

Welcome to SFWR ENG 4aa3 and SFWR ENG 4ga3

- Asghar Ali Bokhari
- Computing & Software Dept.
- Office ITB 101-c
- Ex. 27554
- email: bokhari@mcmaster.ca

Course Outline:

It is your responsibility to read the course outline available at www.elm.mcmaster.ca

Lectures: Mon., Wed., Thurs. 13:30 (HH 302)

Labs (ITB 134):

4GA3:

Tue. L1:08: 30-17:20

4AA3:

Tue. L1: 14:30 -17:20

Mon. L2: 08:30 - 11:20

Mon. L3: 14:30 -17:20

L3 is overcrowded, 10 students to move to other labs

Lab Schedule:

- Next week: no labs, do the **pre-lab**
- Details of work to be submitted as pre-lab is given in *lab109.pdf* on course web page
- Also read *lab-intro.pdf*
- Lab sessions begin the week starting Sept. 21

Lab Groups:

Each student will work individually in the lab.

Lab 4 may require working in groups of two. In such a case TAs will assign groups to students.

InBox Message - Windows Internet Explorer

https://univmail.cis.mcmaster.ca/Session/1670506-Naa7u5splvt5WLLtcfQ/Message.wssp?Mailbox=INBOX&MSG=18095

File Edit View Favorites Tools Help

Mailboxes Compose Rules Settings External Log Out

Next Unread Reply Reply To All Forward
Back to InBox Delete Set Flag Close as Unread

From: "~~██████████~~" <~~██████████@univmail.cis.mcmaster.ca~~>
Subject: Thermocouple from SE4AA3
Date: Thu, 26 Jun 2008 17:41:57 -0400
To: bokhari@mcmaster.ca

Good afternoon,

I was one of your students in SE 4AA3, and was wondering if I could acquire some of your information (pdfs, etc) from one of the labs that we did for this class, specifically that related to LabVIEW and the thermocouple.

I'd like to thank you for introducing both the class and myself to technology (ie: LabVIEW) that are in wide use in many areas of SE. Specifically: I am presently working on a project on internship at Valeo that requires the use of a thermocouple for thermal durability testing of vehicle ECUs used in ultrasonic park assist and automated parking applications.

Again, thank you - whatever information you could lend on the thermocouple could be extremely useful.

Mr. ~~██████████~~
Software Engineering and Management V
Valeo Interior Controls NA SE Intern

Grading:

Test 1	(October 19)	12.5%
Test 2	(November 19)	12.5%
Assignments		10%
Laboratory		25%
Final exam	(3 hours)	40%

TEXTBOOK:

Phillip A. Laplante, *Real-Time Systems Design and Analysis*, (3rd Edition), John Wiley & Sons Inc., 2004. ISBN 0-471-22855-9.

ADDITIONAL REFERENCES:

- Course Web Page: <http://www.elm.mcmaster.ca/>
- G.F. Franklin, J.D. Powell and M. Workman, *Digital Control of Dynamic Systems* (3rd Edition), Addison Wesley Longman, 1998.
- N.S. Nise, *Control Systems Engineering* (5th Edition), John Wiley & Sons Inc., 2008. ISBN-13 978-0471-79475-2
- Jane W. S. Liu, *Real Time Systems*, Prentice Hall, 2000. ISBN 0-13-099651-3
- C. M. Krishna and Kang G. Shin, *Real Time Systems*, McGraw-Hill, 1997. ISBN 0-07-057043-4

Real Time Systems

What is a system?

A system is an assemblage of inter-related elements comprising a unified whole. (wikipedia)

A combination of several components or pieces of equipment integrated to perform a specific function.

thequalityportal.com/glossary/s.htm

A combination of the hardware, software, and firmware.

sparc.airtime.co.uk/users/wysywig/gloss.htm

Black Box Representation



Real Time Systems

- Inputs/outputs of a computer system – digital
 - Data conversion may be required
- Input also called stimulus
- Output also called response
- What is response time?
 - Time between a stimulus and the corresponding response.
- What is a real time system?
- A possible definition: “ A system where a timely response to external stimuli is vital”
- What is meant by timely?

