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	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/z8f0830hh020sg

Email: info@E-XFL.COM

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

Reset Controller

The Z8 Encore! F0830 Series products are reset using any one of the following: the RESET pin, Power-On Reset, Watchdog Timer (WDT) time-out, STOP Mode exit or Voltage Brown-Out (VBO) warning signal. The RESET pin is bidirectional; i.e., it functions as a reset source as well as a reset indicator.

On-Chip Debugger

The Z8 Encore! F0830 Series products feature an integrated On-Chip Debugger (OCD). The OCD provides a rich set of debugging capabilities, such as reading and writing registers, programming Flash memory, setting breakpoints and executing code. The OCD uses one single-pin interface for communication with an external host.

Acronyms and Expansions

This document references a number of acronyms; each is expanded in Table 2 for the reader's understanding.

Acronyms	Expansions
ADC	Analog-to-Digital Converter
NVDS	Nonvolatile Data Storage
WDT	Watchdog Timer
GPIO	General-Purpose Input/Output
OCD	On-Chip Debugger
POR	Power-On Reset
VBO	Voltage Brown-Out
IPO	Internal Precision Oscillator
PDIP	Plastic Dual Inline Package
SOIC	Small Outline Integrated Circuit
SSOP	Small Shrink Outline Package
QFN	Quad Flat No Lead
IRQ	Interrupt request
ISR	Interrupt service routine
MSB	Most significant byte
LSB	Least significant byte
PWM	Pulse Width Modulation
SAR	Successive Approximation Regis-

Table 2. Acronyms and Expansions

Z8 Encore![®] F0830 Series Product Specification



Figure 5. Z8F0830 Series in 28-Pin QFN Package

10

29

Table 12. Reset Status Register (RSTSTA	.T)
---	------------

Bit	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	Reserved			
RESET	See Table 13 0 0 0			0	0	0		
R/W	R	R	R	R	R	R	R	R
Address				FF	0H			
Bit	Description	n						
[7] POR	Power-On Reset Indicator This bit is set to 1 if a Power-On Reset event occurs and is reset to 0, if a WDT time-out or Stop Mode Recovery occurs. Reading this register also reset this bit to 0.							
[6] STOP	Stop Mode Recovery Indicator This bit is set to 1 if a Stop Mode Recovery occurs. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurs because of a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the Stop Mode Recovery is not caused by a WDT time-out. This bit is reset by a Power-On Reset or a WDT time-out that occurred while not in STOP Mode. Reading this register also resets this bit.							
[5] WDT	Watchdog Timer Time-Out Indicator This bit is set to 1 if a WDT time-out occurs. A Power-On Reset resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit. This read must occur before clearing the WDT interrupt.							
[4] EXT	External Reset Indicator If this bit is set to 1, a reset initiated by the external RESET pin occurred. A Power-On Reset or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.							
[3:0]	Reserved These registers are reserved and must be programmed to 0000.							

Table 13. POR Indicator Values

Reset or Stop Mode Recovery Event	POR	STOP	WDT	EXT
Power-On Reset	1	0	0	0
Reset using RESET pin assertion	0	0	0	1
Reset using Watchdog Timer time-out	0	0	1	0
Reset using the On-Chip Debugger (OCTCTL[1] set to 1)	1	0	0	0
Reset from STOP Mode using DBG pin driven Low	1	0	0	0
Stop Mode Recovery using GPIO pin transition	0	1	0	0
Stop Mode Recovery using WDT time-out	0	1	1	0

Low-Power Modes

The Z8 Encore! F0830 Series products contain power saving features. The highest level of power reduction is provided by the STOP Mode. The next level of power reduction is provided by the HALT Mode.

Further power savings can be implemented by disabling the individual peripheral blocks while in NORMAL Mode.

The user must not enable the pull-up register bits for unused GPIO pins, since these ports are default output to VSS. Unused GPIOs include those missing on 20-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

STOP Mode

Executing the eZ8 CPU's STOP instruction places the device into STOP Mode. In STOP Mode, the operating characteristics are:

- Primary crystal oscillator and Internal Precision Oscillator are stopped; XIN and XOUT (if previously enabled) are disabled and PA0/PA1 revert to the states programmed by the GPIO registers
- System clock is stopped
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- Watchdog Timer's internal RC oscillator continues to operate if enabled by the Oscillator Control Register
- If enabled, the Watchdog Timer logic continues to operate
- If enabled for operation in STOP Mode by the associated Flash option bit, the Voltage Brown-Out protection circuit continues to operate
- All other on-chip peripherals are idle

To minimize the current in STOP Mode, all GPIO pins that are configured as digital inputs must be driven to V_{DD} when the pull-up register bit is enabled or to one of power rail (V_{DD} or GND) when the pull-up register bit is disabled. The device can be brought out of STOP Mode using Stop Mode Recovery. For more information about Stop Mode Recovery, see *the* <u>Reset and Stop Mode Recovery</u> *chapter on page 21*.

