**Com 300: Computer architecture:**

What we are going to cover.

1. Classification schemes.
2. Functional units.
3. Bus systems.
4. I/O systems.
5. storage systems.
6. Instructions sets.
7. Microprogramming.
8. Survey of different kinds of computer architecture.
9. Software influence on computer architecture.

# **Introduction.**

**Computer architecture** is the study on the way hardware components are connected together to form a computer system.

* computer architecture explains what the computer does.

**Computer organization** is concerned with the structure and the behavior of a computer system as seen by the user. This explains how the computer does things.

## **Structure of a basic computer system.**

**A computer** can be defined as a fast electronic calculating machine that accepts data (digitized input) processes it as per a list of internally stored instructions and procedures then finally gives out information.

## **Types of computers**

1. **Personal computer (pc).**

This is the most common type of a computer found in homes, schools, businesses, offices etc. It is the most common type of desktop computers with processing and storage units along with the various input and output devices.

1. **Notebook computers.**

These are compact and portable version of the personal computers. E.g., laptops and tablets.

1. **Work stations.**

These are computers with high resolution input output graphics capability but with the same dimensions as of that of a desktop computer.

They are used in engineering applications and architecture that entails interactive design work.

1. **Enterprise systems.**

These are used business data processing in medium to large co-operations that require much more computing power and storage than work station. These systems are usually bigger than the PC’s and are commonly referred to as servers. Servers have become dominant in the world wide web as a source of all types of information.

1. **Super computers.**

These are used for large scale numerical calculations required in applications like weather forecasting, research etc.

# **Functional units of a computer**

A computer consists of 5 functionally independent main parts. These parts are:

1. Input unit.
2. Memory Unit.
3. Arithmetic and Logic unit.
4. Output Unit.
5. Control unit.

## **1.Input unit**

Source programs (high level language programs) and data is fed into a computer using a computer through input devices such as keyboard, joysticks, trackballs, mouse, scanners etc.

With respect to a keyboard:

Whenever a key is pressed, one corresponding word or number is translated into its equivalent binary code and fed into the main memory or processor via a wire or wireless connection.

## **2.memory unit**

The purpose is to store data. There are two types of memory:

* **Primary memory** – These are exclusively associated with the processor and are very fast. Only programs which are being executed are stored in the primary memory. The memory contains a large number of semi-conductor storage cells. Each cell is capable of storing one bit of information. The cells are usually processed in a group of fixed sizes called a word. To provide easy access to a word in memory, a distinct address is associated with each word location. Addresses are therefore numbers that identify memory locations. The number of bits in each word is called the word length. Instructions and data can be written or read out of memory under the control of a processor. Instructions must reside in memory during execution.The memory in which any of its locations can be reached in a short and fixed amount of time after specifying its address is known as **Random Access Memory (RAM).**

The time required to access one word is the **Memory Access Time.**

Memory that is only readable by the user and whose contents cannot be modified is called the **Read Only Memory (ROM).**

Cache are small fast Ram unit which are found in the processor and are often contained in the same IC chip to archive high performance. Primary storage is expensive.

* **Secondary memory.** Secondary memory is used where large amount of data and programs have to stored. Examples include; Magnetic tapes, optical disks, floppy disks.

## 3. Arithmetic logic unit

Most computer operators are executed in the ALU part of the processor. These operations include addition, subtraction, division and multiplication. The operands are brought into the ALU from the memory and stored in high speed storage elements called registers. According the instructions, the operation is performed in the required sequence. The control unit and the ALU are many times faster than other devices connected to the computer thus enabling a single processor to control external devices such as keyboards, displays, magnetic and optical disks and other mechanical controllers.

## 4. Output unit

Output units are the counterparts of the input units and their basic function is to send processed results into the outside world. Examples: Printer, speaker, monitor, projector.

## 5. control unit

This is effectively the nerve center that sends signals to other units and senses back their states. The actual timing signals that govern the transfer of data between input unit, processor, memory and output units are generated by the control unit.

# Basic operational concepts of a computer

To perform a given task, an appropriate program consisting of a list of instructions is stored in memory. Individual instructions are brought from the memory into the processor which then executes the operation. Example:

Add LOCA, R

This instruction adds the operand at memory location *LOCA* to the operand in register *R.* It then finally places the result into the register R. This instruction is performed in several steps.

1. The instruction is fetched from the memory into the processor.
2. The operand at LOCA is fetched and added to the content at R.
3. The resulting sum is stored in register R.

Transfers between the memory and the processor are started by sending the address of the memory location to be accessed to the memory unit and issuing the appropriate control signals. The data is then transferred to and from the memory.

MAR

PC

MAR

MDR

R0

R1

R…N

CONTROL UNIT

ALU

Figure 1. connection between processor and memory

The figure above shows how memory and the processor are connected. In addition to the control and the ALU, the processor contains a number of registers used for several different purposes. These registers include:

* **IR (instruction register):**  Holds instructions that are currently being executed. Its output is available for the control circuits which generates the timing signals that controls the various processing elements in execution of instructions.
* **PC (program counter):** This keeps track of execution of a program. PC usually contains memory address of the next instruction to be fetched and executed.

The other two registers that facilitate communication with memory are:

* **MAR (memory address register):** MAR contains the address of the location to be accessed.
* **MDR (memory data register):** This contains the data to be written into or read out of the address location.

