Digital Logic Systems

Recitation 6: Representations of Boolean Functions by Formulas & Foundations of combinational circuits

Guy Even Moti Medina

School of Electrical Engineering Tel-Aviv Univ.

December 6, 2011

Sum of Products

Recall the following definitions.

Definition

A variable or a negation of a variable is called a literal.

Definition

A formula that is the AND of literals is called a product term.

Definition

A simple product term p is a minterm with respect to a set U of variables if $vars(p) = U$.

A minterm is a simple product term, and therefore, every variable in U appears exactly once in p.

Sum of Products (cont.)

Notation.

With each product term p, we associate the set of variables that appear in p. The set of variables that appear in p is denoted by $vars(p)$. Let $vars^{+}(p)$ denote the set of variables that appear in p that appear without negation. Let $vars^{-}(p)$ denote the set of variables that appear in p that with negation. Let $literals(p)$ denote the set of literals that appear in p.

Minterms of a Boolean Function

Recall the following definitions.

Definition

For a $v \in \{0,1\}^n$, define the minterm p_v to be $p_v \triangleq (\ell_1^v \cdot \ell_2^v \cdots \ell_n^v)$, where:

$$\ell_i^v \triangleq \begin{cases} X_i & \text{if } v_i = 1 \\ \bar{X}_i & \text{if } v_i = 0. \end{cases}$$

Definition

Let $f^{-1}(1)$ denote the set

$$f^{-1}(1) \triangleq \{v \in \{0,1\}^n \mid f(v) = 1\}.$$

Definition

The set of minterms of f is defined by

$$M(f) \triangleq \{p_v \mid v \in f^{-1}(1)\}.$$

Minterms of a Boolean Function (cont.)

Theorem

Every Boolean function $f : \{0,1\}^n \to \{0,1\}$ *that is not a constant zero is represented by the sum of the minterms in* $M(f)$.

Proof.

We have seen a constructive proof in class. □

Back to.. Sum of Products: Example

Example

Represent the following Boolean functions as an SOP formula: (i) $f(a, b) = \max\{a, b\}$, (ii) $g(a, b) = \min\{a, b\}$.

Product of Sums:Example

A similar "ritual"...

Example

Represent the following Boolean functions as an POS formula:
(i) $f(a, b) = \min\{a, b\}$, (ii) $h(a, b) = \max\{a, b\}$.

Minterms of a Boolean Function - a (good) Example

Example

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function defined as follows.

$$f(v) \triangleq \begin{cases} 1 & \text{if } \sum_{i=1}^{n} v_i = n-1 \\ 0 & \text{o.w.} \end{cases}$$

- Compute $M(f)$ for $n = 3$?
- Compute $M(f)$ for general n?
- What is p_v, for $v \in f^{-1}(1)$?
- How many assignments satisfy p_v? write down these assignments.

Foundations of Combinational Circuits: Building Blocks

The building blocks of combinational circuits:

- Combinational gates
- Wires and nets

combinational gates - terminology

- inputs and outputs of a gate are often referred to as *terminals*, *ports*, or even *pins*.
- fan-in of a gate g = number of input terminals of g (i.e., the number of bits in the domain of the Boolean function that specifies the functionality of g).
- basic gates have constant fan-in (2-3).
- The basic gates that we consider are: inverter (NOT-gate), OR-gate, NOR-gate, AND-gate, NAND-gate, XOR-gate, NXOR-gate, multiplexer (MUX). All this gates have a single output.
- fan-out $\neq$ the number of output ports.
- $\{in(g)_i\}_{i=1}^{n}$ = the input ports of a gate g, where n =fan-in(g).
- $\{out(g)_i\}_{i=1}^{k}$ = the output ports of a gate g, where k=number of output ports of g.

Wires and nets

A wire is a connection between two terminals (e.g., an output of one gate and an input of another gate). In the zero-noise model, the signals at both ends of a wire are identical.

Very often we need to connect several terminals (i.e., inputs and outputs of gates) together. We could, of course, use any set of edges (i.e., wires) that connects these terminals together. Instead of specifying how the terminals are physically connected together, we use nets.

Definition

A net is a subset of terminals that are connected by wires. The fan-out of a net N is the number of input terminals that are contained in N.

Example

We may draw a net in any way that we find convenient or aesthetic. The interpretation of the drawing is that terminals that are connected by lines or curves constitute a net.

