211 swap nibbles 211 symbols, additional 207 T T T TCM 209 TCMX 209 test complement under mask 209 test under mask 209 TCMS 209 test under mask 209
Watchdog Timer control (WDTCTL) 30, 96, 141, 196 Watchdog Timer reload high byte (WDTH) 97 Watchdog Timer reload upper byte (WDTU) 97 register file 15 register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL 21
141, 196 Watchdog Timer reload high byte (WDTH) 97 Watchdog Timer reload low byte (WDTL) 98 Watchdog Timer reload upper byte (WDTU) 97 register file 15 register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 REF 211 return 211 RL 211 RLC 211 rotate left 211 rotate left through carry 211 rotate left through carry 211 rotate right 11 RP 207 RR 206, 211 RF 206 RRC 211 SBC 208 SBC 208 SRP 210 stack pointer 207 stack pointer 207 strop mode 32 stop mode 32 stop mode 210 Stop Mode Recovery sources 27 using a GPIO port pin transition 28 using a GPIO port pin transition 29 SUB 208 subtract 208 subtract 208 subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 rotate and shift instuctions 211 swap nibbles 211 symbols, additional 207 TCM 209 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209
Watchdog Timer reload high byte (WDTH) 97 Watchdog Timer reload low byte (WDTL) 98 Watchdog Timer reload low byte (WDTU) 97 register file 15 register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL C 211 rotate left 211 rotate left through carry 211 rotate right 211 r
Watchdog Timer reload low byte (WDTL) 98 Watchdog Timer reload upper byte (WDTU) 97 1
Watchdog Timer reload upper byte (WDTU) 97 register file 15 register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL 211 RL C 211 RL C 211 rotate end shift instuctions 211 rotate right 121 rotate right 121 rotate right through carry 211 RP 207 RR 206, 211 rT 206 RRC 211 STOP mode 32 stop mode 210 Stop Mode Recovery sources 27 using a GPIO port pin transition 28 using a GPIO port pin transition 29 SUB 208 subtract 208 subtract 208 subtract vith carry 208 subtract with carry 208 subtract with carry 208 SUBX 208 SWAP 211 swap nibbles 211 symbols, additional 207 TCM 209 TCMX 209 test complement under mask 209 test complement under mask 209 test complement under mask - extended addressing 209 sets tunder mask 209 test under mask 209
stop mode 210 Stop Mode Recovery sources 27 using a GPIO port pin transition 28 using Watchdog Timer time-out 28 stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 subtract 208 RET 211 subtract 208 REL 211 RL 211 RL 211 SUB 208 SU
register file 15 register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL 208 SUBX 208 SUBX 208 SUBX 208 SUBX 208 TOtate left through carry 211 rotate left through carry 211 rotate right 211 rotate right through carry 211 rotate right through carry 211 RP 207 RR 206, 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 RS C 208 SBC 208 L Stop Mode Recovery sources 27 using a GPIO port pin transition 28 stop mode recovery sources 29 using a GPIO port pin transition 29 SUB 208 Subtract 208 Subtract with carry 208 Subtract with carry 208 SUBX 208 SUBX 208 TCM 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209 test under mask 209
register pair 206 register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL 211 RL C 211 RL C 211 rotate left through carry 211 rotate right 211 rotate right 211 rotate right through carry 211 rotate right through carry 211 rr 206 RRC 211 RRC 211 RRC 211 RRC 211 RRC 211 RRC 206 RRC 211 RRC 208 Sources 29 using a GPIO port pin transition 29 Sources 29 using a GPIO port pin transition 29 Sources 29 using a GPIO port pin transition 29 Sources 29 using a GPIO port pin transition 29 Sources 29 using a GPIO port pin transition 29 SUB 208 subtract 208 subtract 208 subtract 208 subtract 208 subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 SUBX 208 SWAP 211 swap nibbles 211 symbols, additional 207 T T T TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209 test under mask 209
register pointer 207 reset and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 subtract 208 return 211 RL 211 RLC 211 RLC 211 subtract with carry 208 rotate and shift instuctions 211 rotate left through carry 211 rotate right 211 rotate right 211 rotate right through carry 211 RRP 207 RR RR 206, 211 rr 206 RRC 211 SBC 208 using a GPIO port pin transition 28 stop mode recovery sources 29 using a GPIO port pin transition 29 SUB 208 subtract 208 subtract 208 subtract 208 subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 SWAP 211 swap nibbles 211 symbols, additional 207 T T T T TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209 test under mask 209
