ES chapter1

Embedded system design

What is an embedded computing system ?

- It is a system that includes programmable computer but is not a general purpose computing system , used for special purpose.(it has dedicated function)
- It controls the physical operation of the machine.

EX: Washing machines, Fax machines, Automobiles …etc

- Designers in many fields identify where processor can be used ,design a hardware platform with I/O devices and memory that can suited to application
- Design soft ware that performs required tasks
- Many of the challenges encountered are not only computer engineering problems

Ex : Can be mechanical or electrical problems.

- The **applications** of **embedded systems** include home appliances, office automation, security, telecommunication, instrumentation, entertainment, aerospace, banking and finance, automobiles

Characteristics of embedded computing applications:

Single-functioned: An embedded system usually executes only one program, repeatedly.

For example, a pager is always a pager.

In contrast, a desktop system executes a variety of programs, like spreadsheets, word processors, and video games, with new programs added frequently.

- Embedded computing system has to provide sophisticated functionality, Complex algorithms:

Algorithms are complex because operations performed are sophisticated.

EX: Processor that controls an automobile engine must perform complicated filtering operations that optimize the performance of the car.

- Tightly constrained:
- Embedded systems often must cost just a few dollars
- must be sized to fit on a single chip
- must perform fast enough to process data in real-time
- and must consume minimum power to extend battery life ,
- prevents the necessity of a cooling fan.

- Reactive and real-time: Many embedded systems must continually react to changes in the system’s environment, and must compute certain results in real time without delay.
- For example, a car's cruise controller continually monitors and reacts to speed
- It must compute acceleration or decelerations amounts repeatedly within a limited time;
- a delayed computation result could result in a failure to maintain control of the car.

- In contrast, a desktop system typically focuses on computations, with relatively infrequent (from the computer’s perspective) reactions to input devices.

- In addition, a delay in those computations, while perhaps inconvenient to the computer user, typically does not result in a system failure

Embedded computing operations must often be performed to meet dead lines.

Real time:

Many embedded computing system have to perform in real time , if the data is not ready by a certain dead line the system may break .

- Failure to meet dead lines is unsafe and can even endanger lives.

Multi rate: Many embedded computing system have to control several real time activities , running at the same time and also at different rates.

Example : Multimedia applications are the applications of multirate behavior . The audio video portions of a multimedia stream run at different rates but they must be synchronized.

If synchronization not attained either in audio or video portions spoils the presentation.

Manufacturing cost :

The total cost of designing and building the system is very important.

Manufacturing cost determined by processor used , the amount memory required and types of I/O devices used.

Finally ,most embedded computing system are designed by small teams to meet dead lines.

Challenges in embedded computing system design:

How much hardware do we need ?

To meet the performance deadlines and manufacturing cost constraints , the choice of hard ware is important , too little hardware system fails to meet dead lines , too much hard ware system becomes too expensive.

How do we meet dead lines ?

For meeting deadlines , system speed must be higher , so that programs runs fastly . which makes the processor expensive. Processor speed can be enhanced by increasing clock speed , this is not enough, memory speed also must be increased.

How do we minimize power consumption ?

In battery powered applications , power consumption is important. By running the system slowly will consume less power , slowing down the system may ,miss dead lines.

How do we design for upgradability:

The hard ware platform may be used over several product generations(or) no changes. However we want to be able to add features by changing software.

Complex testing:

Exercising an embedded system is difficult . Because we have to test it in real time.

Limited observability and controlability :

Embedded computing systems do not come with key boards and screens. This makes it more difficult to see what is going on and to affect system operations.

Restricted development environment:

Developing tools for embedded system are limited. Generally code is compiled on PC’s and downloaded on to embedded systems.

Performance of embedded computing system:

Embedded system designers must have clear performance goal in mind, their program must meet dead lines, because of real time computing .

Dead line is the time at which a computations must be finished.

If the program doesn’t produce the required output by dead line , program doesn’t work , even though o/p is generated later.

- To understand the real time behavior of an embedded system ,we have to analyze the system at several different levels of abstraction .

Layers include :

CPU: CPU clearly influence the behavior of the system , particularly when the cpu is a pipelined processor with a cache.

Platform:

The platform includes the bus and I/o devices . These peripheral components that surround the cpu are responsible for feeding the cpu and can affect its performance.

