CMP 338: *Fourth Class*

HW 3 solution
Integrate circuit manufacture and cost
Boolean Algebra and Truth Tables
“Black Box” circuit design
Performance metrics
Performance and execution time
Relative performance
The CPU Time equation
The TINY instruction set architecture

For next class:

begin HW 4; review 1.6, 1.10; read A.1-2, 2.1-3

HW 3 part 1: Base Conversion

Convert the following to the indicated base:

$10110111011110_2 = 2DDE_{16}$

$4C1F91_{16} = 0100\ 1100\ 0001\ 1111\ 1001\ 0001_2$

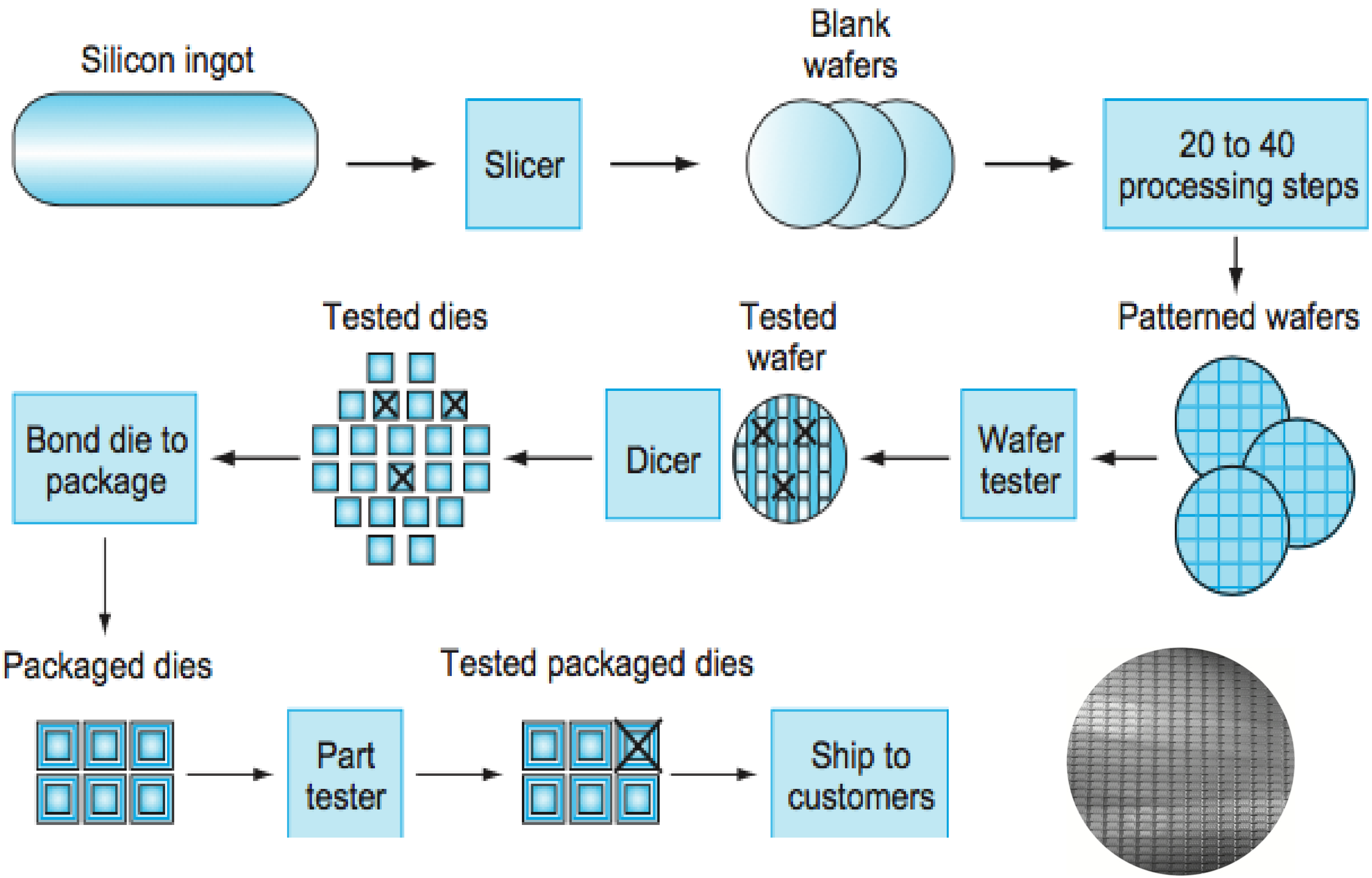
$100_{10} = 64 + 32 + 4 = 2^6 + 2^5 + 2^2 = 01100100_2$

