Faiz Akram (PhD)

Subject and code: Real Time and Embedded Systems CoSc-3026

3rd Year, B.Sc in Computer Science

Learning objectives:

After studying this unit, you will be able to:

- Explain the definition of embedded system and real time system
- Describe the characteristic of embedded system and real time system
- Discuss about the model of real time system
- Discuss some examples of real time system and types of real time system
- Explain the advantages and disadvantages of embedded and real time system
- Know about modeling timing constraints
- Discuss About computer organization concepts and memory
- Explain about design process

CHAPTER-1
INTRODUCTION

System: A system is a way of working, organizing or doing one or many tasks according to a fixed plan, program or set of rules.

A system is also an arrangement in which all its units assemble and work together according to the plan or program.

System examples:
Watch: It is a time display SYSTEM
Parts: Hardware, Needles, Battery, Dial, Chassis
and Strap

INTRODUCTION……CONTD

Rules

1. All needles move clockwise only
2. A thin needle rotates every second
3. A long needle rotates every minute
4. A short needle rotates every hour
5. All needles return to the original position after 12 hours

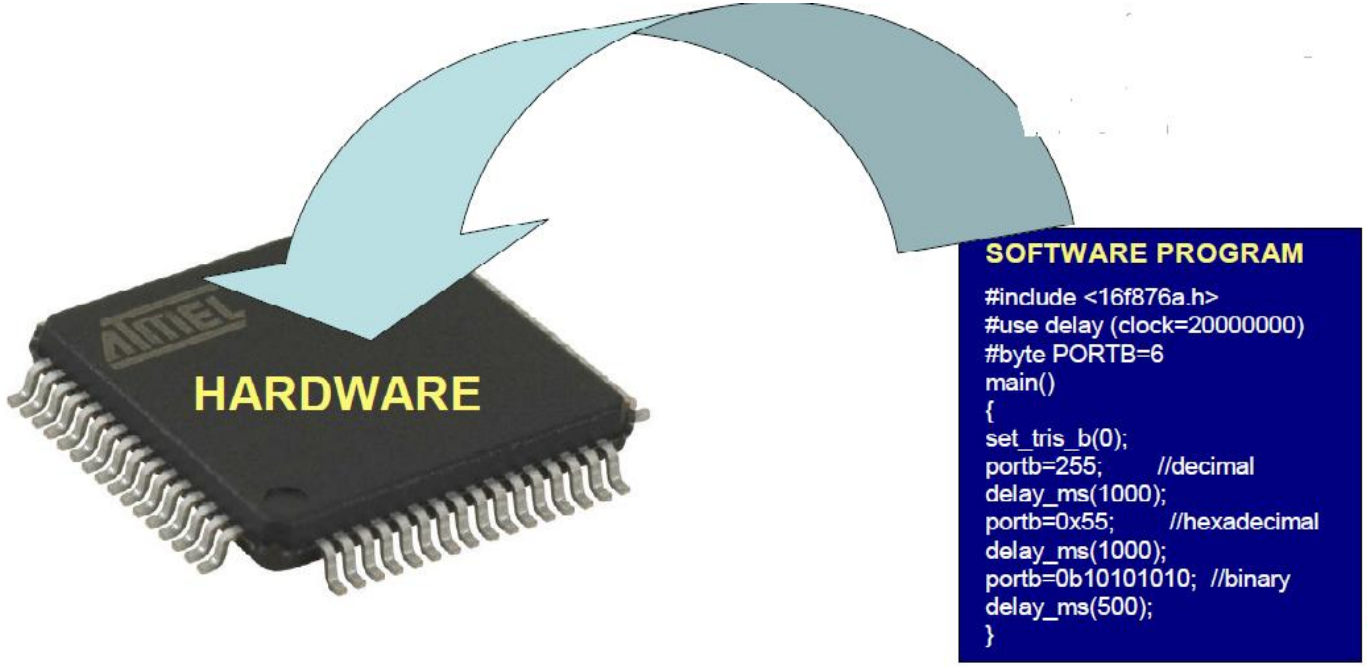
An **embedded system** is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, **a** fire alarm is an embedded system; it will sense only smoke. So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

A specialized computer system which is dedicated to a specific task.