Program:

Programs are very large and the cpu sees only small window of the program at a time. We must consider entire program to determine its over all behavior.

- **Task** : we generally run several programs simultaneously on a cpu , creating a multi tasking system . The tasks interact with each other in ways that have profound implications on performance.
- **Multi processors**: Many embedded systems have more than one processor. The interactions between these processors adds more complexity to the analysis of over all system performance.

Embedded system design process:

Embedded system design process aimed at two objectives:

1. It will give an introduction to various steps in embedded system design.
2. It will allow us to consider the idea of design methodology

In top down view of embedded system design process, design start with system requirements specifications, architecture, components, system integration

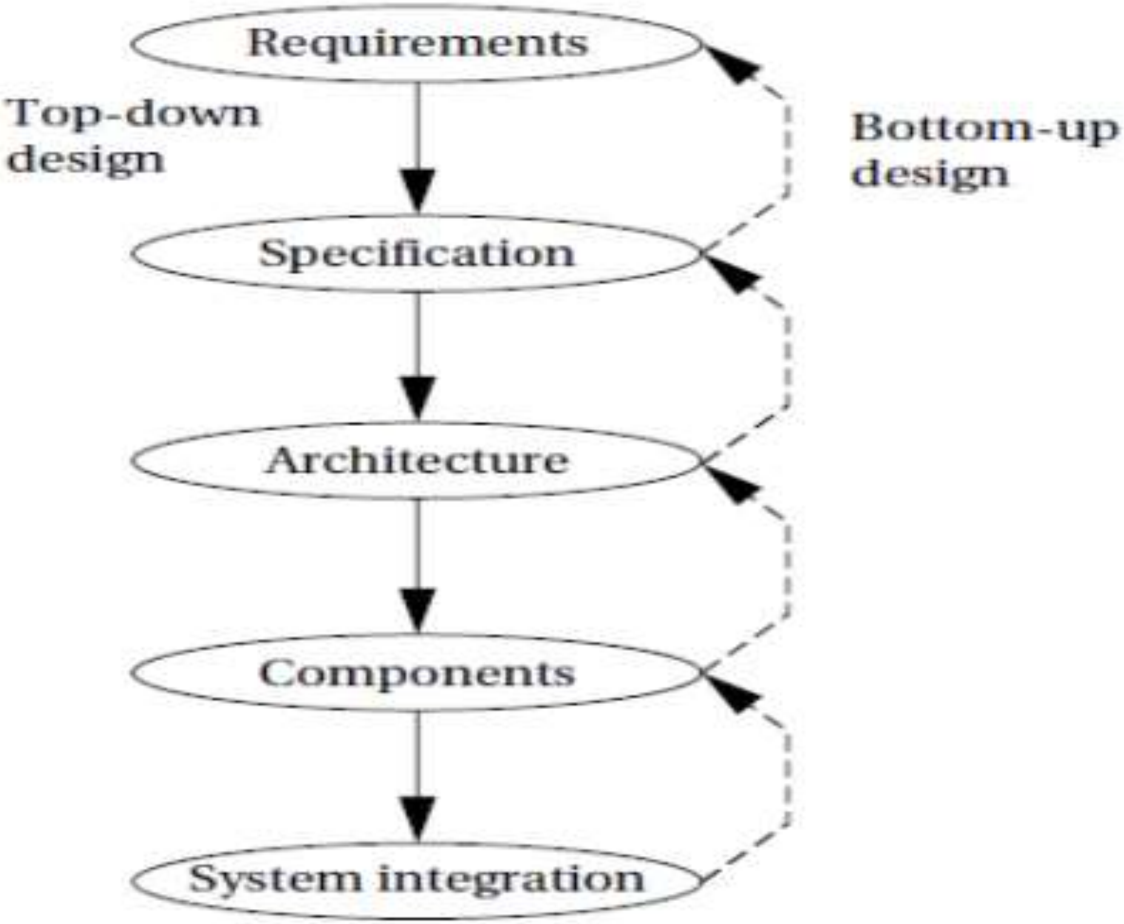

Fig 1.7 Major levels of abstraction in the design process.

Major goals of the design:

1. Manufacturing cost
2. Performance
3. Power consumption

(Refer wolf text book