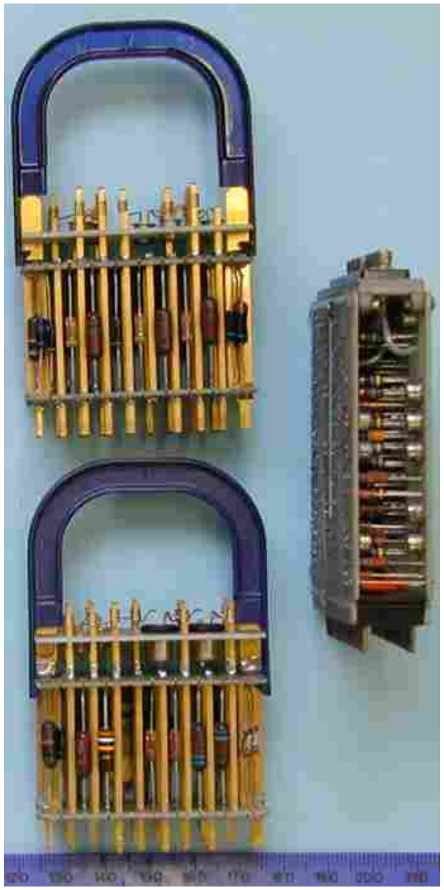
Lecture 10
Digital Logic Design
-- Basics

Logic Design

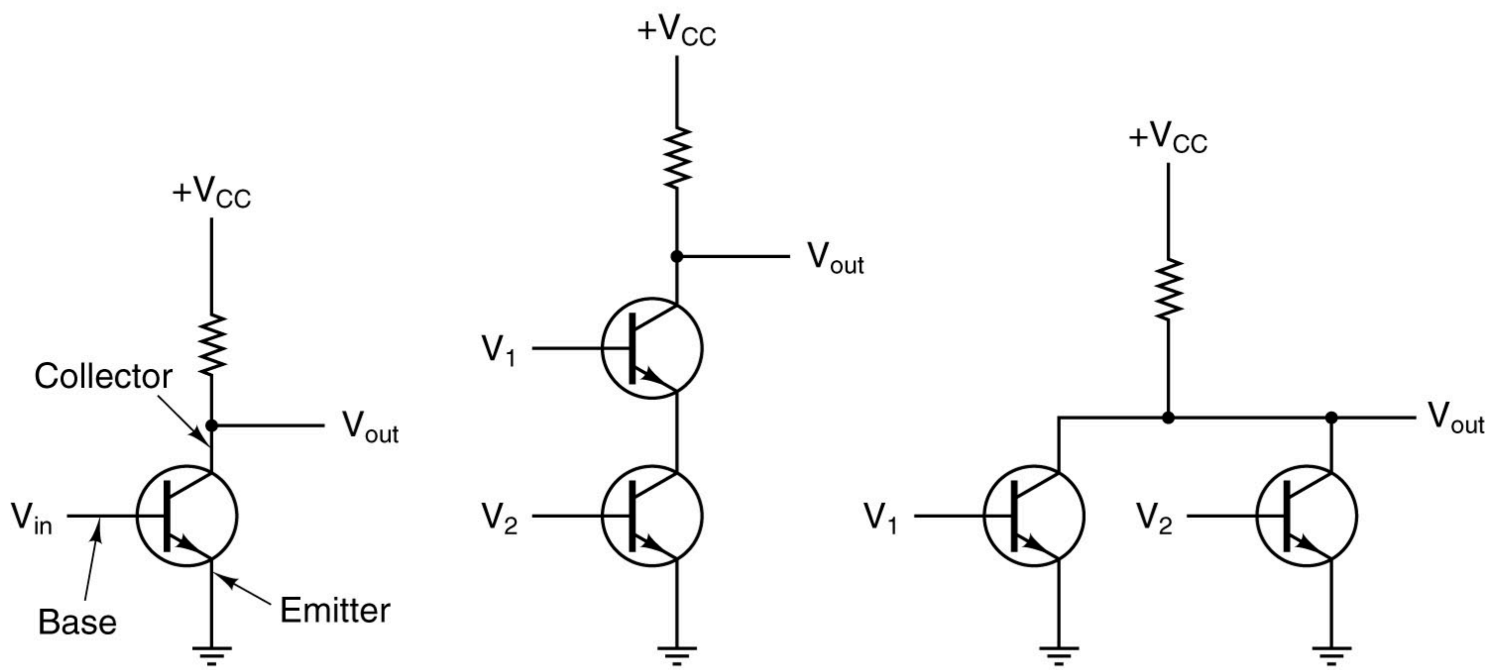
- From now, we'll study how a modern processor is built starting with basic logic elements as building blocks.
- Why study logic design?
  - Understand what processors can do fast and what they can't do fast (avoid slow things if you want your code to run fast!)
  - Background for more detailed hardware courses