$10001010_2 = 2^7 + 2^3 + 2^1 = 128 + 8 + 2 = 138_{10}$

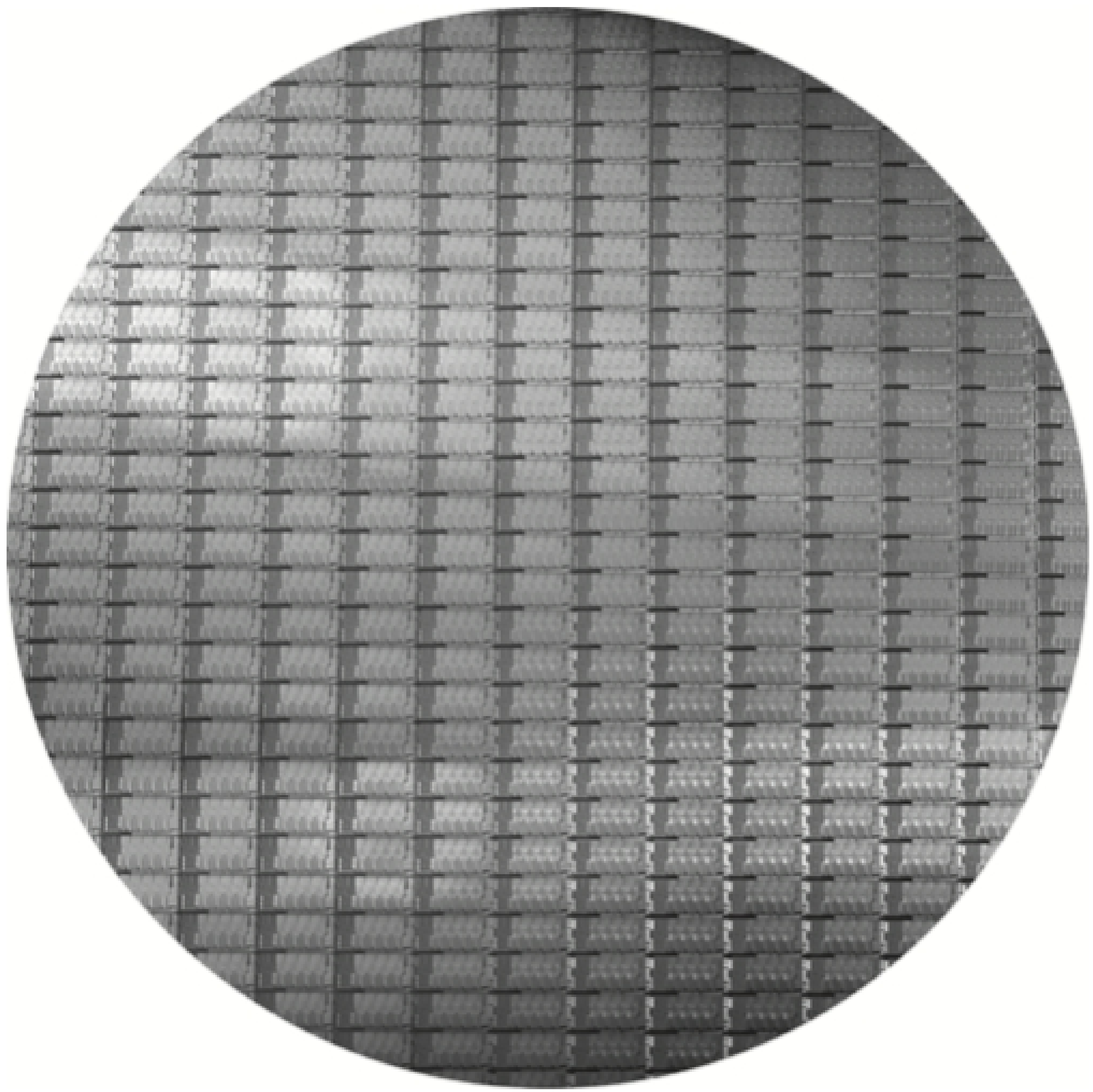
$1F3_{16} = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 256 + 240 + 3 = 499_{10}$