39

GPIO Interrupts

Many of the GPIO port pins can be used as interrupt sources. Some port pins can be configured to generate an interrupt request on either the rising edge or falling edge of the input pin signal. Other port pin interrupt sources, generate an interrupt when any edge occurs (both rising and falling). See the <u>Interrupt Controller</u> chapter on page 53 for more information about interrupts using the GPIO pins.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data and output data; Table 17 lists these port registers. Use the Port A–D Address and Control registers together to provide access to subregisters for port configuration and control.

Port Register Name
Port A–D Address Register (selects subregisters)
Port A–D Control Register (provides access to subregisters)
Port A–D Input Data Register
Port A–D Output Data Register
Port Register Name
Data Direction
Alternate Function
Output Control (open-drain)
High Drive Enable
Stop Mode Recovery Source Enable
Pull-Up Enable
Alternate Function Set 1
Alternate Function Set 2

Table 17. GPIO Port Registers and Subregisters

Z8 Encore![®] F0830 Series Product Specification

Architecture

Figure 9 displays the Interrupt Controller block diagram.



Figure 9. Interrupt Controller Block Diagram

Operation

This section describes the operational aspects of the following functions.

Master Interrupt Enable: see page 55

Interrupt Vectors and Priority: see page 56

Interrupt Assertion: see page 56

Software Interrupt Assertion: see page 57

Master Interrupt Enable

The master interrupt enable bit (IRQE) in the Interrupt Control Register globally enables and disables the interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an EI (enable interrupt) instruction
- Execution of an IRET (return from interrupt) instruction

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	Port A7 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	 Port A Pin x Interrupt Request 0 = No interrupt request is pending for GPIO Port A pin x. 1 = An interrupt request from GPIO Port A pin x is awaiting service. 				
Note: x	lote: x indicates the specific GPIO port pin number (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:

Bit	Description (Continued)
[6] TPOL (cont'd)	 PWM DUAL OUTPUT Mode 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. 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.
	 CAPTURE RESTART Mode 0 = Count is captured on the rising edge of the timer input signal. 1 = Count is captured on the falling edge of the timer input signal. COMPARATOR COUNTER Mode 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.
	Caution: 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 ^{PRES} , 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.

Analog-to-Digital Converter

The Z8 Encore! MCU includes an eight-channel Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC). The ADC converts an analog input signal to a 10-bit binary number. The features of the SAR ADC include:

- Eight analog input sources multiplexed with general purpose I/O ports
- Fast conversion time, less than 11.9µs
- Programmable timing controls
- Interrupt on conversion complete
- Internal voltage reference generator
- Ability to select external reference voltage
- When configuring an ADC using external $V_{\text{REF}}, \text{PB5}$ is used as V_{REF} in the 28-pin package

Architecture

The ADC architecture, displayed in Figure 11, consists of an 8-input multiplexer, sampleand-hold amplifier and 10-bit SAR ADC. The ADC digitizes the signal on a selected channel and stores the digitized data in the ADC data registers. In an environment with high electrical noise, an external RC filter must be added at the input pins to reduce highfrequency noise.

 $T_{CONV} = T_{S/H} + T_{CON}$ $T_{CONV} = T_S + T_H + 13 * SCLK * 16$

where:

$$\begin{split} & \text{SCLK} = \text{System Clock} \\ & \text{T}_{\text{CONV}} = \text{Total conversion time} \\ & \text{T}_{\text{S}} = \text{Sample time} (\text{SCLK} * \text{ADCST}) \\ & \text{T}_{\text{CON}} = \text{Conversion time} (13 * \text{SCLK} * 16) \\ & \text{T}_{\text{H}} = \text{Hold time} (\text{SCLK} * \text{ADCSST}) \\ & \text{DIV} = 16 (\text{fixed to divide by 16 for F0830 Series products}) \end{split}$$

Example: For an F0830 Series MCU running @ 20MHz:

$$\begin{split} T_{CONV} &= 1 \mu s + 0.5 \mu s + 13 * SCLK * DIV \\ T_{CONV} &= 1 \mu s + 0.5 \mu s + 13 * (1/20 \text{ MHz}) * 16 = 11.9 \mu s \end{split}$$





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).