INTRODUCTION….CONTD

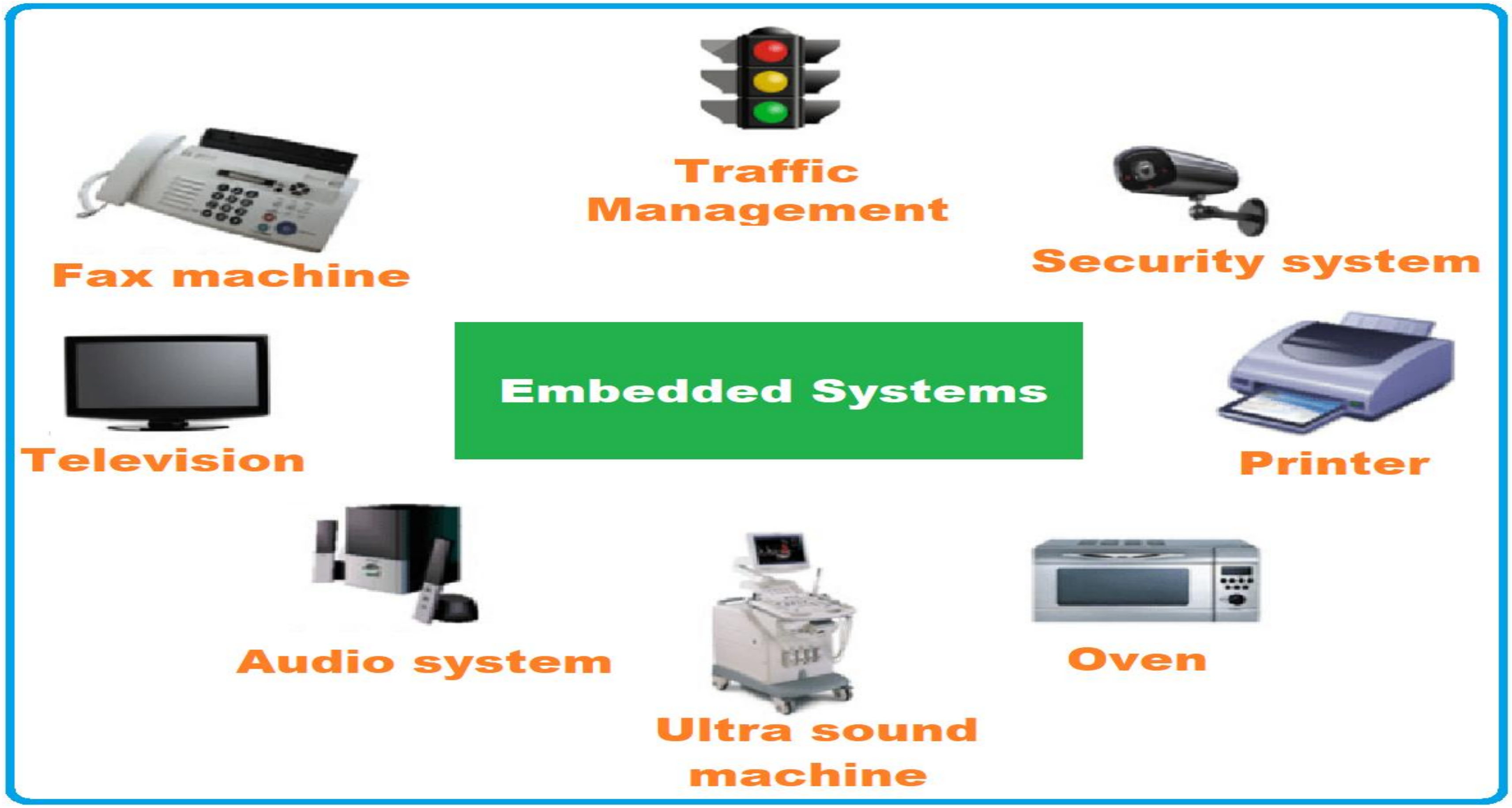
Definition: An Embedded System is one that has computer hardware with software embedded in it as one of its important components.

Its software embeds in ROM (Read Only Memory). It does not need secondary memories as in a computer



```
#include <16f876a.h>
#use delay (clock=20000000)
#byte PORTB=6
main()
{
set_tris_b(0);
portb=255;        //decimal
delay_ms(1000);
portb=0x55;       //hexadecimal
delay_ms(1000);
portb=0b10101010; //binary
delay_ms(500);
}
```

INTRODUCTION….CONTD(EXAMPLE)



Example of Real time embedded systems

INTRODUCTION….CONTD

An ATM is an embedded system which utilizes a crowded computer to set up a network between a bank computer and an ATM itself. It also has a microcontroller to bear both input and output operations.

An embedded system is a combination of hardware and software with some attached peripherals to perform a specific task or a narrow range of tasks with restricted resources.

An embedded system can be thought of as a computer hardware system having software embedded in it.

