Lecture 3.0

Real Time Embedded Systems

- Introduction

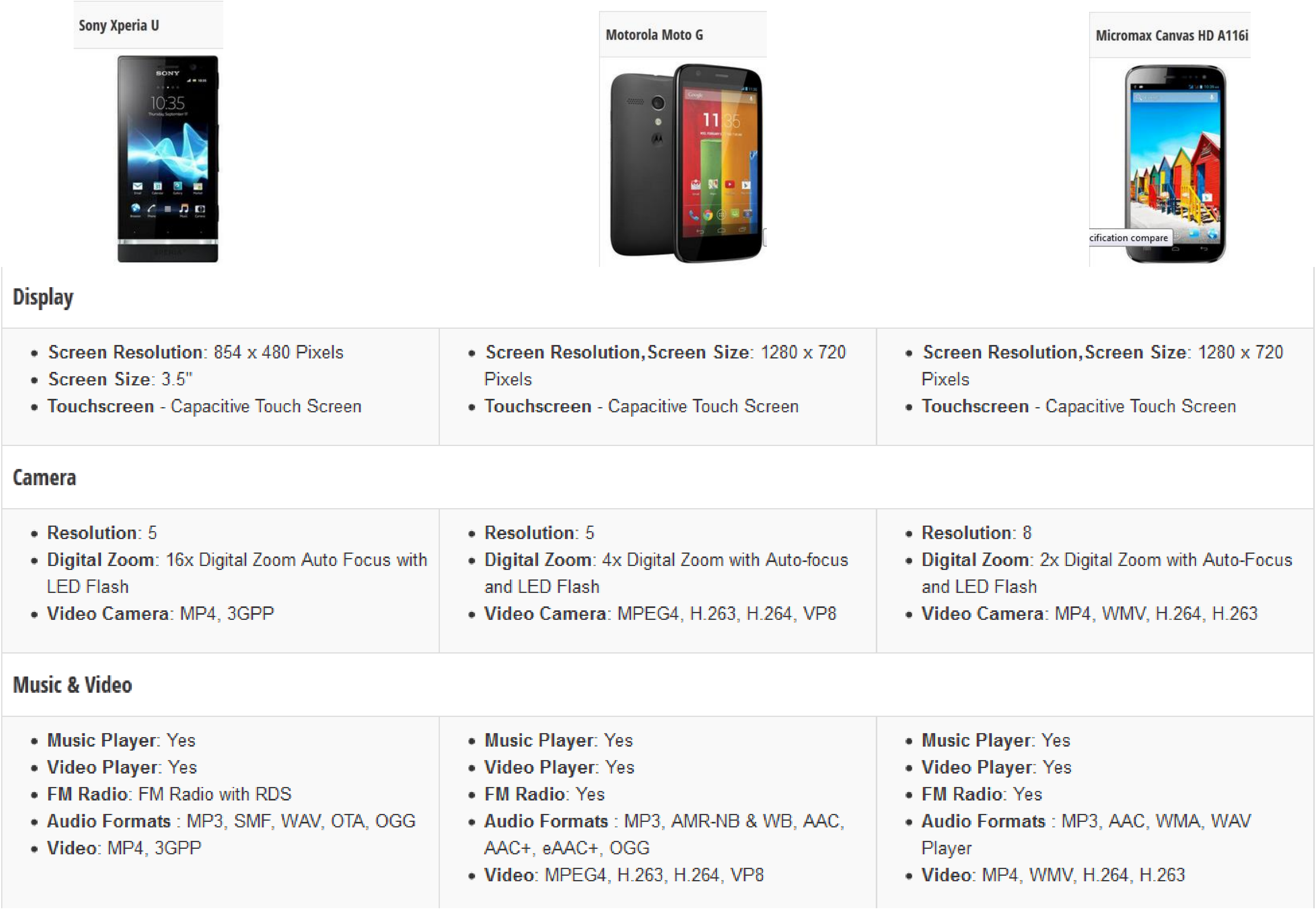
Outline

- What is Real Time?
- What is Real Time Embedded System?
- Application of Real Time Systems
- A Basic Model of Real Time Systems
- Characteristics of Real Time Systems

Books Referred . .

