



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	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	29
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.620", 15.75mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1621pm020ec

Email: info@E-XFL.COM

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

zilog

Introduction

Zilog's Z8 Encore! XP MCU family of products are a line of Zilog[®] microcontroller products based upon the 8-bit eZ8 CPU. The Z8 Encore! XP[®] 64K Series Flash Microcontrollers, hereafter referred to collectively as the Z8 Encore! XP or the 64K Series adds Flash memory to Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming capability allows for faster development time and program changes in the field. The new eZ8TM CPU is upward compatible with existing Z8[®] instructions. The rich-peripheral set of the Z8 Encore! XP makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Features

The features of Z8 Encore! XP 64K Series Flash Microcontrollers include:

- 20 MHz eZ8 CPU
- Up to 64 KB Flash with in-circuit programming capability
- Up to 4 KB register RAM
- 12-channel, 10-bit Analog-to-Digital Converter (ADC)
- Two full-duplex 9-bit UARTs with bus transceiver Driver Enable control
- Inter-integrated circuit (I²C)
- Serial Peripheral Interface (SPI)
- Two Infrared Data Association (IrDA)-compliant infrared encoder/decoders
- Up to four 16-bit timers with capture, compare, and PWM capability
- Watchdog Timer (WDT) with internal RC oscillator
- Three-channel DMA
- Up to 60 input/output (I/O) pins
- 24 interrupts with configurable priority
- On-Chip Debugger
- Voltage Brownout (VBO) Protection
- Power-On Reset (POR)
- Operating voltage of 3.0 V to 3.6 V with 5 V-tolerant inputs
- 0 °C to +70 °C, -40 °C to +105 °C, and -40 °C to +125 °C operating temperature ranges

zilog



Control Register Summary

Timer 0 High Byte T0H (F00H - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 Timer 0 current count value [15:8] **Timer 0 Low Byte** T0L (F01H - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 Timer 0 current count value [7:0] Timer 0 Reload High Byte T0RH (F02H - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 Timer 0 reload value [15:8] **Timer 0 Reload Low Byte** TORL (HF03 - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 Timer 0 reload value [7:0] **Timer 0 PWM High Byte** T0PWMH (F04H - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 — Timer 0 PWM value [15:8] Timer 0 Control 0 T0CTL0 (F06H - Read/Write) D7 D6 D5 D4 D3 D2 D1 D0 Reserved Cascade Timer 0 = Timer 0 Input signal is GPIO pin 1 = Timer 0 Input signal is Timer 3 out Reserved



D7|D6|D5|D4|D3|D2|D1|D0| Timer 1 reload value [7:0]

PS019919-1207







currently converting

zilog

Stop Mode Recovery Using a GPIO Port Pin Transition HALT

Each of the GPIO Port pins may be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. The GPIO Stop Mode Recovery signals are filtered to reject pulses less than 10 ns (typical) in duration. In the Watchdog Timer Control register, the STOP bit is set to 1.



Caution: In STOP mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin through the end of the Stop Mode Recovery delay. Thus, short pulses on the Port pin can initiate Stop Mode Recovery without being written to the Port Input Data register or without initiating an interrupt (if enabled for that pin).



T1I—Timer 1 Interrupt Request

0 = No interrupt request is pending for Timer 1.

1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request

0 = No interrupt request is pending for Timer 0.

1 = An interrupt request from Timer 0 is awaiting service.

U0RXI—UART 0 Receiver Interrupt Request

0 = No interrupt request is pending for the UART 0 receiver.

1 = An interrupt request from the UART 0 receiver is awaiting service.

U0TXI-UART 0 Transmitter Interrupt Request

0 = No interrupt request is pending for the UART 0 transmitter.

1 = An interrupt request from the UART 0 transmitter is awaiting service.

I²CI— I²C Interrupt Request

0 = No interrupt request is pending for the I²C.

1 = An interrupt request from the I²C is awaiting service.

SPII—SPI Interrupt Request

0 = No interrupt request is pending for the SPI.

1 = An interrupt request from the SPI is awaiting service.

ADCI—ADC Interrupt Request

0 = No interrupt request is pending for the Analog-to-Digital Converter.

1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 25) stores interrupt requests for both vectored and polled interrupts. When a request is presented 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.