An embedded system has specific requirements and performs pre-defined tasks, unlike a general-purpose personal computer.

BLOCK DIAGRAM OF GENERIC EMBEDDED SYSTEM

The generic block diagram of an embedded system is shown in Figure 1.1. Every embedded system consists of certain input devices such as: key boards, switches, sensors, actuators; output devices such as: displays, buzzers(used to notify various events such as interrupt generation), sensors processor along with a control program embedded in the off-chip or on-chip memory, and a real time operating system (RTOS).

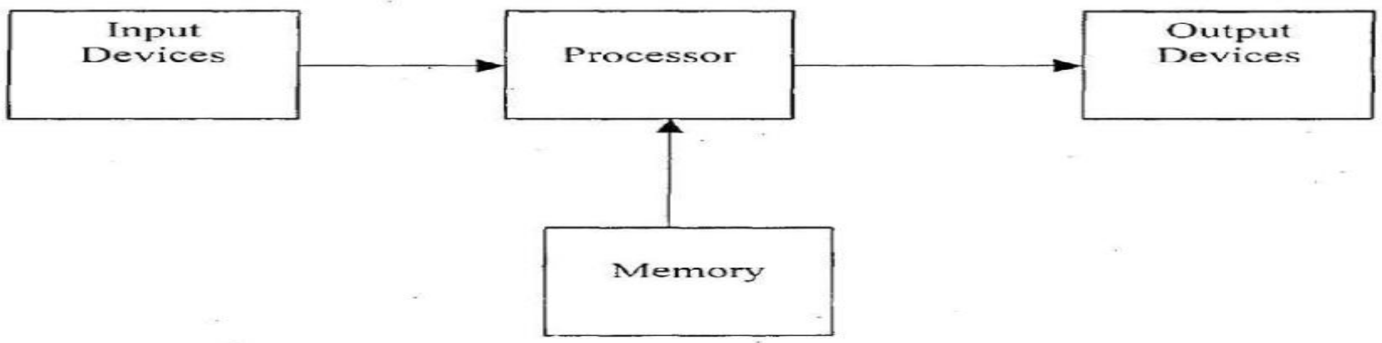


Figure 1.1: Block Diagram of a Generic Embedded System.

COMPONENT OF EMBEDDED SYSTEM

An embedded system has three components

- ✓ It has hardware
- ✓ It has application software
- ✓ It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS.

So we can define an embedded system as a Microcontroller based, software driven, reliable, real-time control system.

