CS 225: Switching Theory

S. Tripathy
IIT Patna

Course mechanics

- Course Instructors
  - S. Tripathy and J. Mathew
- Course webpage
- 172.16.1.252/~som/cs225
- Pl join with
  - MS Team CS225 class
    - MS Team CS 225 using the code u1y6icm
  - CS225 whatsapp group
    - https://chat.whatsapp.com/Bnh2oPVc7v9HdKkkPciIZ3
- My office room
  - Room 407
  - Block III, C.S. E Department
  - Ph. 8036

Evaluation

- Evaluation
  - Up-to-Mid (50)
    - Assignments/ Quiz-tests (20%)
    - Midterm (30%)
  - After-Mid (50)

Tentative

- Your participation in the class has major value

- Quiz
  - Some (all) quizs would be at the starting of the class or at the end of the class

Schedule

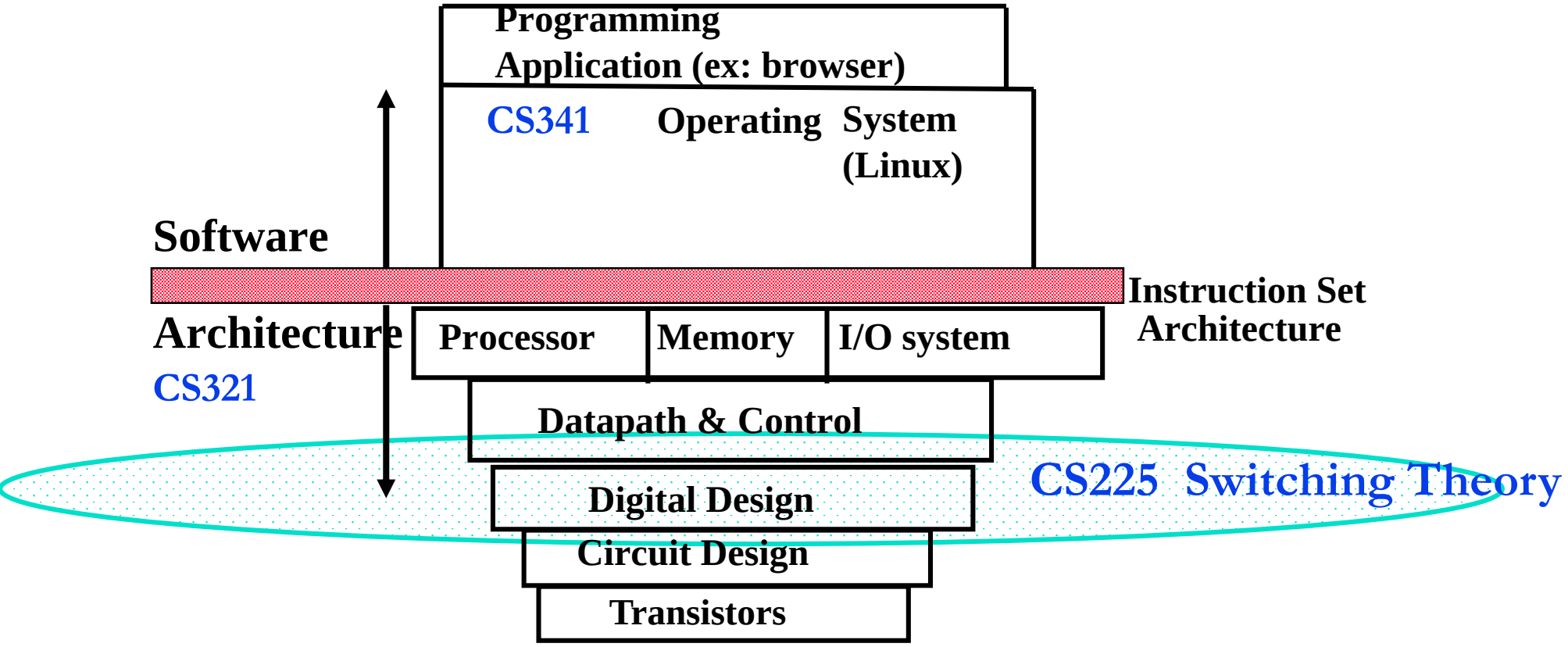
- Lecture Schedule:
  - Tuesday (10-11AM)
  - Wednesday (10-11AM)
  - Thursday (10-11AM)

- For Discussion
  - Tuesday 4.00-5.00PM
    - Nice if you could mail me so will wait\ prepare
    - You can reach me at som@iitp.ac.in
    - 8084717331 (whatsapp)

This class

- What will we study in this course? And Why?

