ENGN2219/COMP6719
Computer Systems & Organization

Convener: Shoaib Akram

shoaib.akram@anu.edu.au

Australian National University

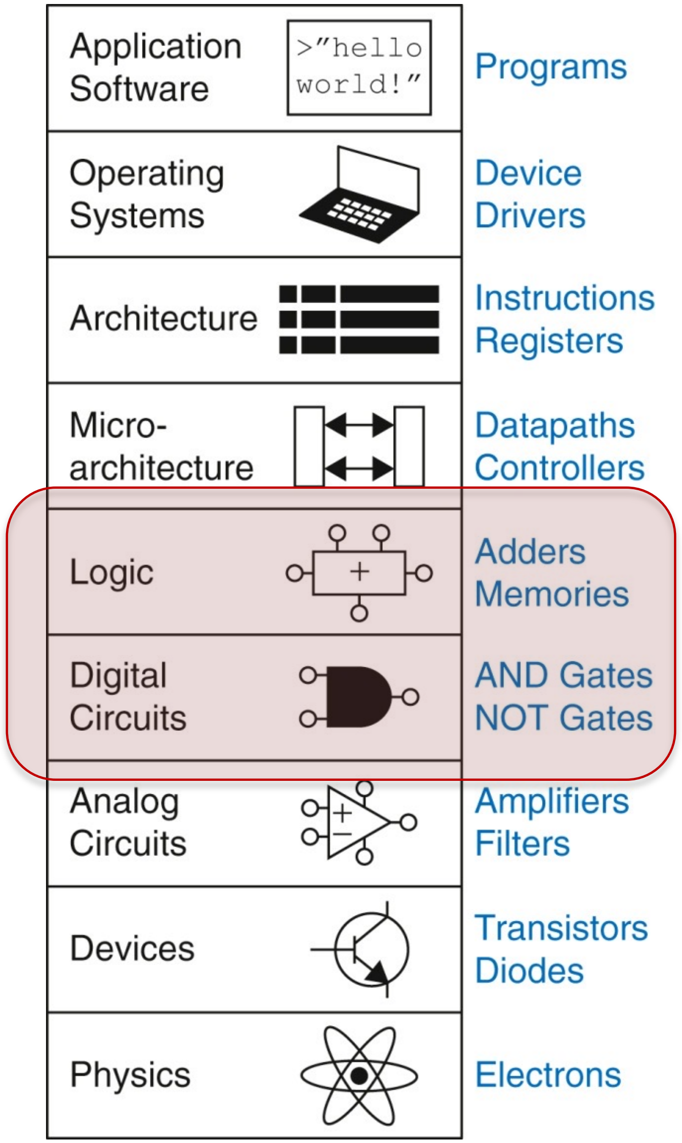
Plan: Week 3

Week 2: Logic gates & Combinational logic

Week 2: Multiplexers, ALU, and decoders

This Week: Boolean algebra (equation minimization)

This Week: Sequential circuits (state & memory)



Broadening our horizon
"one layer at a time"

Implementing Combinational Logic

Steps in implementing combinational Logic

1. Initial specification (e.g., in English)
2. Construct the truth table
3. Derive the Boolean equation

Functional specification

4. *Simplify the Boolean equation (use Boolean algebra)*
5. Implement the equation using logic gates

Boolean Algebra

- The sum-of-products canonical form does not lead to the simplest logic gate implementation
  - In many cases, we can reduce the # AND gates
  - We can reduce the # literals in the equation

- We use Boolean algebra to simplify Boolean equations
  - Think of simplification in ordinary algebra except we are dealing with 0 and 1

Boolean Algebra

- Boolean algebra consists of
  - Axioms (*correct by definition*)
  - Theorems of one variable
  - Theorems of several variables
- Any theorem can be proved via the axioms
  - An axiom is the ground truth (cannot be proven wrong)
- The *Principle of Duality*
  - If the symbols 0 and 1 and the operators AND and OR are interchanged, the statement will still be correct

Boolean Axioms

Number	Axiom	Dual	Name
A1	$B = 0$ if $B \neq 1$	$B = 1$ if $B \neq 0$	Binary Field
A2	$\overline{0} = 1$	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	$1 + 0 = 0 + 1 = 1$	AND/OR

Dual: Replace: $\bullet$ with $+$
0 with 1

Boolean Theorems of One Variable

Number	Theorem	Dual	Name
T1	$B \bullet 1 = B$	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	$B + B = B$	Idempotency
T4	$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	$B + \overline{B} = 1$	Complements