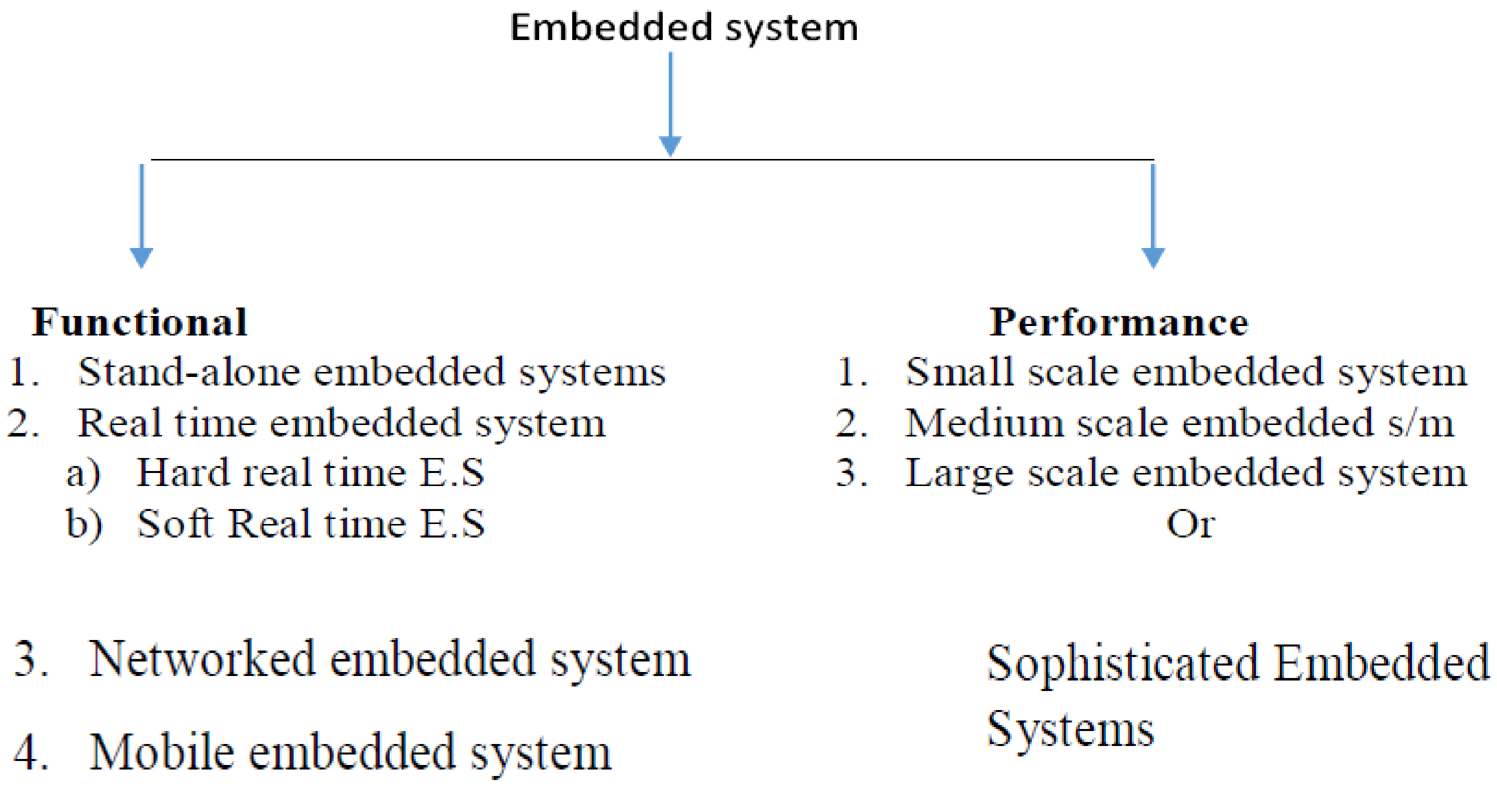
CHARACTERISTICS OF AN EMBEDDED SYSTEM

The important characteristics of an embedded system are:
Speed (bytes/sec) : Should be high speed
Power (watts) : Low power dissipation
Size and weight : As far as possible small in size and low weight
Accuracy (% error) : Must be very accurate
Adaptability: High adaptability and accessibility.
Reliability: Must be reliable over a long period of time.

So, an embedded system must perform the operations at a high speed so that it can be readily used for real time applications and its power consumption must be very low and the size of the system should be as for as possible small and the readings must be accurate with minimum error. The system must be easily adaptable for different situations.

CATEGORIES OF EMBEDDED SYSTEMS

Embedded systems can be classified into the following 4 categories based on their functional and performance requirements.



Embedded system

Functional

1. Stand-alone embedded systems
2. Real time embedded system
   a) Hard real time E.S
   b) Soft Real time E.S
3. Networked embedded system
4. Mobile embedded system

Performance

1. Small scale embedded system
2. Medium scale embedded s/m
3. Large scale embedded system

Or

Sophisticated Embedded Systems

CATEGORIES OF EMBEDDED SYSTEMS….cont

Real-time embedded systems:

An embedded system which gives the required output in a specified time or which strictly follows the time deadlines for completion of a task is known as a Real time system i.e. a Real Time system, in addition to functional correctness, also satisfies the time constraints .

There are two types of Real time systems. (i) Soft real time system and (ii) Hard real time system.

Soft Real-Time system: A Real time system in which, the violation of time constraints will cause only the degraded quality, but the system can continue to operate is known as a Soft real time system. In soft real-time systems, the design focus is to offer a guaranteed bandwidth to each real-time task and to distribute the resources to the tasks.

Ex: A Microwave Oven, washing machine, and TV remote etc.

CATEGORIES OF EMBEDDED SYSTEMS….cont

Hard Real-Time system: A Real time system in which, the violation of time constraints will cause critical failure and loss of life or property damage or catastrophe is known as a Hard Real time system. These systems usually interact directly with physical hardware instead of through a human being.

Ex: Deadline in a missile control embedded system , Delayed alarm during a Gas leakage ,car airbag control system , A delayed response in pacemakers ,Failure in RADAR functioning etc.

CLASSIFICATIONS OF EMBEDDED SYSTEM

1. Small Scale Embedded System:

•Single 8 bit or 16bit Microcontroller.

• Little hardware and software complexity.

• They May even be battery operated.

• Usually "C" is used for developing these system.

• The need to limit power dissipation when system is running continuously.

Programming tools: Editor, Assembler and Cross Assembler

2. Medium Scale Embedded System:

Single or few 16 or 32 bit microcontrollers or Digital Signal Processors (DSP) or Reduced Instructions Set Computers (RISC).

