ME 311: Microprocessors and Automatic Control

Basics of digital logic design
Combinational

P.S. Gandhi
Mechanical Engineering
IIT Bombay

1

Life Skill

Process of Learning or Gaining Knowledge

- Journey from "something known" to "something unknown"
- What it is not?
  - Just information: (a lot of it is on web why we need classroom teaching)
  - Just memory
- So what to do if you feel you don't understand things? Go one step back and connect dots there what was missing..

2

Recap

- Introduction why we should study microprocessors and control fundes?
- Examples of real life automation in industry and in appliances
- Core or brain of all these is microprocessor
- Todays class: start with building blocks for microprocessors

Applications of Logic Design/Microprocessors

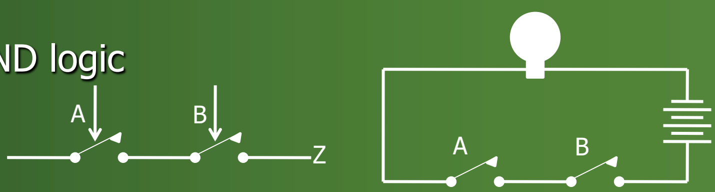
- Embedded products and real-time control systems: as we saw already
- Scientific equipment
  - Testing, sensing, reporting
- Conventional computer design
  - CPUs, memories, peripherals
- Networking and communications
  - Phones, modems, routers, computer

Brief history

- 1850: George Boole invents Boolean algebra
  - Maps logical propositions to symbols
  - Permits manipulation of logic statements using mathematics
- 1938: Claude Shannon links Boolean algebra to switches
  - His Masters' thesis
- 1945: John von Neumann develops first stored program computer
  - Its switching elements are vacuum tubes (a big advance from relays)
- 1946: ENIAC--world's first all electronic computer
  - 18,000 vacuum tubes
  - Several hundred multiplications per minute
- 1947: Shockley, Brattain, and Bardeen invent the transistor
  - replaces vacuum tubes
- enable integration of multiple devices into one package
  - gateway to modern electronics

Switches to Implement Digital Logic

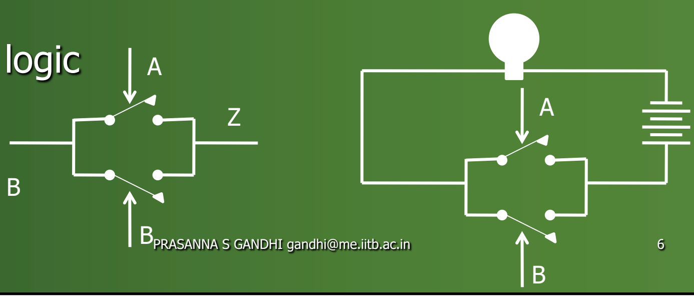
- AND logic



Z = A AND B

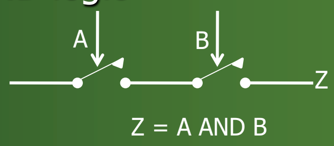
- OR logic

Z = A OR B

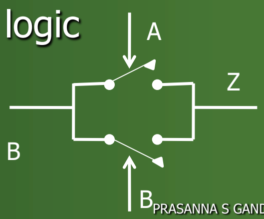


Switches to Implement Digital Logic

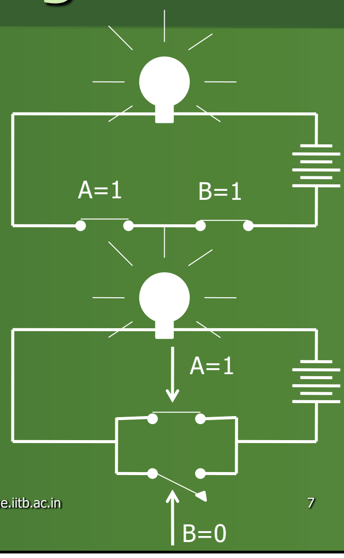
- AND logic



- OR logic

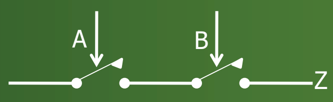


Z = A OR B



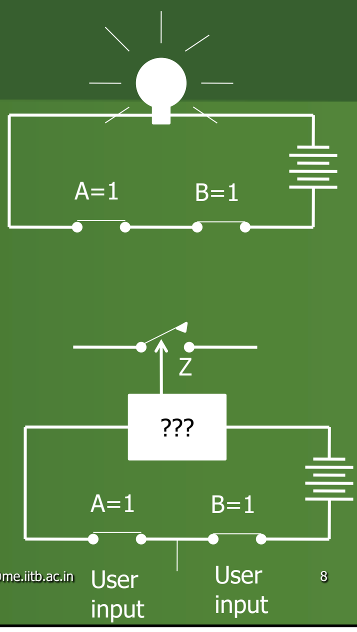
7

Scaling the Logic



Z = A AND B

- Q: How to use output z "as a switch" input to some other logic circuit instead of bulb?

