

17 INTERRUPTS

Most of the interfaces of the ETRAX 100LX generate interrupts. Three types of interrupts are supported:

- Non-maskable interrupts (NMI);
- Maskable interrupts with internally generated vector numbers;
- Maskable interrupts with external vector numbers.

The NMI have the highest priority, and the maskable interrupts with external vector numbers have the lowest priority. All maskable interrupts with internally generated vector numbers have the same priority.

17.1 Interrupt Masks

There are two levels of interrupt masks:

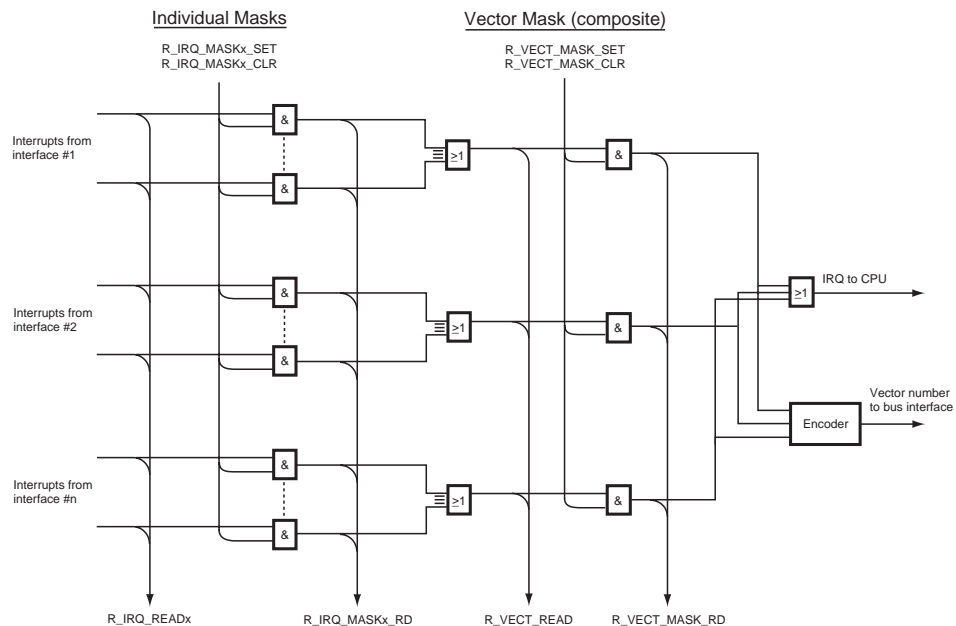


Figure 17-1 *Interrupt Masks*

At one level, the individual bits of all interrupts except the NMI are masked. At the other level there is one composite mask bit for each vector number (i.e. one composite bit for each different internal interface). There are separate addresses for setting and clearing the mask bits and only asserted mask bits (those set to 1), are set or cleared (see section 17.4 *Interrupt Registers*).

It is recommended that the software driver for each interface should control its own mask bits at the individual interrupt masking level, while general system functions control the composite vector number mask. Thus there is one mask bit on each level, even for those vectors that have one interrupt only.

All interrupt masks are undefined after reset. They must be initialised before the interrupt is enabled in the CPU.

17.2 Interrupt Status

The status of the individual interrupts can be read before and after they are gated with the interrupt mask. Registers R_IRQ_READ0 to R_IRQ_READ2 and R_USB_IRQ_RD contain the interrupt bits prior to individual masking. Registers R_IRQ_MASK0_RD to R_IRQ_MASK2_RD and R_USB_IRQ_MASK_RD contain the interrupt bits after the individual mask.

The status of the interrupt vector numbers can be read before and after they are gated with the composite mask. Register R_VECT_READ shows the interrupt vector number bits prior to composite masking. Register R_VECT_MASK_RD shows the interrupt vector number bits after the composite mask.

17.3 USB Interrupts

The USB is capable of generating a relatively large number of maskable interrupts with internally generated vector numbers. Consequently the USB interrupts are handled within the set of registers dedicated to the USB, as distinct from the set of registers entirely dedicated to interrupt handling.

17.4 Interrupt Registers

The interrupt capability of ETRAX 100LX is supported by a dedicated set of interrupt mask and status registers. These registers are arranged in three sub-sets (designated 0 to 2) that contain different interrupt bits at the individual masking level, and a fourth sub-set for the vector number bits at the composite masking level. As previously noted, a sub-set of the R_USB register set contains the USB interrupt bits at the individual masking level. This architecture is summarized in the tables below.

Register Sub-Set 0	Register	Purpose
Handles the NMI and maskable interrupts for: Ethernet error and statistics counters Network ATA Parallel port p0 Parallel port W Shared RAM SCSI-8 port p0 SCSI port W External interrupt External DMA Timers	R_IRQ_MASK0_SET	A 32-bit, write-only register for setting the 29 maskable interrupt fields with internally generated vector numbers. Four fields handle multiple interrupt mask bits by means of the multiplexed interfaces.
	R_IRQ_READ0	A 32-bit, read-only register that contains the NMI and maskable interrupts. It shows the status of these interrupts prior to the individual mask.
	R_IRQ_MASK0_RD	A 32-bit, read-only register that contains the NMI and maskable interrupts. It shows the status of these interrupts after the individual mask.
	R_IRQ_MASK0_CLR	A 32-bit, write-only register for clearing the mask bits of the maskable interrupts that have been set in register R_IRQ_MASK0_SET.

