

14 INTERRUPTS

ETRAX 100 supports three kinds of interrupts:

- Non maskable interrupts (NMI)
- Interrupts with internally generated vector number
- External interrupts with external vector number

NMI has the highest priority and external interrupts with external vector number the lowest. All interrupts with internally generated vector number (except NMI) have the same priority.

14.1 NMI

There are two sources for the non maskable interrupt:

- The $\overline{\text{NMI}}$ pin
- The watchdog timer

The source of the NMI request can be identified by reading the status bits in the interrupt status register.

14.2 INTERRUPTS WITH INTERNALLY GENERATED VECTOR NUMBERS

The internal interfaces have individual vector numbers, except among interfaces that are mutually exclusive (e.g. SCSI port 0 and Parallel port 0) in which case they share the same number. The external $\overline{\text{IRQ}}$ pin can be configured to have either an internal vector number or an external vector number.

Each interface have several different interrupts. All interrupts for a given interface share the same vector number. If two or more interfaces generate an interrupt at the same time, vector number 0x2F is generated and the different interrupts can be sorted out.

If the source interrupt is cleared at the same time as the CPU performs its interrupt acknowledge, the vector number of the interrupting interface will still be generated. However the interrupts will not be present in the interrupt status registers.

14.2.1 Overview of internally generated interrupt vector numbers

The table below shows the different internally generated vector numbers. The names are used in the register descriptions of the vector mask and status registers.

Vector number:	No of interrupts:	Name:	Explanation:
0x20	-	-	Hardware breakpoint
0x21	1	nmi	Non Maskable Interrupt (NMI)
0x22	1	timer0	Timer 0
0x23	1	timer1	Timer 1
0x24	9	scsi0_par0	SCSI0/Par0/ATA/Shared RAM interface
0x25	4	scsi1_par1	SCSI1/Par1
0x26	4	network	Network
0x27	10	snmp	Ethernet error and statistics counters
0x28	8	serial	Serial ports
0x29	8	sw	SW generated interrupts
0x2A	1	irq_intr	IRQ pin used with internal vector
0x2B	8	pa	Port PA
0x2C	1	ext_dma0	External DMA0
0x2D	1	ext_dma1	External DMA1
0x2E	-	-	Reserved
0x2F	-	-	Interrupts for more than one vector are active
0x30	2	dma0	DMA channel 0
0x31	2	dma1	DMA channel 1
0x32	2	dma2	DMA channel 2
0x33	2	dma3	DMA channel 3
0x34	2	dma4	DMA channel 4
0x35	2	dma5	DMA channel 5
0x36	2	dma6	DMA channel 6
0x37	2	dma7	DMA channel 7
0x38	2	dma8	DMA channel 8
0x39	2	dma9	DMA channel 9

Table 14-1 Internally generated interrupt vector numbers

14.3 EXTERNAL INTERRUPT WITH EXTERNAL VECTOR NUMBER

The external interrupt on the $\overline{\text{IRQ}}$ pin is configured for either an external vector number or an internal vector number. This is done by setting the appropriate mask bit in an interrupt mask register and clearing or setting the mask bit for internally generated vector number.

In the acknowledge cycle for an interrupt with external vector number, the $\overline{\text{INTA}}$ output will go low (see Figure 3.8 on page 40). The external device supplies its vector number on the least significant byte of the data bus in response to this.

14.4 INTERRUPT MASKS

There are two levels of interrupt masks. At the first level the individual interrupts for an interface are masked. At the second level there is one mask bit for each interrupt vector number, i.e. for each interface. There are separate addresses for setting and clearing the mask bits and only the mask bits given the value 1 are affected.

It is recommended that the software driver for each interface control its own mask bits on the individual interrupt mask level, while general system functions control the vector number mask. There will therefore be one mask bit on each level even for those vectors that only have one interrupt.

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

14.5 INTERRUPT STATUS

The interrupt status for the individual interrupts can be read before and after gating with the interrupt mask. There is also a status register showing which interrupt vector numbers are active before and after gating with the vector number mask.

14.6 SOFTWARE INTERRUPTS

Among the interrupts with internal vector number, there are 8 software interrupts. The software interrupts are set and cleared with their respective mask bits, that actually constitute the interrupts.

During normal program flow the program occasionally enters a region which is critical, i.e. it can not share resources. This could be a case when, for instance, an incoming flow of data from an I/O unit is filled into a buffer and the amount of space left in the buffer is getting short. This usually implies that interrupts have to be disabled. However, this method offers a solution where the original interrupt routine could be allowed to be enabled more often. The software interrupts supply a simple method for handling the situation when an interrupt requires some simple fast response, but also needs some more extensive but not so time critical action. After the original interrupt has performed the time critical part, it can set a software interrupt and return.

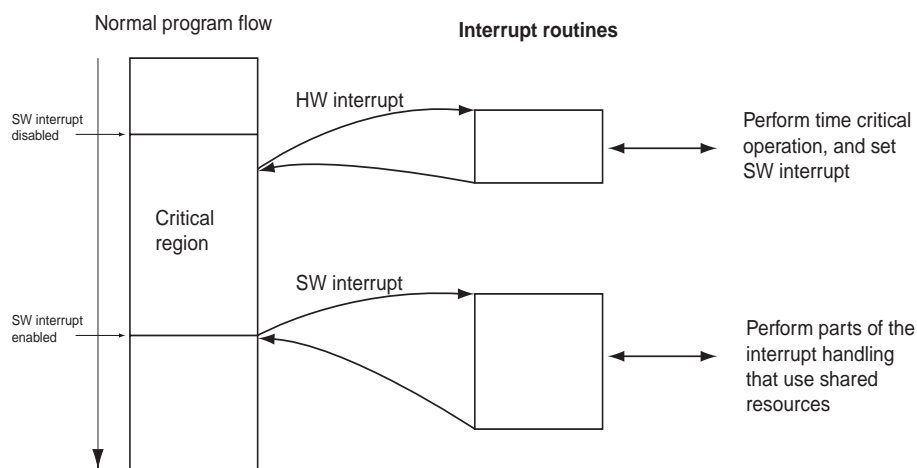


Figure 14-1 Software generated interrupts

Figure 14-1 shows that when the program enters a critical region, the software interrupt is disabled and the hardware interrupt is taking care of the time critical operation and sets the software interrupt. As soon as the software interrupt is enabled the software interrupt routine performs its more extensive action.