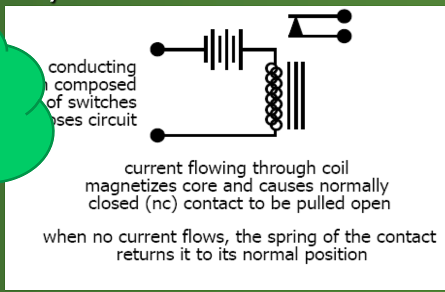


8

Switching networks

- To build larger computations
  - Use a light bulb (output of the network) to set other switches (inputs to another network)→ concept of relay

This is the way older big size computers were created



- Connect one network with another and so on

9

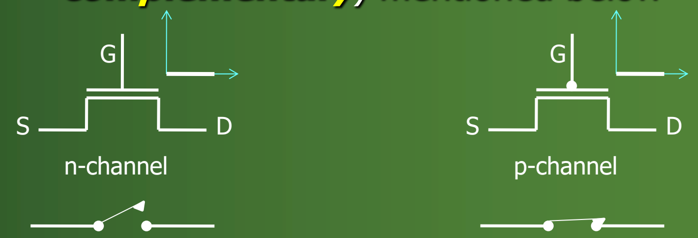
Transistors replacing relays

- Relays: bulky unreliable
- Invention of transistors first in the form of cathode ray tubes and next in the form of semiconductor revolutionized digital circuits
- Current technology: ***Complementary*** metal oxide semiconductor (CMOS) :
  - Several chips including pentium use this
  - Fabrication using VLSI technology (Very large scale integration)
- MOS transistors act as voltage controlled switches

10

CMOS Transistors

- Have 3 terminals source S, gate G and drain D two varieties (hence the word ***complementary***) mentioned below



Normally open (when G is low) Normally closed (when G is low)

11

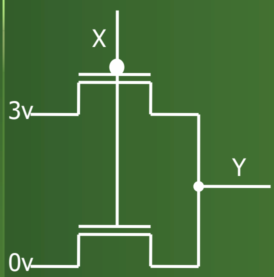
CMOS Transistors

- Have 3 terminals source S, gate G and drain D two varieties (hence the word ***complementory***) mentioned below

Closes when Voltage(G) > voltage(S) Opens when Voltage(G) > voltage(S)

12

CMOS circuit for simple NOT gate

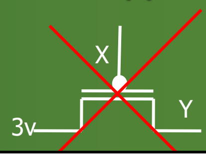


- Input is X and output is Y
- Can you generate a truth table (1 means 3v):

NOT Gate

X	Y
0 volts : 0	3volts: 1
3 volts : 1	0 volts: 0

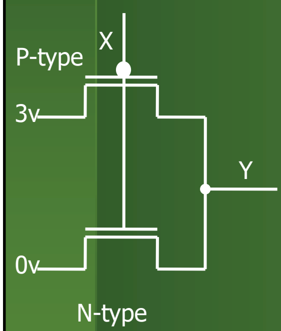
- Why not just the upper MOS???



P type devices not very good in connecting low voltages. Output may not get quite low enough and could not be used appropriately to connect the transistor in the next circuit

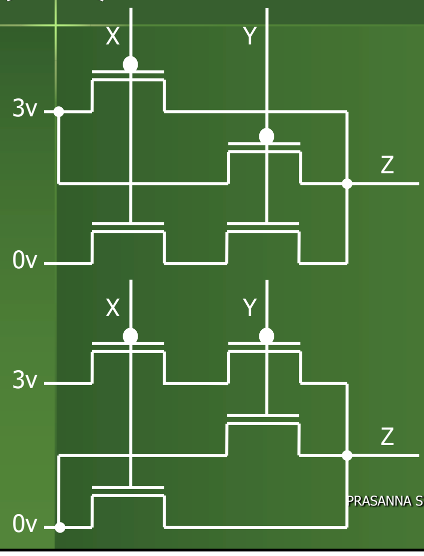
Simple network

Logic Gate: NOT



- Three other possibilities
  - Can we connect 0v at S of p-type transistor and get 0v as output at D when its switched on normally?
  - Can we connect 5v at S of n-type transistor and get 5v as output at D when its switched on by giving G high (needs to be greater than 5v+δ)?
  - Can we have zero at Y when we are not connected by wire (through transistors) to zero?
  - Answers NO for practical reasons: see characteristics of CMOS transistors

Logic gates and their CMOS circuit

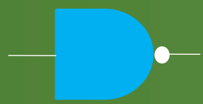


Truth tables

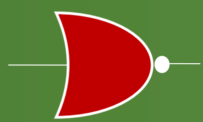
X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

NAND



NOR

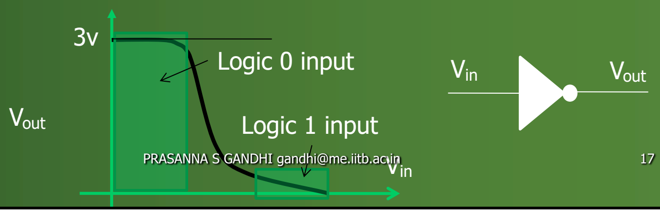


How to construct other gates in similar way?