Logic Gates

- Basic building blocks are logic *gates*.
  - In the beginning, did ad hoc designs, and then saw patterns repeated, gave names
  - Can build gates with transistors and resistors
- Then found theoretical basis for design
  - Can represent and reason about gates with truth tables and Boolean algebra
  - Assume know truth tables and Boolean algebra from a math or circuits course.
  - Section B.2 in the textbook has a review



Gates



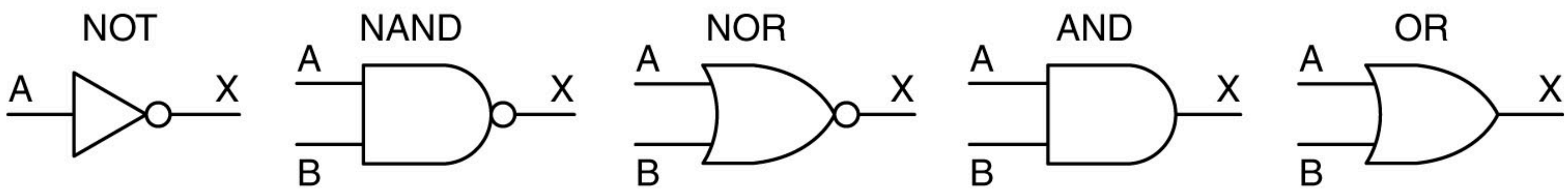
(a) A transistor inverter or A NOT gate

(b) A NAND gate

(c) A NOR gate

Gates and Boolean Algebra

Five basic gates



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

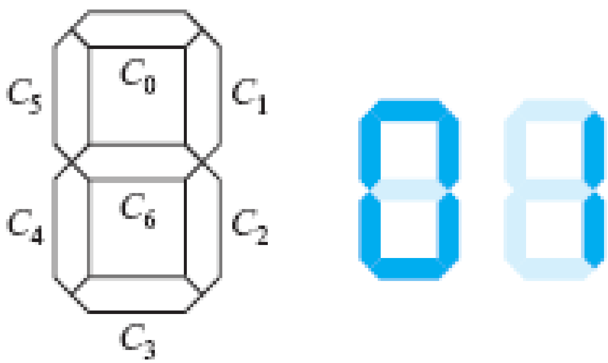
(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

Truth Table

Digit Display



X	C_0	C_1	C_2	C_3	C_4	C_5	C_6
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0

Figure 2.4 Binary digit-display combinational system.

Building a Circuit

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

- Use And/Or gates to implement this expression
- How many gates do we use?
- Can we use less

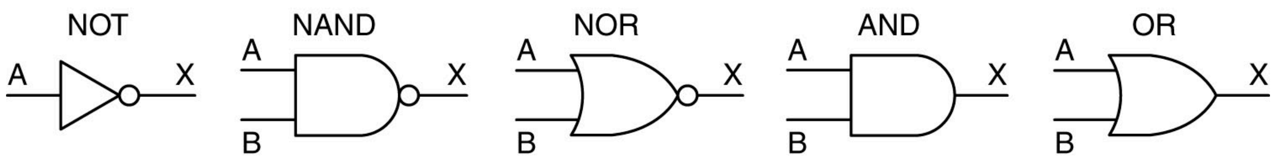
$$\begin{aligned} &= BC_{in} + AC_{in} + AB(\bar{C}_{in} + C_{in}) \\ &= BC_{in} + AC_{in} + AB(1) \\ &= BC_{in} + AC_{in} + AB \end{aligned}$$

Boolean Algebra 101

How to simplify Boolean expression?