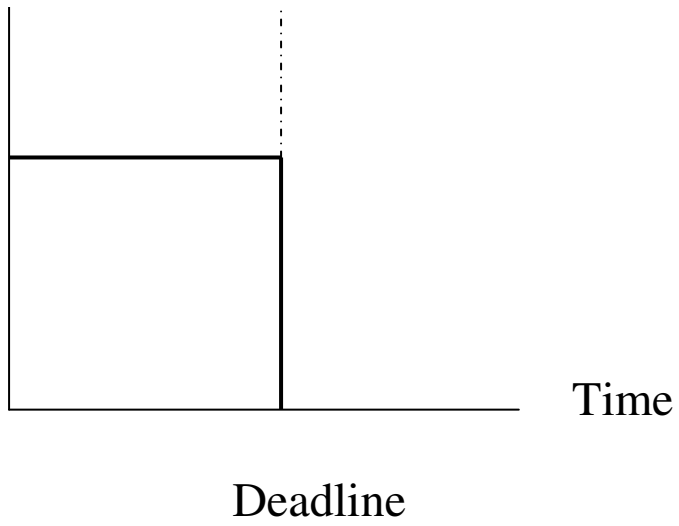
Real Time Systems

Redefine a real time system as:

- A system that must satisfy explicit response-time constraints or risk severe consequences, including failure.
- What happens if the task is not completed by the deadline? Does it make the system useless?
- A robot that welds two sheets of metal in an assembly line
- A computer system that adjusts the temperature inside a building

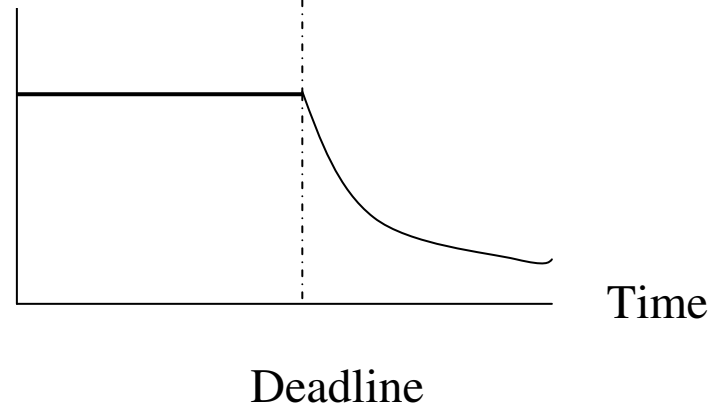
Classification of RTS:

Usefulness



a) Hard RTS

Usefulness



b) Soft RTS

Classification of RTS:

A hard RTS is one in which failure to meet a single deadline may lead to complete and catastrophic system failure.

A soft RTS is one in which performance is degraded but not destroyed by failure to meet response-time constraints.

A firm RTS is one in which missing a few deadlines will not lead to total failure, but missing more than a few may lead to a complete and catastrophic system failure.

A safety critical system is a hard RTS whose failure may cause injury or death to human beings. e.g. an aircraft or nuclear power station control system.

(reading assignment: intro to safety critical systems

<http://www.ipl.com/pdf/p0826.pdf>)

Examples:

- More than 80% of microprocessors used in RT applications!
- Process control systems:
 - usually Hard in some case may be firm
 - Control of power plants
 - Rolling Mills
 - Manufacturing systems with robots
 - Consumer products such as soft drinks
- Safety Critical Systems:
 - Nuclear power plant monitoring system:
 - Vehicle systems for automobiles, submarines, aircraft, railways and ships.
 - Traffic control for highways, airspace, railway tracks, and shipping lanes
 - Railway crossing
 - Medical systems for radiation therapy, patient monitoring etc.

Examples:

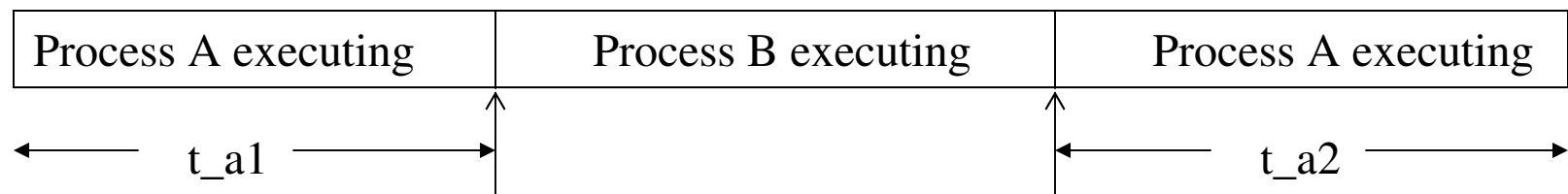
- Multimedia applications:
 - streaming videos
 - Lip synchronization in multimedia applications
 - Computer games
- Telephone, radio and satellite communication
- On line stock price quotation system
- Heat, light, doors and elevator controls in buildings

Note: There is a great deal of latitude for interpretation of hard, firm and soft real time systems

Is there a complete non-real time system?