Table 17-1 Interrupt registers: Register sub-set 0

Register Sub-Set 1	Register	Purpose
Handles SW-generated interrupts and the maskable interrupts for: Parallel port p1 SCSI-8 port p1 Async. serial ports Sync. serial ports General port PA	R_IRQ_MASK1_SET	A 32-bit, write-only register for setting the maskable interrupts with internally generated vector numbers. Of these, one field handles two interrupt mask bits by means of the multiplexed interfaces.
	R_IRQ_READ1	A 32-bit, read-only register that contains the maskable interrupts. It shows the status of these interrupts prior to the individual mask.
	R_IRQ_MASK1_RD	A 32-bit, read-only register that contains the maskable interrupts. It shows the status of these interrupts after the individual mask.
	R_IRQ_MASK1_CLR	A 32-bit, write-only register for clearing the mask bits of the maskable interrupts that have been set in register R_IRQ_MASK1_SET.

Table 17-2 Interrupt registers: Register sub-set 1

Register Sub-Set 2	Register	Purpose
Handles all the interrupts for the DMA channels.	R_IRQ_MASK2_SET	A 32-bit, write-only register for setting the maskable DMA channel interrupts with internally generated vector numbers.
	R_IRQ_READ2	A 32-bit, read-only register for reading the maskable DMA interrupts. It shows the status of these interrupts prior to the individual mask.
	R_IRQ_MASK2_RD	A 32-bit, read-only register that contains the maskable interrupts. It shows the status of these interrupts after the individual mask.
	R_IRQ_MASK2_CLR	A 32-bit, write-only register for clearing the mask bits of the maskable interrupts that have been set in register R_IRQ_MASK2_SET.

Table 17-3 Interrupt registers: Register sub-set 2

USB Register Sub-Set	Register	Purpose
Handles all the interrupts for the USB.	R_USB_IRQ_MASK_SET	A 16-bit, write-only register for setting the maskable USB interrupts with internally generated vector numbers.
	R_USB_IRQ_RD	A 16-bit, read-only register for reading the maskable interrupts. It shows the status of these interrupts prior to the individual mask.
	R_USB_IRQ_MASK_RD	A 16-bit, read-only register that contains the maskable interrupts. It shows the status of these interrupts after the individual mask.
	R_USB_IRQ_MASK_CLR	A 32-bit, write-only register for clearing the mask bits of the maskable interrupts that have been set in register R_USB_IRQ_MASK_SET.

Table 17-4 Interrupt registers: USB Register sub-set

Vector Register Sub-Set	Register	Purpose
Handles the interrupt vector number bits at the composite masking level.	R_VECT_MASK_SET	A 32-bit, write-only register for setting the interrupt vector number bits.
	R_VECT_READ	A 32-bit, read-only register for reading the interrupt vector number bits. It shows the status of these bits prior to the composite mask.
	R_VECT_MASK_RD	A 32-bit, read-only register for reading the interrupt vector number bits. It shows the status of these bits after the composite mask.
	R_VECT_MASK_CLR	A 32-bit, write-only register for clearing the interrupt vector number bits that have been set in register R_VECT_MASK_SET.

Table 17-5 Interrupt registers: Vector register sub-set

Please see chapter 18.13 *Interrupt Mask and Status Registers* for detailed information on the interrupt mask and status registers, and 18.16 *Universal Serial Bus Interface Control Registers* for USB registers.

17.5 Non-Maskable Interrupts

There are two sources of non-maskable interrupt:

- The $\overline{\text{nmi}}$ pin:
- The watchdog timer.

The source of an NMI request can be identified by reading registers R_IRQ_MASK0_RD or R_IRQ_READ0.

Both NMI have the internally generated vector number 0x21.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
nmi_pin	31	This interrupt is from the external $\overline{\text{nmi}}$ pin.	This interrupt is cleared in the external unit.
watchdog_nmi	30	This interrupt is from watchdog timer.	This interrupt is cleared by stopping or restarting the watchdog timer.

Table 17-6 Non-maskable interrupts

17.6 Masked Interrupts with Internally Generated Vector Numbers

The internal interfaces of ETRAX 100LX that are not mutually exclusive have individual, internally generated vector numbers. Mutually exclusive interfaces (e.g. SCSI-8 p0 and parallel port p0), may have the same vector number. Each interface has one or more different interrupts, all of which share the same vector number. If two or more interfaces issue an interrupt at the same time, vector number 0x2F is generated and the different inputs are sorted by the software. See table 17-7 below:

Vector number	Number of interrupts	Name	Explanation
0x20	-	-	Hardware breakpoint/single step bus fault. For more information, refer to chapter 2.9 <i>Hardware Breakpoint Mechanism</i> .
0x21	2	nmi	Non-maskable interrupts (see table 17-6).
0x22	1	timer0	Timer 0 interrupt (see table 17-13)
0x23	1	timer1	Timer 1 interrupt (see table 17-13)
0x24	9	scsi0_par0	Multiple/multiplexed interrupts for ATA, parallel port p0, shared RAM and SCSI-8 p0 (see table 17-10).
0x25	4	scsi1_par1	Multiple/multiplexed interrupts for parallel port p1 and SCSI-8 p1 (see table 17-15).
0x26	4	network	Network interrupts (see table 17-9).
0x27	10	snmp	Ethernet error and statistics counter interrupts (see table 17-8).
0x28	8	serial	Asynchronous and synchronous serial port interrupts (see table 17-16).
0x29	8	sw	Software generated interrupts (see table 17-14).
0x2A	1	irq_intnr	$\overline{\text{irq}}$ pin when used with internally generated vector number (see table 17-11).
0x2B	8	pa	General port PA interrupts (see table 17-17).
0x2C	1	ext_dma0	External DMA channel 0 interrupt (see table 17-12).
0x2D	1	ext_dma1	External DMA channel 1 interrupt (see table 17-12).
0x2E	-	-	MMU bus fault.
0x2F	-	-	Simultaneous interrupts for more than one vector are active.
0x30	2	dma0	DMA channel 0 interrupts (see table 17-19).
0x31	2	dma1	DMA channel 1 interrupts (see table 17-19).
0x32	2	dma2	DMA channel 2 interrupts (see table 17-19).
0x33	2	dma3	DMA channel 3 interrupts (see table 17-19).
0x34	2	dma4	DMA channel 4 interrupts (see table 17-19).
0x35	2	dma5	DMA channel 5 interrupts (see table 17-19).
0x36	2	dma6	DMA channel 6 interrupts (see table 17-19).
0x37	2	dma7	DMA channel 7 interrupts (see table 17-19).
0x38	6	dma8	Interrupts for DMA channel 8 (see table 17-19), and for sub-channels 8.0 to 8.3 (see table 17-18).
0x39	2	dma9	DMA channel 9 interrupts (see table 17-19).
0x3F	10	usb	USB interface interrupts (see table 17-20).