BITS	7	6	5	4	3	2	1	0	
FIELD	PAD7I	PAD6I	PAD5I	PAD4I	PAD3I	PAD2I	PAD1I	PAD0I	
RESET	0								
R/W	R/W								
ADDR	FC3H								

Table 25.	Interrupt	Request 1	Register	(IRQ1)
-----------	-----------	------------------	----------	--------



C1ENL—Port C1 Interrupt Request Enable Low Bit C0ENL—Port C0 Interrupt Request Enable Low Bit

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 36) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port input pin. The Interrupt Port Select register selects between Port A and Port D for the individual interrupts.

Table 36. Interrupt Edge Select Register (IRQES)

BITS	7	6	5	4	3	2	1	0	
FIELD	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0	
RESET	0								
R/W	R/W								
ADDR	FCDH								

IES*x*—Interrupt Edge Select *x*

The minimum pulse width should be greater than 1 system clock to guarantee capture of the edge triggered interrupt. Shorter pulses may be captured but not guaranteed. 0 = An interrupt request is generated on the falling edge of the PAx/PDx input.

1 = An interrupt request is generated on the rising edge of the PAx/PDx input.

where *x* indicates the specific GPIO Port pin number (0 through 7).

Interrupt Port Select Register

The Port Select (IRQPS) register (Table 37) determines the port pin that generates the PAx/PDx interrupts. This register allows either Port A or Port D pins to be used as interrupts. The Interrupt Edge Select register controls the active interrupt edge.

			-						
BITS	7	6	5	4	3	2	1		
FIELD	PAD7S	PAD6S	PAD5S	PAD4S	PAD3S	PAD2S	PAD1S		
RESET	0								

Table 37. Interrupt Port Select Register (IRQPS)

0

PAD0S







Operation

Data Format

The UART always transmits and receives data in an 8-bit data format, least-significant bit first. An even or odd parity bit can be optionally added to the data stream. Each character begins with an active Low Start bit and ends with either 1 or 2 active High Stop bits. Figure 14 and Figure 15 on page 105 displays the asynchronous data format employed by the UART without parity and with parity, respectively.



Table 64. SPI Control Register (SPICTL)

BITS	7	6	5	4	3	2	1	0		
FIELD	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN		
RESET	0									
R/W	R/W									
ADDR	F61H									

IRQE—Interrupt Request Enable

0 = SPI interrupts are disabled. No interrupt requests are sent to the Interrupt Controller.

1 = SPI interrupts are enabled. Interrupt requests are sent to the Interrupt Controller.

STR—Start an SPI Interrupt Request

0 = No effect.

1 = Setting this bit to 1 also sets the IRQ bit in the SPI Status register to 1. Setting this bit forces the SPI to send an interrupt request to the Interrupt Control. This bit can be used by software for a function similar to transmit buffer empty in a UART. Writing a 1 to the IRQ bit in the SPI Status register clears this bit to 0.

BIRQ-BRG Timer Interrupt Request

If the SPI is enabled, this bit has no effect. If the SPI is disabled:

0 = The Baud Rate Generator timer function is disabled.

1 = The Baud Rate Generator timer function and time-out interrupt are enabled.

PHASE—Phase Select

Sets the phase relationship of the data to the clock. For more information on operation of the PHASE bit, see SPI Clock Phase and Polarity Control on page 132.

CLKPOL—Clock Polarity

0 = SCK idles Low (0).

1 = SCK idle High (1).

WOR-Wire-OR (OPEN-DRAIN) Mode Enabled

0 = SPI signal pins not configured for open-drain.

 $1 = \text{All four SPI signal pins (SCK, \overline{SS}, MISO, MOSI)}$ configured for open-drain function. This setting is typically used for multi-master and/or multi-slave configurations.

MMEN—SPI Master Mode Enable

0 = SPI configured in Slave mode.

1 = SPI configured in Master mode.

SPIEN—SPI Enable

0 = SPI disabled.

1 = SPI enabled.



SSIO—Slave Select I/O $0 = \overline{SS}$ pin configured as an input. $1 = \overline{SS}$ pin configured as an output (Master mode only). SSV—Slave Select Value If SSIO = 1 and SPI configured as a Master: $0 = \overline{SS}$ pin driven Low (0). $1 = \overline{SS}$ pin driven High (1).