Typical issues in real time systems

- Consider a computer controlling an aircraft
 - maintaining stability
 - keeping the cabin temperature within acceptable limits
 - What happens if a round robin scheduling policy is used?
- Solution to a problem may be different because of real-time deadlines
 - Consider the use of cache memory
 - Is $t_{a1} + t_{a2} = t_a$?



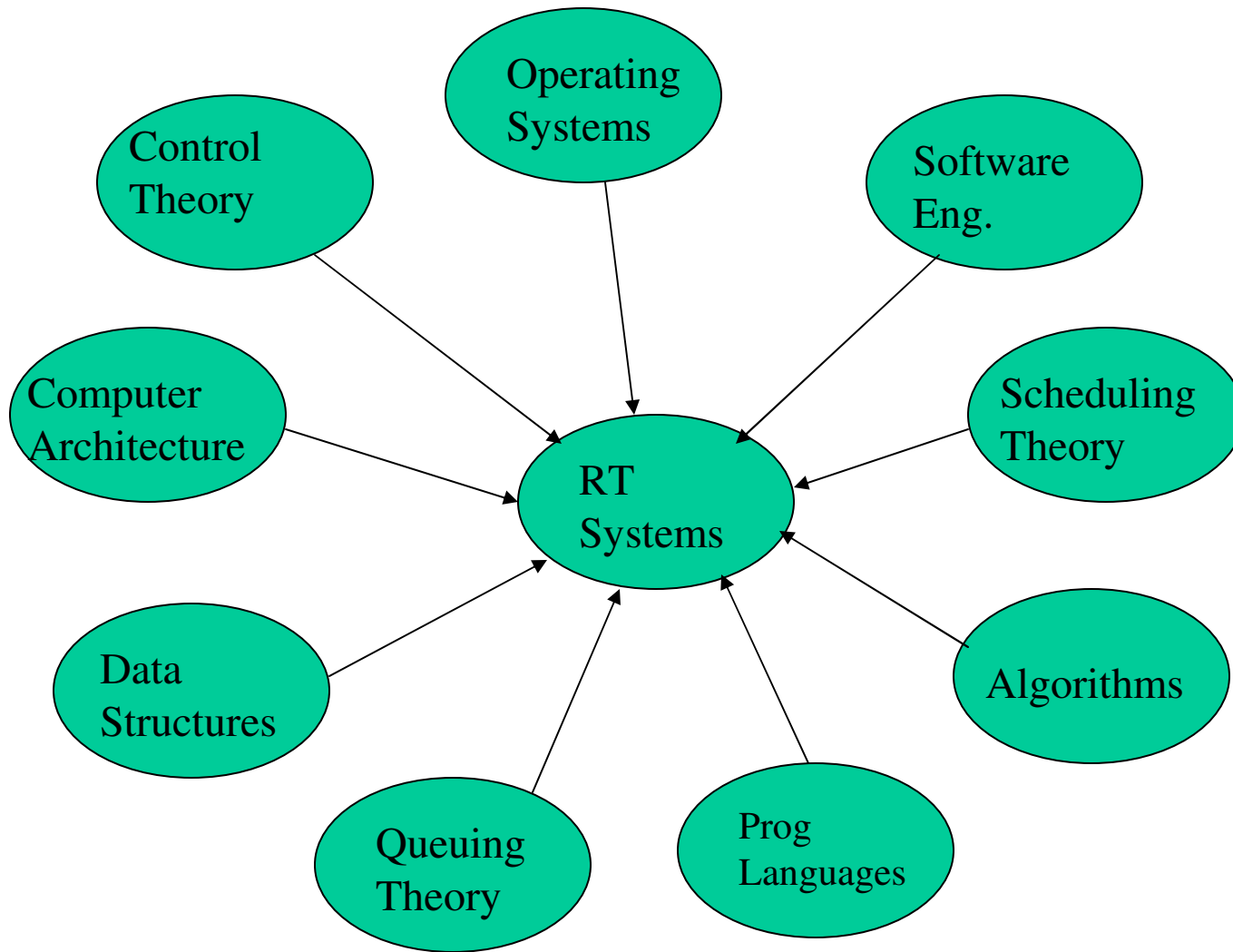
RT Software Design Considerations

- Timing Constraints
 - both logically and temporally correct
 - measure and predict the response time
 - may involve consideration of suitable scheduling algorithms
- Concurrency
 - to improve performance
 - inherent physical concurrency
- Fault tolerance and reliability
 - Reliability is a measure of how often a system will fail and fault tolerance refers to responses to failures that result in as little cost as possible
- Most RT systems are application specific and stand-alone
- Careful design of man-machine interface
- Testing and certification

Myths:

1. Real time systems are synonymous with “fast” systems.
2. Rate monotonic analysis has solved “the real time problem”.
3. There are universal, widely accepted methodologies for real time system specification and design.
4. There is never a need to build a real time operating system, because many commercial products exist.
5. The study of RT systems is mostly about scheduling theory.

Road Map



- Nature of time
 - Essential concept in RTS
 - Keeping and using accurate clocks essential to ensure correct working of RTS.
 - How time is actually measured?
 - The event of sun's reaching its highest apparent point in the sky is called 'transit of the sun'. Keeps on changing
 - Average interval between two consecutive transits of the sun is called the average solar day
 - Mean solar second = $1/86400^{\text{th}}$ of average solar day
 - Atomic clocks, 1948, use cesium 133 atoms, defined a second as the time it takes cesium 133 atom to make exactly 9,192,631,770 oscillations. This was done to make atomic second equal to the mean solar second on midnight of 1/1/1959

