# Noakhali Science & Technology University

## Assignment Title: Computer Architecture and Organization

## Course Code: CSTE 2209

| SUBMITTED TO: | SUBMITTED BY: |
| --- | --- |
| Dr. Nazia Majadi<br>Associate Professor<br>Computer Science and Telecommunication Engineering | Mohammad Borhan Uddin<br>Roll: ASH2101008M<br>Year:02 Term:02<br>Computer Science and Telecommunication Engineering |

Date:10/01/2024

(1) Consider the multiplication of two unsigned numbers 12 and 13 using odd-shift method. Show the process using odd-shift method in a tabular form.

Soln:

Given numbers are 12 and 13.

$12 \rightarrow 1100$ (M)
$13 \rightarrow 1101$ (Q)

Considering a 4 bit computer architecture. So, A will be '0000' and C=0 initially.

| M | C | A | Q | Operation |
|---|---|---|---|---|
| 1100 | 0 | 0000 | 1101 | Initialization |
| 1100 | 0 | 1100 | 1101 | 1st Cycle: Adding (A = A+M) |
| 1100 | 0 | 0110 | 0110 | Right shifting (CAQ) |
| 1100 | 0 | 0011 | 0011 | 2nd Cycle: Right shifting (CAQ) |
| 1100 | 0 | 1111 | 0011 | 3rd Cycle: Adding A = A+M |
| 1100 | 0 | 0111 | 1001 | Right shifting (CAQ) |
| 1100 | 1 | 0011 | 1001 | 4th cycle: Adding A = A+M |
| 1100 | 0 | 1001 | 1100 | Right shifting (CAQ) |

So, the answer will be,

$(1100)_2 \times (1101)_2 = (10011100)_2$

(2) Write a short note on hybrid memory architecture. What do you mean by instruction pipelining? What are the benifit of it? Explain with a real case example.

Soln:

Hybrid memory architecture refers to a computer system that combines both shared memory and distributed memory components. The largest and fastest computers utilize this both types of memory, means hybrid memory architecture.

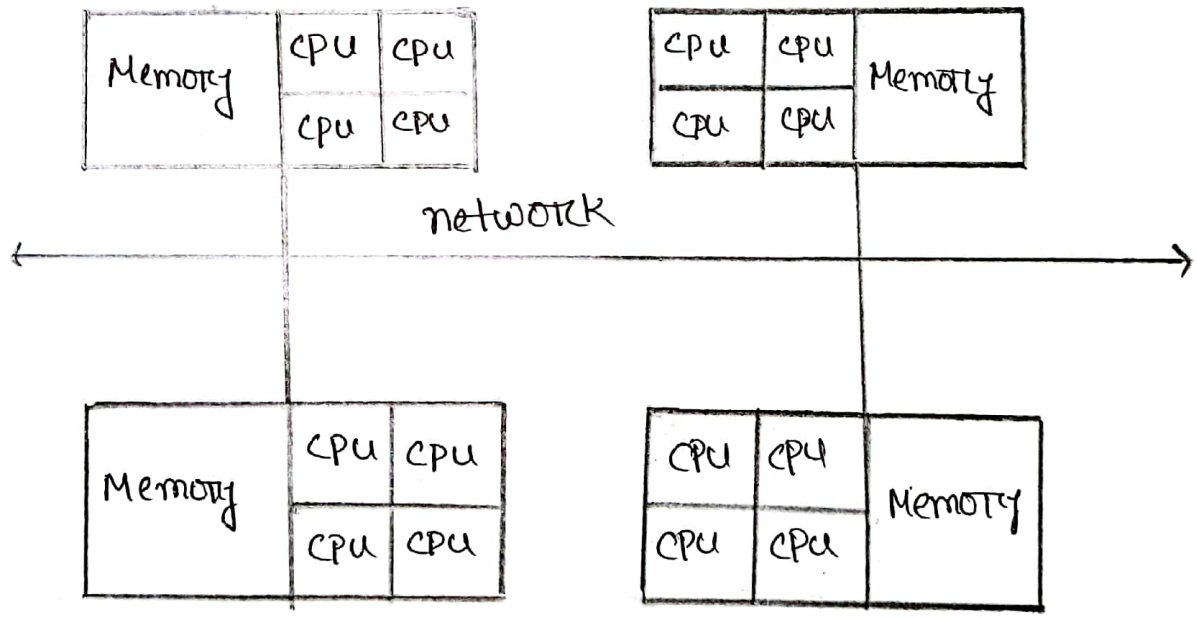


Figure : Hybrid memory architecture

Shared memory component is typically a cache coherent symmetric multiprocessing (SMP) machine, where processor can access the machine's memory as if it were global. On the other hand, the distributed memory components involves the networking of multiple SMP's, where each SMP only knows about its,

own memory and required network communication to move data between SMP's.

