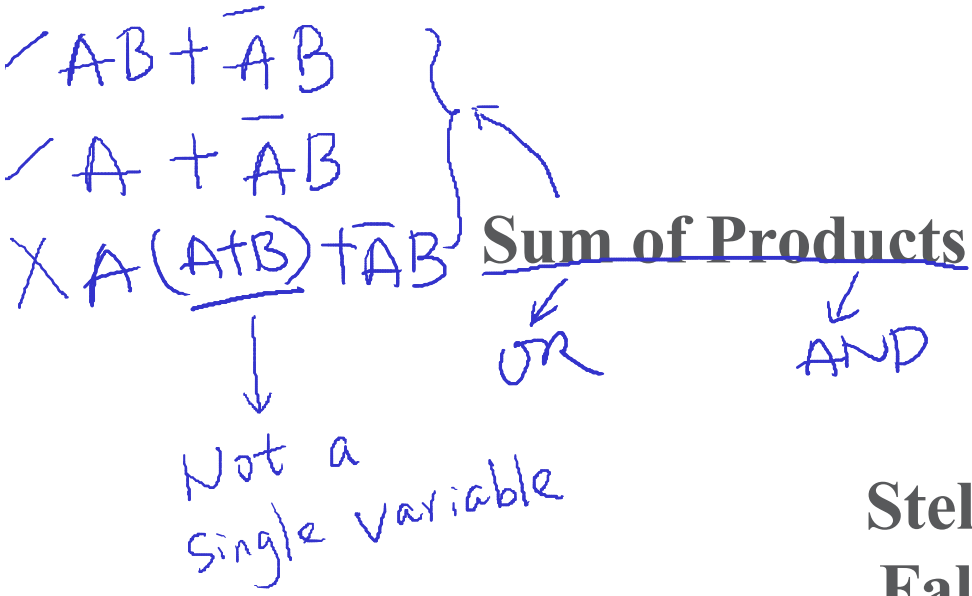
ECE 255
Introduction to Digital Logic Design



Stella Sun

Fall 2023

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Outline

- Review of number conversion
- Boolean expressions in VHDL

- Sum of Products (SOP) Expressions
- Canonical Sum of Products (CSOP) – Formulating Boolean Functions
- Canonical Product of Sums
- Logic Gates and Schematics

Converting Base-16 to Base-10

- Integer *b* serves as the base of a number system (e.g. *b*=16 for base-16)
- Converting base-*b* number to decimal
  - Each base-*b* digit is represented first as its decimal equivalent
    (base-16 digit A represented as 10 for conversion, digit E represented as 14)
  - Each digit for each position *p* is multiplied by b^p
    (A, represented by 10, in position 2 is multiplied by 16^2 or 256)
  - Results for all positions are summed together to reveal the decimal number
- Example: Convert $(2AF)_{16}$ to decimal (base-10)

$d_0 = F = 15$

$d_1 = A = 10$

$d_2 = 2 = 2$

$$D = 2 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0$$

$$= \sum_p d_p \cdot b^p$$

$$= 512 + 160 + 15$$

$$= (687)_{10}$$

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Converting Base-16 to Base-10

- Example: Convert $(F0D)_{16}$ to decimal (base-10)

$$D = 15 \cdot 16^2 + 0 \cdot 16^1 + 13 \cdot 16^0$$

$$= 3840 + 13$$

$$= (3853)_{10}$$

Boolean Functions in VHDL

- VHDL = VHSIC Hardware Description Language
- We'll introduce you to VHDL along the way and your labs will use VHDL

- For basic logic operators in VHDL:
  - Keyword "AND" represents the AND operator – type it out, not ·
  - Keyword "OR" represents the OR operator – type it out, not +
  - Keyword "NOT" for complement or NOT operator – type it out, not ′ or ¯
  - Keyword "XOR for the exclusive-OR operator
- The VHDL assignment operator is <= used to assign right side to left side
- Examples of use:

```
y <= a AND b;       -- implements function y(a,b) = a·b
y <= a OR b;        -- implements function y(a,b) = a + b
y <= NOT(a);        -- implements function y(a,b) = a' = ā
```

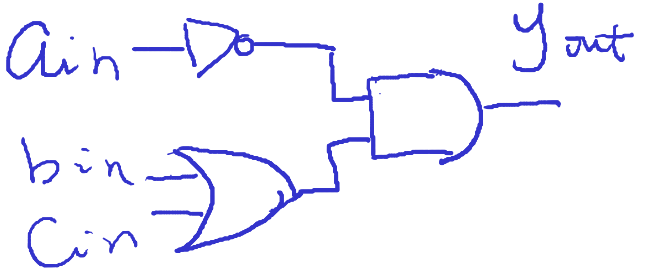
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Boolean Functions in VHDL – Notes & Examples

VHDL ← $y_{out} \Leftarrow$ (NOT a_{in}) AND (b_{in} OR c_{in})

Logic exp. ← $y_{out} = \overline{a_{in}} \cdot (b_{in} + c_{in})$

⇐ assign — value of right drives y_{out} as output



; — end expression

-- Comments

Minterms and Products

- Recall: AND is a function of variables the result of which is TRUE (logic 1) only when *all* input variables are TRUE (logic 1)
- A *minterm* is an AND (a.k.a. "product") in which *each variable appears once* in the term, in either complemented or uncomplemented form
- A minterm *m* is 1 for exactly one combination of variables and 0 for all others

- For a function of *n* variables, how many minterms?

- Let's write all possible minterms for two variables *A*, *B*

Minterms and Products – Truth Tables

- Minterms are AND expressions in a function that yield logic 1 (or TRUE) in the output of that function
- Possible minterms are based on any combination of variables (2^n possible minterms for n variable function)