- Nature of time continued ...
 - Many labs around the world have cesium 133 clocks
 - TAI (International Atomic Time) is the mean number of oscillations of these clocks since Jan 1 1959 divided by 9192631770.
 - Due to constant change in the duration of solar day, 86400 TAI seconds is now about 3 msec less than a mean solar day.
 - The problem is solved by introducing 'leap seconds' whenever the discrepancy between TAI and mean solar time exceeds 800 msec.
 - The adjusted time is called Universal Coordinated Time (UTC)
 - To provide UTC to people, the National Institute of Standard (NIST) operates a short wave radio station (W W V) that broadcasts a short pulse at the start of each UTC second.

- Computer Clocks
 - Quartz crystals – can deviate considerably from standard ones
 - A counter and a holding register
 - Each oscillation of crystal decrements the counter and at zero an interrupt is generated called the **clock tick**.
 - Clocks of different CPUs run at slightly different rates and gradually get out of sync
 - This difference in time values is called **clock skew**
 - Difference between the time shown by a clock and the real time is called **clock drift**
 - When it is important that clocks do not deviate from the real time by more than a certain amount, they are called **physical clocks**.
 - Each machine is assumed to have a timer that causes an interrupt (tick) H times a second. Real timers do not tick exactly h times a second. (e.g. $H=60$ should cause 216000 ticks/hour but actual ticks may vary from 215998 to 216002)

- Clock Synchronization

- Mathematically

- C_i : real time --> clock time

- At real time t , $C_i(t)$ is the time told by a real clock c_i

- Standard clocks TAI and UTC are closest to perfection and it can be assumed that

- $C_s(t) = t$

- Desirable properties of a computer clock c

- Correctness:** At any time t its value should be within some bound ϵ of a standard clock

- $| C(t) - C_s(t) | \leq \epsilon$

- Bounded Drift:** Drift rate is the rate at which a clock gains or loses time w.r.t real time, due to some source of inaccuracy.

