Chapter 11: Input/Output Organisation

Lesson 08:

Interrupts- Part 2

Objective

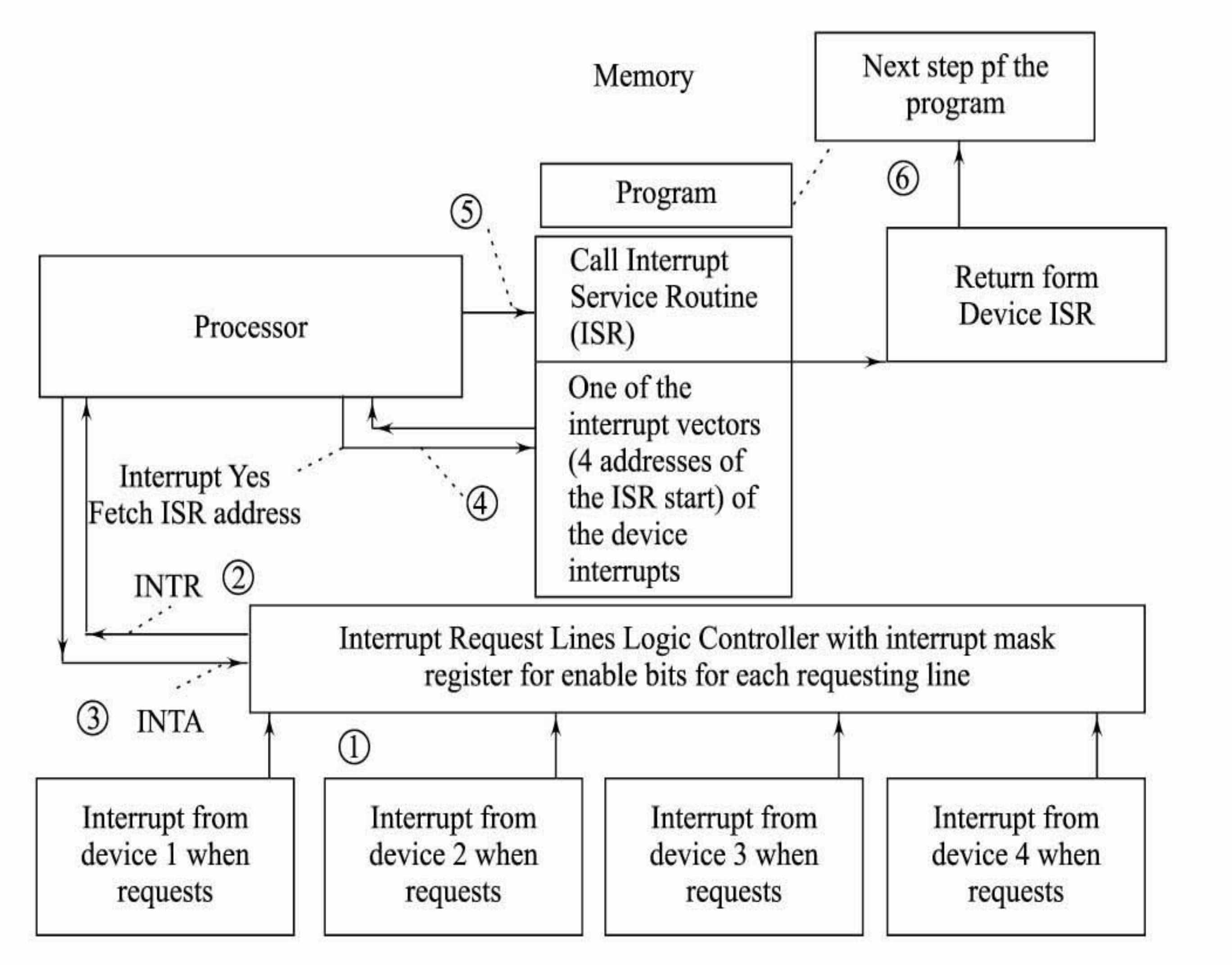
- Learn Enabling and disabling of interrupts
- Learn concept of Interrupt vectors
- Learn concept of Priority resolution in case of multiple requests
- Learn concept of Interrupt service on interrupt request by a software interrupt instruction on exception

Enabling and Disabling the Interrupts

Enabling and Disabling Interrupts

1. Nonmaskable interrupt (The action on interrupt cannot disable it by any instruction)
2. Maskable interrupts─ (The user program instruction can disable (mask) all or a few device interrupts by an)

