Digital Systems
18B11EC213

Module 1: Boolean Function Minimization Techniques and Combinational Circuits-8

Dr. Saurabh Chaturvedi

Logic Gates

- Logic Gates

NOT Gate (Inverter)

AND Gate

OR Gate

NAND Gate

NOR Gate

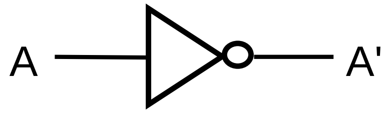
EX-OR (Exclusive-OR) Gate

EX-NOR (Exclusive-NOR) Gate

- Drawing Logic Circuits
- Analysing Logic Circuits
- Universal Gates

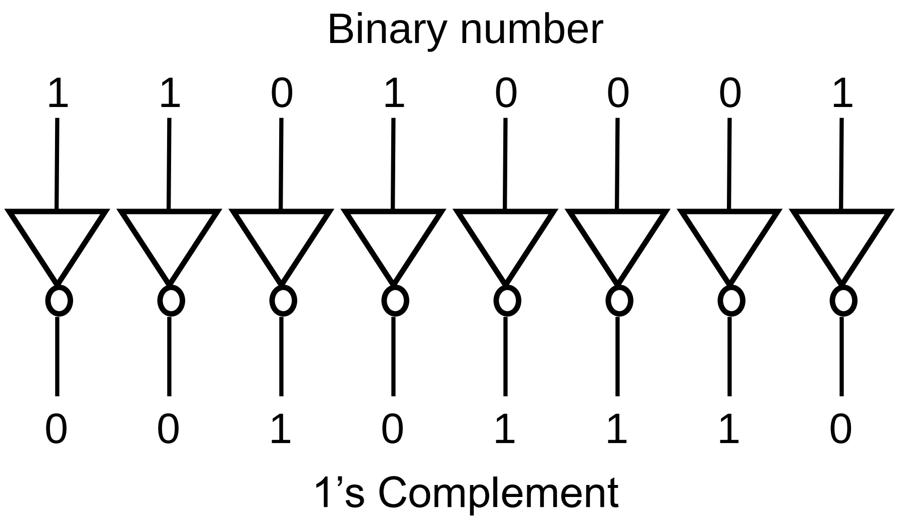
Cont..

❖ NOT Gate (Inverter)



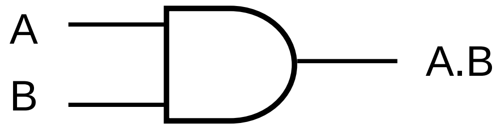
A	A'
0	1
1	0

Application of the inverter: complement



Cont..

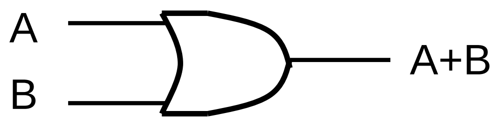
❖ AND Gate



A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

Cont..

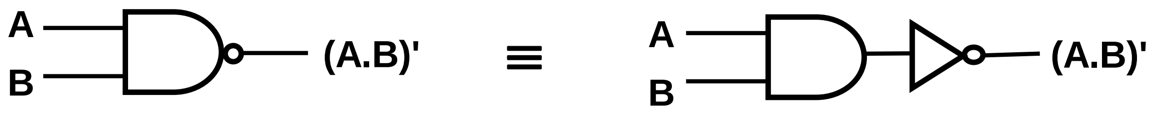
- OR Gate



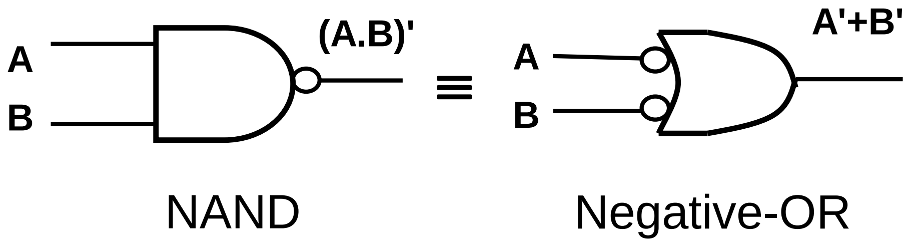
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

Cont..