- AND = NOT(NAND) so invert output of NAND gate (circuit??)
- OR = NOT(NOR) so invert the output of NOR gate (circuit??)

Some practical aspects: CMOS networks

- What influences the speed: wires acting as capacitor and charging discharging delays
- Is zero perfect 0v or is 1 perfect 3v? No. there is a noise margin ➔ transfer characteristics (example: NOT gate)

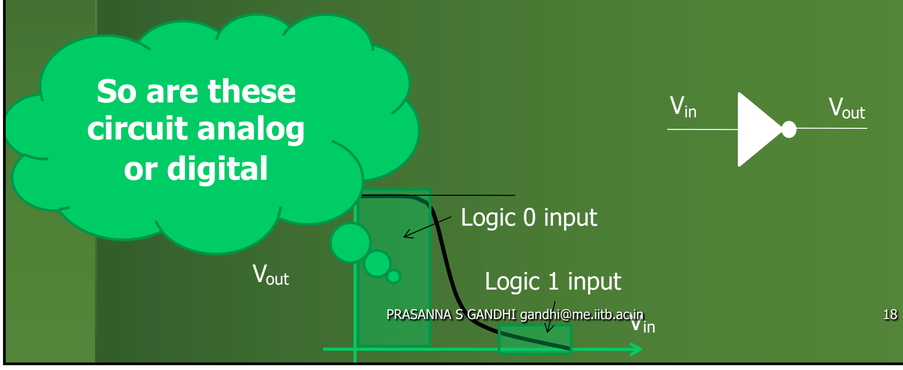


17

Some practical aspects: CMOS networks

- Why do we use digital abstraction??

Robustness!!



18

Logic design

What is design?

- Given a problem develop a solution using available resources to meet some specific design performance parameters

Logic design?

- Converting application task inputs and outputs to specifications in terms of 0s and 1s. (encoding) ➔ several possible ways
- Establishing mathematical relationship and developing combination of basic elements to achieve the goal➔ several possible ways
- Selection of optimum design

PRASANNA S GANDHI gandhi@me.iitb.ac.in 19

19

Example

Inputs			Output/s
A	**B**	**C**	**Z**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

- Say you have a logic table with inputs and outputs specified. How will you come up with some combination of standard gates to realize it in practice? Unique way or several ways
- $Z = (A' \bullet B' \bullet C) + (A \bullet B' \bullet C') + (A \bullet B \bullet C')$
- Further simplification possible ➔ K maps

PRASANNA S GANDHI gandhi@me.iitb.ac.in 20

20

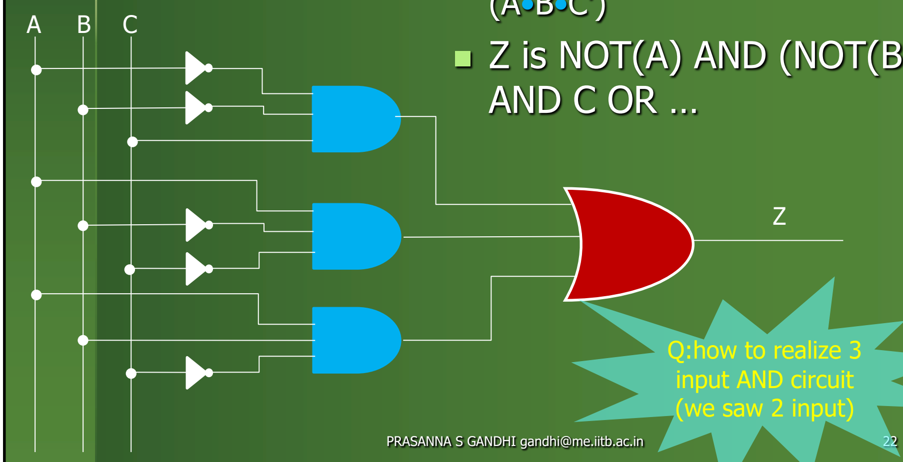
Example

	Inputs		Output/s
A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

- Z = (A′• B′•C)+ (A•B′•C′) + (A•B•C′)
- Further simplification possible ➔ K maps, Boolean identities
- Using the CMOS circuits for AND, OR, and NOT gates this expression can be realized in practice! Isnt it?
- Z is NOT(A) AND (NOT(B)) AND C OR …

Example

- Z = (A′• B′•C)+ (A•B′•C′) + (A•B•C′)
- Z is NOT(A) AND (NOT(B)) AND C OR …



Q:how to realize 3 input AND circuit (we saw 2 input)

THANK YOU

PRASANNA S GANDHI
gandhi@me.iitb.ac.in

23