Both hardware and software complexity.

CLASSIFICATIONS OF EMBEDDED SYSTEM….contd

Programming tools:
RTOS, Source code Engineering Tool, Simulator, Debugger and Integrated Development Environment (IDE).

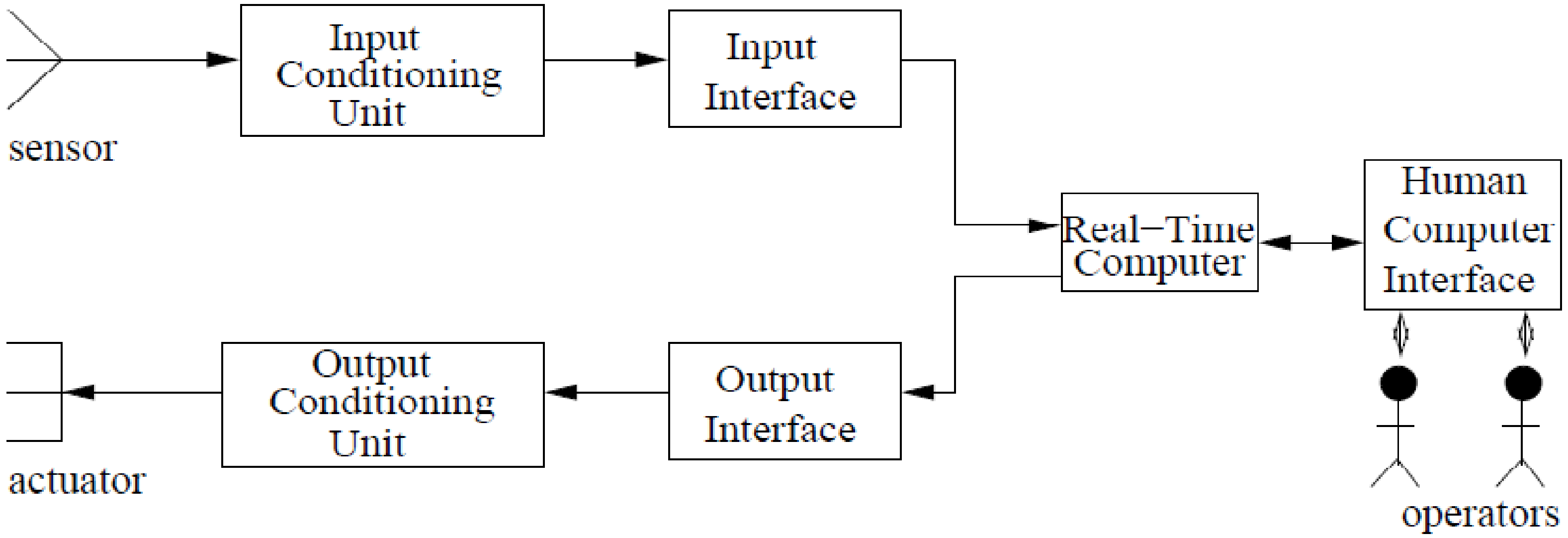
3. Sophisticated Embedded System:

- Enormous hardware and software complexity
- Which may need scalable processor or configurable processor and programming logic arrays.
- Constrained by the processing speed available in their hardware units.

Programming Tools:
For these systems may not be readily available at a reasonable cost or may not be available at all. A compiler or retarget able compiler might have to be developed for this.

A BASIC MODEL OF A REAL TIME EMBEDED SYSTEM

Real-time system includes various hardware and software embedded in a such a way that the specific tasks can be performed in the time constraints allowed. The accuracy and correctness involved in real-time system makes the model complex.



BASIC MODEL OF RTOS

Timing constraints

Before going to Modelling time constraints, first let us understand about timing constraints,

✓Constraints imposed on the timing behaviour of a job

✓Each unit of work that is schedule and execute by system is job

✓Set of related job which jointly provide some system function is a task

- Release time (when job is available for execution)
- Deadline (it should be get executed within that respective time)
- Completion time(Early or late with respect to the deadline)
- Response time (time between release time and completion time)
- Relative deadline(It is nothing but the maximum allowable response time of a job)
- Absolute deadline(It is nothing but a deadline only and is equal to RT+RD)

Modelling timing constraints

Modelling time constraints is very important since once a model of the time constraints in a system is constructed, it can serve as a formal specification of the system. Further, if all the timing constraints in a system are modelled accurately, then it may even be used to automatically generate code from it.