Boolean Algebra (1)

- Using just ‘0’ and ‘1’s (after George Boole)
- Truth Table
  - A Boolean function with n variables has only 2^n possible input combinations
  - A table with 2^n rows



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

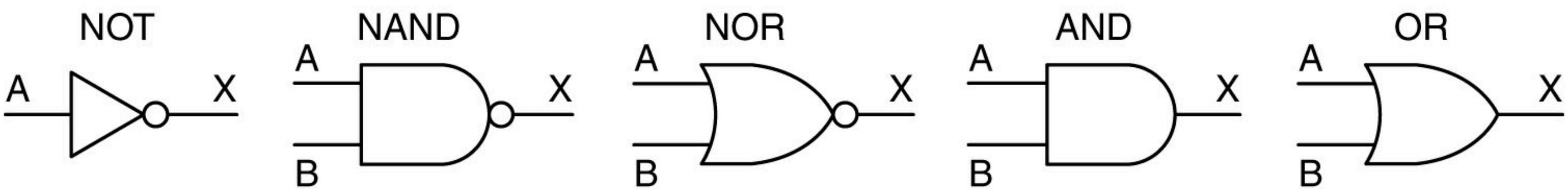
(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

Boolean Algebra (2)

- Algebra representations



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

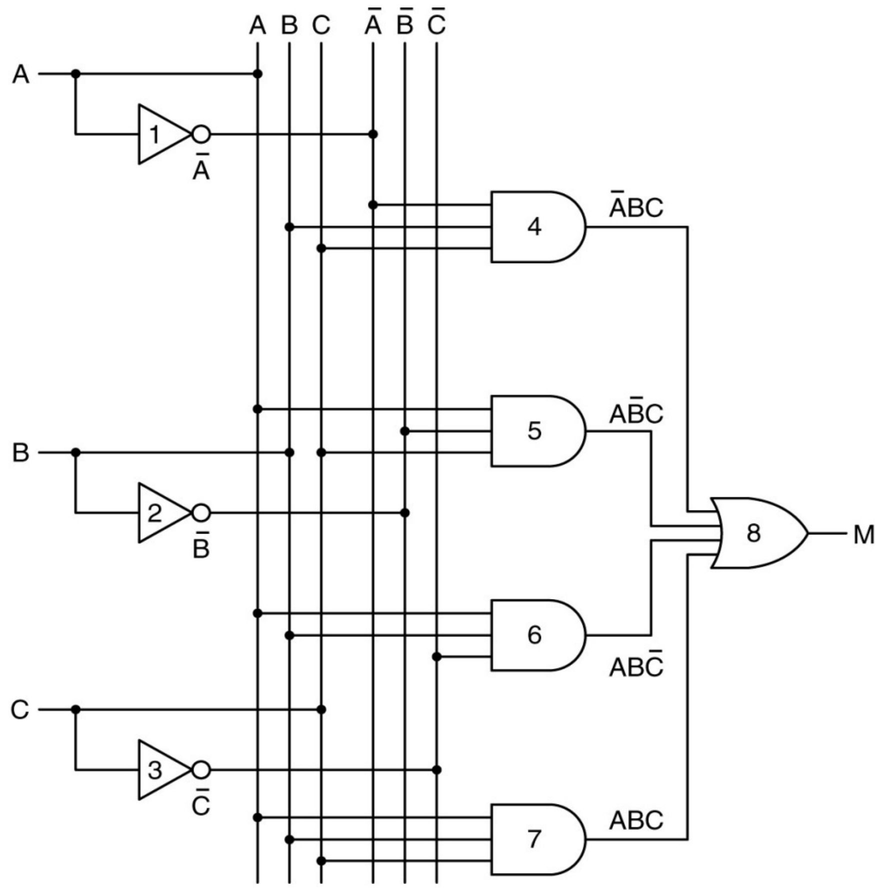
(e)

Boolean Algebra (3)

Another example with 3 inputs

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(a)



(b)

Boolean Algebra (4)

- All Boolean functions can be implemented by logic gates (AND, OR, NOT) and vice versa.