1. Real Time System Theory and Practice - **Rajib Mall (Ch. 1)**
2. Embedded System design: A unified hardware/software introduction by **Frank Vahid and Tony Givargis** *(Wiley Publishers)*
3. Internet resources
4. Embedded System Architecture, Programming and Design **Raj Kamal** *(Tata McGraw Hill Education Private Limited)*
5. Computers as Components Principles of embedded computing system design, by **Wayne Wolf** (*Morgan Kaufmann Publishers)*

What is Real Time?



Sony Xperia U	Motorola Moto G	Micromax Canvas HD A116i
Display		
- **Screen Resolution**: 854 x 480 Pixels - **Screen Size**: 3.5" - **Touchscreen** - Capacitive Touch Screen	- **Screen Resolution, Screen Size**: 1280 x 720 Pixels - **Touchscreen** - Capacitive Touch Screen	- **Screen Resolution, Screen Size**: 1280 x 720 Pixels - **Touchscreen** - Capacitive Touch Screen
Camera		
- **Resolution**: 5 - **Digital Zoom**: 16x Digital Zoom Auto Focus with LED Flash - **Video Camera**: MP4, 3GPP	- **Resolution**: 5 - **Digital Zoom**: 4x Digital Zoom with Auto-focus and LED Flash - **Video Camera**: MPEG4, H.263, H.264, VP8	- **Resolution**: 8 - **Digital Zoom**: 2x Digital Zoom with Auto-Focus and LED Flash - **Video Camera**: MP4, WMV, H.264, H.263
Music & Video		
- **Music Player**: Yes - **Video Player**: Yes - **FM Radio**: FM Radio with RDS - **Audio Formats** : MP3, SMF, WAV, OTA, OGG - **Video**: MP4, 3GPP	- **Music Player**: Yes - **Video Player**: Yes - **FM Radio**: Yes - **Audio Formats** : MP3, AMR-NB & WB, AAC, AAC+, eAAC+, OGG - **Video**: MPEG4, H.263, H.264, VP8	- **Music Player**: Yes - **Video Player**: Yes - **FM Radio**: Yes - **Audio Formats** : MP3, AAC, WMA, WAV Player - **Video**: MP4, WMV, H.264, H.263

What is Real Time?

- This Functionality can only be achieved using very Flexible Device. The Heart of it is Embedded Processor.

- Mobile Phone does many task at the same time.

- Working within a time-constraint, i.e. it has to satisfy everyone with the minimum acceptable delay. We call this as to work in “**Real Time**”.

- The mobile telephone as a **“Real Time Embedded System” (RTES)**

What is Real Time Embedded System?

Real Time

"Real-time" usually means time as prescribed by external sources". It is **quantitative** notion of time. It should not be **qualitative** measurement.

Embedded (Embodiment)

"Embodied phenomena are those that by their very nature occur in real time and real space" . *A number of systems coexist to discharge a specific function in real time*