Use of timing constraints?

Timing constraints are used to specify the timing characteristics of the design. Timing constraints may affect all internal timing interconnections, delays through logic and LUTs and between flip-flops or registers. Area constraints are used to map specific circuitry to a range of resources within the FPGA.

Type of timing constraints in every real time system?

The two general types of timing constraints are **global** and **path-specific**. Global timing constraints cover all paths within the logic design. Path-specific constraints cover specific paths.

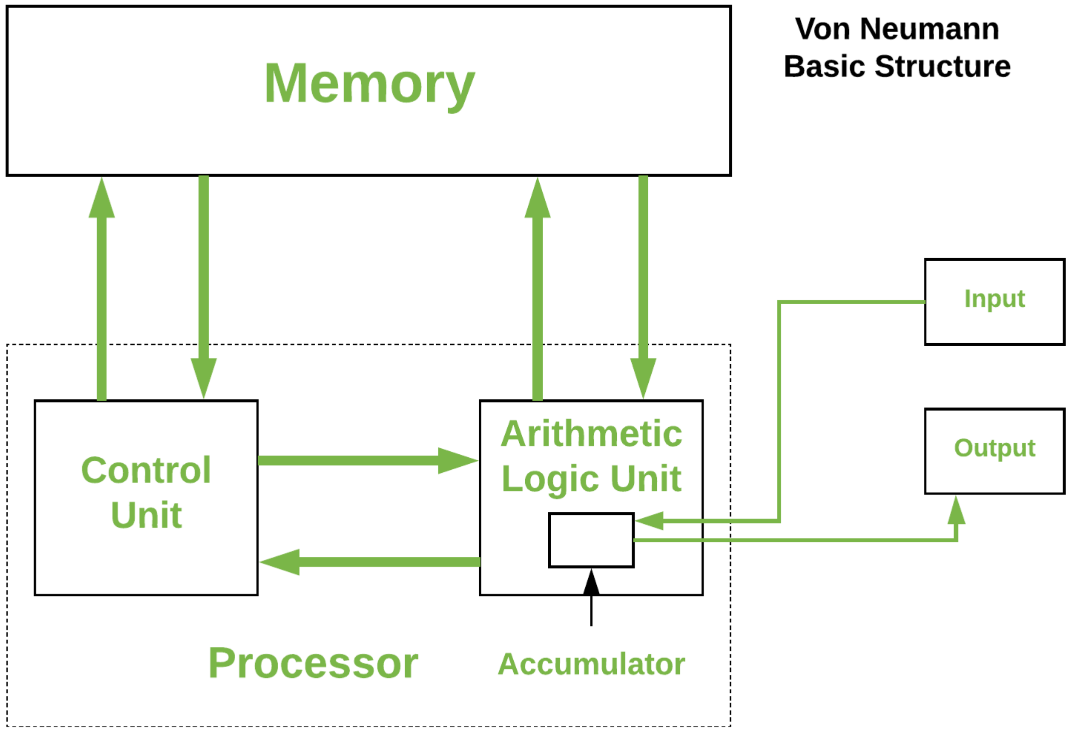
COMPUTER ORGANIZATION CONCEPTS AND MEMORY

The computer organization is concerned with the structure and behaviour of digital computers. It acts as the interface between hardware and software. It deals with the components of a connection in a system. Computer Architecture helps us to understand the functionalities of a system.

Examples of organization attributes include those hardware details transparent to the programmer, such as control signals, interfaces between the computer and peripherals, and the memory technology used. As an example, it is an architectural design issue whether a computer will have a multiply instruction.

A Computer has **five functional independent** units like Input Unit, Memory Unit, Arithmetic & Logic Unit, Output Unit, Control Unit.

COMPUTER ORGANIZATION CONCEPTS AND MEMORY



Computer Organization: Van Neumann architecture

MEMORY ….contd

Basic Operational Concepts:

✓Instructions take a vital role for the proper working of the computer.

✓An appropriate program consisting of a list of instructions is stored in the memory so that the tasks can be started.

✓The memory brings the individual instructions into the processor, which executes the specified operations.

✓Data which is to be used as operands are moreover also stored in the memory.

Example: Add LOCA, R0

✓This instruction adds the operand at memory location LOCA to the operand which will be present in the Register R0.

