ECE 697CE
Foundations of Computer Engineering

Lesson 2

Boolean Switching Functions (Continued)

Rationale

- Threshold logic forms the basis of neural computing
- Minimizing switching functions helps overcome limitations of switching algebra
- Minimization also creates more effective expressions

Objectives

- Understand Electronic Gate Networks
- Differentiate between functions of threshold elements
- Apply principles of minimization to simplify switching functions
- Compare the properties of unate, monotonic, and symmetric functions

Poll: Prior Knowledge

The following is an example of what principle of Boolean Switching Functions?

$$[f(x_1,\ x_2\ \ldots.\ x_n, 0,1,+,.)]' = f(x_1',\ x_2'\ \ldots.\ x_n', 1,0,.,+)$$

1. Combinational Switching Logic
2. De Morgan's Theorem
3. Law of Complementation
4. Consensus Theorem

Prior Knowledge

- Let's review from Lesson 1: Boolean Switching Functions
  - Number representation
  - Switching logic & algebra
  - Simplification of expressions
  - De Morgan's Theorems

Orchestrated Discussion (Hand Raise): Critical Thinking Exercise

- Discuss two questions regarding Lesson 1.

Electronic Gate Networks

- <u>Electronic gates</u>: generally receive voltages as inputs and produce output voltages
- Precise values of voltages not significant: restricted to value ranges – high (value 1) and low (value 0)
- Electronic gates constructed with two-state switching devices: each capable of permitting the flow of current or blocking it
- To implement arbitrary switching functions: gates must be able to implement a functionally complete set of operations

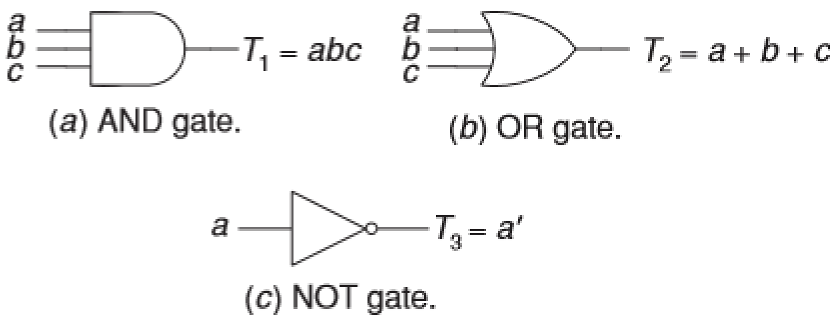


Fig: Gate symbols

Threshold Logic

- **Simplest kind of computing units in Artificial neural networks**
- **Operate by comparing total input with a threshold**
- **Generalization of the common logic gates in conventional computing**

The Threshold Element

- n two valued inputs : $x_1, x_2, \ldots x_n$
- Single two valued output: y
- Internal parameters
  - Threshold: T
  - Weights: $w_1, w_2, \ldots w_n$

} Real, Finite, Positive or Negative numbers

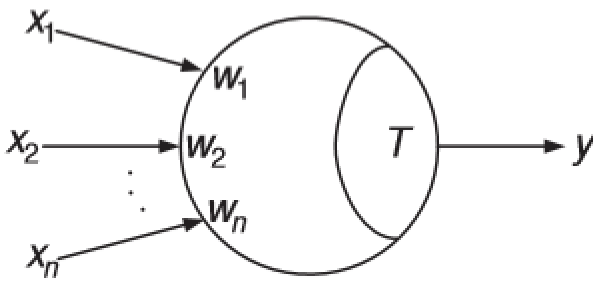
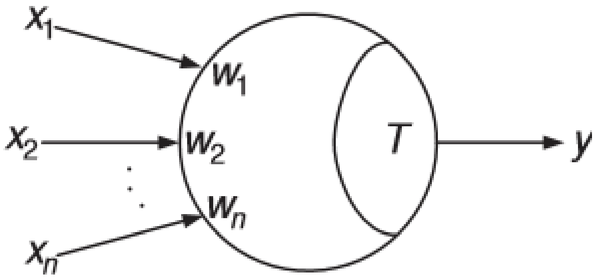


