# ECE 697CE
# Foundations of Computer Engineering

## Lesson 1
## Boolean Switching Functions

# Rationale

- Reviewing Boolean Switching Functions prepares you for Project 1: Finite State Machines in Verilog
- The Verilog project provides real-world experience

# Objectives

- Analyze basic properties of switching logic and algebra
- Apply laws of simplification within switching functions

# Group Discussion and Report Back (Short Answer): Prior Knowledge

**"The interaction between hardware and software at a variety of levels offers a framework for understanding the fundamentals of computing."**

**- (Patterson & Hennessy, 2013, pp. xv)**

- **Undergraduate degree survey**

# Prior Knowledge

- Course concepts build on previous undergraduate courses
- Review prior computer hardware, software, and algebraic math courses

# Orchestrated Discussion (Hand Raise): Critical Thinking Exercise

- Discuss responses to the following questions:
  - What do you desire to get out of this class? What are you interested in?
  - How might you use this knowledge in your future career?
  - How might your future employer benefit from this knowledge?

# Number Representation

- Number representation in different systems
  - Decimal : Base 10
  - Binary : Base 2
  - Octal : Base 8
  - Hexadecimal : Base 16

- Base b number: $(N)_b = a_{q-1}b^{q-1} + \cdots + a_0b^0 + \cdots + a_{-p}b^{-p}$
  $= \sum_{i=-p}^{q-1} a_i b^i$

- Base b $> 1, 0 \leq a_i \leq b-1$
- $a_{q-1}, a_{q-2} \ldots a_0 \rightarrow$ Integer part of N
- $a_{-1}, a_{-2} \ldots a_{-p} \rightarrow$ Fractional part of N
- $a_{q-1} \rightarrow$ Most significant digit
- $a_{-p} \rightarrow$ Least significant digit

# Number Representation

- For eg: Number 18 in
  1. Decimal system, $(18)_{10} = 1 \times 10^1 + 8 \times 10^0$
  2. Binary system, $(10010)_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
  3. Octal system, $(22)_8 = 0 \times 8^2 + 2 \times 8^1 + 2 \times 8^0$
  4. Hexadecimal system, $(12)_{16} = 0 \times 16^2 + 1 \times 16^1 + 2 \times$

Table: Representation of first 15 numbers

| Base | | | | |
|---|---|---|---|---|
| 2 | 4 | 8 | 10 | 12 |
| 0000 | 0 | 0 | 0 | 0 |
| 0001 | 1 | 1 | 1 | 1 |
| 0010 | 2 | 2 | 2 | 2 |
| 0011 | 3 | 3 | 3 | 3 |
| 0100 | 10 | 4 | 4 | 4 |
| 0101 | 11 | 5 | 5 | 5 |
| 0110 | 12 | 6 | 6 | 6 |
| 0111 | 13 | 7 | 7 | 7 |
| 1000 | 20 | 10 | 8 | 8 |
| 1001 | 21 | 11 | 9 | 9 |
| 1010 | 22 | 12 | 10 | $\alpha$ |
| 1011 | 23 | 13 | 11 | $\beta$ |
| 1100 | 30 | 14 | 12 | 10 |
| 1101 | 31 | 15 | 13 | 11 |
| 1110 | 32 | 16 | 14 | 12 |
| 1111 | 33 | 17 | 15 | 13 |

# Orchestrated Discussion (Hand Raise): Number Representation

- Why do we use hexadecimal numbers in computer output when computers use binary numbers?

- Why does a byte have 8 bits when we only need 7 bits to represent all the characters on a keyboard?

# Combinational Switching Logic

- **Combinational switching logic**: Outputs are function of **present** circuit inputs
- **Switching algebra**: Algebraic system consisting of
  - Set {0,1}
  - Binary operations : OR (logical sum) , AND (logical product)
  - Unary operation : NOT (complementation)

| OR operation | AND Operation | NOT operation |
|---|---|---|
| $0+0=0$ | $0*0=0$ | |
| $0+1=1$ | $0*1=0$ | $0'=1$ |
| $1+0=1$ | $1*0=0$ | $1'=0$ |
| $1+1=1$ | $1*1=1$ | |

# Basic Properties of Boolean Logic

- **Idempotent law:**
$$x + x = x$$
$$xx = x$$
- **Commutativity:**
$$x + y = y + x,$$
$$xy = yx$$
- **Associativity:**
$$(x + y) + z = x + (y + z),$$
$$(xy)z = x(yz)$$
- **Complementation:**
$$x + x' = 1,$$
$$xx' = 0$$
- **Distributivity:**
$$x(y + z) = xy + x.z,$$
$$x + yz = (x + y)(x + z)$$

# Basic Properties of Boolean Logic

- Principle of Duality
  - All the preceding properties are grouped in pairs
  - One statement can be obtained from other by I
    - Interchanging the OR , AND operations
    - Replacing the constants 0 and 1 by 1 and 0
  - The statements that have this property are called <u>dual</u>

$$x + x' = 1, \quad xx' = 0 \longrightarrow \text{Duals}$$

$$(x + y) + z = x + (y + z), \quad (xy)z = x(yz) \longrightarrow \text{Duals}$$

# Poll: Basic Properties of Boolean Logic

1. Idempotent law
2. Commutativity
3. Associativity
4. Complementation
5. Distributivity

- What type of law does this equation demonstrate?

$$(A + B) + C = A + (B + C)$$

$$(AB)C = A(BC)$$

# Switching Expression

**Switching expression** : Combination of finite number of switching variables (x, y,..) and constants (0, 1) by means of switching operations (AND,OR,NOT)