$1055_{10} = 1024 + 16 + 8 + 4 + 2 + 1 = 10000011111_2 = 41F_{16}$

Chip Fabrication



Intel Core I7 Wafer



Integrated Circuit Fabriction Costs

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{yield}}$$

$$\text{Dies per wafer} \approx \frac{\text{Wafer area}}{\text{Die area}}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

11.8 inch (300mm) patterned wafer

~325 (20.7 x 10.5 mm) dies per wafer

~23% of dies are defective (yield = ~0.77)

If a wafer costs $20,000

what is the fabrication cost of a die (chip)?

HW 3 part 2: Fabrication Cost

A processor fabrication plant gets 400 processor chips to a wafer with a yield of 80%. If the chip fabrication cost is \$90, what is the cost of the wafer?

$$\mathrm{chip}_{\mathrm{cost}} = \frac{\mathrm{wafer}_{\mathrm{cost}}}{\mathrm{chipsPerWafer} \cdot \mathrm{yield}}$$

$$\begin{aligned}\mathrm{wafer}_{\mathrm{cost}} &= \mathrm{chip}_{\mathrm{cost}} \cdot \mathrm{chipsPerWafer} \cdot \mathrm{yield} \\ &= \$90 \cdot 400 \cdot 0.8 = \$28{,}800\end{aligned}$$

Computer Design Big Picture

A computer is one big ***sequential*** circuit

- **Abstract** into discrete sequential components
  - ***Combinational*** circuits + memory + clock

Combinational circuit design

1. Specify semantics
   - Black Box input and output
   - *Truth Table* (Input determines output)
2. Truth table → *Boolean formula*
3. Minimize boolean formula (Karnaugh Maps)
4. Boolean formula → combinational circuit

Boolean Algebra

Constants: **0** and **1** (**F**alse, **T**rue)

Operators: *not* ($\overline{\ }$, ~), *and* (●, &), *or* (**+**, |)

A	$\overline{\mathbf{A}}$
0	**1**
1	**0**

A	**B**	**A + B**	**A ● B**
0	0	**0**	**0**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**1**	**1**

Boolean Formulas

Constant — 0 (False) or 1 (True)

Formula — either

- Constant , or ***Variable***, or
- Conjunction, Disjunction, or Negation of formulas

Literal — a variable or its negation

Term — conjunction of literals

Clause — disjunction of terms

Disjunctive Normal Form theorem:

- Every formula can be written as a single clause

Truth Tables for Boolean Formulas

Columns

- Input — variables
- Output — formula(s)
- Intermediate — sub-formulas

Rows

- 1 for every possible combination of input values
  - (in ascending order of input values)

Cells — constants (0 or 1)

- Use ***not***, ***and***, or ***or*** table on cell(s) in same row
- For Boolean identities — add ***equality*** operator

$$\mathit{formula}_1 \mathbf{=} \mathit{formula}_2$$

Truth Tables for Boolean Identities

A	B	A = B
0	0	1
0	1	0
1	0	0
1	1	1

Columns

- Input — variables
- Output — formula(s)
- Intermediate — sub-formulas

Rows

- 1 for every possible combination of input values
  - (in ascending order of input values)

Cells — constants (0 or 1)