- NAND Gate

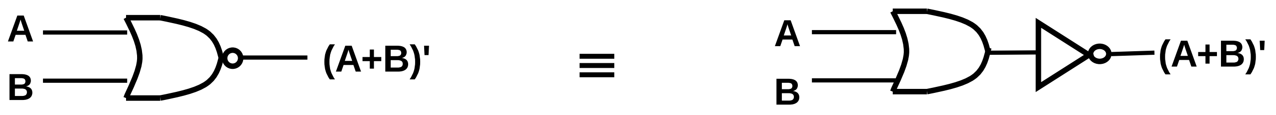


A	B	(A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

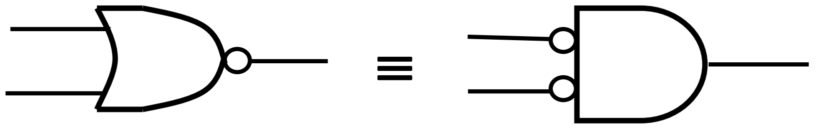


Cont..

- NOR Gate



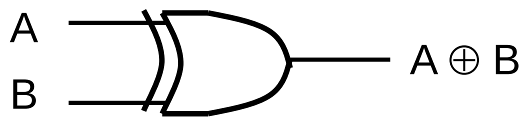
A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0



NOR   Negative-AND

Cont..

- XOR Gate

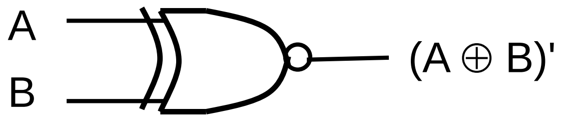


$A \oplus B = A'.B + A.B'$

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Cont..

❖ XNOR Gate



$(A \oplus B)' = A.B + A'.B'$

A	B	(A ⊕ B) '
0	0	1
0	1	0
1	0	0
1	1	1

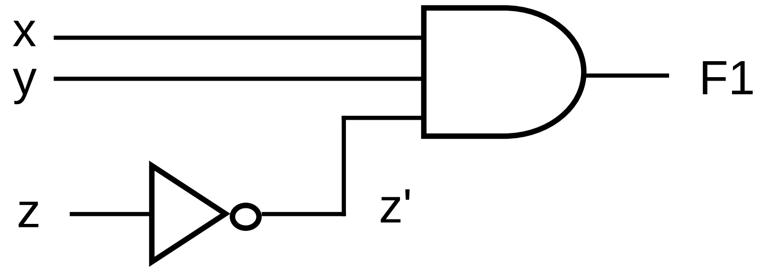
Drawing Logic Circuits

- When a Boolean expression is provided, we can easily draw the logic circuit.

Example-1:

F1(x, y, z) = x.y.z'

Note the use of a 3-input AND gate.

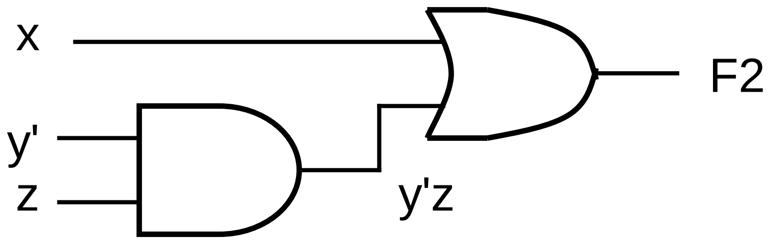


Cont..

Example-2:

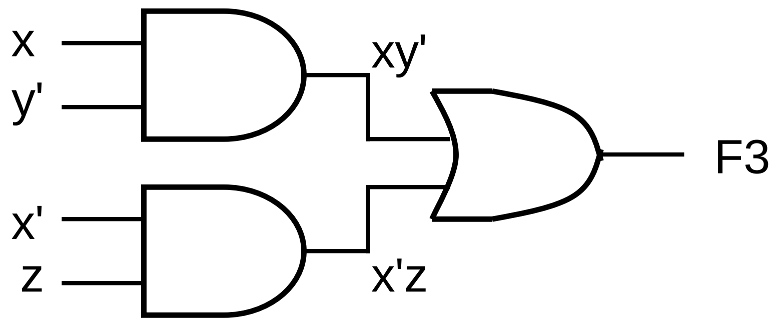
F2(x, y, z) = x + y’.z

We can assume that the variables and their complements are available.