- Boolean operations
  - AND: XY, X ^ Y, X &Y
  - OR: X+Y, X v Y, X | Y
  - NOT: $\overline{X}$, X’, /X, \X

$$\overline{A} \bullet B + C = ((\overline{A}) \bullet B) + C$$

$$\overline{A} + B \bullet C = (\overline{A}) + (B \bullet C)$$

- Order: NOT, AND, OR

Basics

- P1: X = 0 or X = 1
- P2: 0 . 0 = 0
- P3: 1 + 1 = 1
- P4: 0 + 0 = 0
- P5: 1 . 1 = 1
- P6: 1 . 0 = 0 . 1 = 0
- P7: 1 + 0 = 0 + 1 = 1

Boolean Laws

- T1 : Commutative Law
  - A + B = B + A
  - A B = B A
- T2 : Associate Law
  - (A + B) + C = A + (B + C)
  - (A B) C = A (B C)

Boolean Laws

- T3 : Distributive Law
  - A (B + C) = A B + A C
  - A + (B C) = (A + B) (A + C)
- T4 : Identity Law
  - A + A = A
  - A A = A

Laws

- T6 : Redundancy Law
  - A + A B = A
  - A (A + B) = A
- T7
  - 0 + A = A
  - 0 A = 0
- T8
  - 1 + A = 1
  - 1 A = A

Other Laws

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Example

$$C_{out} = \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

$$= \bar{A}BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in} + ABC_{in}$$

$$= \bar{A}BC_{in} + ABC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

Idempotent theorem + commutative law

$$= (\bar{A} + A)BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

distributive law

$$= (1)BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

complementarity law

$$= BC_{in} + A\bar{B}C_{in} + AB\bar{C}_{in} + ABC_{in}$$

identity law

$$= BC_{in} + AC_{in} + AB(\bar{C}_{in} + C_{in})$$

$$= BC_{in} + AC_{in} + AB(1)$$

$$= BC_{in} + AC_{in} + AB$$

Question 1

Simplify the Boolean expression

(A+B+C)(D+E)' + (A+B+C)(D+E) and choose the best answer.

- A + B + C
- D + E
- A'B'C'
- D'E'
- None of the above

Question 2

Given the function $F(X,Y,Z) = XZ + Z(X' + XY)$, the equivalent most simplified Boolean representation for F is:

- $Z + YZ$
- $Z + XYZ$
- XZ
- $X + YZ$
- None of the above

Question 3

An equivalent representation for the Boolean expression $A' + 1$ is

- A
- A'
- 1
- 0

Question 4

Simplification of the Boolean expression

AB + ABC + ABCD + ABCDE + ABCDEF

yields which of the following results?

- ABCDEF
- AB
- AB + CD + EF
- A + B + C + D + E + F
- A + B(C+D(E+F))

Question 5

Given that F = A'B'+ C'+ D'+ E', which of the following represent the only correct expression for **F'**?

- F'= A+B+C+D+E
- F'= ABCDE
- F'= AB(C+D+E)
- F'= AB+C'+D'+E'
- F'= (A+B)CDE

Deriving Complement of a Boolean Expression

- DeMorgan's LAW: **The complement of a Boolean expression is formed by replacing all literals by their compliments; ANDs become ORs, ORs become ANDs**

$$\overline{X + Y} = \overline{X} \bullet \overline{Y} \qquad \overline{X \bullet Y} = \overline{X} + \overline{Y}$$

Using DeMorgan's Law

$$\overline{X + Y} = \overline{X} \bullet \overline{Y} \qquad \overline{X \bullet Y} = \overline{X} + \overline{Y}$$

$$\overline{X + Y \bullet Z} = ?$$

Question 5

Given that F = A'B'+ C'+ D'+ E', which of the following represent the only correct expression for F'?

- F'= A+B+C+D+E
- F'= ABCDE
- F'= AB(C+D+E)
- F'= AB+C'+D'+E'
- F'= (A+B)CDE

Question 6

Given the function F(A,B,X,Y) = AB + X'Y, the most simplified Boolean representation for F' is

- (AB)' + (X'Y)
- A'B' + XY'
- (A'+ B')(X + Y')
- (AB + X'Y)'
- (AB)'(X'Y)'

Question 7

Given the function F(X,Y,Z) = XZ + Z(X'+ XY), the equivalent, most simplified Boolean representation for F is

- Z + YZ
- Z + XYZ
- XZ
- Z
- none of the above

Question 8

Simplification of the Boolean expression AB + A (BC)' yields which of following results?

- A
- BC
- B
- AB
- (BC)'