Position among CSE Courses



- Big idea: Coordination of many levels of abstraction

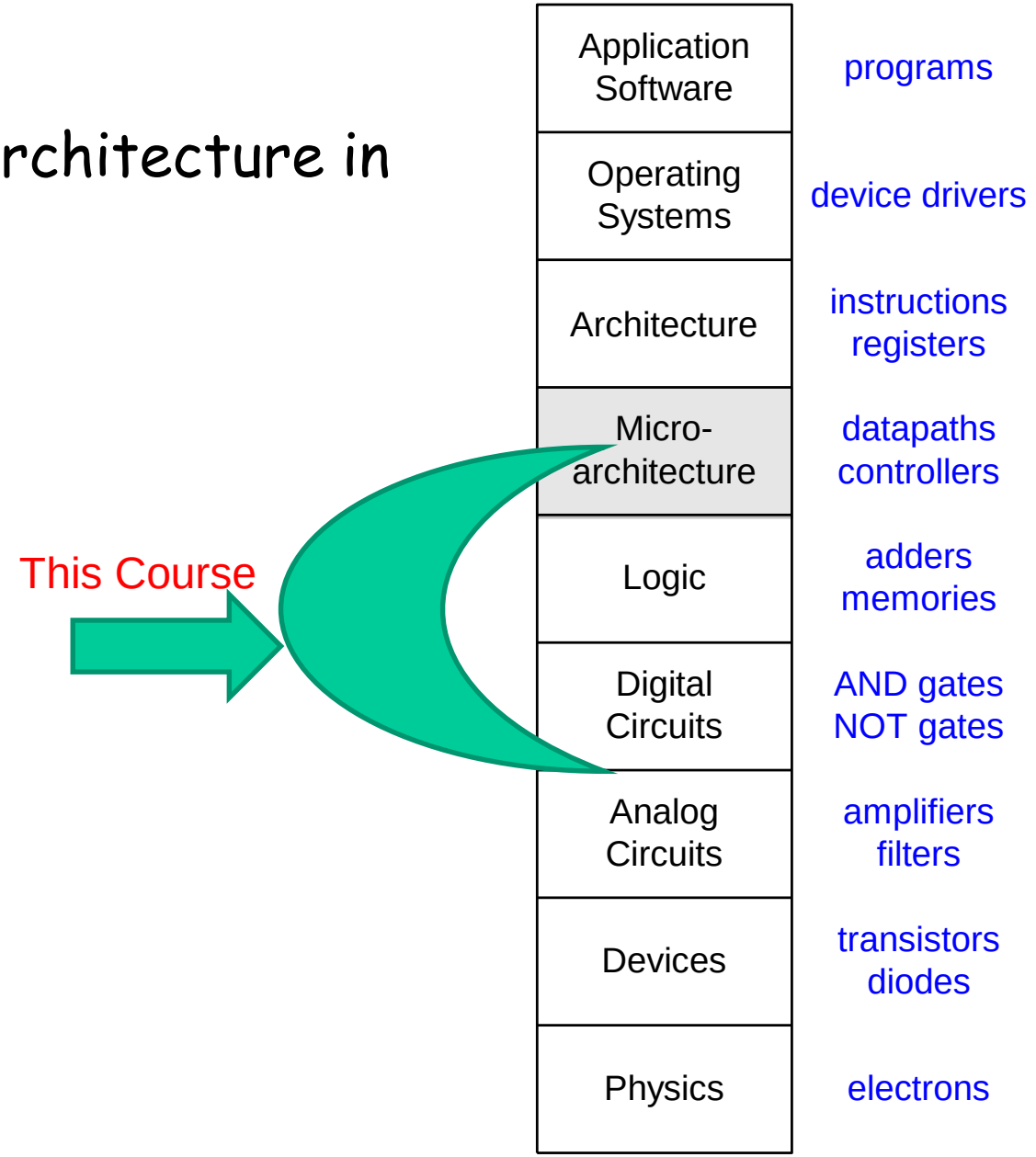
Objectives

- To know the fundamental concept of logic circuit
- Extends the idea to build complex digital systems
- At the end of the course the student will be capable enough to understand the concepts required in the design of digital circuits like computers and communication systems
- CS226: can write a Verilog and VHDL description of basic hardware blocks and simulate them using standard simulators.

This Course

- Micro-architecture:
  - how to implement an architecture in hardware Processor:

- Datapath: functional blocks
- Control: control signals



Course Outline

Number Systems, Boolean algebra, logic gates, minimization of completely and incompletely specified switching functions, Karnaugh map and Quine-McCluskey method, multiple output minimization, two- level and multi-level logic circuit synthesis.

