Boolean Algebra

- Also known as *Switching Algebra*
  - › Invented by mathematician George Boole in 1849
  - › Used by Claude Shannon at Bell Labs in 1938
    - To describe digital circuits built from relays
- Digital circuit design is based on
  - › Boolean Algebra
    - Attributes
    - Postulates
    - Theorems
  - › These allow minimization and manipulation of logic gates for optimizing digital circuits

Boolean Algebra Attributes

- Binary
  - A1a: $X=0$ if $X\neq 1$
  - A1b: $X=1$ if $X\neq 0$
- Complement
  - aka *invert*, *NOT*
  - A2a: if $X=0$, $X'=1$
  - A2b: if $X=1$, $X'=0$
    - The tick mark ’ means complement, invert, or NOT
    - Other symbol for complement: $X'=\overline{X}$

X	X’
0	1
1	0

- AND operation
  - A3a: 0•0=0
  - A4a: 1•1=1
  - A5a: 0•1=1•0=0
    - The dot • means AND
    - Other symbol for AND: X•Y=XY (*no symbol*)

X	Y	X•Y
0	0	0
0	1	0
1	0	0
1	1	1

- OR operation
  - A3b: 1+1=1
  - A4b: 0+0=0
  - A5b: 1+0=0+1=1
    - The plus + means OR

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Algebra Postulates

- Identity Elements
  - › P2a: X+0=X
  - › P2b: X•1=X
- Commutativity
  - › P3a: X+Y=Y+X
  - › P3b: X•Y=Y•X
- Complements
  - › P6a: X+X'=1
  - › P6b: X•X'=0

OR operation

X	Y	X+0	X+Y	Y+X	X'	X+X'
0	0	0	0	0	1	1
0	1	0	1	1	1	1
1	0	1	1	1	0	1
1	1	1	1	1	0	1

AND operation

X	Y	X•1	X•Y	Y•X	X'	X•X'
0	0	0	0	0	1	0
0	1	0	0	0	1	0
1	0	1	0	0	0	0
1	1	1	1	1	0	0

Boolean Algebra Postulates

- Associativity
  - › P4a: (X+Y)+Z=X+(Y+Z)
  - › P4b: (X•Y)•Z=X•(Y•Z)

X	Y	Z	X+Y	(X+Y)+Z	Y+Z	X+(Y+Z)	X•Y	(X•Y)•Z	Y•Z	X•(Y•Z)
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0	0	0	0
0	1	0	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	0	0	1	0
1	0	0	1	1	0	1	0	0	0	0
1	0	1	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1

Boolean Algebra Postulates

- Distributivity
  - › P5a: X+(Y•Z) = (X+Y)•(X+Z)
  - › P5b: X•(Y+Z) = (X•Y)+(X•Z)

X	Y	Z	X+Y	X+Z	(X+Y)•(X+Z)	Y•Z	X+(Y•Z)	X•Y	X•Z	X•Y+X•Z	Y+Z	X•(Y+Z)
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0	0	1	0
1	0	0	1	1	1	0	1	0	0	0	0	0
1	0	1	1	1	1	0	1	0	1	1	1	1
1	1	0	1	1	1	0	1	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Boolean Algebra Theorems

- Idempotency
  - › T1a: X+X=X
  - › T1b: X•X=X
- Null elements
  - › T2a: X+1=1
  - › T2b: X•0=0
- Involution
  - › T3: (X')'=$\overline{\overline{X}}$=X

X	Y	X+Y (OR)	X•Y (AND)	X+X	X•X	X+1	X•0	X'	X''
0	0	0	0	0	0	1	0	1	0
0	1	1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0	0	1
1	1	1	1	1	1	1	0	0	1

Boolean Algebra Theorems

- Absorption (aka *covering*)
  - T4a: X+(X•Y)=X
  - T4b: X•(X+Y)=X
  - T5a: X+(X'•Y)=X+Y
  - T5b: X•(X'+Y)=X•Y

OR AND

X	Y	X+Y	X•Y	X+ (X•Y)	X• (X+Y)	X'	X'•Y	X+ (X'•Y)	X'+Y	X• (X'+Y)
0	0	0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0	0
1	1	1	1	1	1	0	0	1	1	1

Boolean Algebra Theorems

- Absorption (aka *combining*)
  - T6a: $(X \bullet Y)+(X \bullet Y')=X$
  - T6b: $(X+Y) \bullet (X+Y')=X$