The operating steps during execution include:

1. Programs reside in the memory and usually obtained through the input output unit.
2. Execution of a program starts when the program counter is set to point at the first instruction of the program.
3. Contents of the Program counter are transferred to the MAR and a Read control signal is send to the memory.
4. After the time required to access the memory location lapses, the address word is read out of the memory and loaded into the MDR.
5. The contents of the MDR are then transferred into IR(instruction register) making the instruction ready to be decoded and executed.
6. If the instruction involves an operation by the ALU, then it is necessary to obtain the required operands.
7. An operand in the memory is fetched by sending its address to the MAR and initiating a READ cycle.
8. When the operand has been read from memory to MDR, it is then transferred to the ALU.
9. After one or two repeated cycles, the ALU can then perform the desired operations.
10. If the result of the operation is to be stored in memory, the result if then send to the MDR.
11. Address of the location where the result is to be stored is send to the MAR and a WRITE cycle is initiated.
12. The contents of the program counter are incremented such that it points to the next instruction to be executed.

Normal execution of a program may be preempted (temporarily interrupted) if some devices require urgent servicing. To do this, the device that requires urgent action raises an interrupt signal. An interrupt signal **is a request from an I/O device for service by the processor.**  The Processor provides the requested service by executing an appropriate interrupt service routine. This diversion may change the internal stag(t)e of the processor forcing it to save its state in a memory location before the interruption. When the interrupt routine service is completed, the state of the processor is restored so that the interrupted program may continue its execution.

# BUS STRUCTURE

A bus is the simplest and most common way of interconnecting various parts of a computer. To archive reasonable speeds of operation, a computer must be organized so that all its units can handle at least one full word of data at a given time. In addition to the lines that carry data, the bus must have lines for address and control. The simplest way to interconnect various parts of a computer is by using

**single bus.**

INPUT

MEMORY

PROCESSOR

OUTPUT

COMMON BUS SINGLE BUS

A single bus structure can be for only one transfer at a time implying only two units can actively use the bus at any given time. Bus control lines are therefore use to arbitrate multiple requests for the use of the bus.

**Advantages of a single bus structure.**

* Low cost.
* Very flexible for attaching peripheral devices.

**Multiple bus structures**

These exists to increase the performance but they also increase the cost significantly. All interconnected devices do not all communicated at the same speed and time leading to a problem that is usually solved by using **cache registers i.e., buffer registers**: These are electronic registers of small capacity as compared to the main memory but are of comparable speeds. Instructions from the processor are loaded into these buffers and then once complete, the transfer of data as a whole occurs at a fast rate.

**Pipelining**

There are two basic techniques to increase the instruction execution rate of a processor.

* Increase the clock rate of the processor therefore decreasing the decryption time.
* Increase the number of instructions that can be executed simultaneously. (pipelining)

The idea of pipelining is to have more than one instruction been processed at the same time.

The success of a pipeline depends on dividing the execution of an instruction to a number of served units (stages) each performing part of the required operation.

A common division of instructions is to consider instruction fetch (IF), instruction decode (ID), operand fetch (OF), instruction execution (IE), storage of results (IS) as the serve tasks needed for the execution of an instruction. In this case, it is possible to have five instructions in a pipeline at the same time thus reducing instruction execution latency.

**Pipelining refers to** the technique I which a given task is divided into a number of subtasks, that need to be performed in a sequence.

Each subtask is performed by a given functional unit. The units are connected in serial fashion and all of them operate simultaneously. The use of a pipeline improves the performance compared to traditional sequential execution of tasks.

*See the diagram in the next page….*

F1

D1

E1

W1

F2

D2

E2

W2

F3

D3

E3

W3

F1

D1

E1

W1

F2

D2

E2

W2

F3

D3

E3

W3

1 2 3 4 5 6 7

I1

I2

I3

The figure above shows an illustration of the difference between executing four subtasks of a given instruction.

In the sequential processing, 12 units of time required to complete the execution of all instructions where as only 6 units are used in pipelining.

A Gant chart is used to formulate some performance measures for the goodness of a pipelining in processing a series of tasks. In developing a Gant chart, we can assume that the time taken by each subtask is the same.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | F3 | D3 | E3 | W3 |
|  | F2 | D2 | E2 | W2 |  |
| F1 | D1 | E1 | W1 |  |  |

0 1 2 3 4 5 6

From the above Grant chart, three performance measures for goodness of pipelining can be drawn.

The performance measures are:

1. Speed-up S(n).
2. Throughput U(n).
3. Efficiency E(n).
4. **Speed-up S(n).**

As can be seen from above, n + m – 1 time units are required to complete m tasks. ( considering m as the number of tasks [instructions] and n as the number of stages or units in a pipeline.

Speedup = Time using sequential processing

Time using pipeline processing

= m \* n \* t

(n+m-1) \* t

1. **Throughput U(n).**

U(n) = number of tasks executed per unit time

U(n) = m

(n+m-1) \* t

1. **Efficiency E(n)**

E(n) = ratio of actual speedup to maximum speed up.

E(n) = speed-up

n

The above analysis simply ignores important aspects that can affect the performance of a pipeline creating a pipeline stall. For example, an instruction can encounter a cache miss when fetching an operand from the memory. A pipeline is said to have stalled if one unit (stage) requires more time to perform its function therefore causing other stages to become idle.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | F3 | D3 |  |  |  | E3 | W3 |  |
|  | F2 | D2 |  |  |  | E2 | W2 |  |  |
| F1 | D1 | E1 |  |  | W1 |  |  |  |  |
| 0 | 1 | 2 | 3 | 4  Units of time | 5 | 6 | 7 | 8 | 9 |

Assuming the execution of instruction 1 encounters a cache miss and requires 2 extra units of time to complete execution, the entire pipeline will stall for two units of time creating a bubble.

Pipeline hazards (stalling) occurs for a number of reasons including;

* Instruction Dependency
* Data Dependency.