Handling Multiple Devices



Memory
Next step pf the program
Program
⑥
⑤
Call Interrupt Service Routine (ISR)
Return form Device ISR
Processor
One of the interrupt vectors (4 addresses of the ISR start) of the device interrupts
Interrupt Yes Fetch ISR address
④
INTR ②
Interrupt Request Lines Logic Controller with interrupt mask register for enable bits for each requesting line
③ INTA
①
Interrupt from device 1 when requests
Interrupt from device 2 when requests
Interrupt from device 3 when requests
Interrupt from device 4 when requests

Schaum's Outline of Theory and Problems of Computer Architecture

Interrupts Nesting on Multiple Device Interrupts

Interrupts nesting in some processors

- Like subroutines call nesting
- During the execution of one routine another one is called
- The return from the subroutine called is always to the calling program or routine processors, which auto disable during the running of an ISR

Interrupts nesting when not provided

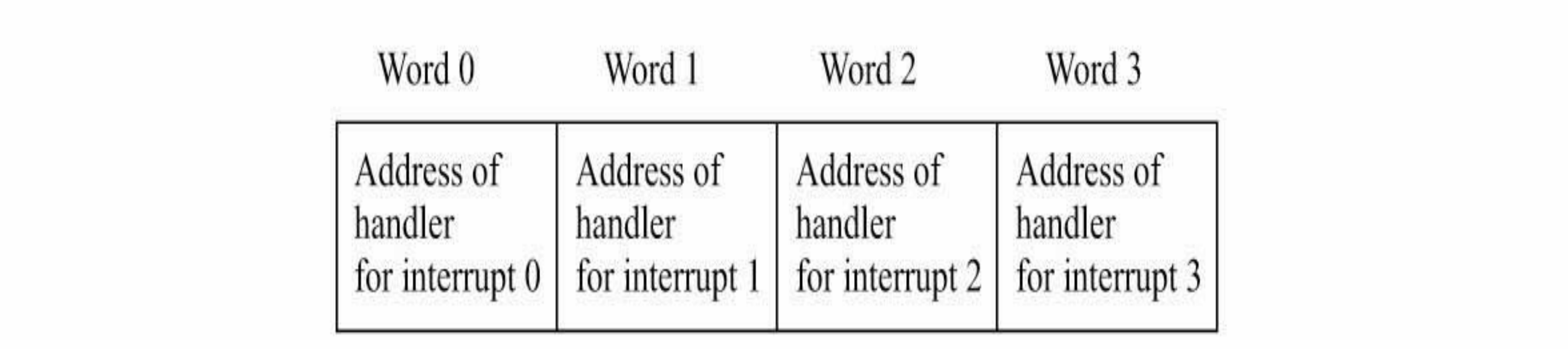
- In cases where interrupt nesting is not provided, there should be some mechanism to handle multiple devices' request
- One is making the ISRs short
- Another is to design a priority mechanism that operates only on return from the ISR
- The highest priority interrupt among the pending ones executes on the return

Processor that do not auto disable

- Some processors that do not auto disable except for a brief interval of execution of first ISR instruction provide for the priority mechanism, which operates during execution of the each ISR instruction after the first one

The Interrupt vectors

Exemplary Four Interrupt-Vector Addresses for a Processor with Four Interrupts



Word 0	Word 1	Word 2	Word 3
Address of handler for interrupt 0	Address of handler for interrupt 1	Address of handler for interrupt 2	Address of handler for interrupt 3

The Interrupt priority resolution

Priority Interrupts Organisation

- Daisy chaining of priorities─ System allocating the fixed priorities
- Priority resolution (which ISR to execute first in case multiple device interrupts that are pending) in the processor interrupt controller logic or external interrupt handler logic

Priority

- CASE 1: System Allocating the Priority Groups
- CASE 2: Priority Resolver-System Allocating the Specific Rotation Priorities

The exceptions (software interrupts)

Exceptions

- Not only the I/O devices but certain exceptional conditions can also generate *asynchronous* events— events that occur at times that the processor cannot predict or control, but which the processor must respond to reasonably quickly to provide acceptable performance
- Exceptional condition during program run detected and the software interrupt instruction initiates an interrupt

Summary

We Learnt

- Interrupt driven IO
- Enabling and disabling of interrupts
- Interrupt vectors
- Priority resolution on multiple interrupts
- Interrupt service on interrupt request by software interrupt instruction on exception

End of Lesson 08 on

Interrupts- Part 2