This bit has no effect if SSIO = 0 or SPI configured as a Slave.

SPI Diagnostic State Register

The SPI Diagnostic State register (Table 67) provides observability of internal state. This is a read only register used for SPI diagnostics.

Table 67. SPI Diagnostic State Register (SPIDST)

BITS	7	6	5	4	3	2	1	0			
FIELD	SCKEN	TCKEN		SPISTATE							
RESET		0									
R/W		R									
ADDR	F64H										

SCKEN—Shift Clock Enable

- 0 = The internal Shift Clock Enable signal is deasserted
- 1 = The internal Shift Clock Enable signal is asserted (shift register is updates on next system clock)

TCKEN—Transmit Clock Enable

- 0 = The internal Transmit Clock Enable signal is deasserted.
- 1 = The internal Transmit Clock Enable signal is asserted. When this is asserted the serial data out is updated on the next system clock (MOSI or MISO).

SPISTATE—SPI State Machine

Defines the current state of the internal SPI State Machine.



145

- Master receives from a 7-bit slave
- Master receives from a 10-bit slave

SDA and SCL Signals

 I^2C sends all addresses, data and acknowledge signals over the SDA line, most-significant bit first. SCL is the common clock for the I^2C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I^2C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a high level. When the slave releases the clock, the I²C Controller continues the transaction. All data is transferred in bytes and there is no limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the high period of SCL.

I²C Interrupts

The I²C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The Transmit interrupt is enabled by the IEN and TXI bits of the Control register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I²C Controller and neither the START or STOP bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status register and can only be cleared by setting the START or STOP bit in the I²C Control register. When this interrupt occurs, the I²C Controller waits until either the STOP or START bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I^2C Controller (master reading data from slave). This procedure sets the RDRF bit of the I^2C Status register. The RDRF bit is cleared by reading the I^2C Data register. The RDRF bit is set during the acknowledge phase. The I^2C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.



In order for a receive (read) DMA transaction to send a Not Acknowledge on the last byte, the receive DMA must be set up to receive n-1 bytes, then software must set the NAK bit and receive the last (nth) byte directly.

Start and Stop Conditions

The master (I^2C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I²C Controller generates a START condition by pulling the SDA signal Low while SCL is High. To complete a transaction, the I²C Controller generates a Stop condition by creating a low-to-high transition of the SDA signal while the SCL signal is high. The START and STOP bits in the I²C Control register control the sending of the Start and Stop conditions. A master is also allowed to end one transaction and begin a new one by issuing a Restart. This is accomplished by setting the START bit at the end of a transaction, rather than the STOP bit. Note that the Start condition not sent until the START bit is set and data has been written to the I²C Data register.

Master Write and Read Transactions

The following sections provide a recommended procedure for performing I²C write and read transactions from the I²C Controller (master) to slave I²C devices. In general software should rely on the TDRE, RDRF and NCKI bits of the status register (these bits generate interrupts) to initiate software actions. When using interrupts or DMA, the TXI bit is set to start each transaction and cleared at the end of each transaction to eliminate a 'trailing' Transmit interrupt.

Caution should be used in using the ACK status bit within a transaction because it is difficult for software to tell when it is updated by hardware.

When writing data to a slave, the I²C pauses at the beginning of the Acknowledge cycle if the data register has not been written with the next value to be sent (TDRE bit in the I²C Status register = 1). In this scenario where software is not keeping up with the I²C bus (TDRE asserted longer than one byte time), the Acknowledge clock cycle for byte n is delayed until the Data register is written with byte n + 1, and appears to be grouped with the data clock cycles for byte n+1. If either the START or STOP bit is set, the I²C does not pause prior to the Acknowledge cycle because no additional data is sent.

When a Not Acknowledge condition is received during a write (either during the address or data phases), the I²C Controller generates the Not Acknowledge interrupt (NCKI = 1) and pause until either the STOP or START bit is set. Unless the Not Acknowledge was received on the last byte, the Data register will already have been written with the next address or data byte to send. In this case the FLUSH bit of the Control register should be set at the same time the STOP or START bit is set to remove the stale transmit data and enable subsequent Transmit interrupts.