X	Y	X+Y (OR)	X•Y (AND)	Y'	X•Y'	(X•Y)+(X•Y')	X+Y'	(X+Y)•(X+Y')
0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0
1	0	1	0	1	1	1	1	1
1	1	1	1	0	0	1	1	1

Boolean Algebra Theorems

- Absorption (aka *combining*)
  - › T7a: $(X \cdot Y)+(X \cdot Y' \cdot Z)=(X \cdot Y)+(X \cdot Z)$
  - › T7b: $(X+Y) \cdot (X+Y'+Z) = (X+Y) \cdot (X+Z)$

X Y Z	Y’	XY	XY’Z	(XY)+ (XY’Z)	XZ	(XY)+ (XZ)	X+Y	X+Y’ +Z	(X+Y)• (X+Y’+Z)	X+Z	(X+Y)• (X+Z)
0 0 0	1	0	0	0	0	0	0	1	0	0	0
0 0 1	1	0	0	0	0	0	0	1	0	1	0
0 1 0	0	0	0	0	0	0	1	0	0	0	0
0 1 1	0	0	0	0	0	0	1	1	1	1	1
1 0 0	1	0	0	0	0	0	1	1	1	1	1
1 0 1	1	0	1	1	1	1	1	1	1	1	1
1 1 0	0	1	0	1	0	1	1	1	1	1	1
1 1 1	0	1	0	1	1	1	1	1	1	1	1

Boolean Algebra Theorems

- DeMorgan's theorem (very important!)
  - T8a: (X+Y)'= X'•Y'
    - $\overline{X+Y} = \overline{X} \bullet \overline{Y}$ break (or connect) the bar & change the sign
  - T8b: (X•Y)'= X'+Y'
    - $\overline{X \bullet Y} = \overline{X} + \overline{Y}$ break (or connect) the bar & change the sign
  - Generalized DeMorgan's theorem:
    - GT8a: $(X_1+X_2+\ldots+X_{n-1}+X_n)' = X_1'\bullet X_2'\bullet\ldots\bullet X_{n-1}'\bullet X_n'$
    - GT8b: $(X_1\bullet X_2\bullet\ldots\bullet X_{n-1}\bullet X_n)' = X_1'+X_2'+\ldots+X_{n-1}'+X_n'$

X	Y	X+Y (OR)	X•Y (AND)	X'	Y'	(X+Y)'	X'•Y'	(X•Y)'	X'+Y'
0	0	0	0	1	1	1	1	1	1
0	1	1	0	1	0	0	0	1	1
1	0	1	0	0	1	0	0	1	1
1	1	1	1	0	0	0	0	0	0

Boolean Algebra Theorems

- Consensus Theorem
  - T9a: (X•Y)+(X'•Z)+(Y•Z) = (X•Y)+(X'•Z)
  - T9b: (X+Y)•(X'+Z)•(Y+Z) = (X+Y)•(X'+Z)

X Y Z	X'	XY	X'Z	YZ	(XY)+ (X'Z)+ (YZ)	(XY)+ (X'Z)	X+Y	X'+Z	Y+Z	(X+Y)• (X'+Z)• (Y+Z)	(X+Y)• (X'+Z)
0 0 0	1	0	0	0	0	0	0	1	0	0	0
0 0 1	1	0	1	0	1	1	0	1	1	0	0
0 1 0	1	0	0	0	0	0	1	1	1	1	1
0 1 1	1	0	1	1	1	1	1	1	1	1	1
1 0 0	0	0	0	0	0	0	1	0	0	0	0
1 0 1	0	0	0	0	0	0	1	1	1	1	1
1 1 0	0	1	0	0	1	1	1	0	1	0	0
1 1 1	0	1	0	1	1	1	1	1	1	1	1

More Theorems?