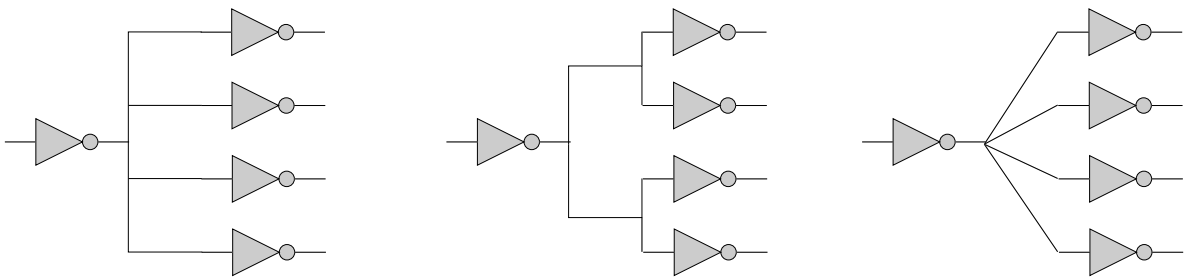
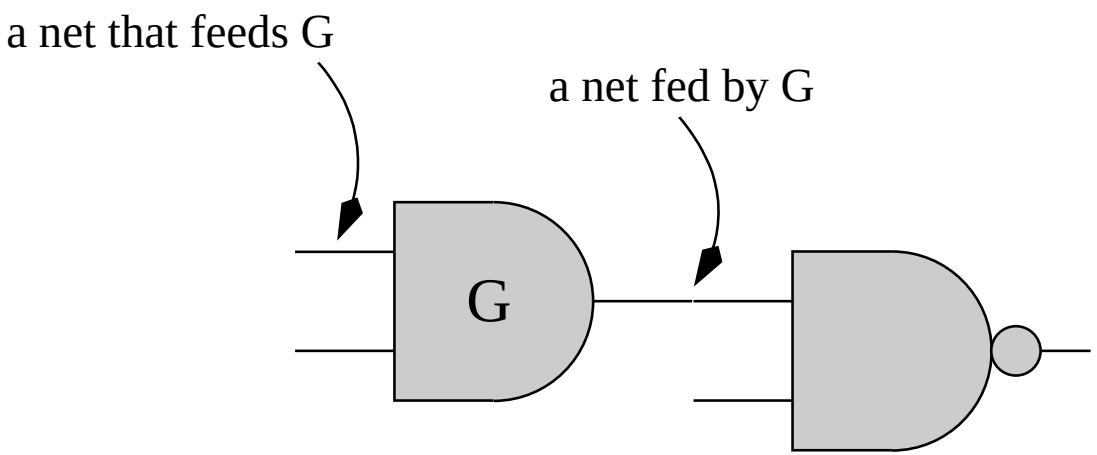


Figure: Three equivalent nets.

Direction in nets

We say that a net N feeds an input terminal t if the input terminal t is in N.
We say that a net N is fed by an output terminal t if t is in N.

Direction of signals along nets is obtained in "pure" CMOS gates as follows. Output terminals are connected (via low resistance) to the ground or to the power (but not both!). Input terminals, on the other hand, are connected only to capacitors.



Simple nets

The following definition captures the type of nets we would like to use. We call these nets *simple*.

Definition

A net N is *simple* if (i) N is fed by exactly one output terminal, and (ii) N feeds at least one input terminal.

A simple net N that is fed by the output terminal t and feeds the input terminals $\{t_i\}_{i\in I}$ can be modeled by the wires $\{w_i\}_{i\in I}$. Each wire w_i connects t and t_i. In fact, since information flows in one direction, we may regard each wire w_i as a directed edge $t \to t_i$. To simplify the discussion, we model simple nets by a "star" of wires emanating from a common output terminal.

Each such wire connects an output terminal of a gate to input terminal of a gate. Thus, a full description of a wire is of the form $(g_1, t_1) \longrightarrow (g_2, t_2)$, where t_1 is an output terminal of gate g_1 and t_2 is an input terminal of gate t_2.

Wire notation

Often, we abbreviate and describe the wire $(g_1, t_1) \longrightarrow (g_2, t_2)$ by $g_1 \longrightarrow g_2$. This abbreviation is not ambiguous if the following holds:

- The gate g_1 has a single output terminal. Since there is only one output terminal, we need not specify to which output terminal the wire is connected.
- (i) Only two wires are directed toward g_2, (ii) g_2 has two input terminals, and (iii) the Boolean function of g_2 is commutative. In this case, we connect each wire to a different input terminal.

Input/Output gates

Definition (input and output gates)

An *input gate* is a gate with zero inputs and a single output. An *output gate* is a gate with one input and zero outputs.

Input Gate Output Gate

- Inputs from the "external world" are fed to a circuit via input gates.
- Outputs to the "external world" are fed by the circuit via output gates.
- an input gate is labeled (IN, x_i), where x_i is the name of the signal along the wire that emanates from it.
- an output gate is labeled (OUT, y_i), where y_i is the name of the signal along the wire that enters it.

Library of gates

Let Γ denote a library of combinational gates that contains standard combinational gates such as an inverter, OR-gate, AND-gate, et cetera.
Let *IO* denote a library that contains two special types of gates: input-gates (IN, x_i) and output-gates (OUT, y_j).

Combinational gate - definition

For simplicity, we assume that Γ contains combinational gates with a single output terminal, two input terminals, and implement commutative Boolean functions.

Definition

A combinational circuit C is a pair (G, π), where $G = (V, E)$ is a directed acyclic graph and $\pi : V \to \Gamma \cup IO$ is a labeling function such that:

1. $\pi(v) \in IO$ iff v is a source or a sink in G.
2. For every vertex v, the in-degree of v equals the fan-in of $\pi(v)$.
3. The restriction of π to sources and sinks is one-to-one. (Namely, the names of input-gates and output-gates are distinct.)