Example-3:

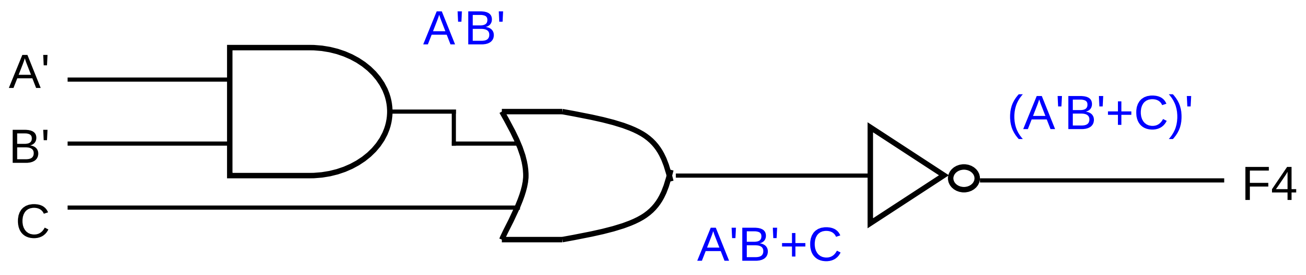
F3 = xy' + x'z



Analysing Logic Circuits

- When a logic circuit is given, we can analyse the circuit to obtain the logic expression.

Example: What is the Boolean (logic) expression of F4?



F4 = (A'B'+C)' = (A+B).C'

Universal Gates: NAND and NOR

- AND/OR/NOT gates are sufficient for building any Boolean (logic) functions.

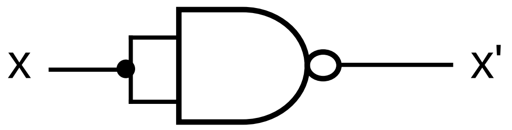
- We call the set {AND, OR, NOT} a complete set of logic.

Cont..

❑ NAND Gate

- NAND gate is self-sufficient, i.e., it can implement any logic circuit with it.
- Therefore, {NAND} is also a complete set of logic.
- It can be used to implement AND/OR/NOT.

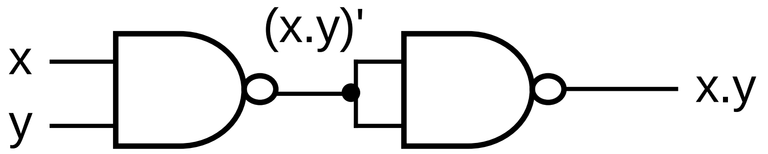
❖ Implementing an inverter (NOT gate) using NAND gate:



$(x.x)' = x'$ Idempotency

Cont..

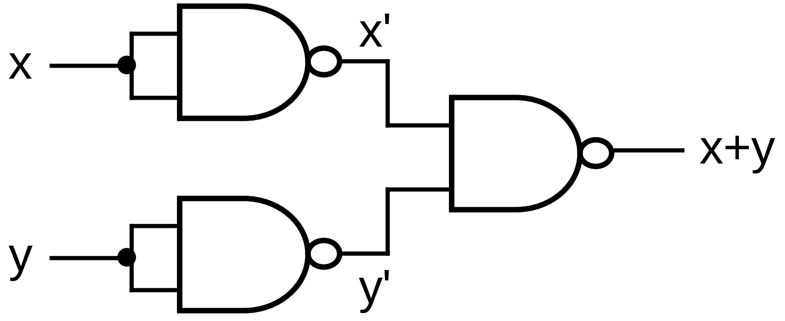
❖ Implementing AND gate using NAND gates:



((xy)'(xy)')' = ((xy)')'   Idempotency

= (xy)   Involution

❖ Implementing OR gate using NAND gates:



((xx)'(yy)')' = (x'y')'   Idempotency

= (x')'+(y')'   DeMorgan

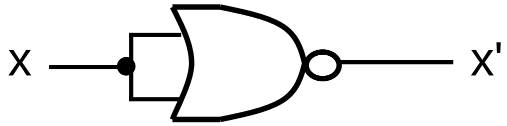
= x+y   Involution

Cont..