Combinational Ckt

Clocks, flip-flops, latches, counters and shift registers, finite-state machine model, synthesis of synchronous sequential circuits, minimization and state assignment, asynchronous sequential circuit synthesis.
Programmable logic devices: memory, PLA, PAL. Representation of sequential circuits using ASM charts, synthesis of output and next state functions, data path control path partition-based design.

Sequential Ckt

Text Books:
1. Z. Kohavi, Switching and Finite Automata Theory,
2. M. M. Mano, Digital Design, 3rd Ed, Pearson Education

What Digital Design is?

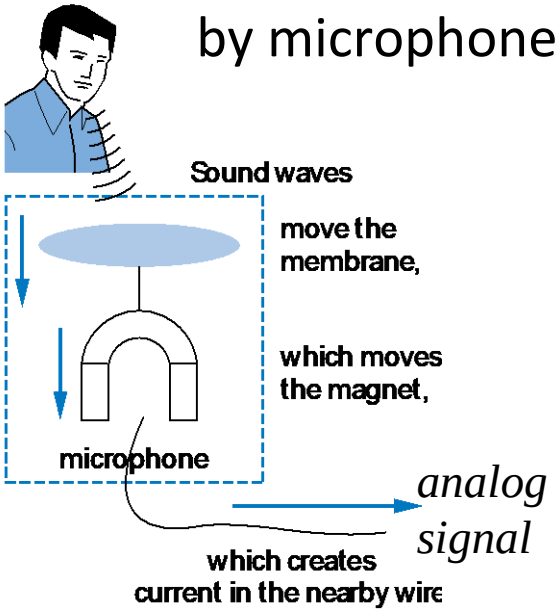
- Design:?
  - Plan that governs the development of a particular system
- Digital Design:?
  - Design of Digital system
  - Plan that governs the development of a digital system.
  - Usually Plan for building digital system is
    - represented by block diagrams and
    - Inter connection of a number of simple elements callled logic circuit or logic blocks
- Digital blocks are used for
  - Arithmetic operations in digital computer
  - Sequence and timing of traffic lights in traffic system
  - Represent time of a day in digital watch

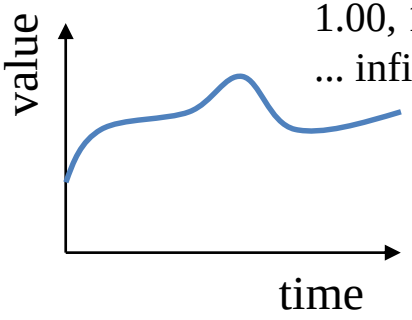
What Does "Digital" Mean?

Analog signal

Inifinite possible values

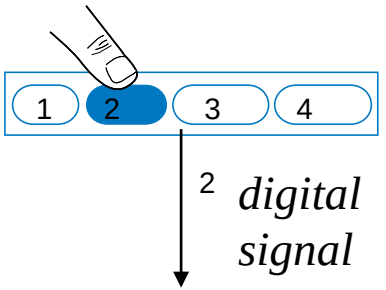
Ex: voltage on a wire created by microphone

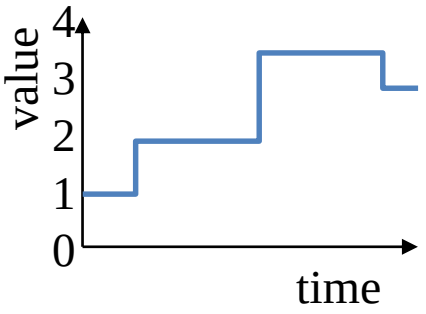




Possible values:
1.00, 1.01, 2.0000009, ... infinite possibilities

- Digital signal
  - Finite possible values
    - Ex: button pressed on a keypad

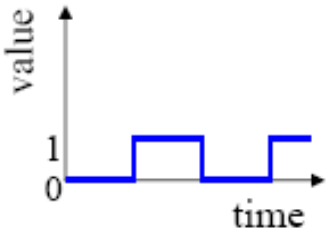




Possible values:
0, 1, 2, 3, or 4.
That's it.

Binary : Digital Signals with Only Two Values

- ***Binary digital signal -- only two*** possible values
  - Typically represented as **0 and 1**
  - One *binary digit is a* ***bit***
  - We'll only consider *binary digital sig*



- Binary is popular because
  - Storing/transmitting one of two values is easier than three or more (e.g., loud beep or quiet beep, reflection or no reflection)
  - Easy to store with bistable elements
  - Reliably transmitted on noisy and inaccurate wires
  -
- Transistors, the basic digital electric component, operate using two voltage

Representations of Digital Design: Switches

A switch connects two points under control signal.