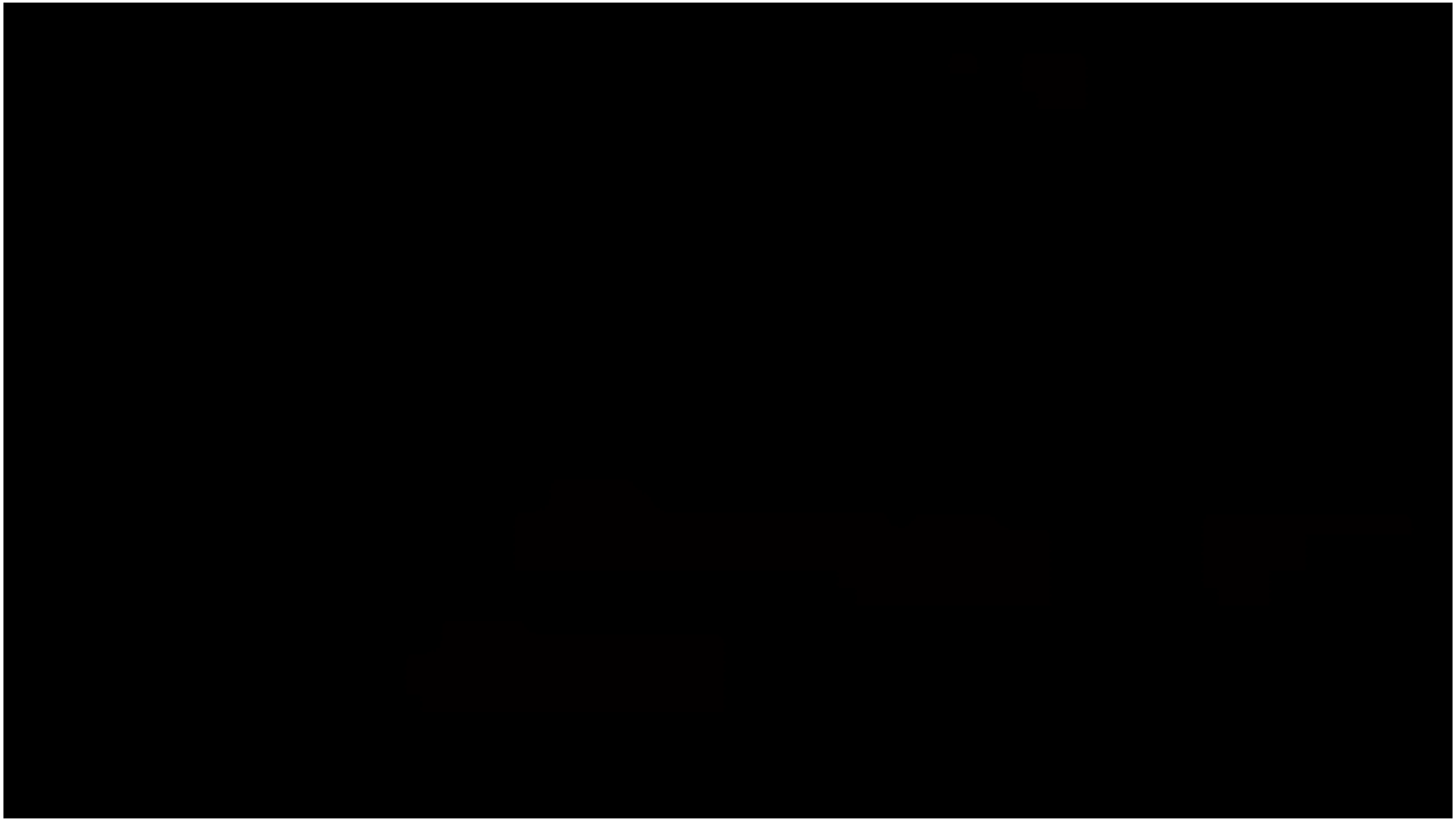
Real Time Embedded Systems

A Real Time Embedded System" (RTES) is precisely the union of subsystems to discharge a specific task coherently.

It is a system, when we need quantitative notion of time to describe the behaviour of the system.

Application of Real Time ES (RTES)