- Shannon's expansion theorem (also very important!)
  - T10a: $f(X_1,X_2,\ldots,X_{n-1},X_n)=$
    $(X_1'\bullet f(0,X_2,\ldots,X_{n-1},X_n))+(X_1\bullet f(1,X_2,\ldots,X_{n-1},X_n))$
    - Can be taken further:
      - $f(X_1,X_2,\ldots,X_{n-1},X_n)= (X_1'\bullet X_2'\bullet f(0,0,\ldots,X_{n-1},X_n))$
        $+ (X_1\bullet X_2'\bullet f(1,0,\ldots,X_{n-1},X_n)) + (X_1'\bullet X_2\bullet f(0,1,\ldots,X_{n-1},X_n))$
        $+(X_1\bullet X_2\bullet f(1,1,\ldots,X_{n-1},X_n))$
    - Can be taken even further:
      - $f(X_1,X_2,\ldots,X_{n-1},X_n)= (X_1'\bullet X_2'\bullet\ldots\bullet X_{n-1}'\bullet X_n'\bullet f(0,0,\ldots,0,0))$
        $+ (X_1\bullet X_2'\bullet\ldots\bullet X_{n-1}'\bullet X_n'\bullet f(1,0,\ldots,0,0)) + \ldots$
        $+ (X_1\bullet X_2\bullet\ldots\bullet X_{n-1}\bullet X_n\bullet f(1,1,\ldots,1,1))$
  - T10b: $f(X_1,X_2,\ldots,X_{n-1},X_n)=$
    $(X_1+f(0,X_2,\ldots,X_{n-1},X_n))\bullet(X_1'+ f(1,X_2,\ldots,X_{n-1},X_n))$
    - Can be taken further as in the case of T10a
- We'll see significance of Shannon's expansion theorem later

Principle of Duality

- Any theorem or postulate in Boolean algebra remains true if:
  - › 0 and 1 are swapped, ***and***
  - › • and + are swapped
    - \- **BUT**, be careful about operator precedence!!!
- Operator precedence order:
  1) Left-to-right
  2) Complement (NOT)
  3) AND
  4) OR
- Use parentheses liberally to ensure correct Boolean logic equation

Postulates w/ Precedence & Duality

P.	a. expression	b. dual
2	a+0=a	a•1=a
3	a+b=b+a	a•b=b•a
4	(a+b)+c=a+(b+c)	(a•b)•c=a•(b•c)
5	a+(b•c) = (a+b)•(a+c)	a•(b+c) = (a•b)+(a•c)
6	a+a’=1	a•a’=0

Theorems w/ Precedence & Duality

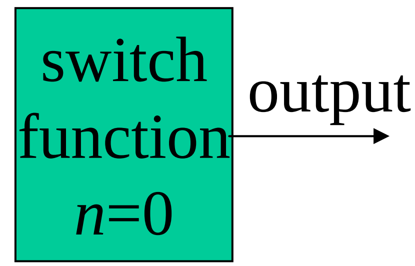
Th.	a. expression	b. dual
1	a+a=a	a•a=a
2	a+1=1	a•0=0
3	a''=a	
4	a+ab=a	a(a+b)=a
5	a+a'b=a+b	a(a'+b)=ab
6	ab+ab'=a	(a+b)(a+b')=a
7	ab+ab'c=ab+ac	(a+b)(a+b'+c)=(a+b)(a+c)
8	(a+b)'=a'b'	(ab)'=a'+b'
9	ab+a'c+bc=ab+a'c	(a+b)(a'+c)(b+c)= (a+b)(a'+c)
10	$f(X)=x_1{}'f(0,\ldots,x_n)+x_1 f(1, \ldots,x_n)$	$f(X)=(x_1+f(0,\ldots,x_n))(x_1{}'+ f(1, \ldots,x_n))$

Switching Functions

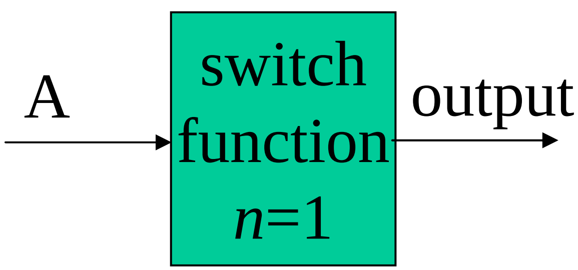
- For *n* variables, there are 2^n possible combinations of values
  - › From all 0s to all 1s
- There are 2 possible values for the output of a function of a given combination of values of *n* variables
  - › 0 and 1
- There are 2^{2^n} different switching functions for *n* variables

Switching Function Examples

- n=0 (no inputs) ⇨ $2^{2^n} = 2^{2^0} = 2^1 = 2$
  - › Output can be either 0 or 1



- n=1 (1 input, A) ⇨ $2^{2^n} = 2^{2^1} = 2^2 = 4$
  - › Output can be 0, 1, A, or A'

