NUMBER CONVERSIONS:

- Decimal to any other base:
  - Divide by the ‘base’ and keep track of the remainder.
  - Let the remainders topple over to the right. I.E. -> Start writing from the bottom.
- Any base to decimal:
  - Take the value times the base to the corresponding power; the power of 0 starts from the right.
- Any base other than decimal to binary:
  - Convert to decimal and then convert.
  - For octal, each number is represented by THREE binary digits.
  - For hexadecimal, each number is represented by FOUR binary digits.
- Hexadecimal to octal:
  - Write each digit of hexadecimal in binary, then split each four digit-binary to three digit binary -> octal.

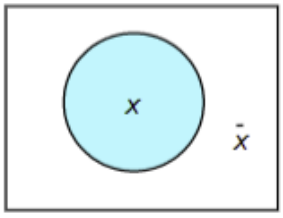
BOOLEAN ALGEBRA LAWS AND THEOREMS:

- $x \bullet y = \overline{x} + \overline{y}$ **DUALITY**
- $x + y = \overline{x} \bullet \overline{y}$ **DUALITY**
- $x \bullet 1 = x$
- $x + 1 = 1$
- $x \bullet 0 = 0$
- $x + 0 = x$
- $x \bullet x = x$
- $x + x = x$
- $x \bullet (x + y) = x$ **ABSORPTION**
- $x + x \bullet y = x$ **ABSORPTION**
- $x \bullet y + x \bullet \overline{y} = x$ **COMBINING**
- $(x + y) \bullet (x + \overline{y}) = x$ **COMBINING**
- $\overline{x \bullet y} = \overline{x} + \overline{y}$ **DEMOGRAN’S**
- $\overline{x + y} = \overline{x} \bullet \overline{y}$ **DEMOGRAN’S**
- $x + \overline{x} \bullet y = x + y$
- $x \bullet (\overline{x} + y) = x \bullet y$
- $x \bullet y + y \bullet z + \overline{x} \bullet z = x \bullet y + \overline{x} \bullet z$ **CONSENSUS**
- $(x + y) \bullet (y + z) \bullet (\overline{x} + z) = (x + y) \bullet (\overline{x} + z)$ **CONSENSUS**

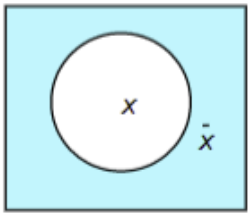
VENN DIAGRAMS:

(a) Constant 1

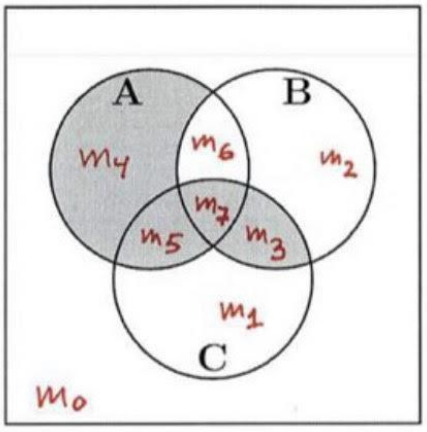
(b) Constant 0

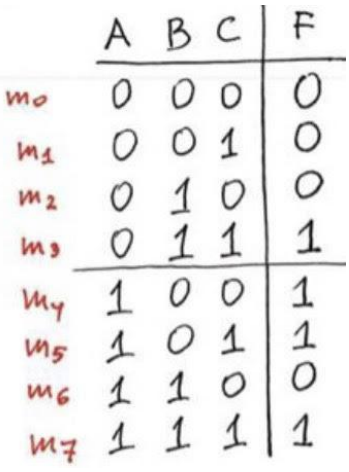


(c) Variable x



(d) $\bar{x}$





	A	B	C	F
m_0	0	0	0	0
m_1	0	0	1	0
m_2	0	1	0	0
m_3	0	1	1	1
m_4	1	0	0	1
m_5	1	0	1	1
m_6	1	1	0	0
m_7	1	1	1	1

- AND - combining only INTERSECTING ‘filled’ portions..
- OR - combining all ‘filled’ portions.