❑ NOR Gate

- NOR gate is also self-sufficient.
- Therefore, {NOR} is also a complete set of logic.
- It can be used to implement AND/OR/NOT.

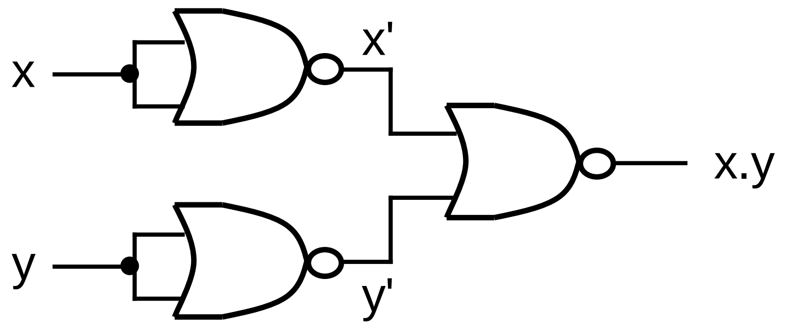
❖ Implementing an inverter (NOT gate) using NOR gate:



(x+x)' = x' Idempotency

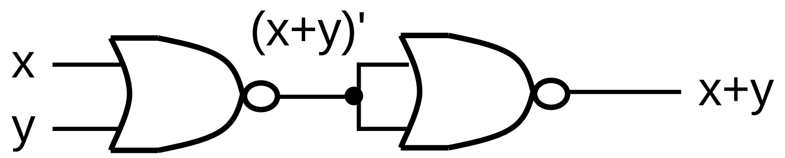
Cont..

❖ Implementing AND gate using NOR gates:



$((x+x)'+(y+y)')'=(x'+y')'$	Idempotency
$= (x')'.(y')'$	DeMorgan
$= x.y$	Involution

❖ Implementing OR gate using NOR gates:



$((x+y)'+(x+y)')' = ((x+y)')'$	Idempotency
$= (x+y)$	Involution

Implementation Using NAND Gates

- It is possible to implement any Boolean expression using NAND gates.

  Procedure:

  Step-1: Obtain the sum-of-products (SOP) Boolean expression.

  Let the Boolean function is $F3(x, y, z) = xy'+x'z$

  Step-2: Use the DeMorgan's theorem to obtain the expression using two-level NAND gates.

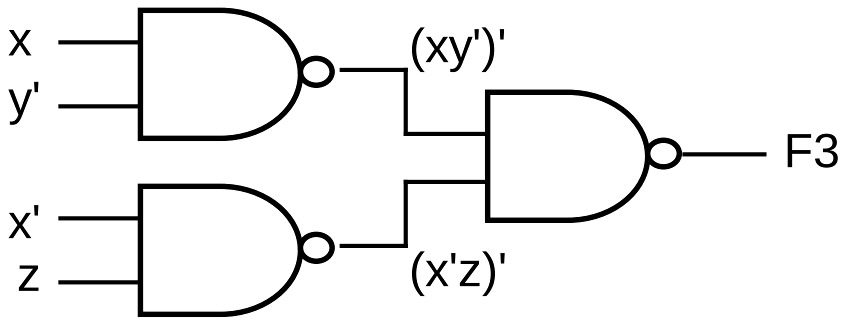
  Given $F3 = xy'+x'z$

  $= ((xy'+x'z)')'$ Involution

  => $F3 = ((xy')' . (x'z)')'$ DeMorgan

Cont..

F3 = ((xy')'.(x'z)')' = xy' + x'z



Implementation Using NOR Gates

- It is possible to implement any Boolean expression using NOR gates.

  Procedure:

  Step-1: Obtain the product-of-sums (POS) Boolean expression.

  Let the Boolean function is F6 = (x+y').(x'+z)

  Step-2: Use the DeMorgan's theorem to obtain the expression using two-level NOR gates.