- **Automated Car Assembly Plant**



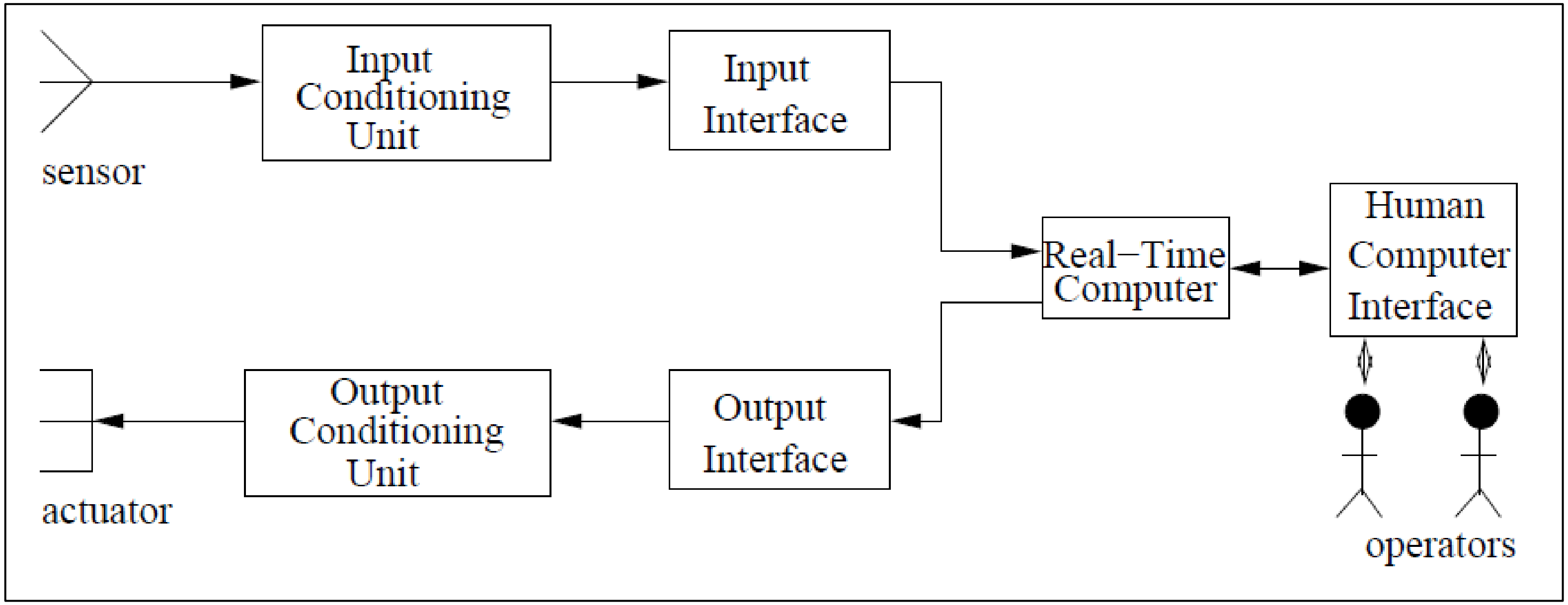
Application of Real Time ES (RTES)

- Industrial Application
  - Automated Car Assembly, Chemical Plant Control, supervisory control and data acquisition(SCADA)
- Medical
- Peripheral equipment
  - Laser Printer
- Automotive & Transport
- Telecommunication
  - A Cellular System, Cell Phones
- Aerospace
- Defence Application–Missile Guidance System, Antimissile System
- Consumer Application
- Computer on Board an Aircraft
- Miscellaneous Application
  - Railway Reservation System, Video Conferencing

Think – Pair - Share

- Design one Real Time Embedded System based on Applications discussed (Group Activity)

Basic Model of RTES



- **Sensor**
- **Actuators**
- **Signal Conditioning Unit**
  - Voltage Amplification
  - Voltage Level Shifting
  - Freq Range Shifting
  - Signal Mode Conversion
- **Interface Unit**
  - Analog to Digital Conversion
  - Digital to Analog Conversion
- **Real Time Computer**
- **Human Computer Interface**

Characteristics of RTES

- Time Constraints
- New Correctness Criterion
- Embedded
- Safety - Criticality
- Concurrency
- Distributed and Feedback Structure
- Task Criticality
- Custom Hardware
- Reactive
- Stability
- Exception Handling

Characteristics of RTES

1. Time Constraints

- Real Time task is associated with ***time constraints***, generally known as ***Deadline***.
- Deadline specifies the time before which the task must be complete and produce the results.
- Real Time Operating System ***ensures*** the task meet their respective timing constraints using ***task scheduling*** strategies.