reset using Watchdog Timer time-out 28 stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL 211 RL 211 RL 211 Subtract 208 subtract vith carry 208 soutcate left 211 rotate left through carry 211 rotate right through carry 211 RP 207 RR 206, 211 RR 206 RRC 211 SBC 208 stop mode recovery sources 29 using a GPIO port pin transition 29 SUB 208 subtract 208 subtract 208 subtract vith carry 208 subtract with carry 208 SWAP 211 swap nibbles 211 symbols, additional 207 T T T TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209 test under mask 209 test under mask 209 test under mask 209
and stop mode characteristics 23 and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL
and Stop Mode Recovery 22 carry flag 209 sources 24 RET 211 return 211 RL 211 RLC 211 RLC 211 rotate left through carry 211 rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 RC 211 RC 211 RC 206 RRC 211 RC 207 RR 206 RRC 211 RC 208 SUBX
using a GPIO port pin transition 29 sources 24 RET 211 subtract 208 RET 211 subtract 208 RL 211 RL 211 subtract - extended addressing 208 REC 211 subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 SUBX 208 SUBX 208 SUBX 208 SUBX 208 Totate left 211 swap nibbles 211 symbols, additional 207 Table 207 RR 206, 211 Tr 206 RRC 211 TCMX 209 test complement under mask 209 test under mask 209 Test under mask 209 Test under mask 209 test under mask 209
sources 24 RET 211 return 211 RL 211 RL 211 RL 211 Subtract 208 subtract 208 subtract - extended addressing 208 subtract with carry 208 subtract with carry - extended addressing 208 rotate and shift instuctions 211 rotate left 211 SWAP 211 swap nibbles 211 rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 rr 206 RRC 211 T TCM 209 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 208 SBC 208 SUBX 208 SUBX 208 SUBX 208 SUBX 208 TOMA 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209 test under mask 209
subtract 208 return 211 return 21
subtract - extended addressing 208 RL 211 RLC 211 subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 rotate and shift instuctions 211 subtract with carry - extended addressing 208 SUBX
subtract with carry 208 subtract with carry 208 subtract with carry - extended addressing 208 SUBX 208 SUBX 208 SWAP 211 swap nibbles 211 symbols, additional 207 Totate right through carry 211 RP 207 RR 206, 211 Tr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 208 SBC 208 SBC 208 Subtract with carry - extended addressing 209 swap nibbles 211 symbols, additional 207 symbols, additional 207 symbols additio
subtract with carry - extended addressing 208 rotate and shift instructions 211 rotate left 211 swap nibbles 211 rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 rr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 208 SBC 208 Subx 208 SUBX 208 SU
rotate and shift instuctions 211 rotate left 211 rotate left through carry 211 rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 rr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 SBC 208 TSUBX 208 SWAP 211 swap nibbles 211 symbols, additional 207 TCM 209 TCMX 209 test complement under mask 209 test complement under mask 209 test under mask 209
rotate left 211 rotate left through carry 211 rotate right 211 rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 rr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 test under mask 209 test under mask 209
rotate left through carry 211 rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 rr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 Section 11 Swap nibbles 211 symbols, additional 207 T T T T T T T T T T T T T T T T T T
rotate right 211 rotate right through carry 211 RP 207 RR 206, 211 Tr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
rotate right through carry 211 RP 207 RR 206, 211 Tr 206 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
RP 207 RR 206, 211 Tr 206 RRC 211 TCM 209 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
TCM 209 RRC 211 TCMX 209 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
TCM 209 RRC 211 TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
TCMX 209 test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
test complement under mask 209 test complement under mask - extended addressing 209 SBC 208 test under mask 209
test complement under mask - extended addressing 209 SBC 208 test under mask 209
SBC 208 209 test under mask 209
SBC 208 209 test under mask 209
000 000 010
SCF 209, 210 test under mask - extended addressing 209
second opcode map after 1FH 225 timer signals 10
set carry flag 209, 210 timers 70
set register pointer 210 architecture 70
shift right arithmatic 211 block diagram 71