Instruction pipelining, on the othe hand, is a technique used in computer processor to improve performance by allowing multiple instructions into smaller stages and each stage is performed by a different part of the processor. This allow instructions to overlap in execution, resulting in increased throughput and faster processing

One of benifits of instruction pipelining is improved efficiency. By breaking down instructions into smaller stages and executing them concurrently. the processor can make better use of its resources and complete instruction more quickly. The parallelism is also invisible to the programmers.

In modern CPU's instruction pipelining is used. In a pipelined processor, different stages work simultaneously to fetch, decode, execute and write back instruction. The overlapping improves performance and resource utilization.

Example of instruction pipelining with no branch operation :

| Step | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | FI | DA | FO | EX | | | | | |
| 2 | | FI | DA | FO | EX | | | | |
| 3 | | | FI | DA | FO | EX | | | |
| 4 | | | | FI | DA | FO | EX | | |
| 5 | | | | | FI | DA | FO | EX | |
| 6 | | | | | | FI | DA | FO | EX |

* FI = Fetch Instruction
* FO = Fetch Operation
* DA = Decode Instruction
* EX = Execute Instruction

(iii) What are the differences among read-mostly memory ?

Soln:

Read-mostly memory is designed for applications where read operations are much more frequent than write operation. It offers non-volatile, ensuring data persistance. There are commonly three types of read mostly memory: EPROM, EEPROM and Flash memory. The difference between among memory are written below:

| EPROM | EEPROM | Flash memory |
|---|---|---|
| Writing to an EPROM can take up to 20 minutes. | Writing to an EEPROM can take Reversal hundred microsecond per byte. | It takes one or few seconds to be written to. |
| Its have a high density means a lot of information can be stored in small space. | It has less density than EPROM's. | Flash memory have a high density compare to EPROM's. |
| EPROM is erased by exposing it to ultraviolate radiation. | EEPROM is electrically erasedble, allows it to reprogrammed without being removed from the device. | Also electrically erasable, but it is done in large blocks. |
| Its expensive. | It's more expensive than EPROM's. | It's intermediate in cost between than two. |
| It's have a long lifespan but are limited by the numbers of times, they can be erased and reprogrammed. | It's have shorter lifespan than EPROM's. | Flash memory has a lifespan comparable to EEPROM's but more durable because it doesn't degrade with read operations. |

(4) Draw a flowchart for single interrupt processing. Explain in details.

Soln:

The occurence of an interrupt fires a numbers of events both in processor hardware and software. Figure below displays a sequence.