- For a perfect clock c_p

- $d(C_p(t))/dt = 1$, as $C_p(t) = t$

- Clock Synchronization

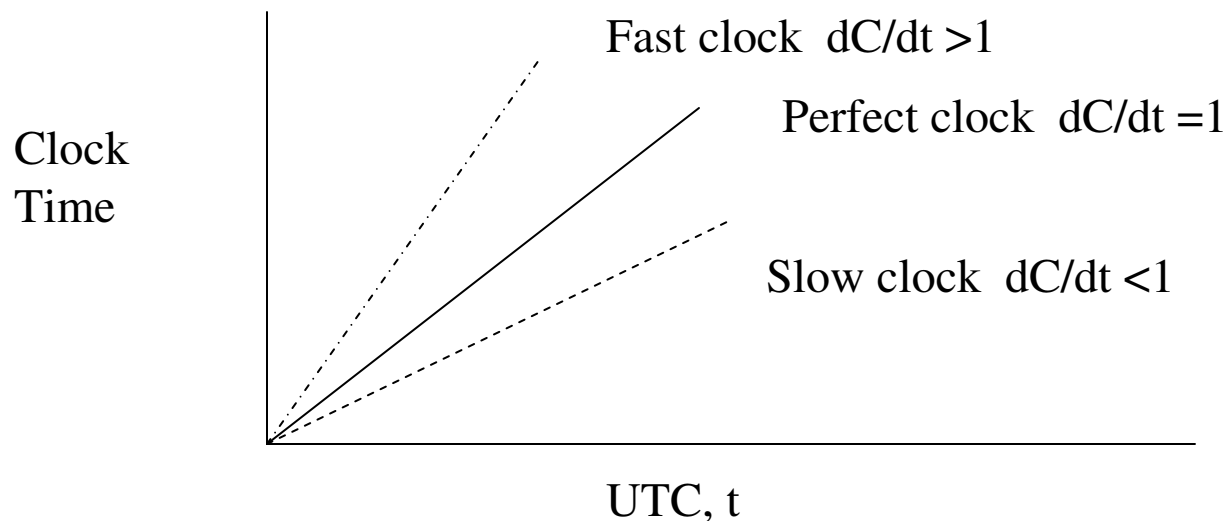
- To maintain accuracy of a working clock the maximum drift rate should not exceed a given bound ρ

$$| d(C(t))/dt - 1 | \leq \rho$$

or

$$1 - \rho \leq d(C(t))/dt \leq 1 + \rho$$

The constant ρ is specified by manufacturer and is called the **maximum drift rate**



- Clock Synchronization

- Consider two clocks drifting from UTC in opposite directions at time Δt after they were synchronized, they may be $(2 \rho \Delta t)$ apart

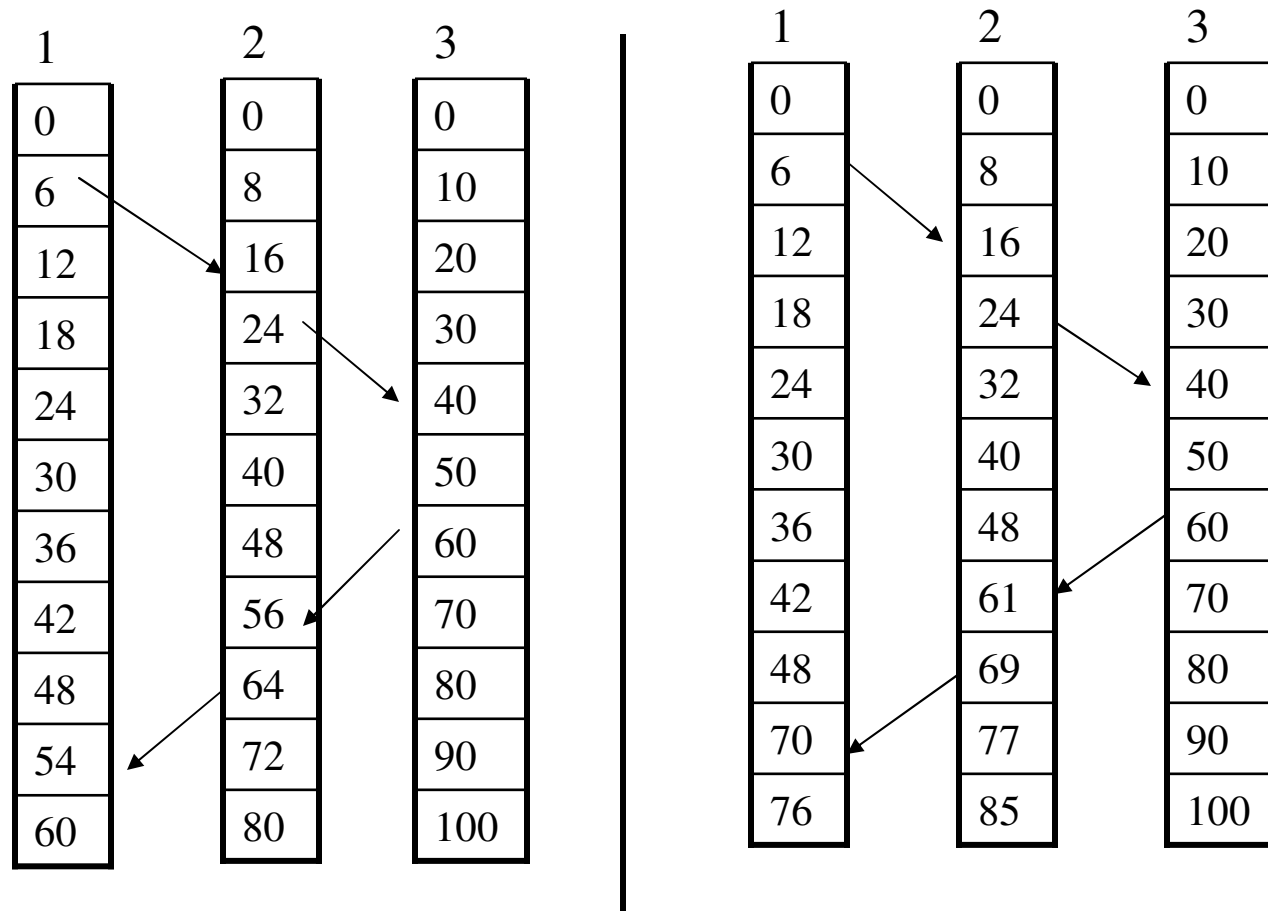
To guarantee that two clocks will never drift more than δ secs, resynchronize every $(\delta / 2\rho)$ seconds