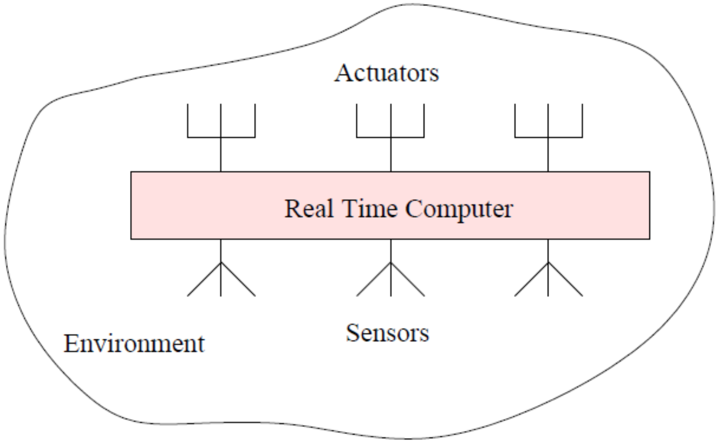
Characteristics of RTES

2. New Correctness Criteria

- Not Only ***Logical*** Correctness.
- But also ***in-time***
- Means, RTES should be able to produce the ***correct output on time*** as per the change in input condition.
- The Result Produced after the Deadline would be considered as incorrect ***result***.

Characteristics of RTES

3. Embedded



- RTES are generally ***Embedded in nature and control*** the physical environment on which it is EMBEDDED.

Characteristics of RTES

4. Safety-Criticality

- Traditional system safety and reliability are independent issue.
- A ***Safe System*** does not cause any damage even when it fails.
- A ***Reliable system***, is one that operate for long duration without exhibiting any failure.
- In RTES ***Safety & Reliability*** are generally ***bounded together***

Characteristics of RTES

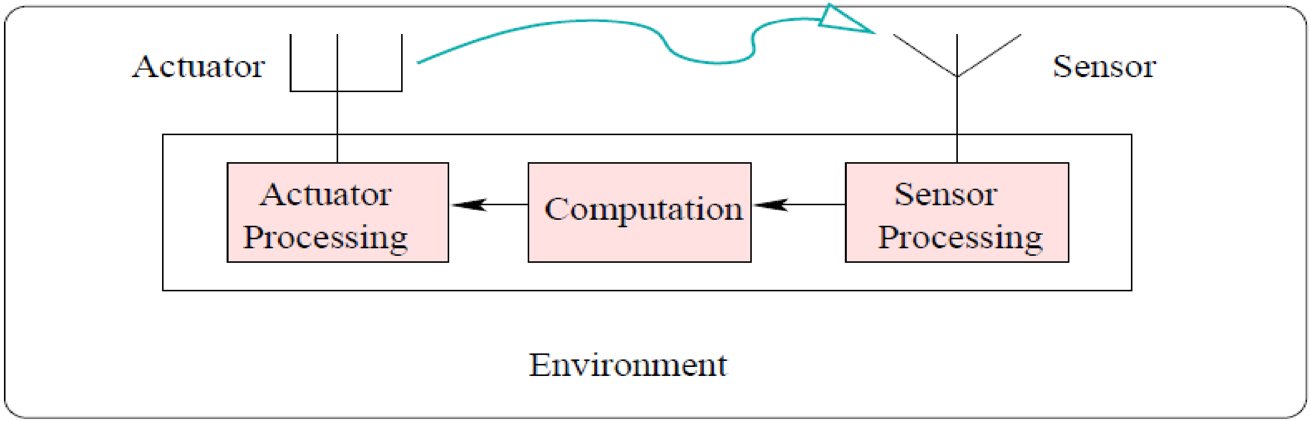
5. Concurrency

- An RTES should respond ***several Independent events*** within ***short and strict time bounds***. e.g. chemical plant automation system
- Chemical plant automation system *monitors the chemical reaction* and *control the rate of reaction* by *changing the different parameters* of reaction such as Temperature and Chemical reaction.

Characteristics of RTES

6. Distributed and Feedback structure

- May located in large geographical region (e.g. refinery)
- Processing at each node – Locally and send it to Central processing => Reduce the burden on BUS
- E.g. Petroleum refinery



Characteristics of RTES

7. Task Criticality

- ***Measure of the cost of failure of a task.***
- Task criticality is determined by examining ***how critical are the results*** produced by task for proper functioning of the system
- ***Higher the criticality*** of task, the ***more reliable it*** should be made
- In the event of failure of such task, immediate failure detection and recovery are important
- Task Priority ⇔ Task Criticality