- Use ***not***, ***and***, or ***or*** table on cell(s) in same row
- For Boolean identities — add ***equality*** operator

$$formula_1 \mathbf{=} formula_2$$

Proof by Truth Table

$$X \bullet (Y + Z) = (X \bullet Y) + (X \bullet Z)$$

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

Proof by Truth Table

$X \bullet (Y + Z) = (X \bullet Y) + (X \bullet Z)$

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

Some Boolean Identities

$$\overline{\overline{X}} = X$$

$X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z$

$X + (Y + Z) = (X + Y) + Z$

$X \bullet 1 = X$

$X + 0 = X$

$X \bullet \overline{X} = 0$

$X + \overline{X} = 1$

$X \bullet Y = Y \bullet X$

$X + Y = Y + X$

$X \bullet X = X$

$X + X = X$

$X \bullet 0 = 0$

$X + 1 = 1$

$(X \bullet Y) \bullet X = X \bullet Y$

$(X + Y) + X = X + Y$

$X \bullet (Y + Z) = (X \bullet Y) + (X \bullet Z)$

$X + (Y \bullet Z) = (X + Y) \bullet (X + Z)$

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

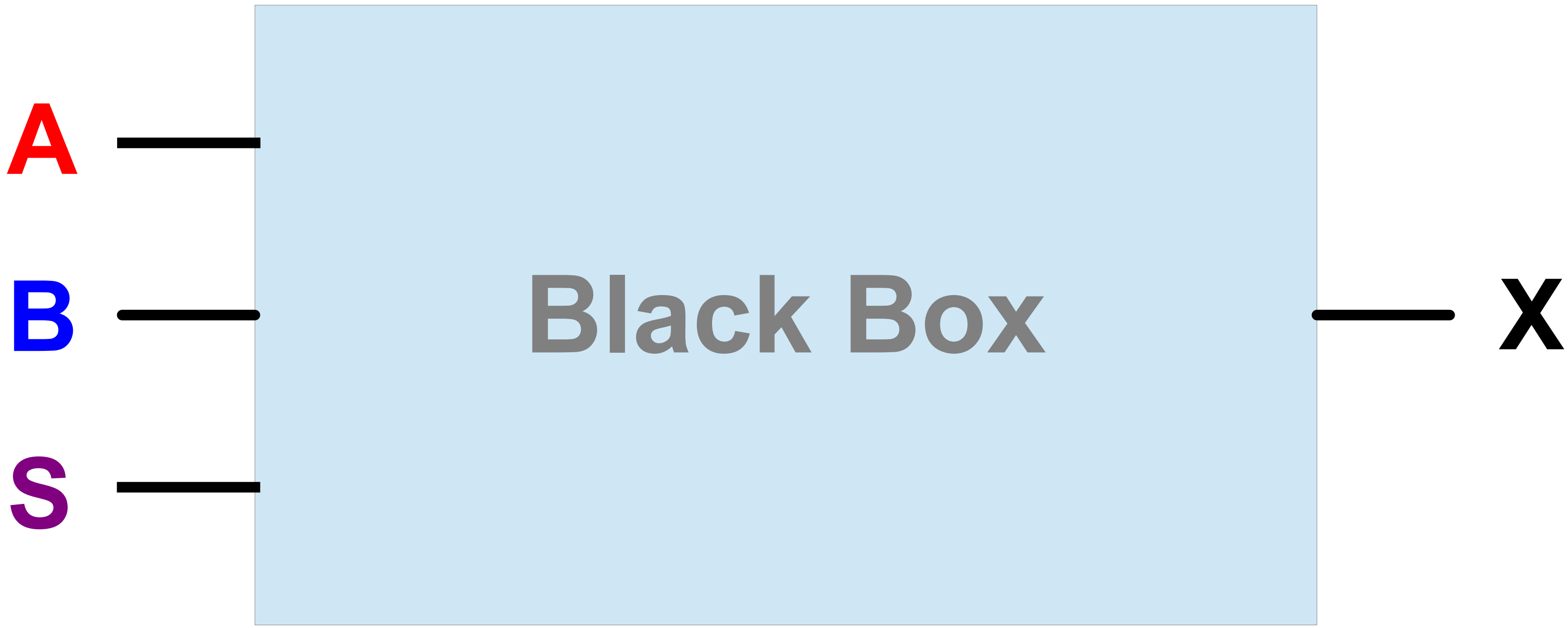
$\overline{X + Y} = \overline{X} \bullet \overline{Y}$