Dual: Replace: • with +
0 with 1

Theorems: Several Variable

#	Theorem	Dual	Name
T6	$B \bullet C = C \bullet B$	$B+C = C+B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	$(B + C) + D = B + (C + D)$	Associativity
T8	$B \bullet (C + D) = (B \bullet C) + (B \bullet D)$	$B + (C \bullet D) = (B+C)\,(B+D)$	Distributivity
T9	$B \bullet (B+C) = B$	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	$(B+C) \bullet (B+\overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D) =$ $(B \bullet C) + (\overline{B} \bullet D)$	$(B+C) \bullet (\overline{B}+D) \bullet (C+D) =$ $(B+C) \bullet (\overline{B}+D)$	Consensus

Warning: T8’ (dual of T8) differs from traditional algebra: OR (+) distributes over AND (•)

Proving Theorems

- **Method 1:** Perfect induction
  - Check all possible input combinations (proof by exhaustion)
  - *Two expressions are equal if they produce the same value for every possible input combination*

- **Method 2:** Use other theorems/axioms to simplify equations
  - As in ordinary algebra, make one side of the equation look like the other

Example: Perfect Induction

Number	Theorem	Name
T6	B•C = C•B	Commutativity

B	*C*	*BC*	*CB*
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Example: Perfect Induction

Number	Theorem	Name
T9	$B \bullet (B+C) = B$	Covering

B	*C*	*(B+C)*	*B(B+C)*
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

Method 2: T9 (Covering)

Number	Theorem	Name
T9	$B \bullet (B+C) = B$	Covering

Method 2: Prove true using other axioms and theorems.

$B \bullet (B+C)$	$= B \bullet B + B \bullet C$	T8: Distributivity
	$= \mathbf{B} + B \bullet C$	T3: Idempotency
	$= B \bullet (1 + C)$	T8: Distributivity
	$= B \bullet (\mathbf{1})$	T2: Null element
	$= B$	T1: Identity

Method 2: T10 (Combining)

Number	Theorem	Name
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	Combining

Prove true using other axioms and theorems:

$B \bullet C + B \bullet \overline{C}$	$= B \bullet (C + \overline{C})$	T8: Distributivity
	$= B \bullet (\mathbf{1})$	T5': Complements
	$= B$	T1: Identity

Simplifying Boolean Equations

- A basic principle for simplifying sum-of-product equations
  - PA + PA’ = P
  - *P is any implicant*
  - *Y = A’B + AB = B(A’+A) = B(1) = B*
- An equation is *minimized* if
  - *it uses the fewest number of implicants*
  - *if there are multiple equations with the same number of implicants, then the one with the fewest literals*

Simplification Example – 1

$Y = AB + AB'$

$Y = A$ T10: Combining

or

$= A(B + B')$	T8: Distributivity
$= A(1)$	T5': Complements
$= A$	T1: Identity

Simplification Example – 2

$Y = A(AB + ABC)$

$= A(AB(1 + C))$	T8: Distributivity
$= A(AB(1))$	T2’: Null Element
$= A(AB)$	T1: Identity
$= (AA)B$	T7: Associativity
$= AB$	T3: Idempotency

Simplification Example – 3A

$Y = AB'C + ABC + A'BC$

$= AC(B + B') + A'BC$	T8: Distributivity
$= AC(1) + A'BC$	T5: Complements
$= AC + A'BC$	T1: Identity

- The two implicants AC and BC share the minterm ABC
- *Are we stuck with simplifying only one of the minterm pairs?*

Simplification Example – 3B

Y = AB’C + ABC + A’BC

= AB’C + **ABC + ABC** + A’BC	T3’: Idempotency
= (AB’C+ABC) + (ABC+A’BC)	T7’: Associativity
= AC + BC	T10: Combining

- *The two implicants AC and BC are called prime implicants*
- *They cannot be combined with any other implicants in the equation to get a new implicant with fewer literals*

Simplification Example – 4

$Y = A'B'C' + AB'C' + AB'C$

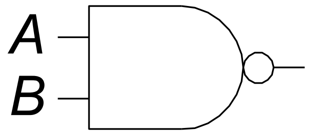
De Morgan’s Theorem

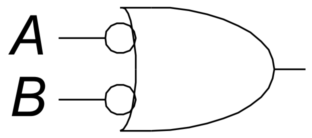
#	Theorem	Dual	Name
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots} = \overline{B_0} + \overline{B_1} + \overline{B_2} \ldots$	$\overline{B_0 + B_1 + B_2 \ldots} = \overline{B_0} \bullet \overline{B_1} \bullet \overline{B_2} \ldots$	DeMorgan’s Theorem

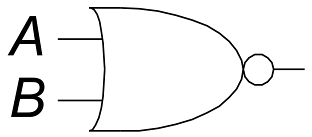
- *The complement of the product is the sum of the complements*
- ***Dual:** The complement of the sum is the product of the complements*

De Morgan’s Theorem

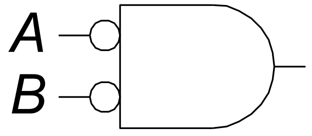
- $Y = \overline{AB} = \overline{A} + \overline{B}$