Characteristics of RTES

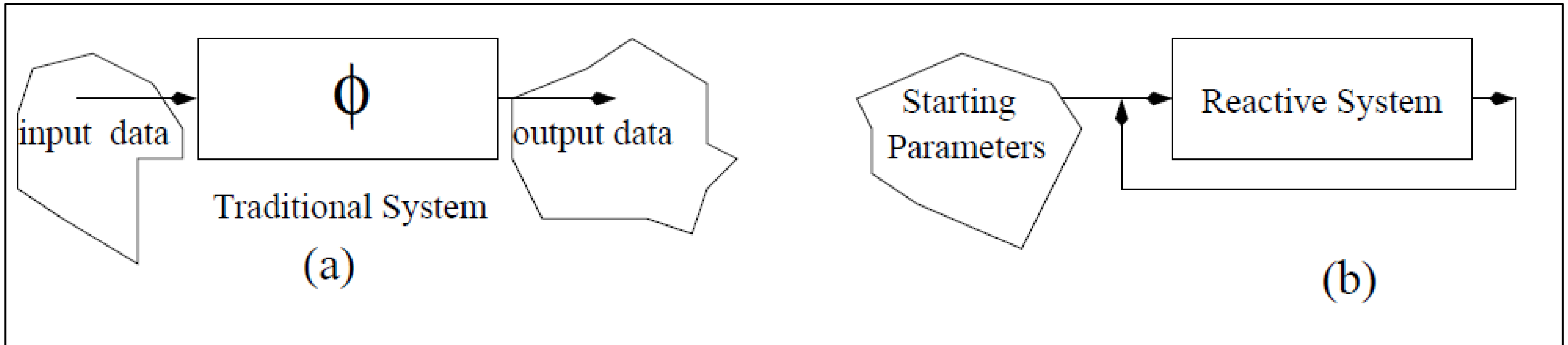
8. Custom Hardware

- RTES is often/normally implemented on ***custom hardware*** that is specifically designed and developed for the purpose
- e.g. – Cell Phone Processor – ***does not use tradition processor*** [e.g. 8051]
- e.g. – MPFI normally uses the ***16 or 32-bit processor*** having limited frequency of operation [40-100MHz] – not uses the processor like 1GHz.

Characteristics of RTES

9. Reactive

- RTES are often ***reactive***
- In which an ***on-going interaction*** between the computer and the environment is maintained
- Normal System *Out = f(Input)*
- Normal System [e.g. Library]
- Reactive System [computation in the reactive system is non-terminating]



Traditional versus Reactive Systems

Characteristics of RTES

10. Stability

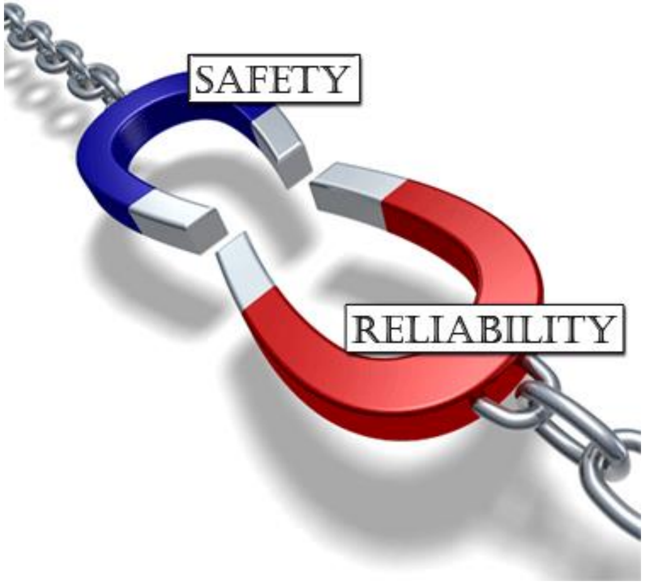
- Under Overload conditions, real-time system need to continue to meet the deadlines of the most critical task, though the deadlines of non-critical system is not met.