Table 17-7 Masked Interrupts with Internally Generated Vector Numbers

If a source interrupt is cleared while the CPU is executing its interrupt acknowledgement sequence, the vector number of the interrupting interface is nevertheless generated. However the interrupts are not present in the interrupt status registers.

17.6.1 Interrupts in Register Sub-Set 0

In register sub-set 0, the interrupt mask bits are set in register `R_IRQ_MASK0_SET`, where only the asserted bits are set (bits written with 0 are not affected). The masked interrupts are read in registers `R_IRQ_MASK0_RD`. Register `R_IRQ_READ0` contains the interrupt bits before they are individually masked. The asserted mask bits are cleared in register `R_IRQ_MASK0_CLR`.

Each interrupt field occupies the same bit number in all the registers of sub-set 0.

Ethernet Error and Statistics Counter Interrupts

Register sub-set 0 handles ten interrupts associated with Ethernet error and statistics counters. They all have the internally generated vector number 0x27.

Name of interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
<code>sqe_test_error</code>	29	This interrupt is set when the <code>sqe_test_error</code> counter attains the value 128.	This interrupt is cleared by reading the <code>sqe_test_error</code> field of network interface register <code>R_PHY_COUNTERS</code> .
<code>carrier_loss</code>	28	This interrupt is set when the <code>carrier_loss</code> counter attains the value 128.	This interrupt is cleared by reading the <code>carrier_loss</code> field of network interface register <code>R_PHY_COUNTERS</code> .
<code>deferred</code>	27	This interrupt is set when the <code>deferred</code> counter attains the value 128.	This interrupt is cleared by reading the <code>deferred</code> field of network interface register <code>R_TR_COUNTERS</code> .
<code>late_col</code>	26	This interrupt is set when the <code>late_col</code> counter attains the value 128.	This interrupt is cleared by reading the <code>late_col</code> field of the <code>R_TR_COUNTERS</code> register.
<code>multiple_col</code>	25	This interrupt is set when the <code>multiple_col</code> counter attains the value 128.	This interrupt is cleared by reading the <code>multiple_col</code> field of network interface register <code>R_TR_COUNTERS</code> .
<code>single_col</code>	24	This interrupt is set when the <code>single_col</code> counter attains the value 128.	This interrupt is cleared by reading the <code>single_col</code> field of network interface register <code>R_TR_COUNTERS</code> .
<code>congestion</code>	23	This interrupt is set when the <code>congestion</code> counter attains the value 128.	This interrupt is cleared by reading the <code>congestion</code> field of network interface register <code>R_REC_COUNTERS</code> , an action which also clears the <code>overrun</code> interrupt.
<code>oversize</code>	22	This interrupt is set when the <code>oversize</code> counter attains the value 128.	This interrupt is cleared by reading the <code>oversize</code> field of network interface register <code>R_REC_COUNTERS</code> .
<code>alignment_error</code>	21	This interrupt is set when the <code>alignment_error</code> counter attains the value 128.	This interrupt is cleared by reading the <code>alignment_error</code> field of network interface register <code>R_REC_COUNTERS</code> .
<code>crc_error</code>	20	This interrupt is set when the <code>crc_error</code> counter attains the value 128.	This interrupt is cleared by reading the <code>crc_error</code> field of network interface register <code>R_REC_COUNTERS</code> .

Table 17-8 Ethernet Error and Statistics Counter Interrupts

Network Interrupts

Register sub-set 0 contains four network interrupts, all with the internally generated vector number 0x26.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
overrun	19	This interrupt is set when the network receiver experiences a FIFO overrun condition (congestion error). Two interrupts are available (overrun and congestion), but usually only one of them should be enabled. The overrun interrupt should be used if software intervention is necessary when an overrun error occurs. The congestion interrupt should be used if the only action needed is an error count.	This interrupt is cleared by reading the congestion field of network interface register R_REC_COUNTERS, an action which also clears the congestion interrupt.
underrun	18	This interrupt is set when the network transmitter experiences a FIFO underrun condition.	This interrupt is cleared by setting the clr_error field in network interface register R_NETWORK_TR_CTRL.
excessive_col	17	This interrupt is set when the network transmitter experiences collisions for 16 consecutive transmission attempts. It is set after the first collision if the retry field in network interface register R_NETWORK_TR_CTRL is set to disable , or when the transmitter stops after the cancel field of R_NETWORK_TR_CTRL has been set.	This interrupt is cleared by setting the clr_error field in network interface register R_NETWORK_TR_CTRL.
mdio	16	This interrupt is from the MII mdio pin. It is generated when the mdio pin is low. The interrupt should be masked off during normal data transfers over the mdio interface.	This interrupt should be cleared in the external unit that is driving the mdio pin.