✓The above mentioned example can be written as follows:

```
Load LOCA, R1
Add R1, R0
```

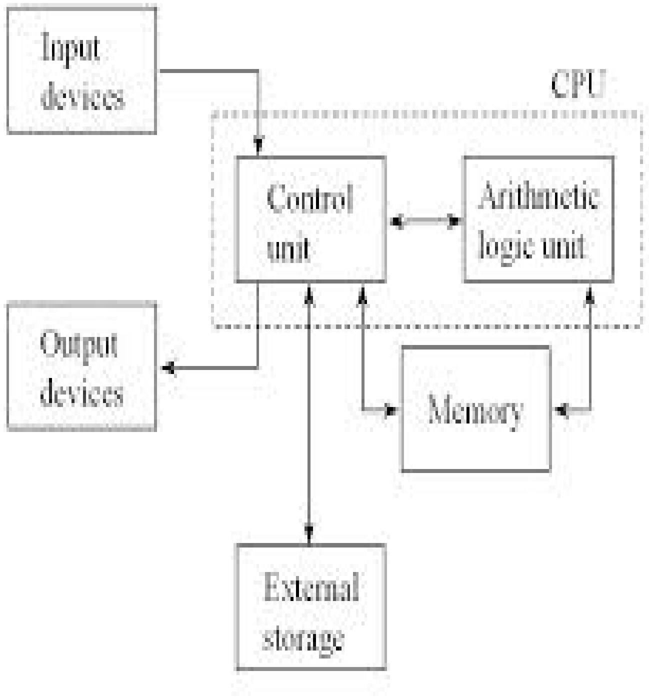
✓First instruction sends the contents of the memory location LOCA into processor Register R0, and meanwhile the second instruction adds the contents of Register R1 and R0 and places the output in the Register R1.

MEMORY ….contd

Performance :-

Performance means how quickly a program can be executed.

In order to get the best performance it is required to design the compiler, machine instruction set & hardware in a coordinated manner.

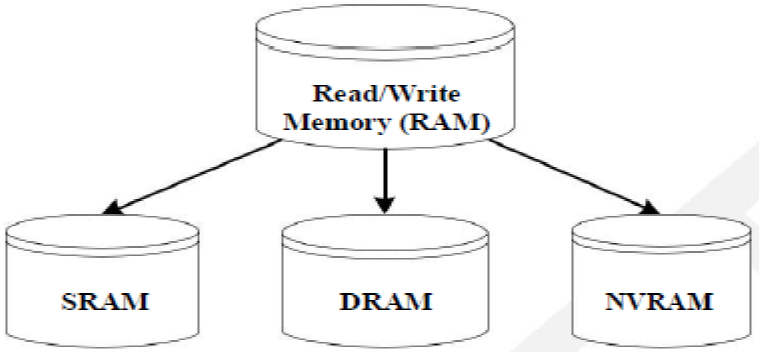


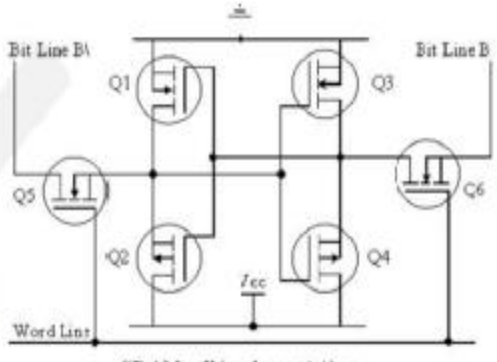
Connection B / W Processor & Memory

The above mentioned block diagram consists of the following components

MEMORY ….contd

RAM is a direct access memory, meaning we can access the desired memory location directly without the need for traversing through the entire memory locations to reach the desired memory position (i.e. Random Access of memory location).





SRAM cell implementation

1. Static RAM (SRAM):

✓Static RAM stores data in the form of Voltage.

✓They are made up of flip-flops

✓SRAM cell (bit) is realized using 6 transistors (or 6 MOSFETs).

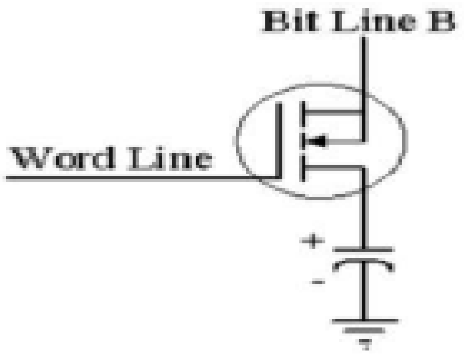
✓ Four of the transistors are used for building the latch (flip-flop) part of the memory cell and 2 for controlling the access.

✓ Static RAM is the fastest form of RAM available.