Proof by Truth Table

$$(X \bullet \overline{Y}) + (Y \bullet \overline{X}) = (X + Y) \bullet (\overline{X} + \overline{Y})$$

x	y	$\overline{x}$	$\overline{y}$	$x \bullet \overline{y}$	$y \bullet \overline{x}$	$(x \bullet \overline{y}) + (y \bullet \overline{x})$	$(x+y) \bullet (\overline{y} + \overline{x})$	$x+y$	$\overline{y} + \overline{x}$
0	0	1	1	0	0	**0**	**0**	0	1
0	1	1	0	0	1	**1**	**1**	1	1
1	0	0	1	1	0	**1**	**1**	1	1
1	1	0	0	0	0	**0**	**0**	1	0

Two Way Multiplexer Design



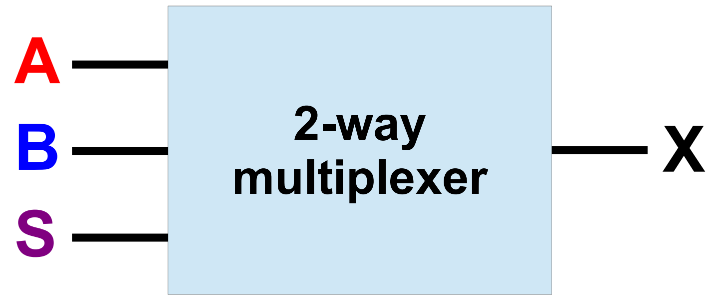
Informal semantics:

X = **A** — *if* **S** = 0

X = **B** — *if* **S** = 1

Two Way Multiplexer Truth Table

S	A	B	X
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**



not S and A and not B, or
not S and A and B, or
S and not A and B, or
S and A and B

$$X = \overline{S}A\overline{B} + \overline{S}AB + S\overline{A}B + SAB$$
$$= \overline{S}A + SB$$

Truth Table → Boolean Formula

Ignore table rows where output is 0

For each remaining row

- Construct a term that is true only for that row
  - For each input variable *v* include a literal that is
    - ***v*** — if the input for *v* in that row is 1
    - $\bar{\boldsymbol{v}}$ — if the input for *v* in that row is 0

The formula is the disjunction of these row terms

Note: formula is in disjunctive normal form

- (***full*** DNF — each term has literal for each variable)

Proves the disjunctive normal form theorem

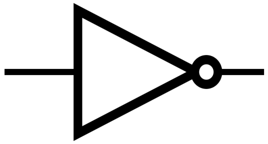
- Arbitrary formula ~> truth table ~> DNF formula