Table 17-9 Network Interrupts

Masked Interrupts for ATA, Parallel Port p0, Shared RAM and SCSI-8 Port p0

In register sub-set 0 there are nine different interrupt fields that have the internally generated vector number 0x24. Four of these fields are multi-functional (bits 11 to 8), containing different interface-dependent interrupts.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
ata_drq3	15	This interrupt is set when a unit on ATA bus 3 requests a DMA transfer.	This interrupt is cleared at the end of the DMA transfer on ATA bus 3.
ata_drq2	14	This interrupt is set when a unit on ATA bus 2 requests a DMA transfer.	This interrupt is cleared at the end of the DMA transfer on ATA bus 2.
ata_drq1	13	This interrupt is set when a unit on ATA bus 1 requests a DMA transfer.	This interrupt is cleared at the end of the DMA transfer on ATA bus 1.
ata_drq0	12	This interrupt is set when a unit on ATA bus 0 requests a DMA transfer.	This interrupt is cleared at the end of the DMA transfer on ATA bus 0.
par0_ecp_cmd	11	When parallel port p0 is in ECP mode, this interrupt is set when an ECP command is received at the port.	This interrupt is cleared by reading the ecp_cmd_bit field in parallel port p0 register R_PAR0_DATA.
ata_irq3		When ATA is in use, this interrupt is set when a unit on ATA bus 3 requests an interrupt.	This interrupt is cleared in the external unit on ATA bus 3.
par0_peri	10	When parallel port p0 is in use, this interrupt is set by the peripheral connected to the port.	This interrupt is cleared by acknowledging the peri_int field in parallel port p0 register R_PAR0_CTRL_DATA.
ata_irq2		When ATA is in use, this interrupt is set when a unit on ATA bus 2 requests an interrupt.	This interrupt is cleared in the external unit on ATA bus 2.
par0_data	9	When parallel port p0 is in use, this interrupt is set when data is available on the port. When DMA is used for the data transfer, this interrupt indicates that at least one byte was received since the interrupt was last cleared.	This interrupt is cleared by reading the data field of parallel port p0 register R_PAR0_DATA.
ata_irq1		When ATA is in use, this interrupt is set when a unit on ATA bus 1 requests an interrupt.	This interrupt is cleared in the external unit on ATA bus 1.
par0_ready	8	When parallel port par0 is in use, this interrupt is set when the port is ready to get new data for transmission.	The interrupt is cleared by writing new data to the data field of parallel port register R_PAR0_DATA. This field should be masked when the DMA is used for data transfers.
ata_irq0		When ATA is in use, this interrupt is set when a unit on ATA bus 0 requests an interrupt.	The interrupt is cleared in the external unit on ATA bus 0.
mio		This interrupt is set on the $\overline{\text{intio}}$ pin of the shared RAM interface.	This interrupt is cleared by setting the i field of register R_SHARED_RAM_CONFIG.
scsi0		When SCSI-8 port p0 is in use, this interrupt is set when the SCSI controller requests service from the CPU.	The interrupt is cleared by writing to the clr_status field (bit 24) of register R_SCSI0_CMD_DATA.
ata_dmaend	7	This interrupt is set when the selected ATA unit releases its DMA request (transfer completed). This interrupt should be masked, except when an ATA DMA transfer is started.	The interrupt is cleared when next ATA DMA transfer commences.

Table 17-10 Masked Interrupts for ATA, Parallel Port p0, Shared RAM and SCSI-8 Port p0

External Interrupt Configuration Fields

In register sub-set 0 there are two fields for configuring the external interrupt on the $\overline{\text{irq}}$ pin, as shown in the table below.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
irq_ext_vector_nr	5	This field configures the external interrupt on the $\overline{\text{irq}}$ pin to have an external vector number. The mask bit is only effective if the irq_int_vector_nr mask bit (4) is cleared.	This interrupt is cleared in the external unit. This can be done by software or the $\overline{\text{inta}}$ output.
irq_int_vector_nr	4	This field configures the external interrupt on the $\overline{\text{irq}}$ pin to have the internally generated vector number 0x2A. When the corresponding mask bit is set, the internally generated interrupt vector number is used and the irq_ext_vector_nr mask bit (5) has no effect.	This interrupt is cleared in the external unit. This can be done by software only.

Table 17-11 External Interrupt Configuration Fields

External DMA Interrupts

In register sub-set 0 there are two fields for handling external DMA interrupts.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
ext_dma1	3	This interrupt has the internally generated vector number 0x2D. It is set when external DMA channel 1 has stopped.	This interrupt should be masked, except when waiting for the completion of a transfer on external DMA channel 1.
ext_dma0	2	This interrupt has the internally generated vector number 0x2C. It is set when external DMA channel 0 has stopped.	This interrupt should be masked, except when waiting for the completion of a transfer on external DMA channel 0.

Table 17-12 External DMA Interrupts

Timer Interrupts

In register sub-set 0 there are two fields for handling the timer interrupts.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
timer1	1	This interrupt has the internally generated vector number 0x23. It is set whenever timer1 reaches its terminal count.	This interrupt is cleared by setting the i1 bit in timer register R_TIMER_CTRL.
timer0	0	This interrupt has the internally generated vector number 0x22. It is set whenever timer0 reaches its terminal count.	This interrupt is cleared by setting the i0 bit in timer register R_TIMER_CTRL.

Table 17-13 Timer Interrupts

17.6.2 Interrupts in Register Sub-Set 1

In this sub-set the masked interrupts are set in register R_IRQ_MASK1_SET and read in register R_IRQ_MASK1_RD. Register R_IRQ_READ1 contains the interrupt bits before they are individually masked. The asserted mask bits are cleared in register R_IRQ_MASK1_CLR.

Each interrupt field occupies the same bit number in all the registers of sub-set 1.

