ENEE244-010x
Digital Logic Design

Lecture 6

Announcements

- Homework 2 due today
- Homework 3 up on webpage (tonight), due Wed, 9/30.
- Class is canceled on Wed, 9/23
- Substitute on Mon, 9/28

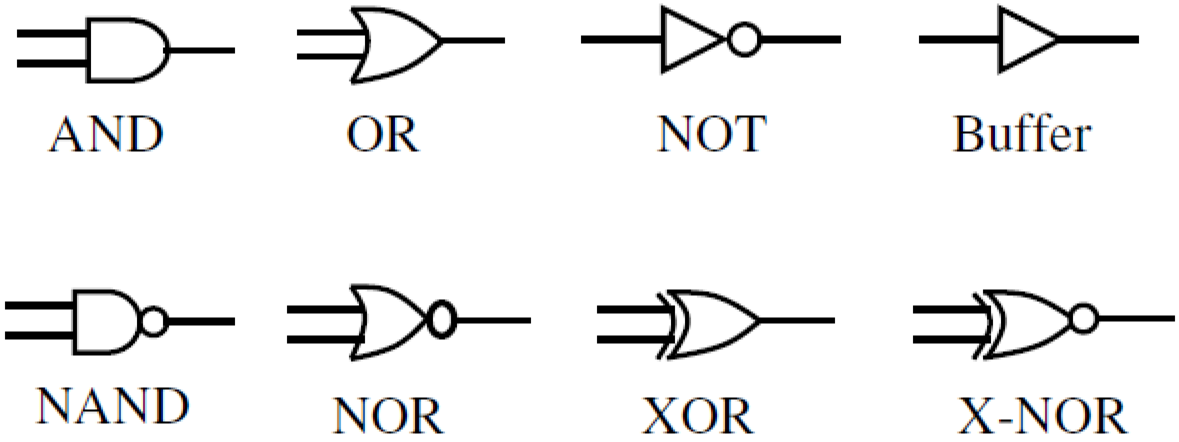
Agenda

- Last time:
  - Manipulations of Boolean Formulas (3.6)
- This time:
  - Gates and Combinational Networks (3.7)
  - Incomplete Boolean Functions and Don’t Care Conditions (3.8)
  - Universal Gates (3.9.3)

Digital Logic Gates

- AND $f(x, y) = xy$
- OR $f(x, y) = x + y$
- NOT (Inverter) $f(x) = \overline{x}$
- Buffer (Transfer) $f(x) = x$
- NAND $f(x, y) = \overline{(xy)} = \overline{x} + \overline{y}$
- NOR $f(x, y) = \overline{x + y} = \overline{x}\,\overline{y}$
- XOR $f(x, y) = x\,\overline{y} + \overline{x}\,y = x \oplus y$
- X-NOR (Equivalence) $f(x, y) = xy + \overline{x}\,\overline{y}$

Gate Symbols



- Multiple input gates:

  AND (1 if all 1), NAND (not AND)

  OR (1 if at least one 1), NOR (not OR)

Some Terminology

- System specification: A description of the function of a system and of other characteristics required for its use.
  - A function (table, algebraic) on a finite set of inputs.
- Double-rail logic*: both variables and their complements are considered as primary inputs.
- Single-rail logic: only variables are considered as primary inputs. Need inverters for their complements.

Gates and Combinational Networks

- Synthesis Procedure
- Example: Truth table for parity function on three variables

X	Y	Z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Synthesis Procedure

X	Y	Z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

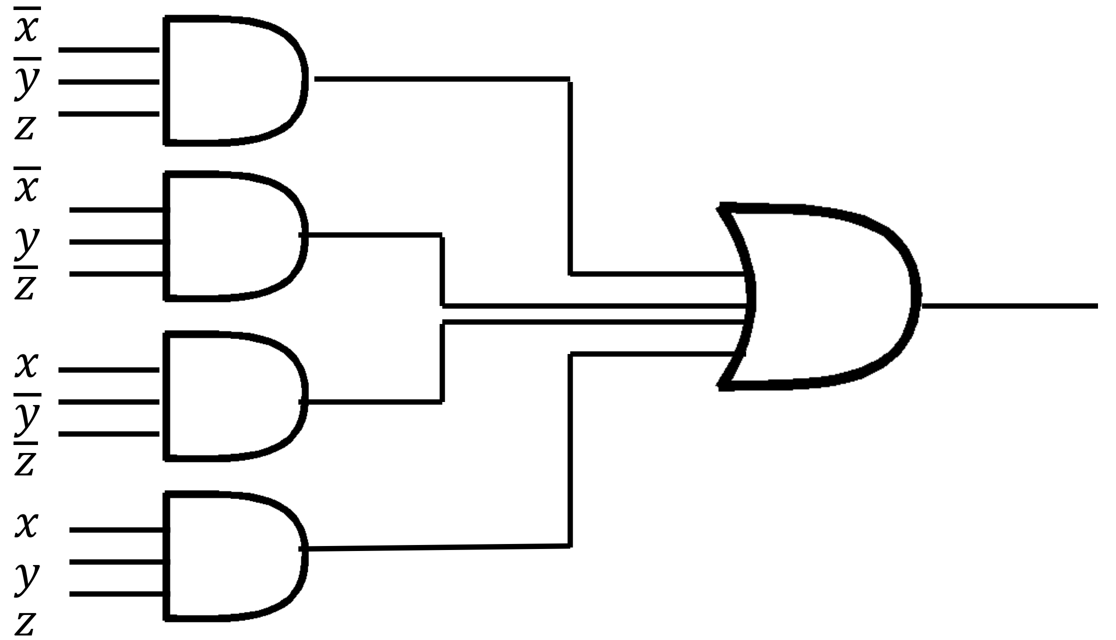
Minterm Canonical Form:

$$f(x, y, z) = \overline{x}\,\overline{y}z + \overline{x}y\,\overline{z} + x\overline{y}\,\overline{z} + \mathrm{xyz}$$