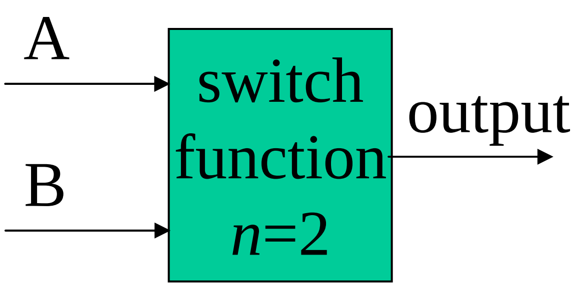


A	f_0	f_1	f_2	f_3
0	0	1	0	1
1	0	0	1	1

$f_0 = 0$
$f_1 = A'$
$f_2 = A$
$f_3 = 1$

Switching Function Examples

- n=2 (2 inputs, A and B) ➡ $2^{2^n} = 2^{2^2} = 2^4 = 16$



A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

$f_0 = 0$ — logic 0

$f_1 = A'B' = (A+B)'$ — NOT-OR or NOR

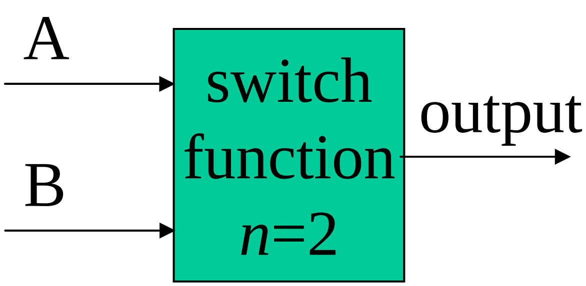
$f_2 = A'B$

$f_3 = A'B'+A'B = A'(B'+B) = A'$ — invert A

Most frequently used *Less frequently used* *Least frequently used*

Switching Function Examples

- n=2 (2 inputs, A and B) ➡ $2^{2^n} = 2^{2^2} = 2^4 = 16$



A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

f_4 = AB'

f_5 = A'B'+AB' = (A'+A)B' = B' — invert B

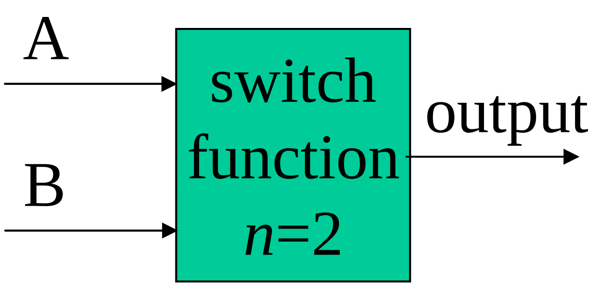
f_6 = A'B+AB' — exclusive-OR

f_7 = A'B'+A'B+AB' = A'(B'+B)+(A'+A)B'
= A'+B' = (AB)' — NOT-AND or NAND

Most frequently used *Less frequently used* *Least frequently used*

Switching Function Examples

- n=2 (2 inputs, A and B) ➡ $2^{2^n} = 2^{2^2} = 2^4 = 16$



A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

f_8 = AB — AND

f_9 = A'B'+AB — exclusive-NOR

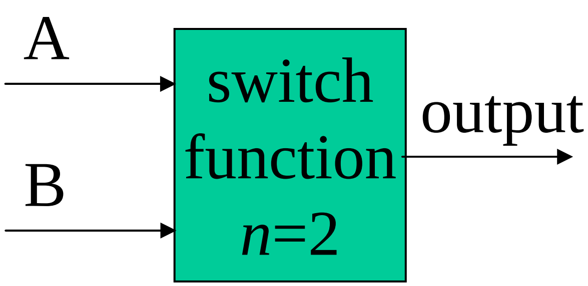
f_{10} = A'B+AB = (A'+A)B = B — buffer B

f_{11} = A'B'+A'B+AB = A'(B'+B)+(A'+A)B = A'+B

Most frequently used *Less frequently used* *Least frequently used*

Switching Function Examples

- n=2 (2 inputs, A and B) ⇨ $2^{2^n} = 2^{2^2} = 2^4 = 16$



A B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0 0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0 1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

f_{12} = AB'+AB = A(B'+B) = A — buffer A

f_{13} = A'B'+AB'+AB = A(B'+B)+A'B' = A+A'B' = A+B'

f_{14} = A'B+AB'+AB = A(B'+B)+(A'+A)B = A+B — OR

f_{15} = A'B'+A'B+AB'+AB = A'(B'+B)+A(B'+B)
= A'+A = 1 — logic 1

Most frequently used *Less frequently used* *Least frequently used*