MEMORY ….contd

2. Dynamic RAM (DRAM)

- ✓ Dynamic RAM stores data in the form of charge.
- ✓The advantages of DRAM are its high density and low cost compared to SRAM
- ✓ The disadvantage is that since the information is stored as charge it gets leaked off with time and to prevent this they need to be refreshed periodically
- ✓ Special circuits called DRAM controllers are used for the refreshing operation. The refresh operation is done periodically in milliseconds interval



DRAM cell implementation

Memory selection for Embedded Systems

•Selection of suitable memory is very much essential step in high performance applications, because the challenges and limitations of the system performance are often decided upon the type of memory architecture.

•Systems memory requirement depend primarily on the nature of the application that is planned to run on the system.

•Memory performance and capacity requirement for low cost systems are small, whereas memory throughput can be the most critical requirement in a complex, high performance system.

Following are the factors that are to be considered while selecting the memory devices,

✓Speed
✓Data storage size and capacity
✓Bus width
✓Power consumption
✓Cost

EMBEDDED SYSTEM DESIGN PROCESS

The embedded system design process aimed at two objectives.

(A) First, it will give us an introduction to the various steps in embedded system design

(B) Second, it will allow us to consider the design methodology itself.

A design methodology is important for three reasons.

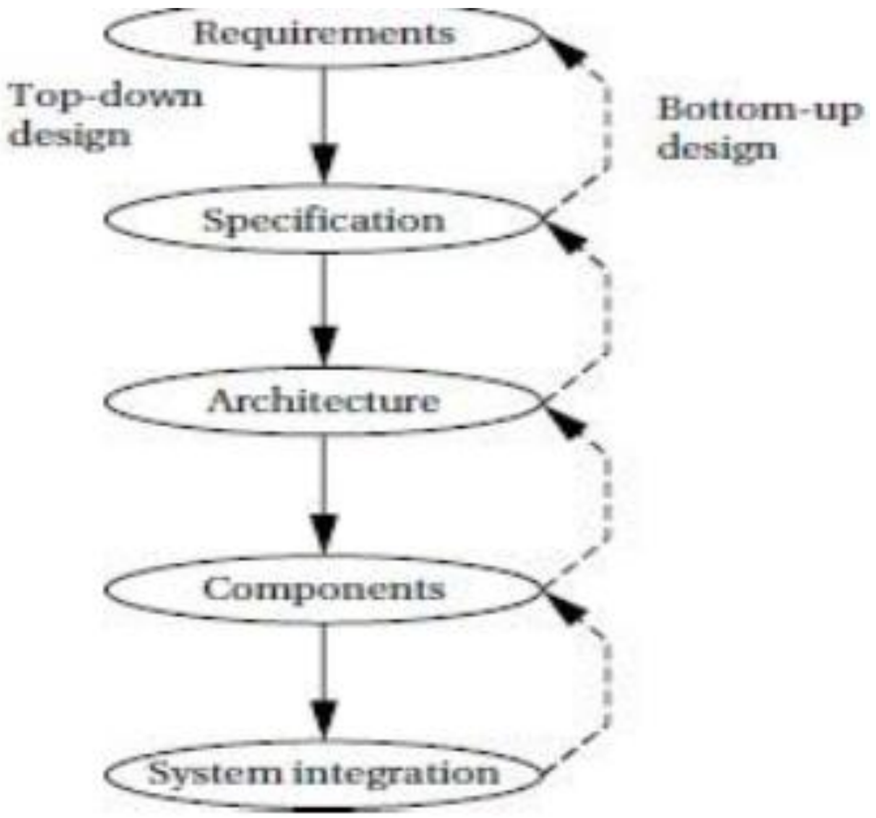
First, it allows us to keep a scorecard on a design to ensure that we have done everything we need to do, such as optimizing performance or performing functional tests.

Second, it allows us to develop computer-aided design tools.

Third, a design methodology makes it much easier for members of a design team to communicate.

The below Figure summarizes the major steps in the embedded system design process. In this top–down view, we start with the system requirements.

EMBEDDED SYSTEM DESIGN PROCESS….contd



Major levels of abstraction in the design process

DESIGN PROCESS OF EMBEDDED SYSTEM….contd

The traditional design approach has been traverse the two sides of the accompanying diagram separately, that is,

- ✓Design the hardware components
- ✓Design the software components
- ✓Bring the two together
- ✓Spend time testing and
- ✓Debugging the system

The major areas of the design process are

- ✓Ensuring a sound software and hardware specification.
- ✓Formulating the architecture for the system to be designed.
- ✓Partitioning the h/w and s/w.
- ✓Providing an iterative approach to the design of h/w and s/w

THANK YOU