Normally Open

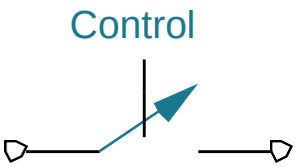
when it is 1 (true), the switch is closed

when the control signal is 0 (false), the switch is open

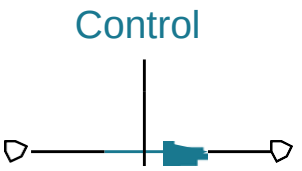
Normally Closed

when control is 1 (true), the switch is open

when control is 0 (false), the switch is closed



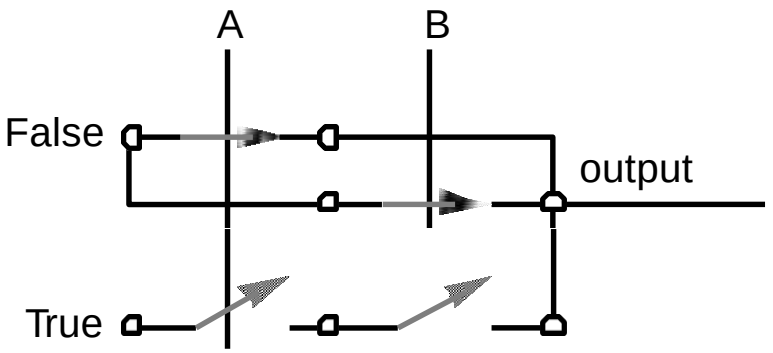
Normally Open
Switch



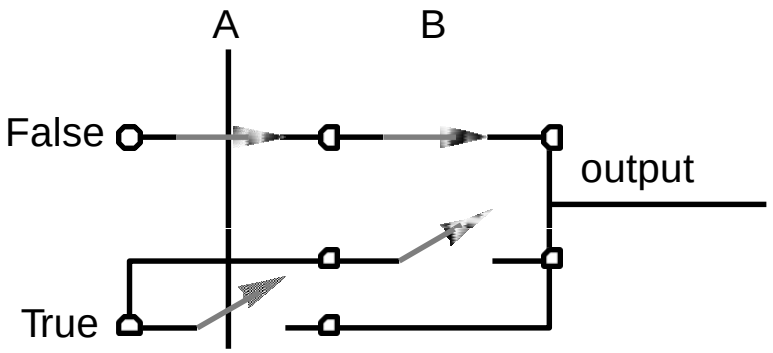
Normally Closed
Switch

Switch Representations

Implementation of AND and OR Functions with Switches



AND function Series connection to TRUE



OR function Parallel connection to TRUE

Advantages of Digital signal

- Better Quality:
  - Analog signal (e.g., audio) may lose quality if Voltage levels are not saved/ copied/ transmitted perfectly
  - Digitized version enables save/ copy/ transmit near perfectly
  - But we can distinguish 0s from 1s
  - Ex.: 1 V: "01" 2 V: "10" 3 V: "11"
- Compressable:
- Ex: digital data: 000000001111100011111
- Can be compressed into x0y1

Real World Example -Digitization Benefit

Analog signal (e.g., audio) may lose quality

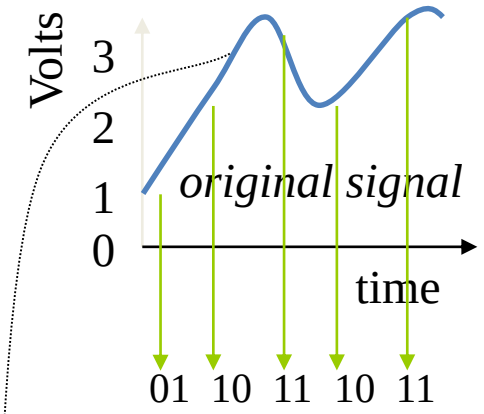
Voltage levels not saved/copied/transmitted perfectly

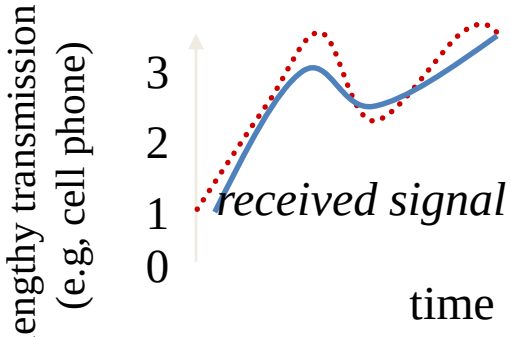
Digitized version enables near-perfect save/cpy/trn.