Fig: Symbol for a threshold element

Input-Output Relation of a Threshold Element

- **Input Output relation of a threshold element**

$$y = 1 \; if \; and \; only \; if \sum_{i=1}^{n} w_i x_i \geq T$$



$$y = 0 \; if \; and \; only \; if \sum_{i=1}^{n} w_i x_i < T$$

- $\sum_{i=1}^{n} w_i x_i$ **: Weighted sum of the element**

Example: A Threshold Element

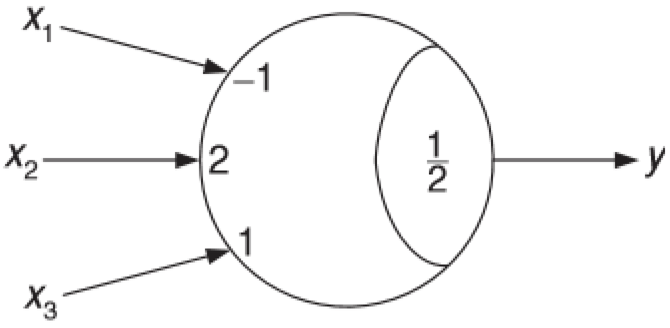


Fig: A threshold element

Input variables			Weighted sum	Output
x_1	x_2	x_3	$-x_1 + 2x_2 + x_3$	y
0	0	0	0	0
0	0	1	1	1
0	1	0	2	1
0	1	1	3	1
1	0	0	−1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	2	1

Table: Input Output relation of the gate

Switching function realized

$$f(x_1, x_2, x_3) = \sum(1, 2, 3, 6, 7) \Rightarrow simplify \Rightarrow x_1' x_3 + x_2$$

Majority Gates

- Special type of threshold element
- Property of a 3 input majority gate:

Produces a 1 if majority of inputs are 1

$$M(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_3x_1$$

- Implementation as a threshold element

$$w_1 = w_2 = w_3 = 1 \; and \; T = 2$$

- To implement an AND gate: Tie one input to 0
- To implement an OR gate: Tie one input to 1

Minority Gates

- Property of a 3 input minority gate:

Produces a 1 if majority of inputs are 0

$$m(x_1, x_2, x_3) = x_1'x_2' + x_2'x_3' + x_3'x_1'$$

- Complement of Majority function

- To implement a NAND gate: Tie one input to 0
- To implement a NOR gate: Tie one input to 1

Minimization of Switching Function

- Aim: Simplify switching function $f(x_1, x_2, \dots x_n)$ to find an expression $g(x_1, x_2, \dots x_n)$ which is equal to f and minimizes some cost criteria

- Criteria to determine minimal cost:
  - Minimum number of appearances of literals
  - Minimum number of literals in SOP or POS expression
  - Minimum number of terms in SOP expression (Ensure there is no other expression with same number of terms and fewer literals)

Irredundant Expression

- Irredundant expression: A SOP expression from which no term can be deleted without altering its logic

- $f(x, y, z) = x'yz' + x'y'z' + xy'z' + x'yz + xyz + xy'z$ -> Redundant
  $= x'z' + y'z' + yz + xz$ -> Irredundant

- An irredundant expression is not necessarily minimal
- Minimal expression not always unique

Whiteboard: Irredundant Expression

- **a + a’b is irredundant but not minimal. a+a’b = a+b. a+b is minimal.**

- **ab’ + ac’ + a’c + a’bc’ = b’c + a’b + ac’ realize the same function but different expressions.**

Karnaugh Map: The Map Method

- Algebraic method to combine terms using the rule

$$Aa + Aa'$$

- Modified form of truth table
- Arrangement of combinations is conven
- n-variable map consists of 2^n cells
- Cells represent all possible combinations of variables

yz \ wx	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

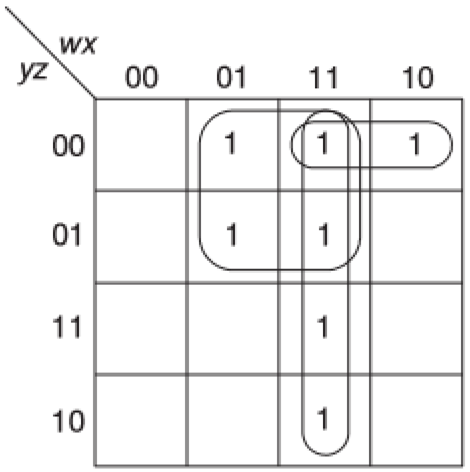
Fig: Location of minterms in 4 variable map

yz \ wx	00	01	11	10
00		1	1	1
01		1	1	
11			1	
10			1	

Map for $f(w, x, y, z) = \sum(4, 5, 8, 12, 13, 14, 15)$

Minimization of Boolean Functions

- Box: Collection of 2^m cells, each adjacent to m cells of the collection.
- Box is said to cover these cells
- Expressed as a product of n-m literals for a function containing n variables
- Eliminate m literals that are not in the product
- Make as many boxes as possible and as large as possible.
- Boxes can only have dimensions of 1, 2, or 4.
- Each box represents 1 term.
- Doubling the size of a box reduces 1 literal from that term.