FHSWP	FWP	Flash Code Protection Description
0	0	Programming and erasing disabled for all Flash program memory. In user code pro- gramming, page erase and mass erase are all disabled. Mass erase is available through the On-Chip Debugger.
0 or 1	1	Programming, page erase and mass erase are enabled for all of the Flash program memory.

Table 71. Flash Code Protection using the Flash Option Bits

At reset, the Flash Controller is locked to prevent accidental program or erasure of Flash memory. To program or erase Flash memory, first write the target page to the page select register. 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 same page previously stored there. 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 19 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 is allowed through the debug port.

After unlocking a specific page, the user 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 causes 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! devices are divided into maximum number of eight sectors. A sector is oneeighth of the total size of Flash memory, unless this value is smaller than the page size, in which case the sector and page sizes are equal. On Z8 Encore! F0830 Series devices, the sector size is varied according to the Z8 Encore! F0830 Series Flash Memory Configuration shown in Table 69 on page 108 and in Figures 14 through 18, which follow the table

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 is lost. User code must write this register in their initialization routine if they want 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 Register 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,

Page Erase

Flash memory can be erased one page (512 bytes) at a time. Page erasing Flash memory sets all bytes in that page to the value FFH. The Flash Page Select Register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control Register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the page erase operation completes. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash Status Register to determine when the Page Erase operation is complete. When the page erase is complete, the Flash Controller returns to its Locked state.

Mass Erase

Flash memory can also be mass erased using the Flash Controller, but only by using the On-Chip Debugger. Mass erasing Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the mass erase successfully enabled, writing the value 63H to the Flash Control Register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash Status Register to determine when the Mass Erase operation is complete. When the mass erase is complete, the Flash Controller returns to its Locked state.

Flash Controller Bypass

The Flash Controller can be bypassed; instead, the control signals for Flash memory can be brought out to the GPIO pins. Bypassing the Flash Controller allows faster row programming algorithms by controlling the Flash programming signals directly.

Row programing is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of Flash memory. Mass Erase and Page Erase operations are also supported, when the Flash Controller is bypassed.

For more information about bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore!*. This document is available for download at <u>www.zilog.com</u>.

Flash Controller Behavior in Debug Mode

The following behavioral changes can be observed in the Flash Controller when the Flash Controller is accessed using the On-Chip Debugger:

• The Flash write protect option bit is ignored.

Bit	Description (Continued)
[1:0]	Filter Select
FilterSely	2-bit selection for the clock filter mode.
	00 = No filter.
	01 = Filter low level noise on high level signal.
	10 = Filter high level noise on low level signal.
	11 = Filter both.
Notes: x ir	ndicates bit values 3–1; y indicates bit values 1–0.

Note: The bit values used in Table 89 are set at factory and no calibration is required.

DlyCtl3, DlyCtl2, DlyCtl1	Low Noise Pulse on High Signal (ns)	High Noise Pulse on Low Signal (ns)			
000	5	5			
001	7	7			
010	9	9			
011	11	11			
100	13	13			
101	17	17			
110	20	20			
111	25	25			
Note: The variation is about 30%.					

Table 90. ClkFlt Delay Control Definition

157

Crystal Oscillator

The products in the Z8 Encore! F0830 Series contain an on-chip crystal oscillator for use with external crystals with 32 kHz to 20 MHz frequencies. In addition, the oscillator supports external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with frequencies up to 8 MHz. The on-chip crystal oscillator can be used to generate the primary system clock for the internal eZ8 CPU and the majority of its on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32 kHz–20 MHz). If an external clock generator is used, the X_{OUT} pin must remain unconnected. The on-chip crystal oscillator also contains a clock filter function. To see the settings for this clock filter, see <u>Table 90</u> on page 133. By default, however, this clock filter is disabled; therefore, no divide to the input clock (namely, the frequency of the signal on the X_{IN} input pin) can determine the frequency of the system clock when using the default settings.