"Sample" voltage at particular rate, save sample using bit encoding

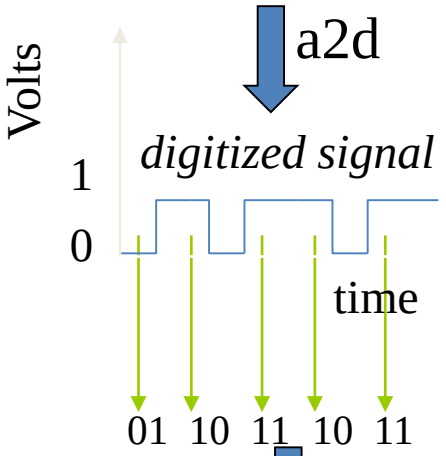
Voltage levels still not kept perfectly

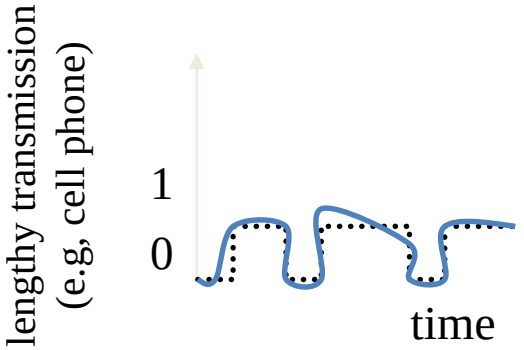
But we can distinguish 0s from 1s





How fix -- higher, lower, ?





Can fix -- easily distinguish 0s and 1s, restore

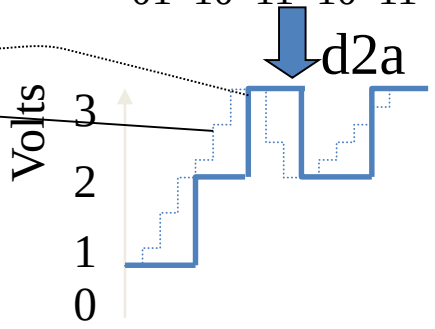
Let bit encoding be:

1 V: "01"

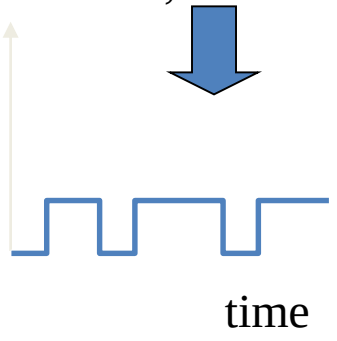
2 V: "10"

3 V: "11"

Digitized signal not perfect re-creation, but higher sampling rate and more bits per encoding brings closer.



same



Digitized Audio: Compression Benefit

Digitized audio can be compressed

- e.g., MP3s
- A CD can hold about 20 songs uncompressed, but about 200 compressed

Compression also done on digitized pictures (jpeg), movies (mpeg), and more

Digitization has many other benefits too

Example compression scheme:

00 --> 0000000000

01 --> 1111111111

1X --> X

0000000000 0000000000 0000001111 1111111111

00 00 10000001111 01

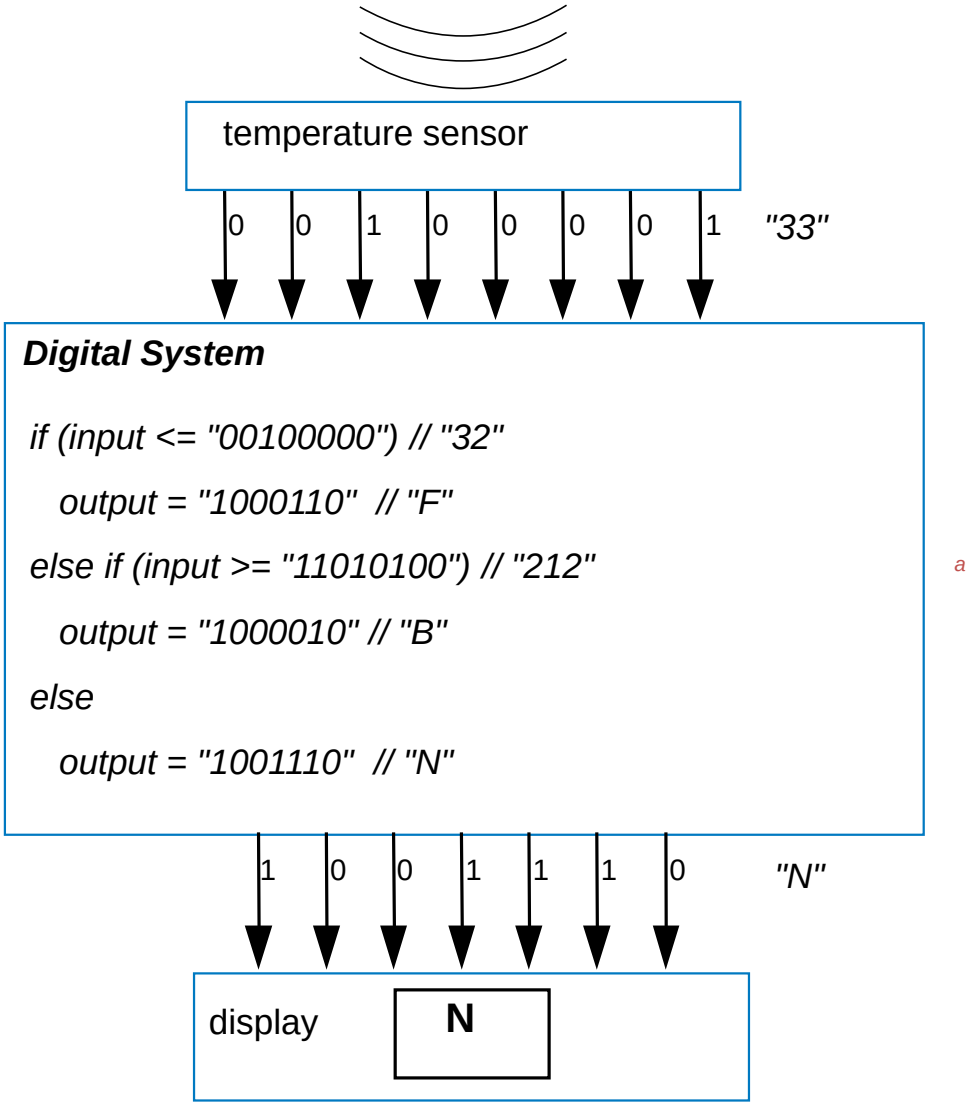
Using Digital Data in a Digital System

A temperature sensor outputs temperature in binary

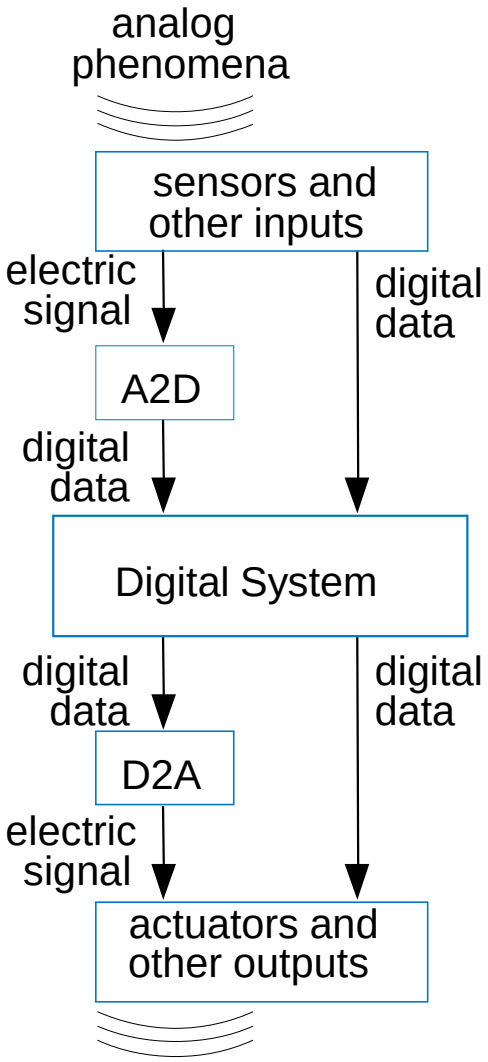
The system reads the temperature, outputs ASCII code:

- "F" for freezing (0-32)
- "B" for boiling (212 or more)
- "N" for normal

A display converts its ASCII input to the corresponding letter



How Do We Encode Data as Binary for Our Digital System?



Some inputs inherently binary

Button: not pressed (0), pressed (1)

Some inputs inherently digital

Just need encoding in binary

e.g., multi-button input: encode red=001, blue=010, ...