TRUTH TABLE & K-MAP TEMPLATES:

x_1	x_2	
0	0	m_0
0	1	m_1
1	0	m_2
1	1	m_3

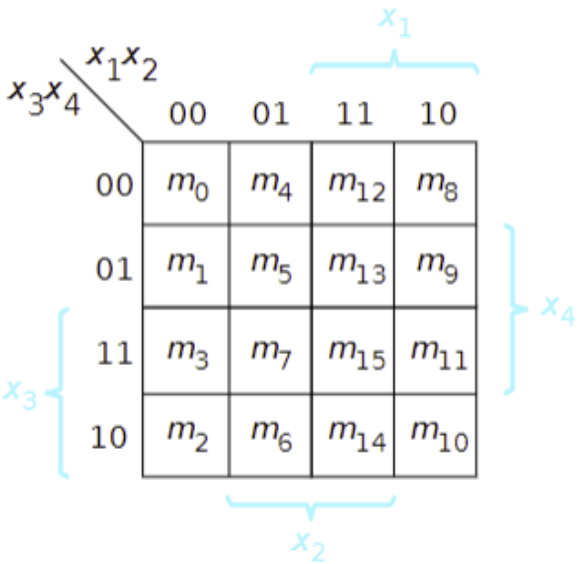
x_2 \ x_1	0	1
0	m_0	m_2
1	m_1	m_3

x_1	x_2	x_3	
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7

x_3 \ x_1x_2	00	01	11	10
0	m_0	m_2	m_6	m_4
1	m_1	m_3	m_7	m_5

(b) Karnaugh map

x1	x2	x3	x4	
0	0	0	0	m0
0	0	0	1	m1
0	0	1	0	m2
0	0	1	1	m3
0	1	0	0	m4
0	1	0	1	m5
0	1	1	0	m6
0	1	1	1	m7
1	0	0	0	m8
1	0	0	1	m9
1	0	1	0	m10
1	0	1	1	m11
1	1	0	0	m12
1	1	0	1	m13
1	1	1	0	m14
1	1	1	1	m15

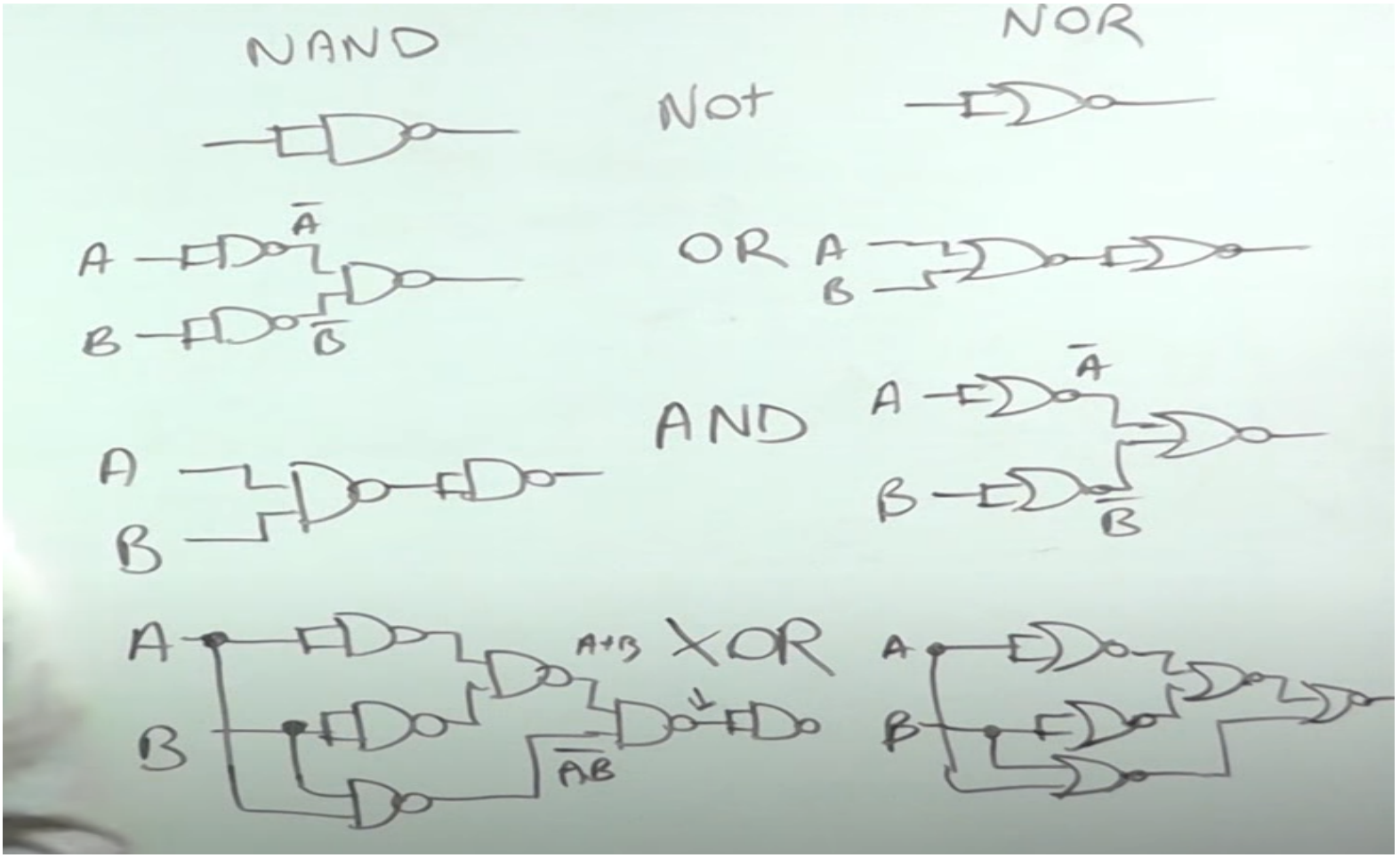


COST:

- Count the number of the gates.
- Count all wires to the gates (including NOT gates)..