Note: Although the X_{IN} pin can be used as an input for an external clock generator, the CLKIN pin is better suited for such use. See *the* System Clock Selection section on page 151 for more information.

Operating Modes

The Z8 Encore! F0830 Series products support the following four OSCILLATOR Modes:

- Minimum power for use with very low frequency crystals (32kHz to 1MHz)
- Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 8 MHz)
- Maximum power for use with high frequency crystals (8MHz to 20MHz)
- On-chip oscillator configured for use with external RC networks (<4MHz)

The OSCILLATOR Mode is selected using user-programmable Flash option bits. See the <u>Flash Option Bits</u> chapter on page 124 for more information.

Crystal Oscillator Operation

The XTLDIS Flash option bit controls whether the crystal oscillator is enabled during reset. The crystal may later be disabled after reset if a new oscillator has been selected as the system clock. If the crystal is manually enabled after reset through the OSCCTL Reg-

Hex Address: F83

Table 153. LED Drive Level High Register (LEDLVLH)

Bit	7	6	5	4	3	2	1	0
Field				LEDLV	LH[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						

Hex Address: F84

Table 154. LED Drive Level Low Register (LEDLVLL)

Bit	7	6	5	4	3	2	1	0
Field				LEDLV	LL[7:0]			
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address		F84H						

Hex Address: F85

This address range is reserved.

Oscillator Control

For more information about the Oscillator Control registers, see the <u>Oscillator Control</u> <u>Register Definitions</u> section on page 154.

Hex Address: F86

Table 155.	Oscillator	Control	Register	(OSCCTL	.)
------------	------------	---------	----------	---------	----

Bit	7	6	5	4	3	2	1	0
Field	INTEN	XTLEN	WDTEN	POFEN	WDFEN		SCKSEL	
RESET	1	0	1	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	6H			

Hex Address: FC5

Table 162. IF	RQ1 Enable	Low Bit R	Register (I	RQ1ENL)
---------------	------------	-----------	-------------	---------

Bit	7	6	5	4	3	2	1	0
Field	PA7ENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address		FC5H						

Hex Address: FC6

Table 163. Interrupt Request 2 Register (IRQ2)

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		FC6H						

Hex Address: FC7

Table 164. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	7	6	5	4	3	2	1	0
Field		Rese	erved		C3ENH	C2ENH	C1ENH	C0ENH
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		FC7H						

Hex Address: FC8

Table 165. IRQ2 Enable Low Bit Register (IRQ2ENL)

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

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				PADD	R[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				PC	TL				
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								

Hex Address: FFB

Table 196. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0		
Field	FFREQL									
RESET	0									
R/W	R/W									
Address	FFBH									

watchdog timer reload upper byte (WDTU) 96 register file 14 register pair 165 register pointer 165 registers ADC channel 1 102 ADC data high byte 103 ADC data low bit 103, 104, 105 reset and stop mode characteristics 22 and stop mode recovery 21 carry flag 167 sources 23 **RET 169** return 169 RL 169 **RLC 169** rotate and shift instuctions 169 rotate left 169 rotate left through carry 169 rotate right 170 rotate right through carry 170 RP 165 RR 165, 170 rr 165 **RRC** 170

S

SBC 167 SCF 167, 168 second opcode map after 1FH 183 set carry flag 167, 168 set register pointer 168 shift right arithmatic 170 shift right logical 170 signal descriptions 11 software trap 169 source operand 165 SP 165 SRA 170 src 165 SRL 170 SRP 168 stack pointer 165 **STOP 168** stop mode 30, 168 stop mode recovery sources 26 using a GPIO port pin transition 27, 28 using watch-dog timer time-out 27 **SUB 167** subtract 167 subtract - extended addressing 167 subtract with carry 167 subtract with carry - extended addressing 167 **SUBX 167 SWAP 170** swap nibbles 170 symbols, additional 165

Т

Table 134. Power Consumption Reference Table 197 **TCM 167 TCMX 167** test complement under mask 167 test complement under mask - extended addressing 167 test under mask 167 test under mask - extended addressing 167 tiing diagram, voltage measurement 100 timer signals 11 timers 68 architecture 68 block diagram 69 capture mode 77, 78, 89, 90 capture/compare mode 81, 89 compare mode 79, 89 continuous mode 70, 89 counter mode 71.72 counter modes 89 gated mode 80, 89 one-shot mode 69, 89 operating mode 69 PWM mode 74, 75, 89, 90 reading the timer count values 82