When reading data from the slave, the I²C pauses after the data Acknowledge cycle until the receive interrupt is serviced and the RDRF bit of the status register is cleared by

zilog

- 13. The I²C Controller shifts the data out of using the SDA signal. After the first bit is sent, the Transmit interrupt is asserted.
- 14. If more bytes remain to be sent, return to step 9.
- 15. Software responds by setting the STOP bit of the I²C Control register (or START bit to initiate a new transaction). In the STOP case, software clears the TXI bit of the I²C Control register at the same time.
- 16. The I²C Controller completes transmission of the data on the SDA signal.
- 17. The slave may either Acknowledge or Not Acknowledge the last byte. Because either the STOP or START bit is already set, the NCKI interrupt does not occur.
- The I²C Controller sends the STOP (or RESTART) condition to the I²C bus. The STOP or START bit is cleared.

Address Only Transaction with a 10-bit Address

In the situation where software wants to determine if a slave with a 10-bit address is responding without sending or receiving data, a transaction can be done which only consists of an address phase. Figure 30 displays this 'address only' transaction to determine if a slave with 10-bit address will acknowledge. As an example, this transaction can be used after a 'write' has been done to a EEPROM to determine when the EEPROM completes its internal write operation and is once again responding to I²C transactions. If the slave does not Acknowledge the transaction can be repeated until the slave is able to Acknowledge.



Figure 30. 10-Bit Address Only Transaction Format

Follow the steps below for an address only transaction to a 10-bit addressed slave:

- 1. Software asserts the IEN bit in the I^2C Control register.
- 2. Software asserts the TXI bit of the I^2C Control register to enable Transmit interrupts.
- 3. The I²C interrupt asserts, because the I²C Data register is empty (TDRE = 1)
- 4. Software responds to the TDRE interrupt by writing the first slave address byte. The least-significant bit must be 0 for the write operation.
- 5. Software asserts the START bit of the I^2C Control register.
- 6. The I^2C Controller sends the START condition to the I^2C slave.



1 = DMAx is enabled and initiates a data transfer upon receipt of a request from the trigger source.

DLE—DMAx Loop Enable

- 0 = DMAx reloads the original Start Address and is then disabled after the End Address data is transferred.
- 1 = DMAx, after the End Address data is transferred, reloads the original Start Address and continues operating.

DDIR—DMAx Data Transfer Direction

- $0 = \text{Register File} \rightarrow \text{on-chip peripheral control register.}$
- 1 = on-chip peripheral control register \rightarrow Register File.

IRQEN—DMAx Interrupt Enable

0 = DMAx does not generate any interrupts.

1 = DMAx generates an interrupt when the End Address data is transferred.

WSEL—Word Select

- 0 = DMAx transfers a single byte per request.
- 1 = DMA*x* transfers a two-byte word per request. The address for the on-chip peripheral control register must be an even address.

RSS—Request Trigger Source Select

The Request Trigger Source Select field determines the peripheral that can initiate a DMA transfer. The corresponding interrupts do not need to be enabled within the Interrupt Controller to initiate a DMA transfer. However, if the Request Trigger Source can enable or disable the interrupt request sent to the Interrupt Controller, the interrupt request must be enabled within the Request Trigger Source block.

- 000 = Timer 0.
- 001 = Timer 1.
- 010 = Timer 2.
- 011 = Timer 3.

100 = DMA0 Control register: UART0 Received Data register contains valid data. DMA1 Control register: UART0 Transmit Data register empty.

101 = DMA0 Control register: UART1 Received Data register contains valid data. DMA1 Control register: UART1 Transmit Data register empty.

- 110 = DMA0 Control register: I²C Receiver Interrupt. DMA1 Control register: I²C
- Transmitter Interrupt register empty.
- 111 = Reserved.

DMAx I/O Address Register

The DMAx I/O Address register (Table 78) contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is given by {FH,

168



Reserved These bits are reserved and must be 0.

FSTAT—Flash Controller Status

 $00_{0000} =$ Flash Controller locked

00_0001 = First unlock command received

 $00_{010} =$ Second unlock command received

 $00_{011} =$ Flash Controller unlocked

00_0100 = Flash Sector Protect register selected

00_1xxx = Program operation in progress

01_0xxx = Page erase operation in progress

10_0xxx = Mass erase operation in progress

Page Select Register

The Page Select (FPS) register (Table 94) selects one of the 128 available Flash memory pages to be erased or programmed. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory locations with the 7 most significant bits of the address given by the PAGE field are erased to FFH.