Two-level Gate Network

Minterm Canonical Form:

$$f(x, y, z) = \overline{x}\,\overline{y}z + \overline{x}y\,\overline{z} + x\overline{y}\,\overline{z} + xyz$$



Incomplete Boolean Functions and Don’t Care Conditions

Incomplete Boolean Functions and Don’t Care Conditions

- n-variable incomplete Boolean function is represented by a truth table with n+1 columns and 2^n rows.
- For those combinations of values in which a functional value is not to be specified, a symbol, --, is entered.
- The complement of an incomplete Boolean function is also an incomplete Boolean function having the same unspecified rows of the truth table.

Describing Incomplete Boolean Functions

X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	--
1	0	0	0
1	0	1	--
1	1	0	0
1	1	1	1

Minterm canonical formula:

$$f(x, y, z) = \sum m(0,1,7) + dc(3,5)$$

Describing Incomplete Boolean Functions

X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	--
1	0	0	0
1	0	1	--
1	1	0	0
1	1	1	1

Maxterm canonical formula:

$$f(x, y, z) = \Pi M(2,4,6) + dc(3,5)$$

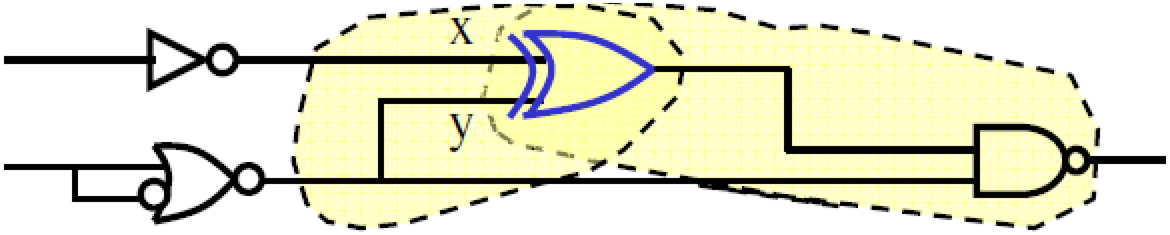
Describing Incomplete Boolean Functions

- Manipulating Boolean equations derived from incomplete Boolean functions is a very difficult task.
- In the next chapter, there are procedures for obtaining minimal expressions that can handle the don’t care conditions.
- Can leverage don’t care conditions to get simplified expressions for functions (smaller gate networks).

More on Don't Care Conditions

- Satisfiability don't cares of a subcircuit/subsystem consist of all input patterns that will never occur.
- Observability don't cares of a subcircuit/subsystem are the input patterns that represent situations when an output is not observed.

Example: $\{xy, \overline{x}y\}$ $\{x\overline{y}, \overline{x}\,\overline{y}\}$



Universal Gates

- A gate or set of gates is called universal if it can implement all Boolean functions.
- Standard universal gates:
  - AND, OR, NOT
  - Proof?
- Theorem:
  - A set of gates $\{G_1, \ldots, G_n\}$ is universal if it can implement AND, OR, NOT.

NAND is Universal

- Recall $NAND(x, y) = \overline{x\,y} = \overline{x} + \overline{y}$
- $NOT(x) =$

$$\overline{x} = \overline{x} + \overline{x} = NAND(x, x)$$

- $AND(x, y) =$

$$xy = \overline{\overline{(xy)}} = NOT((\overline{xy})) \\ = NOT\big(NAND(x, y)\big) \\ = NAND\big(NAND(x, y), NAND(x, y)\big)$$

- $OR(x, y) =$

$$x + y = \overline{\overline{(x + y)}} = \overline{(\overline{x}\,\overline{y})} \\ = NAND\big(NOT(x), NOT(y)\big) \\ = NAND\big(NAND(x, x), NAND(y, y)\big)$$

NOR is Universal

- Recall $NOR = \overline{x+y} = \overline{x}\,\overline{y}$
- $NOT(x) =$
  $$\overline{x} = \overline{x}\,\overline{x} = NOR(x, x)$$
- $AND(x) =$
  $$xy = \overline{\overline{(xy)}} = \overline{(\overline{x} + \overline{y})}$$
  $$= NOR\big(NOT(x), NOT(y)\big)$$
  $$= NOR\big(NOR(x, x), NOR(y, y)\big)$$
- $OR(x) =$
  $$x + y = \overline{\overline{(x+y)}} = NOT\big(\overline{x+y}\big)$$
  $$= NOT\big(NOR(x, y)\big)$$
  $$= NOR\big(NOR(x, y), NOR(x, y)\big)$$

Class Exercise