Software Generated Interrupts

Register sub-set 1 has eight fields for handling software generated interrupts. They all have the internally generated vector number 0x29.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
sw_int7	31	This interrupt is generated in software when mask bit sw_int7 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int7 field in register R_IRQ_MASK1_CLR is set.
sw_int6	30	This interrupt is generated in software when mask bit sw_int6 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int6 field in register R_IRQ_MASK1_CLR is set.
sw_int5	29	This interrupt is generated in software when mask bit sw_int5 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int5 field in register R_IRQ_MASK1_CLR is set.
sw_int4	28	This interrupt is generated in software when mask bit sw_int4 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int4 field in register R_IRQ_MASK1_CLR is set.
sw_int3	27	This interrupt is generated in software when mask bit sw_int3 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int3 field in register R_IRQ_MASK1_CLR is set.
sw_int2	26	This interrupt is generated in software when mask bit sw_int2 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int2 field in register R_IRQ_MASK1_CLR is set.
sw_int1	25	This interrupt is generated in software when mask bit sw_int1 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int1 field in register R_IRQ_MASK1_CLR is set.
sw_int0	24	This interrupt is generated in software when mask bit sw_int0 in register R_IRQ_MASK1_SET is asserted.	This interrupt is cleared when the sw_int0 field in register R_IRQ_MASK1_CLR is set.

Table 17-14 Software Generated Interrupts

Please refer to section 17.8 *Software Interrupts* below for more information on the software interrupts.

Parallel Port p1 and SCSI-8 Port p1 Interrupts

Register sub-set 1 has four fields for handling the interrupts at parallel port p1. One field (bit 16), contains an interrupt bit for SCSI-8 port p1 also. All these interrupts have the internally generated vector number 0x25.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
par1_ecp_cmd	19	When parallel port p1 is in ECP mode, this interrupt is set when an ECP command is received at the port.	This interrupt is cleared by reading the ecp_cmd_bit field in parallel port p1 register R_PAR1_CTRL_DATA.
par1_peri	18	When parallel port p1 is in use, this interrupt is set by the peripheral connected to the port.	This interrupt is cleared by acknowledging the peri_int field in parallel port p1 register R_PAR1_CTRL_DATA.
par1_data	17	When parallel port p1 is in use, this interrupt is set when data is available on the port. When DMA is used for the data transfer, this interrupt indicates that at least one byte was received since the interrupt was last cleared.	This interrupt is cleared by reading the data field of parallel port p1 register R_PAR1_CTRL_DATA.
par1_ready	16	When parallel port p1 is in use, this interrupt is set when the port is ready to get new data for transmission.	The interrupt is cleared by writing new data to the data field of parallel port p1 register R_PAR1_CTRL_DATA. This field should be masked when the DMA is used for data transfers.
scsi1		When SCSI-8 port p1 is in use, this interrupt is set when the SCSI controller requests service from the CPU.	The interrupt is cleared by writing to the clr_status field (bit 24) of register R_SCSI1_CMD_DATA.

Table 17-15 Parallel Port p1 and SCSI-8 Port p1 Interrupts

Asynchronous and Synchronous Serial Port Interrupts

Register sub-set 1 has eight fields for handling the interrupts of the four asynchronous serial ports (p3 to p0). All of these interrupts have the internally generated vector number 0x28.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
ser3_ready	15	This interrupt is set when Async/Sync serial port p3 is ready to acquire new data for transmission. It should be masked when the DMA is used for data transfers.	This interrupt is cleared by writing new data to the R_SERIAL3_TR_DATA register, or to the data_out field of R_SERIAL3_CTRL.
ser3_data	14	When Async/Sync serial port p3 is in use, this interrupt is set when data is available on the port. When DMA is used for the data transfer, this interrupt indicates that at least one byte was received since the interrupt was last cleared.	This interrupt is cleared when data is read from the R_SERIAL3_REC_DATA register, or from the data_in field of R_SERIAL3_READ.
ser2_ready	13	This interrupt is set when asynchronous serial port p2 is ready to acquire new data for transmission. It should be masked when the DMA is used for data transfers.	This interrupt is cleared by writing new data to the R_SERIAL2_TR_DATA register, or to the data_out field of R_SERIAL2_CTRL.
ser2_data	12	When asynchronous serial port p2 is in use, this interrupt is set when data is available on the port. When DMA is used for the data transfer, this interrupt indicates that at least one byte was received since the interrupt was last cleared.	This interrupt is cleared when data is read from the R_SERIAL2_REC_DATA register, or from the data_in field of R_SERIAL2_READ.
ser1_ready	11	This interrupt is set when Async/Sync serial port p1 is ready to acquire new data for transmission. It should be masked when the DMA is used for data transfers.	This interrupt is cleared by writing new data to the R_SERIAL1_TR_DATA register, or to the data_out field of R_SERIAL1_CTRL.
ser1_data	10	When Async/Sync serial port p1 is in use, this interrupt is set when data is available on the port. When DMA is used for the data transfer, this interrupt indicates that at least one byte was received since the interrupt was last cleared.	This interrupt is cleared when data is read from the R_SERIAL1_REC_DATA register, or from the data_in field of R_SERIAL1_READ.
ser0_ready	9	This interrupt is set when asynchronous serial port p0 is ready to acquire new data for transmission. It should be masked when the DMA is used for data transfers.	This interrupt is cleared by writing new data to the R_SERIAL0_TR_DATA register, or to the data_out field of R_SERIAL0_CTRL.
ser0_data	8	When asynchronous serial port p0 is in use, this interrupt is set when data is available on the port. When DMA is used for the data transfer, this interrupt indicates that at least one byte was received since the interrupt was last cleared.	This interrupt is cleared when data is read from the R_SERIAL0_REC_DATA register, or from the data_in field of R_SERIAL0_READ.