The Page Select register shares its Register File address with the Flash Sector Protect Register. The Page Select register cannot be accessed when the Flash Sector Protect register is enabled.

BITS	7	6	5	4	3	2	1	0			
FIELD	INFO_EN		PAGE								
RESET	0										
R/W	R/W										
ADDR	FF9H										

Table 94. Page Select Register (FPS)

INFO_EN—Information Area Enable

0 = Information Area is not selected.

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

PAGE—Page Select

This 7-bit field selects the Flash memory page for Programming and Page Erase operations. Flash Memory Address[15:9] = PAGE[6:0].





Oscillator Operation with an External RC Network

The External RC oscillator mode is applicable to timing insensitive applications. Figure 41 displays a recommended configuration for connection with an external resistorcapacitor (RC) network.



Figure 41. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45 k Ω 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 42 displays the typical (3.3 V and 25 °C) oscillator frequency as a function of the capacitor (*C* in pF) employed in the RC network assuming a 45 k Ω external resistor. For very small values of C, the parasitic capacitance of the oscillator XIN pin and the printed circuit board should 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 20 pF are recommended.





ADC Magnitude Transfer Function (Linear Scale)

Figure 49. Analog-to-Digital Converter Frequency Response



AC Characteristics

The section provides information on the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs. Table 113 lists the 64K Series AC characteristics and timing.

Table 113. AC Characteristics

		V _{DD} = 3.0–3.6V T _A = -40 °C to 125 °C			
Symbol	Parameter	Minimum	Maximum	Units	Conditions
F _{syscik}	System Clock Frequency	_	20.0	MHz	Read-only from Flash memory.
		0.032768	20.0	MHz	Program or erasure of the Flash memory.
F _{XTAL}	Crystal Oscillator Frequency	0.032768	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
T _{XIN}	Crystal Oscillator Clock Period	50	-	ns	T _{CLK} = 1/F _{syscik}
T _{XINH}	System Clock High Time	20		ns	
T _{XINL}	System Clock Low Time	20		ns	
T _{XINR}	System Clock Rise Time	_	3	ns	T _{CLK} = 50 ns. Slower rise times can be tolerated with longer clock periods.
T _{XINF}	System Clock Fall Time	_	3	ns	T _{CLK} = 50 ns. Slower fall times can be tolerated with longer clock periods.



264



Figure 61. Second Opcode Map after 1FH



281

SUB 246 SUBX 246 SWAP 250 TCM 247 **TCMX 247** TM 247 TMX 247 **TRAP 249** watch-dog timer refresh 248 XOR 249 **XORX 249** instructions, eZ8 classes of 245 interrupt control register 79 interrupt controller 5, 67 architecture 67 interrupt assertion types 70 interrupt vectors and priority 70 operation 69 register definitions 71 software interrupt assertion 70 interrupt edge select register 78 interrupt port select register 78 interrupt request 0 register 71 interrupt request 1 register 72 interrupt request 2 register 73 interrupt return 249 interrupt vector listing 67 interrupts not acknowledge 145 receive 145 **SPI 135** transmit 145 **UART 111** introduction 1 IR 243 Ir 243 IrDA architecture 125 block diagram 125 control register definitions 128 operation 126 receiving data 127 transmitting data 126 **IRET 249**

IRQ0 enable high and low bit registers 74 IRQ1 enable high and low bit registers 75 IRQ2 enable high and low bit registers 76 IRR 243 Irr 243

J

JP 249 jump, conditional, relative, and relative conditional 249

L

LD 248 LDC 248 LDCI 247, 248 LDE 248 LDEI 247, 248 LDX 248 LEA 248 load 248 load constant 247 load constant to/from program memory 248 load constant with auto-increment addresses 248 load effective address 248 load external data 248 load external data to/from data memory and autoincrement addresses 247 load external to/from data memory and auto-increment addresses 248 load instructions 248 load using extended addressing 248 logical AND 248 logical AND/extended addressing 248 logical exclusive OR 249 logical exclusive OR/extended addressing 249 logical instructions 248 logical OR 248 logical OR/extended addressing 248 low power modes 55 LOFP 44 lead 266 64 lead 267