Map for $f(w,x,y,z) = \sum(4,5,8,12,13,14,15)$

UMASS AMHERST

Minimization of Boolean Functions

- **Example :** $f = yz' + xy$

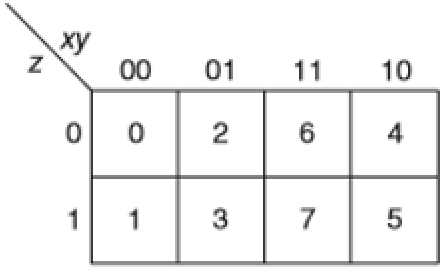


Fig: Location of minterms in a three variable map

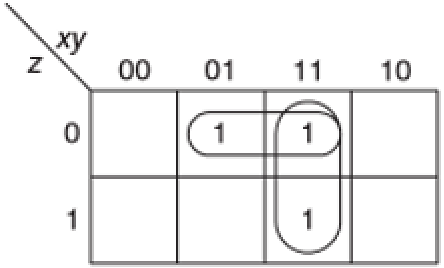


Fig: Map for function $f(x, y, z) = yz' + xy$

- **Algebraic manipulations:**

$$\begin{aligned} f &= x'yz' + xyz' + xyz \\ &= x'yz' + xyz' + xyz' + xyz \\ &= yz'(x' + x) + xy(z' + z) \\ &= yz' + xy \end{aligned}$$

Rules for Minimizations of Boolean Functions

- Cover with boxes those 1-cells that cannot be combined with any other 1-cell.
- Continue to those which have only a single adjacent 1-cell and thus can form boxes of only two 1-cells
- Combine those 1-cells that yield boxes of four but not part of any box of eight cells, and so on
- Minimal expression:
  - A collection of boxes that are as large and as few in number as possible
  - Every 1-cell in the map of the function is covered by at least one box.

Whiteboard: Minimization Example

- Two irredundant expressions $f(w,x,y,z) = \sum(0,4,5,7,8,9,13,15)$

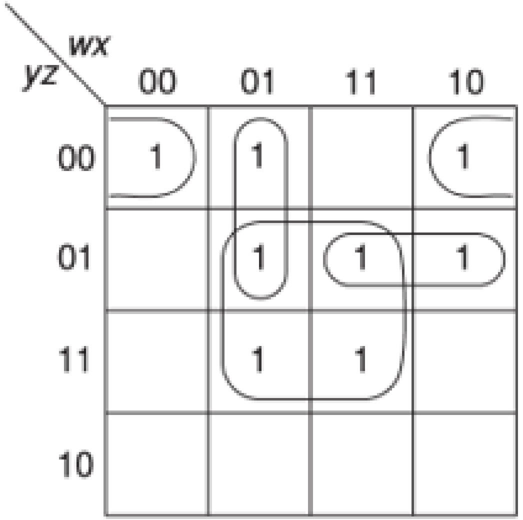


Fig: $f = x'y'z' + w'xy' + wy'z + xz$ is an irredundant expression

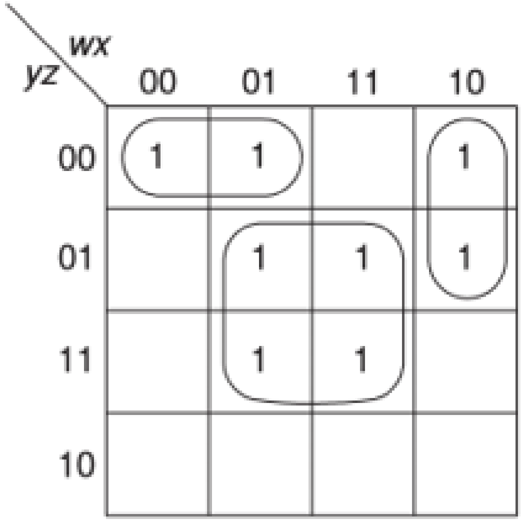


Fig: $f = w'y'z' + wx'y' + xz$ is the unique minimal expression

Summary of this Lesson

- **Learned about simplification of logic functions**

To Prepare for the Next Lesson

- Read the Required Readings for Lesson 3.
- Complete the Pre-work for Lesson 3.

Go to the online classroom for details.