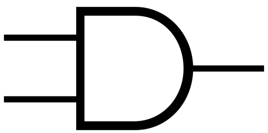
Boolean Formula → CombinationalCircuit

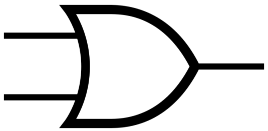
Input wire for each variable

For each sub-formula

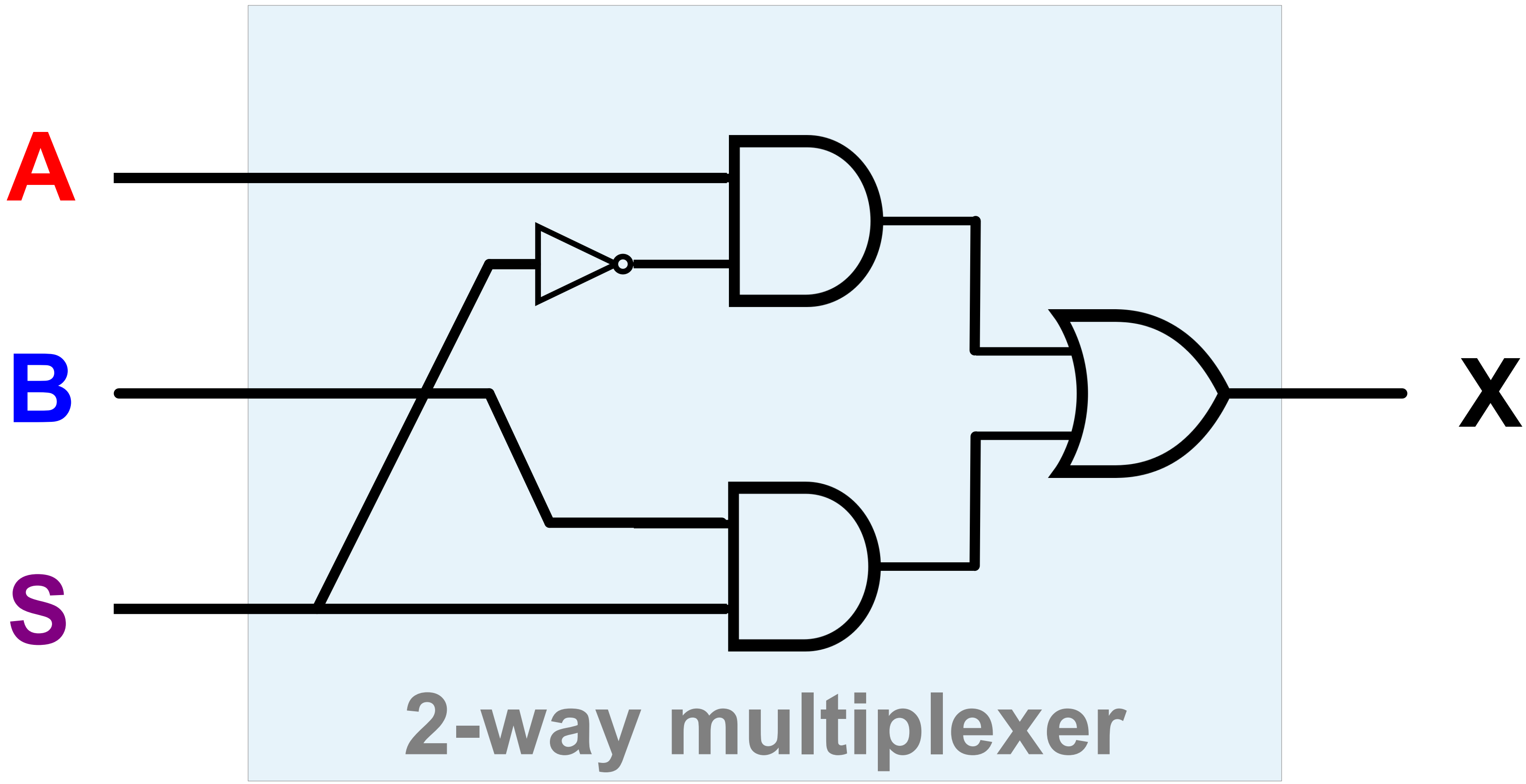
- Replace operand with wire (output from its sub-circuit)
- Replace operator with gate with output wire

~ becomes 

& becomes 

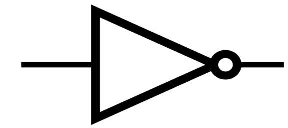
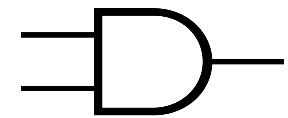
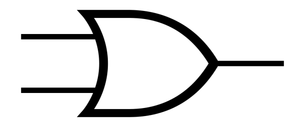
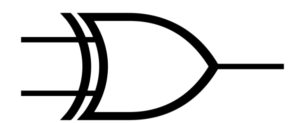
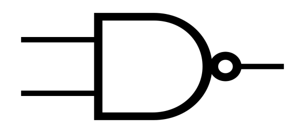
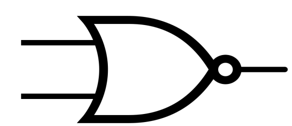
| becomes 

Two Way Multiplexer Circuit



$$X = \overline{S}A + SB$$

Boolean Operators & Gates

not A	**~ A**	$\overline{\mathbf{A}}$	**A** 
A and B	**A & B**	**A • B**	**A B** 
A or B	**A | B**	**A + B**	**A B** 
A xor B	**A ^ B**	**A ≠ B**	**A B** 
A nand B	**A ↑ B**	**A ↑ B**	**A B** 
A nor B	**A ↓ B**	**A ↓ B**	**A B** 
A xnor B	**A = B**	**A = B**	**A B**