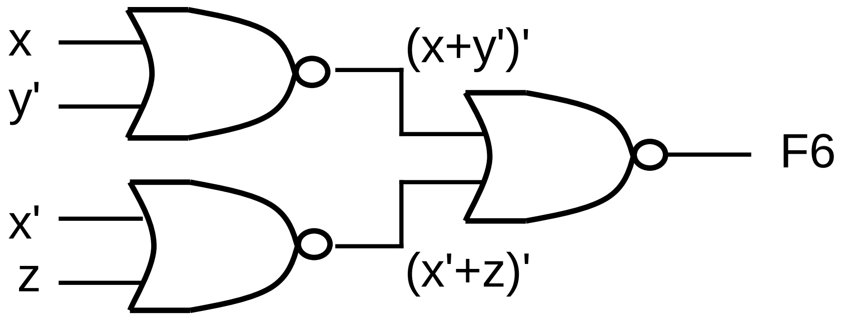
  Given F6 = (x+y').(x'+z)

  = (((x+y').(x'+z))')' Involution

  => F6 = ((x+y')'+(x'+z)')' DeMorgan

Cont..

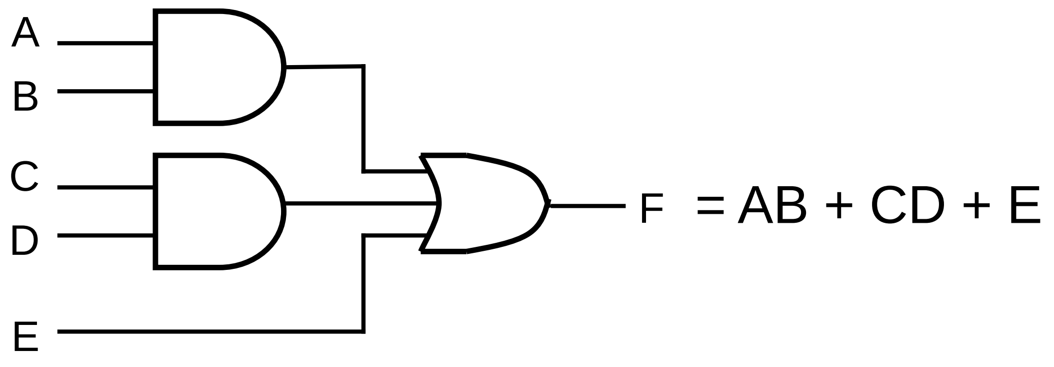
$$F6 = ((x+y')'+(x'+z)')' = (x+y').(x'+z)$$



Implementation of SOP Expressions

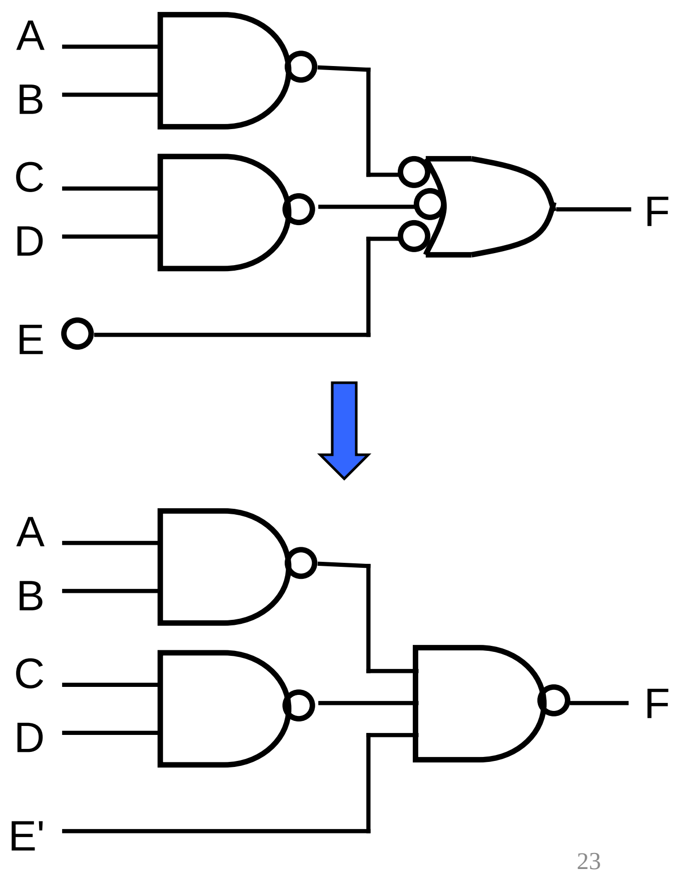
- Sum-of-Products (SOP) expressions can be implemented using:
  - ❖ Two-level AND-OR logic circuits
  - ❖ Two-level NAND logic circuits

❖ AND-OR logic circuit



Cont..