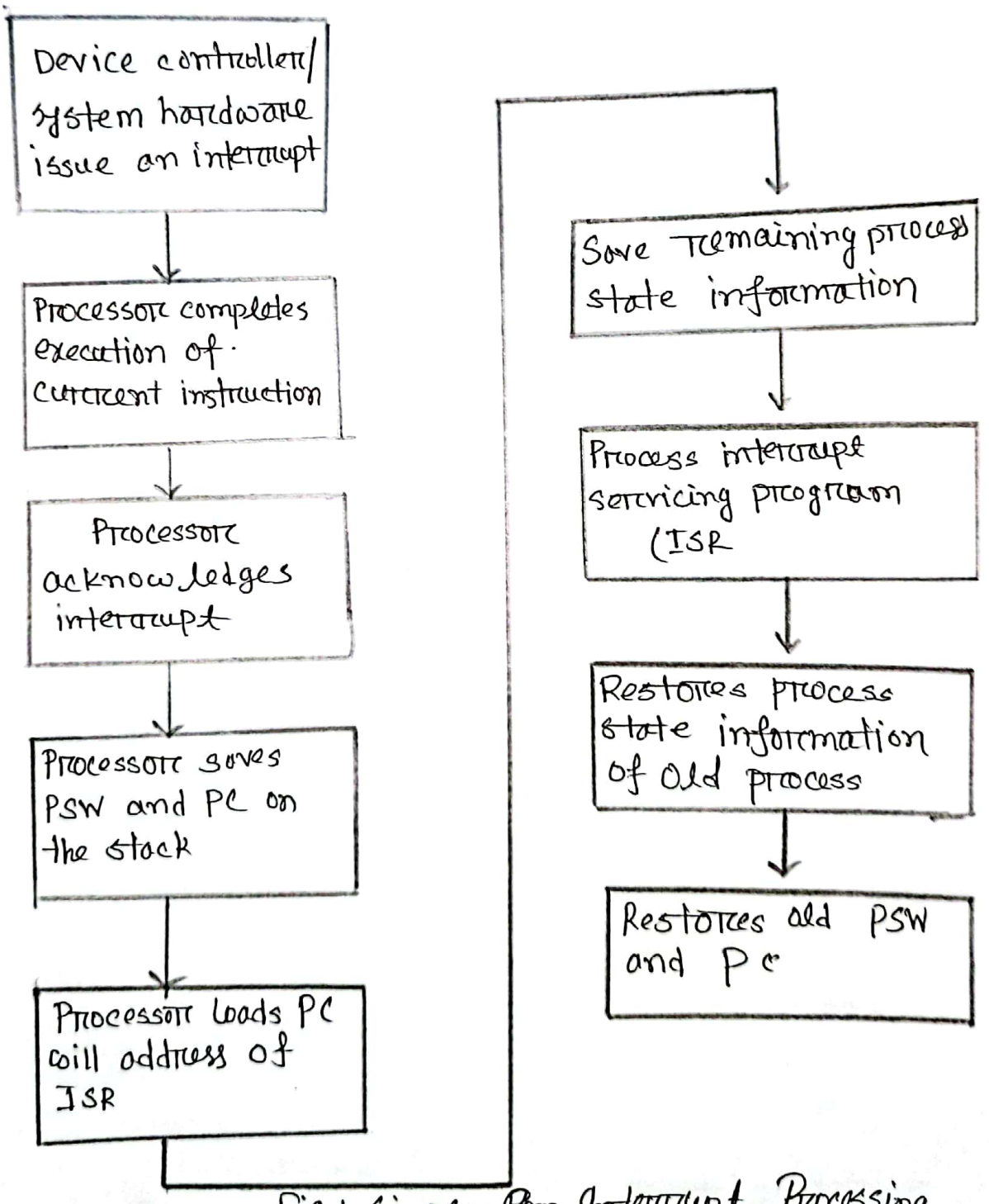


Fig: Simple Interrupt Processing.

06

When an I/O device completes an I/O operation, the below sequence of hardware events take place:

1) The device issue an interrupt signal to processor.

2) Processor completes execution of current instruction before responding to interrupt.

3) Processor tests for interrupts and sends an acknowledgement signal to device that issued interrupt.

4) The minimum information needed to be stored for task being currently executed before CPU starts executing interrupt routine are:

(a) status of processor that in contained in register known as program ~~that is~~ status word (PSW)

(b) Location of next instruction to be executed, of currently executing program that is contained in Program counter (PC).

5) Processor now loads PC with entry location of interrupt-handling program which will respond to this interrupt condition. Once PC has been loaded, processor proceeds to execute next instruction, which is the next instruction cycle that begins with an instruction fetch. Since the instruction fetch is determined by contents of the PC, result is that control is transferred to interrupt handler program. The execution result in the subsequent operations.

6) PC and PSW relating to interrupted program have already been saved on system stack. Additionally the contents of processor registers are also needed to be saved on stack which are used by called interrupt servicing Routine since these registers may be modified by interrupt-handler.

7) Interrupt handler next processes interrupt. The

8) When interrupt processing is finish, saved register values are retrieved from stack and restored to registers that are displayed in figure.

9) Final step is to restore values of PSW and PC from stack. Consquently, the instruction to be executed will be from previously interrupted program.