- There are various algorithms for clock synchronization that differ precisely how the above is achieved.

- **Clock Synchronization**
 - For many applications it is sufficient that all CPUs agree on the same time irrespective of whether they are close to the real time or not. **Logical Clocks**
 - **Lamport's Algorithm**
 1. If a happens before b in the same process $C(a) < C(b)$
 2. If a and b are sending and receiving processes $C(a) < C(b)$
 3. For all events a and b , $C(a) \neq C(b)$
 - When a message arrives and the receiver's clock shows a value prior to the time the message was sent, the receiver fast forwards its clock so that it shows a time that is one unit more than the sending time.

- Clock Synchronization

- To appreciate the need for synchronization, consider an example involving logical clocks



Terminology

- Where do the deadlines come from?
 - Based on the underlying physical phenomena of the system under control
 - Animated displays
 - Navigation systems
 - Deadlines imposed in requirements
- Events
 - Any occurrence that causes the program counter to change non-sequentially
- Release time
 - The time at which an instance of a scheduled task is ready to run – generally associated with an interrupt.

- Synchronous and asynchronous events
 - Synchronous events occur at predictable times in the flow of control
 - Asynchronous events occur at unpredictable points in the flow of control and are usually caused by external sources.
- Event driven systems
 - Use I/O completion or timer events to trigger schedulable tasks
- Determinism
 - For each possible state and each possible set of inputs a unique set of outputs and next state of the system can be determined
- Transducer
 - A device that converts energy from one form to another
- Sensor
 - A device that measures or detects a real-world condition, such as motion, heat or light and converts the condition into an analog or digital representation.
 - e.g. temperature, pressure, altitude, motion sensors

- **Actuator**

- An actuator is a transducer that accepts a signal and converts it to a physical action.

- **Accelerometers**

convert the compression or stretching of a spring to electrical signal

- **Position resolvers**

are sensors that provide angular measurements relating to orientation or altitude of airplanes

- **Micro-controller**

- An integrated chip that has a CPU, RAM, ROM, I/O ports and timers similar to those of a computer but designed to control a single system, much smaller in size, all functions on a single chip.

- Programmable Logic Devices (PLD's)
 - PLA (programmable logic arrays), PAL (programmable array logics) simple digital logic
 - ASIC (application specific integrated circuits) a core processor with some simple peripheral support.
 - FPGA (field programmable gate arrays) similar to ASIC but re-programmable in place.

NOTE: You are supposed to know all terms defined in chapter 2.