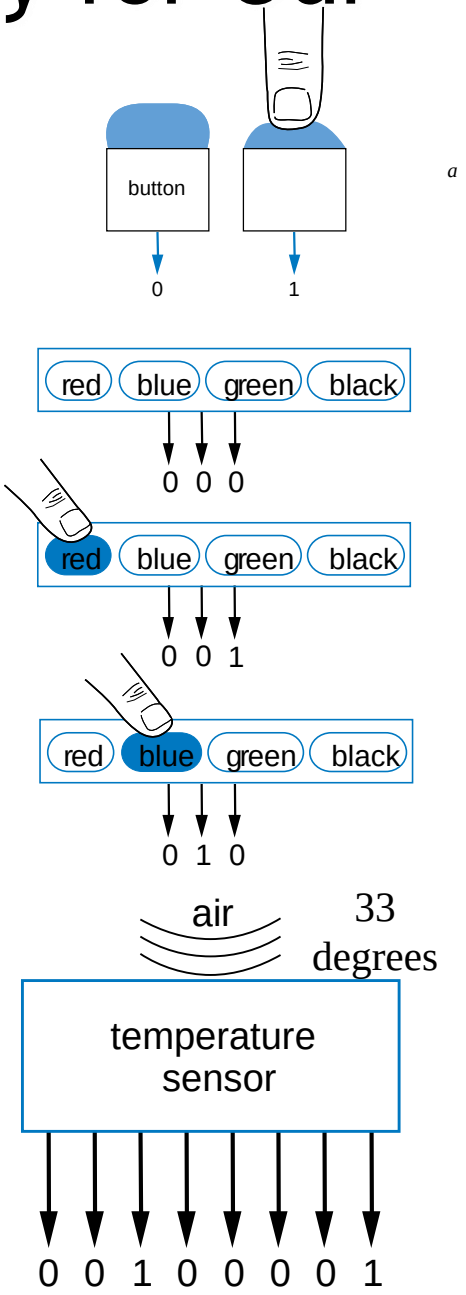
Some inputs analog

Need analog-to-digital conversion

As done in earlier slide -- sample and encode with bits

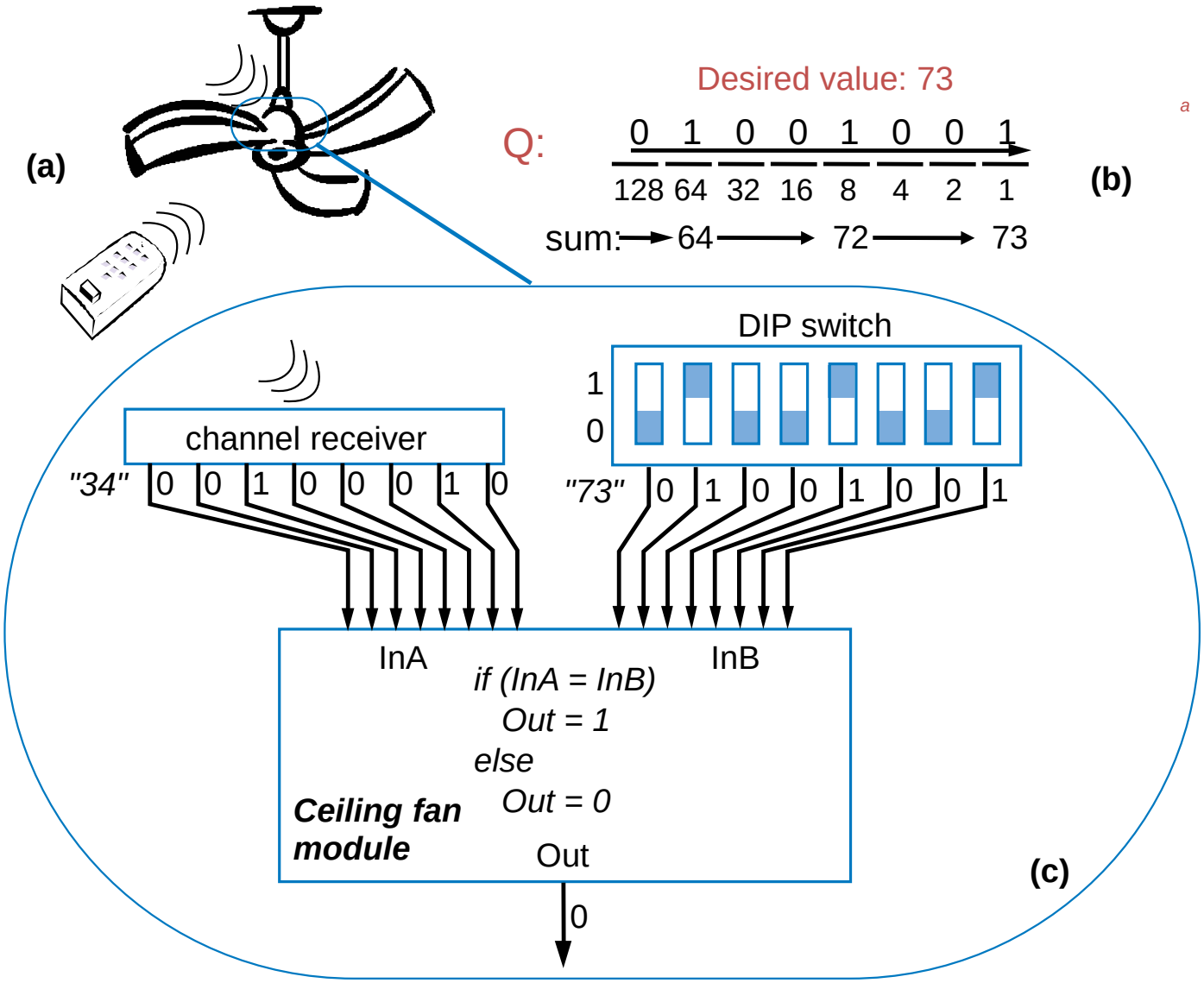
A2D -Analog to digital converter

D2A-Digital to Analog converter

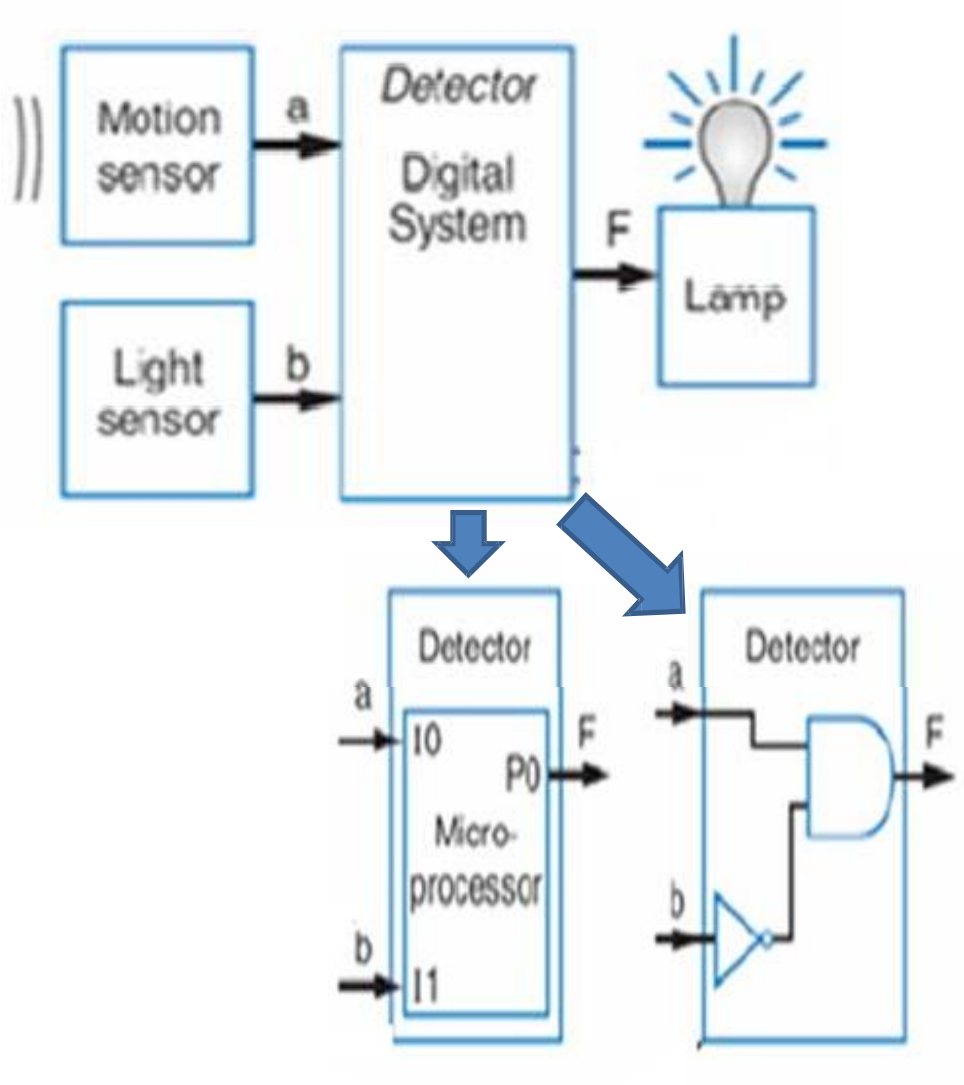


Example: DIP-Switch Controlled Channel

Ceiling fan receiver should be set in factory to respond to channel "73" Convert 73 to binary, set DIP switch accordingly



Digital System: Automatic lightening



Summary

Digital systems surround us

- Inside computers
- Inside many other electronic devices (embedded systems)

Digital systems use 0s and 1s

- Encoding analog signals to digital can provide many benefits
  - e.g., audio—higher-quality storage/transmission, compression, etc.
- Encoding integers as 0s and 1s: Binary numbers

Microprocessors (themselves digital) can implement many digital systems easily and inexpensively

- But often not good enough—need custom digital circuits

Thanks