NAND AND NOR GATES:

- NAND - SOP; just replace normal gates with NAND gates.
- NOR - POS; just replace normal gates with NOR gates.



VERILOG:

- Verilog HDL and VHDL are different hardware description languages.
- AND (&), OR (|), NOT (~), XOR (^).
- Ex)

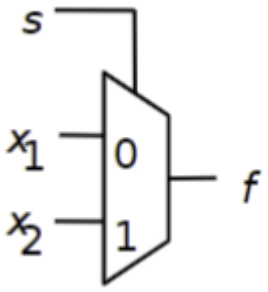
```
module question4(x1, x2, x3, x4, f);
        input x1, x2, x3, x4;
        output f;

        assign a = x1 & x2;
        assign b = ~x3 & ~x4;
        assign f = a | b;
endmodule
```

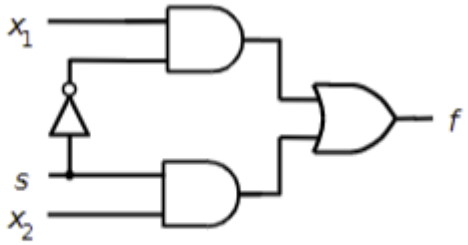
JOINT OPTIMIZATION:

- When given two K-maps, prioritize similar implicants so gates can be reused.

MULTIPLEXERS:

- 2-to-1, 4-to-1, '8-to-1'.
- Number of select lines = Number of inputs = 2^x
- 2-to-1:

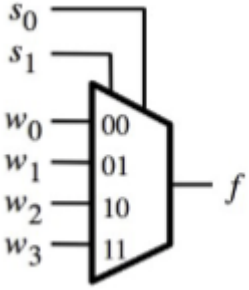




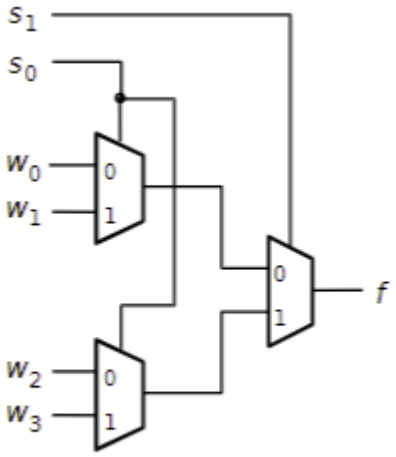
(b) Circuit

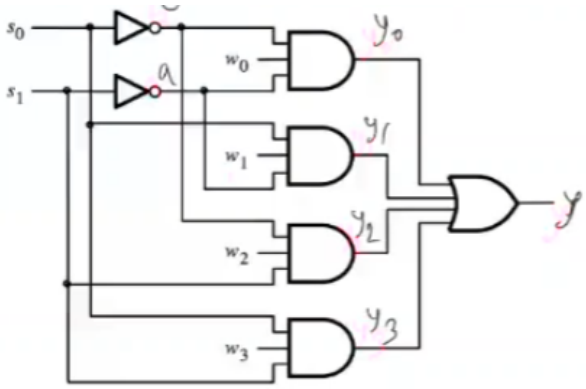
AND THEN IN CIRCUIT FORM: $\overline{S} \bullet X_1 + S \bullet X_2$

- 4-to-1

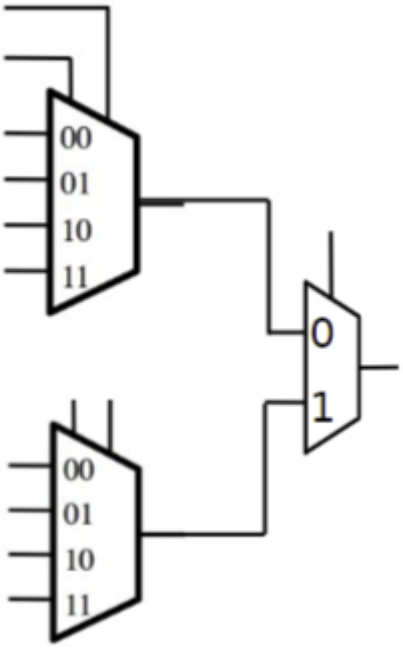


OR





- 8-to-1



OR