Characteristics of RTES

11. Exception Handling

- Many real-time systems work round the clock ***with out human*** operators [e.g. chemical plant]
- In this situation taking ***the corrective action on a failure*** becomes difficult

Safety and Reliability



Safety and Reliability

- **Traditional Systems** Normally Safety and Reliability **are independent issue**.
  - Reliable but unsafe
    - **A Hand Gun**
    - Unsafe: Handgun can misfire or explode and cause significant damage
    - Reliable: Handgun rarely fails
  - Unreliable but safe
    - **Word Processing Software**
    - Safe: software failure does not cause significant damage
    - Unreliable: It may not be very reliable.
- It proves Safety and Reliability are different issue for traditional system
- So, We can improve *Safety* of system without affecting *Reliability* and vice versa.

Safety and Reliability

- **For Real-Time System**
  - **Safety and Reliability are coupled together**
- They are not independent issue for real-time
- **Fail-Safe State**

  "A Fail-Safe State of a system is one which if entered when the system Fails, No damage would result"

- Example,
  - In word processing System, DOC being processed and saved on disk.
- If no damage can result if system enter into fail-safe state before it fails, then it is possible to turn an extremely unreliable and unsafe system into safe system.

Safety and Reliability

- Traditional System fail-safe state technique used to turn an unreliable system into safe system
- Example,
  - Consider Traffic Light controller
  - If it fails then known as highly unreliable
  - Though it can be considered safe….how?
    - It enters into fails-safe state after failing
    - Where all traffic light are **ORANGE** and blinking. Not **RED**/**GREEN**
- However in many real time system does not have fail-safe states.
  - Failure of system cause severe damage and this type of system said to be **safety-critical systems.**

"Safety-Critical System is one whose failure can cause severe damage"

Safety and Reliability

- **Safety-Critical System**
- Example,
  - ***on-board navigation system of an aircraft***
- It has no fail safe state
- Engine can not be Switched OFF
- For this system, issue of safety and reliability become interrelated and this system need to be **highly reliable**.
- In a safety-critical system, Safety can be only ensured **through increased reliability**

How to achieve reliability?

- High reliable software can be developed by adopting these three techniques:
  - **Error Avoidance**
  - **Error Detection and removal**
  - **Fault tolerance**

Error Avoidance

- Every Possibility of **occurrence of errors should be minimized** during product development
- How? :- Using well-founded software engineering practice, Sound design methodology, **adopting suitable CASE tools** and so on.

Error Detection and removal

- In Spite of error avoidance error may generate.
- It needs to be detected and removed.
- It can be **done by reviewing an testing**.
- Errors can easily fixed after detection.

Fault tolerance:

- Even after use of error avoidance and error detection techniques it is virtually impossible to make system error-free.
- **Errors cause failures**
- To achieve high reliability even after error in system, System should be able to tolerate a faults and compute the correct results.
- This is **fault tolerance**.
- It can be achieved by carefully incorporating redundancy.

Types of Real Time Tasks

Types of Real Time Tasks

Real Time class can be classified into three broad category (Depends on consequences of a task missing deadline)

1. Hard Real Time Tasks
2. Firm Real Time Tasks
3. Soft Real Time Tasks
4. Non-Real Time Tasks

Types of Real Time Tasks

Hard Real Time Task

- "constrained to produce its results within certain predefined time bound"
- System considered to be failed if system does not produce its result before specified time bound
- Example:
  - *Robot: needs to detect obstacle and respond it to very quickly, otherwise it would collide with it. And robot would considered to have failed.*

Types of Real Time Tasks

- Example,
  - *Anti missile system*
  - *It needs to detect another missile ,angle it to fire, and destroy missile before damage. All these tasks are hard real time*