- Any constant or switching variable is a switching expression
- If $T_1$ and $T_2$ are switching expressions, so are $T_1$', $T_2$' , $T_1$+ $T_2$ and $T_1 T_2$
- No other combination of constants and variables is a switching expression

# Laws for Simplification in Boolean Algebra

- **Absorption Law of Switching Algebra**

$$x + xy = x$$
$$x(x + y) = x$$

- **Consensus Theorem**

$$xy + x'z + yz = xy + x'z$$
$$(x + y)(x' + z)(y + z) = (x + y)(x' + z)$$

- **No Name Law**

$$x + x'y = x + y$$
$$x(x' + y) = xy$$

# De Morgan's Theorems

**De Morgan's Theorem:** complement of any expression can be obtained by replacing each variable and element with its complement and, at the same time, interchanging the OR and AND operations

- DeMorgan's Theorem for two variables:

$$(x+y)' = x'y'$$
$$(xy)' = x' + y'$$

- Proof by perfect induction:

| $x$ | $y$ | $x'$ | $y'$ | $x+y$ | $(x+y)'$ | $x'y'$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

- DeMorgan's Theorem for n variables

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

# Whiteboard: De Morgan's Theorems

**What is the DeMorgan equivalent of the following expression?**

$$f = abc + a' + c'$$

# Switching Functions

- A <u>switching function</u> $f(x_1, x_2, \dots x_n)$ is a correspondence that associates an element of the algebra with each of the $2^n$ combinations of variables $x_1, x_2, \dots x_n$

- Determining the value of an expression for an input combination:
- Example : $T(x, y, z) = x'z + xz' + x'y'$,
  $T(0,0,1) = 0'1 + 01' + 0'0' = 1$

| $x$ | $y$ | $z$ | $T$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Table: Truth table for T

# Switching Functions

- **Complement function : $f'(x_1, x_2, \dots x_n)$ assumes value 0(1) whenever $f(x_1, x_2, \dots x_n)$ assumes value 1(0)**
- **Logical sum of two functions : $f(x_1, x_2, \dots x_n) + g(x_1, x_2, \dots x_n) = 1$ for every combination in which either *f* or *g* or both equal 1**
- **Logical product of two functions : $f(x_1, x_2, \dots x_n).\, g(x_1, x_2, \dots x_n) = 1$ for every combination for which both *f* and *g* equal 1**

| $x$ | $y$ | $z$ | $f$ | $g$ | $f'$ | $f+g$ | $fg$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

**Table: Illustration of sum, product and complementation of functions**

# Simplification of Expressions

- Example: Simplify $T(x, y, z) = A'C' + ABD + BC'D + AB'D' + ABCD'$
  - Apply consensus theorem to first three terms -> $BC'D$ is redundant
  - Apply distributive law to last two terms -> $AD'(B' + BC) -> AD'(B' + C)$
  - Thus, $T = A'C' + A[BD + D'(B' + C)]$

- Example: Simplify $T(A, B, C, D) = A'B + ABD + AB'CD' + BC$
  - $A'B + ABD = B(A' + AD) = B(A' + D)$
  - $AB'CD' + BC = C(B + AB'D') = C(B + AD')$
  - Thus, $T = A'B + BD + ACD' + BC$
  - Expand $BC$ to $(A + A')BC$ to obtain $T = A'B + BD + ACD' + ABC + A'BC$
  - From absorption law: $A'B + A'BC = A'B$
  - From consensus theorem: $BD + ACD' + ABC = BD + ACD'$
  - Thus, $T = A'B + BD + ACD'$

# Whiteboard: Simplification of Expressions

- Simplify the following algebraic expression:

$$xy + xy' + wx$$

# Canonical Sum-of-Products

- **Minterm** : a product term that contains each of the n variables as factors in either complemented or uncomplemented form
- It assumes value 1 for exactly one combination of variables
- **Canonical sum of products** : Sum of all minterms derived from combinations for which function is 1

| Decimal code | $x$ | $y$ | $z$ | $f$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

Table: Truth table for a function f

- Compact representation of switching functions

$$f(x,y,z) = x'y'z' + x'yz' + x'yz + xyz' + xyz = \sum(0,2,3,6,7)$$

# Canonical Product-of-Sums

- **Maxterm** : a sum terms that contains each of the n variables in either complemented or uncomplemented form
- It assumes value 0 for exactly one combination of variables
- **Canonical product of sums**: Product of all maxterms derived from combinations for which function is 0.
- Compact representation of switching functions

$$f(x,y,z) = (x+y+z')(x'+y+z)(x'+y+z') = \prod(1,4,5)$$

| Decimal code | $x$ | $y$ | $z$ | $f$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

Table: Truth table for a function f

# Whiteboard: Canonical Product-of-Sums

- **Apply De Morgan's Law to go from sum-of-product canonical form to product-of-sum canonical form**

| Decimal code | $x$ | $y$ | $z$ | $f$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

**Table: Truth table for a function f**

# Summary of this Course

- This course features three skill-building Projects
  - Project 1: Finite State Machines in Verilog – Due by Lesson 10
  - Project 2: Programming in C++ – Due by Lesson 15
  - Project 3: MIPS Simulation – Due for presentation in Lesson 20

- This course will also include a Midterm Exam and a Final Exam.

# To Prepare for the Next Lesson

- Complete the Post-work for Lesson 1.
- Read the Required Readings for Lesson 2.
- Complete the Pre-work for Lesson 2.

Go to the online classroom for details.