Table 17-16 Asynchronous and Synchronous Serial Port Interrupts

General Port PA Interrupts

Register sub-set 1 has eight fields for handling the active-high, level-triggered interrupts at general I/O port PA. They all have the internally generated vector number 0x2B.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
pa7	7	This interrupt is on bit 7 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 7 of general port PA.
pa6	6	This interrupt is on bit 6 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 6 of general port PA.
pa5	5	This interrupt is on bit 5 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 5 of general port PA.
pa4	4	This interrupt is on bit 4 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 4 of general port PA.
pa3	3	This interrupt is on bit 3 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 3 of general port PA.
pa2	2	This interrupt is on bit 2 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 2 of general port PA.
pa1	1	This interrupt is on bit 1 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 1 of general port PA.
pa0	0	This interrupt is on bit 0 of general port PA when the port is used for interrupt handling.	This interrupt is cleared in the external unit that is driving bit 0 of general port PA.

Table 17-17 General Port PA Interrupts

17.6.3 Interrupts in Register Sub-Set 2

Register sub-set 2 handles all the interrupts for the DMA channels. The masked interrupts are set in register R_IRQ_MASK2_SET and read in register R_IRQ_MASK2_RD. Register R_IRQ_READ2 contains the interrupt bits before they are individually masked. The asserted mask bits are cleared in register R_IRQ_MASK2_CLR.

Each interrupt field occupies the same bit number in all the registers of sub-set 2.

Sub-Channel Interrupts for DMA Channel 8

Register sub-set 2 has four fields for handling the descriptor interrupts of the four sub-channels in DMA channel 8.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
dma8_sub3_descr	23	This is the descriptor interrupt for DMA channel 8 sub-channel 3. It has the internally generated vector number 0x38.	This interrupt is cleared by asserting the <code>clr_descr</code> bit in DMA register R_DMA_CH8_SUB3_CLR_INTR.
dma8_sub2_descr	22	This is the descriptor interrupt for DMA channel 8 sub-channel 2. It has the internally generated vector number 0x38.	This interrupt is cleared by asserting the <code>clr_descr</code> bit in DMA register R_DMA_CH8_SUB2_CLR_INTR.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
dma8_sub1_descr	21	This is the descriptor interrupt for DMA channel 8 sub-channel 1. It has the internally generated vector number 0x38.	This interrupt is cleared by asserting the clr_descr bit in DMA register R_DMA_CH8_SUB1_CLR_INTR.
dma8_sub0_descr	20	This is the descriptor interrupt for DMA channel 8 sub-channel 0. It has the internally generated vector number 0x38.	This interrupt is cleared by asserting the clr_descr bit in DMA register R_DMA_CH8_SUB0_CLR_INTR.

Table 17-18 Sub-Channel Interrupts for DMA Channel 8

DMA End-of-Packet and Descriptor Interrupts

Register sub-set 2 handles all the end-of-packet and descriptor interrupts for the ten DMA channels.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
dma9_eop	19	This is the end-of-packet interrupt for DMA channel 9. It has the internally generated vector number 0x39.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH8_CLR_INTR.
dma9_descr	18	This is the descriptor interrupt for DMA channel 9. It has the internally generated vector number 0x39.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH8_CLR_INTR.
dma8_eop	17	This is the end-of-packet interrupt for DMA channel 8. It has the internally generated vector number 0x38.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH8_CLR_INTR.
dma8_descr	16	This is the descriptor interrupt for DMA channel 8. It has the internally generated vector number 0x38.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH8_CLR_INTR.
dma7_eop	15	This is the end-of-packet interrupt for DMA channel 7. It has the internally generated vector number 0x37.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH7_CLR_INTR.
dma7_descr	14	This is the descriptor interrupt for DMA channel 7. It has the internally generated vector number 0x37.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH7_CLR_INTR.
dma6_eop	13	This is the end-of-packet interrupt for DMA channel 6. It has the internally generated vector number 0x36.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH6_CLR_INTR.
dma6_descr	12	This is the descriptor interrupt for DMA channel 6. It has the internally generated vector number 0x36.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH6_CLR_INTR.
dma5_eop	11	This is the end-of-packet interrupt for DMA channel 5. It has the internally generated vector number 0x35.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH5_CLR_INTR.
dma5_descr	10	This is the descriptor interrupt for DMA channel 5. It has the internally generated vector number 0x35.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH5_CLR_INTR.
dma4_eop	9	This is the end-of-packet interrupt for DMA channel 4. It has the internally generated vector number 0x34.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH4_CLR_INTR.
dma4_descr	8	This is the descriptor interrupt for DMA channel 4. It has the internally generated vector number 0x34.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH4_CLR_INTR.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
dma3_eop	7	This is the end-of-packet interrupt for DMA channel 3. It has the internally generated vector number 0x33.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH3_CLR_INTR.
dma3_descr	6	This is the descriptor interrupt for DMA channel 3. It has the internally generated vector number 0x33.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH3_CLR_INTR.
dma2_eop	5	This is the end-of-packet interrupt for DMA channel 2. It has the internally generated vector number 0x32.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH2_CLR_INTR.
dma2_descr	4	This is the descriptor interrupt for DMA channel 2. It has the internally generated vector number 0x32.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH2_CLR_INTR.
dma1_eop	3	This is the end-of-packet interrupt for DMA channel 1. It has the internally generated vector number 0x31.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH1_CLR_INTR.
dma1_descr	2	This is the descriptor interrupt for DMA channel 1. It has the internally generated vector number 0x31.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH1_CLR_INTR.
dma0_eop	1	This is the end-of-packet interrupt for DMA channel 0. It has the internally generated vector number 0x30.	This interrupt is cleared by asserting the clr_eop bit in register R_DMA_CH0_CLR_INTR.
dma0_descr	0	This is the descriptor interrupt for DMA channel 0. It has the internally generated vector number 0x30.	This interrupt is cleared by asserting the clr_descr bit in register R_DMA_CH0_CLR_INTR.