- ❖ NAND-NAND circuit (by circuit transformation)
  - a) Add double bubbles
  - b) Change OR-with-inverted-inputs to NAND and bubbles at inputs to their complements

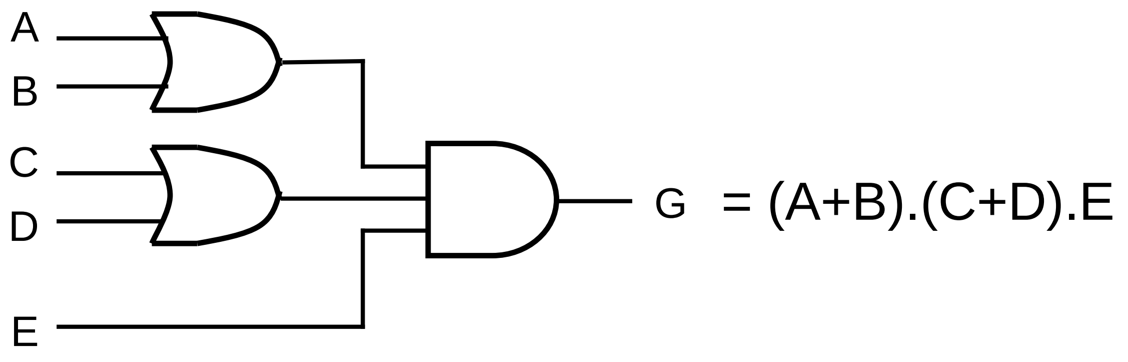


Implementation of POS Expressions

- Product-of-Sums (POS) expressions can be implemented using:
  - ❖ Two-level OR-AND logic circuits
  - ❖ Two-level NOR logic circuits

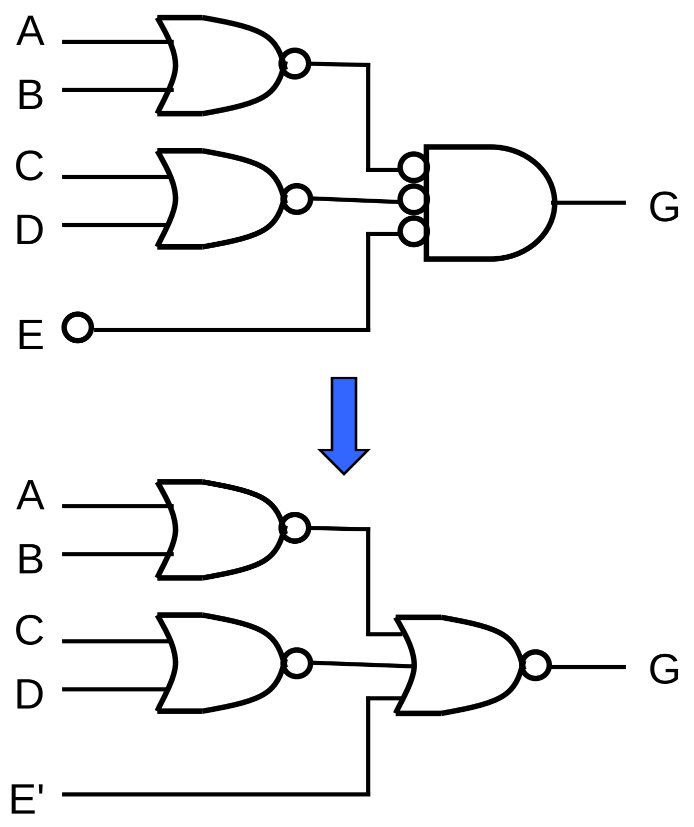
❖ OR-AND logic circuit

Let G = (A+B).(C+D).E



Cont..

- ❖ NOR-NOR circuit (by circuit transformation):
  - a) Add double bubbles
  - b) Change AND-with-inverted-inputs to NOR and bubbles at inputs to their complements



References

- M. M. Mano, *Digital Logic and Computer Design*, 5th ed., Pearson Prentice Hall, 2013.
- R. P. Jain, *Modern Digital Electronics*, 4th ed., Tata McGraw-Hill Education, 2009.