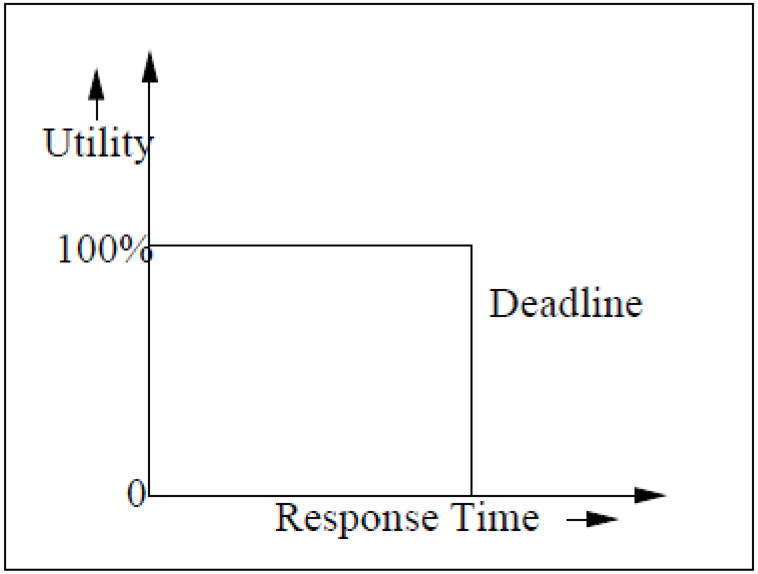
- Applications are having hard real time tasks are ***safety-critical***
- Means Failure of real time tasks would result in severe consequences
- This Makes hard real time tasks extremely critical
- Time bound

[***several micro sec ⇔ a few milli sec***]

Types of Real Time Tasks

Firm Real Time Task

- Task is associated with some predefined deadline before which it is require to produce its result.
- Unlike hard real time task, if it does not complete within deadline, **the system does not fail**.
- The late result **merely discarded**.
- Utility of results become zero after deadline



Types of Real Time Tasks

Firm Real Time Task

- Mostly used in multimedia application
- Example,
  - Video Conferencing
    - Video frames and audio frames converted into packets and transmitted to receiver.
    - Late arriving packets discarded at receiver because of its no use.
  - Satellite-based tracking of enemy movements
    - Satellite sends images of enemy territory to ground station.
    - If ground station is overloaded, new images are received before olds are processed.
    - So old images may discarded by ground station

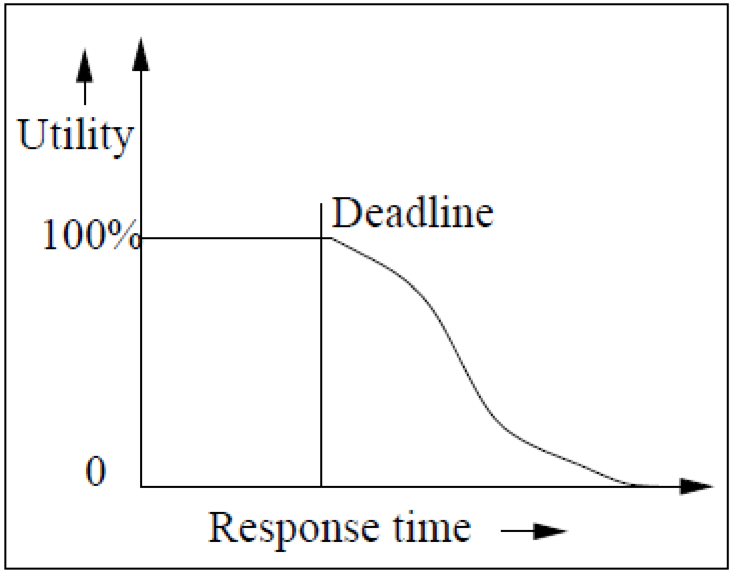
Time bound range from

[*few milli seconds ⇔ several hundred of milli seconds*]

Types of Real Time Tasks

Soft Real Time Task

- It also has time bound with it.
- Unlike above both real time tasks, these tasks are **not expressed as absolute values**.



- Instead constrains are expressed either in term of average response time are required.

Types of Real Time Tasks

Soft Real Time Task

- Example,
  - Web browsing
    - After URL is clicked web page fetched and displayed within few seconds.
    - It may take long time, does not mean that system have failed.
  - Railway Reservation
    - In terms of handling request
    - Task is to produce ticket within 20 seconds or apology message of unavailability of seats after request.
    - If task fails to produce ticket in 20 seconds and misses the deadline, does not result in system failure.
    - But the **utility of the results slowly falls off with the time**.
- Time bound usually range from ***fraction of seconds to few seconds.***

Types of Real Time Tasks

- **Non-Real Time Task**
- Task is not associated with time bounds.

- Can you think of any example of non-real time tasks?

Thank You . . .