Table 17-19 DMA End-of-Packet and Descriptor Interrupts

17.6.4 Interrupts in the USB Register Set

Register sub-set R_USB_IRQ handles all the interrupts for the USB interface. The masked interrupts are set in register R_USB_IRQ_MASK_SET and read in register R_USB_IRQ_MASK_RD. Register R_USB_IRQ_READ contains the interrupt bits before they are individually masked. The asserted mask bits are cleared in register R_USB_IRQ_MASK_CLR.

All USB interrupts have the internally generated vector number 0x3F. Each interrupt field occupies the same bit number in all the registers of this sub-set.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
iso_eof	13	This interrupt occurs when the DMA detects a set eof bit in an endpoint descriptor in the isochronous endpoint DMA list.	This interrupt is cleared when the iso_eof bit in register R_USB_EPID_ATTEN is read.
intr_eof	12	This interrupt occurs when the DMA detects a set eof bit in an endpoint descriptor in the interrupt endpoint DMA list.	This interrupt is cleared when the intr_eof bit in register R_USB_EPID_ATTEN is read.
iso_eot	11	This interrupt occurs when the DMA detects a set eof bit in an endpoint descriptor in the isochronous endpoint DMA list, and the USB interface has finished the last transaction of the transfer.	This interrupt is cleared when the iso_eot bit in register R_USB_EPID_ATTEN is read.

intr_eot	10	This interrupt occurs when the DMA detects a set eof bit in an endpoint descriptor in the interrupt endpoint DMA list, and the USB interface has finished the last transaction of the transfer.	This interrupt is cleared when the intr_eot bit in register R_USB_EPID_ATTEN is read.
ctl_eot	9	This interrupt occurs when the DMA detects a set eof bit in an endpoint descriptor in the control endpoint DMA list, and the USB interface has finished the last transaction of the transfer.	This interrupt is cleared when the ctrl_eot bit in register R_USB_EPID_ATTEN is read.
bulk_eot	8	This interrupt occurs when the DMA detects a set eof bit in an endpoint descriptor in the bulk endpoint DMA list, and the USB interface has finished the last transaction of the transfer.	This interrupt is cleared when bulk_eot bit in register R_USB_EPID_ATTEN is read.
epid_attn	3	This interrupt is triggered whenever a significant endpoint event occurs.	This interrupt is cleared when register R_USB_EPID_ATTEN is read.
sof	2	This interrupt is triggered each time the USB outputs a start-of-frame.	This interrupt is cleared when register R_USB_FM_NUMBER is read.
port_status	1	This interrupt signals a change in any of the configured port status registers.	This interrupt is cleared after all changed port status registers (R_USB_RH_PORT_STATUS_{12}) have been read.
ctl_status	0	This interrupt indicates a change of USB interface controller status.	This interrupt is cleared when register R_USB_STATUS is read.

Table 17-20 Interrupts in the USB Register Set

17.6.5 Vector Number Register Sub-Set

In this sub-set the composite masked interrupts are set in register R_VECT_MASK_SET and read in register R_VECT_MASK_RD. Register R_VECT_READ contains the composite interrupt bits before they are masked. The asserted composite mask bits are cleared in register R_VECT_MASK_CLR.

Each composite field occupies the same bit number in all the registers of the vector number sub-set.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
usb	31	This is the composed bit for interrupts from the USB. Vector number 0x3F.	This field is cleared by asserting the usb bit in register R_VECT_MASK_CLR.
dma9	25	This is the composed bit for interrupts from DMA channel 9. Vector number 0x39.	This field is cleared by asserting the dma9 bit in register R_VECT_MASK_CLR.
dma8	24	This is the composed bit for interrupts from DMA channel 8. Vector number 0x38.	This field is cleared by asserting the dma8 bit in register R_VECT_MASK_CLR.
dma7	23	This is the composed bit for interrupts from DMA channel 7. Vector number 0x37.	This field is cleared by asserting the dma7 bit in register R_VECT_MASK_CLR.
dma6	22	This is the composed bit for interrupts from DMA channel 6. Vector number 0x36.	This field is cleared by asserting the dma6 bit in register R_VECT_MASK_CLR.
dma5	21	This is the composed bit for interrupts from DMA channel 5. Vector number 0x35.	This field is cleared by asserting the dma5 bit in register R_VECT_MASK_CLR.
dma4	20	This is the composed bit for interrupts from DMA channel 4. Vector number 0x34.	This field is cleared by asserting the dma4 bit in register R_VECT_MASK_CLR.
dma3	19	This is the composed bit for interrupts from DMA channel 3. Vector number 0x33.	This field is cleared by asserting the dma3 bit in register R_VECT_MASK_CLR.