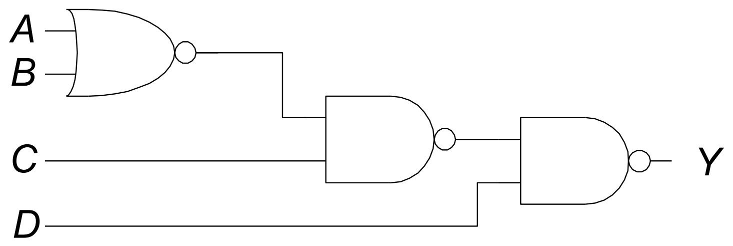
- $Y = \overline{A + B} = \overline{A} \bullet \overline{B}$

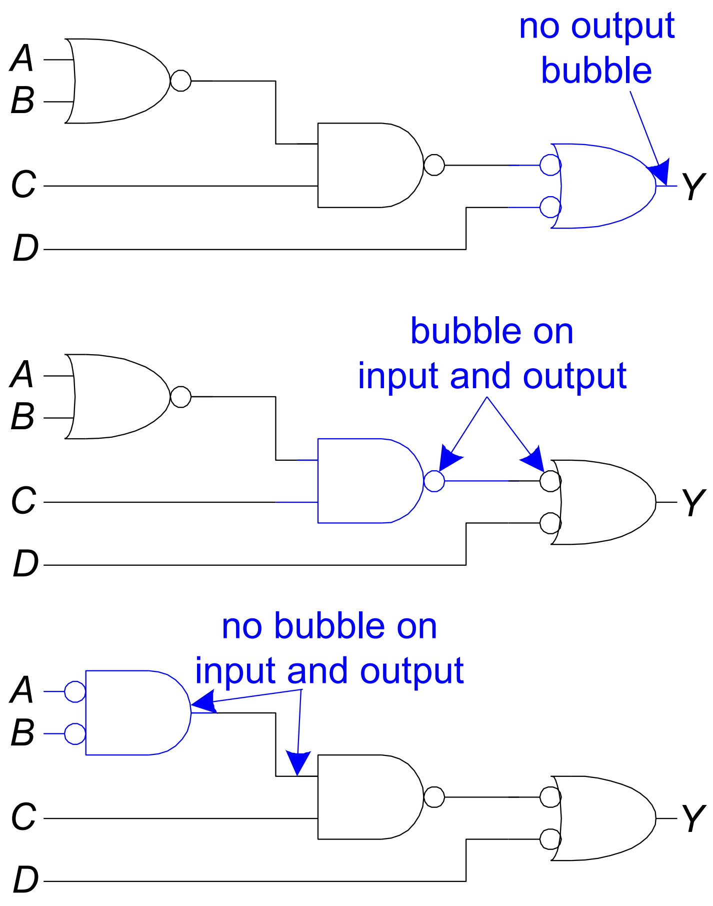


Bubble Pushing Rules

- Pushing bubbles backward/forward changes the body of the gate from AND/OR to OR/AND
- Pushing a bubble from output back to inputs put bubbles on all gate inputs
- Pushing bubbles on all gate inputs forward towards the output puts a bubble on the output

Bubble Pushing Example





$Y = \overline{A}\overline{B}C + \overline{D}$

Priority Circuit

Consider a theater reservation system. The system has four inputs, A_3, . . . , A_0, and four outputs, Y_3, . . . , Y_0. These signals can also be written as $A_{3:0}$ and $Y_{3:0}$. Each user asserts their input when they request the theater for the next day. The system asserts at most one output, granting the theater to the highest priority user. The dean, who is paying for the system, demands highest priority (**3**). The department chair, teaching assistant, and dorm social chair have decreasing priority. *Write a truth table and Boolean equations for the system. Sketch a circuit that performs this function.*

Note: The system is called a four-input priority circuit. We can write equations and simplify them using Boolean algebra. Fortunately, we can find the simplified equations via inspection

Priority Circuit

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0

$Y_3 = A_3$

$Y_2 = A_3'A_2$

$Y_1 = A_3'A_2'A_1$

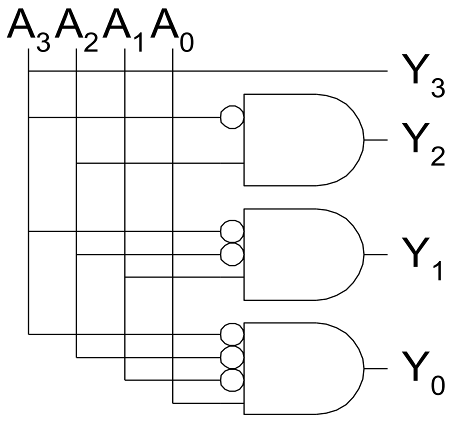
$Y_0 = A_3'A_2'A_1'A_0$

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

Priority Circuit

X = don’t care (value does not impact output)

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0



A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0