Name of Interrupt	Register Bit	Description of Interrupt	Interrupt Clearance
usb	31	This is the composed bit for interrupts from the USB. Vector number 0x3F.	This field is cleared by asserting the usb bit in register R_VECT_MASK_CLR.
dma2	18	This is the composed bit interrupts from DMA channel 2. Vector number 0x32.	This field is cleared by asserting the dma2 bit in register R_VECT_MASK_CLR.
dma1	17	This is the composed bit interrupts from DMA channel 1. Vector number 0x31.	This field is cleared by asserting the dma1 bit in register R_VECT_MASK_CLR.
dma0	16	This is the composed bit for interrupts from DMA channel 0. Vector number 0x30.	This field is cleared by asserting the dma0 bit in register R_VECT_MASK_CLR.
ext_dma1	13	This is the composed bit for interrupts from external DMA channel 1. Vector number 0x2D.	This field is cleared by asserting the ext_dma1 bit in register R_VECT_MASK_CLR.
ext_dma0	12	This is the composed bit for interrupts from external DMA channel 0. Vector number 0x2C.	This field is cleared by asserting the ext_dma0 bit in register R_VECT_MASK_CLR.
pa	11	This is the composed bit for interrupts from general port PA. Vector number 0x2B.	This field is cleared by asserting the pa bit in register R_VECT_MASK_CLR.
irq_intnr	10	This is the composed bit for interrupts from the irq pin, vector number 0x2A.	This field is cleared by asserting the irq_intnr bit in register R_VECT_MASK_CLR.
sw	9	This is the composed bit for software generated interrupts. Vector number 0x29.	This field is cleared by asserting the sw bit in register R_VECT_MASK_CLR.
serial	8	This is the composed bit for interrupts from the asynchronous and synchronous serial ports. Vector number 0x28.	This field is cleared by asserting the serial bit in register R_VECT_MASK_CLR.
snmp	7	This is the composed bit for Ethernet error and statistics counter interrupts. Vector number 0x27.	This field is cleared by asserting the snmp bit in register R_VECT_MASK_CLR.
network	6	This is the composed bit for interrupts from the network interface. Vector number 0x26.	This field is cleared by asserting the network bit in register R_VECT_MASK_CLR.
scsi1 par1	5	This is the composed bit for interrupts from SCSI-8 Port p1 and parallel port p1. Vector number 0x25.	This field is cleared by asserting bit 5 in register R_VECT_MASK_CLR.
scsi0 par0 ata mio	4	This is the composed bit for interrupts from SCSI-8 Port p0, parallel port p0, the ATA port and the shared RAM port. Vector number 0x24.	This field is cleared by asserting bit 4 in register R_VECT_MASK_CLR.
timer1	3	This is the composed bit for interrupts from timer 1. Vector number 0x23.	This field is cleared by asserting bit 3 in register R_VECT_MASK_CLR.
timer0	2	This is the composed bit for interrupts from timer 0. Vector number 0x22.	This field is cleared by asserting bit 2 in register R_VECT_MASK_CLR.
nmi	1	This is the composed bit for interrupts from the NMI. It cannot be masked and therefore the mask bit is always set to 1. Vector number 0x21.	This field cannot be cleared: it is always enabled.
some	0	This is the composed bit for interrupts (except NMI but including irq with an external vector number), that are active after the individual masks.	This field cannot be cleared: it is always enabled.

Table 17-21 Vector Number Register Sub-Set

17.7 External Maskable Interrupt with an External Vector Number

The external interrupt on the $\overline{\text{irq}}$ pin can be configured for an external vector number or an internally generated vector number.

To configure the external interrupt for an external vector number, the **irq_ext_vector_nr** field (bit 5) must be set in R_IRQ_MASK0_SET, and the **irq_int_vector_nr** field (bit 4) must be cleared in interrupt mask register R_IRQ_MASK0_CLR.

In the acknowledgement cycle for an interrupt with an external vector number, the $\overline{\text{inta}}$ output is asserted. In response, the external device supplies its vector number in the least significant byte on the data bus. Please refer to chapter 5.8 *External Interrupt Acknowledge* for more information.

17.8 Software Interrupts

As previously noted, the eight software interrupts with internally generated vector numbers are handled in register sub-set 1. They are set in register R_IRQ_MASK1_SET and cleared with their respective mask bits in register R_IRQ_MASK1_CLR, which actually constitutes the interrupts.

During normal program flow, the software may occasionally enter a critical region in which it cannot share resources. Typically this could occur when incoming data from an interface almost entirely fills a buffer, and usually requires that the interrupts are disabled. However ETRAX 100LX has a solution wherein the original interrupt routine can be enabled more often.

The software interrupts offer a simple and rapid response to hardware interrupts requiring actions that are extensive, but not so time-critical. When the original interrupt has executed the time-critical operation, it can set a software interrupt and return.

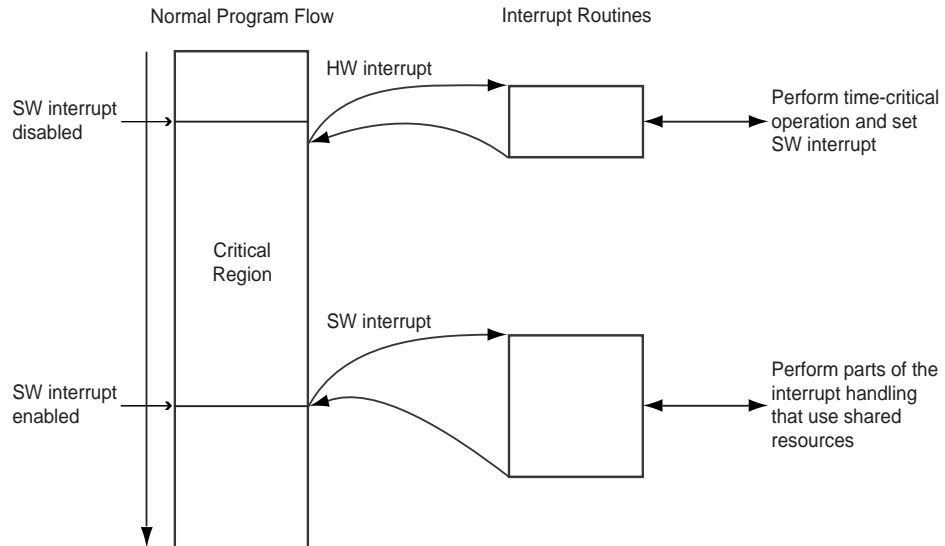


Figure 17-2 Software Generated Interrupt Routines

Figure 17-2 shows that, when the program enters a critical region, the software interrupt is disabled. The hardware interrupt then takes care of the time-critical operation and sets the software interrupt. As soon as the software interrupt is enabled, the software interrupt routine performs the more extensive action.

