TUTORIAL 1 COMBINATIONAL LOGIC CIRCUIT

Overview

- **We will review the following concept in this tutorial:**
- **Combinational logic circuit**
  - No memory, output(s) solely determined by input(s)
- **Truth table and logic function**
- **Two-level logic and PLA**
- **Simplification with Boolean Algebra and K-map**
- **Circuit design**

- **Work with two practical examples**
  - Bit comparator
  - Encoder

Digital Logic Circuit

- Two types of digital logic circuits inside a computer:
  - **Combinational logic circuits**
    - Logic circuits that do not have memory
    - The output depends only on the current inputs and the circuit
    - They can be specified fully with a truth table or a logic equation
  - **Sequential logic circuits**
    - Logic circuits that have memory
    - The output depends on both the current inputs and the value stored in memory (called state)

Circuit Design Process

- **A simple logic design process involves**
  - Problem specification
  - Truth table derivation
  - Derivation of logical expression
  - Simplification of logical expression
  - Implementation

Review of Boolean Algebra

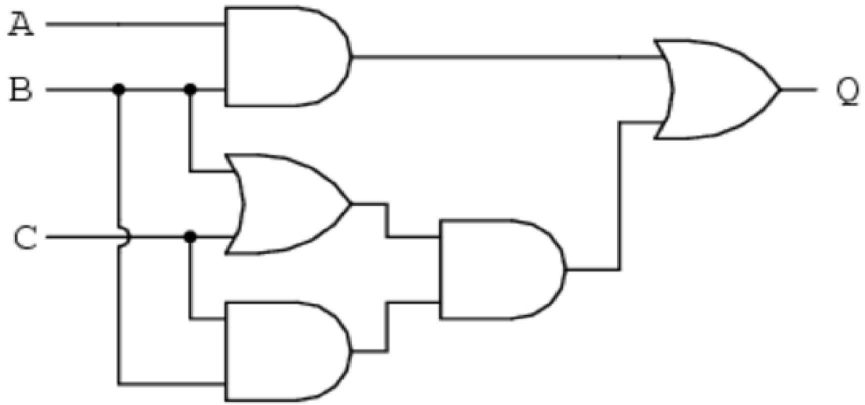
- **Boolean algebra consists of**
  - Boolean variables (with values equal to either '0' or '1')
  - Binary operators: AND (·), OR (+), NOT (')
- **Any logic function can be expressed as a two-level logic expression, either as**
  - Sum-of-Products (SoP) representation, or
  - Product-of-Sums (PoS) representation
- **The AND, OR, and NOT operations form a functionally complete set (namely, universal gates), as they can specify any logic function.**

Basic Laws of Boolean Algebra

Name	AND Form	OR Form
Identity Law	$1A = A$	$0 + A = A$
Null Law	$0A = 0$	$1 + A = A$
Idempotent Law	$AA = A$	$A + A = A$
Inverse Law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative Law	$AB = BA$	$A + B = B + A$
Associative Law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive Law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption Law	$A(A + B) = A$	$A + AB = A$
De Morgan's Law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

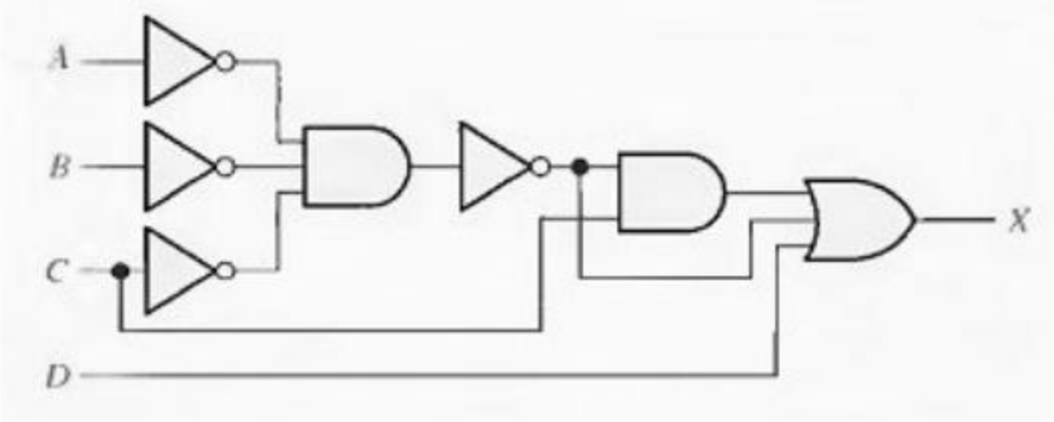
Boolean Algebra Exercise 1

- Find the direct Boolean expression for the following circuit
- Simplify the Boolean expression using Boolean algebra
- Draw the new circuit for the simplified Boolean expression



Boolean Algebra Exercise 2

- Simplify the combinational logic circuit shown below to a minimum form



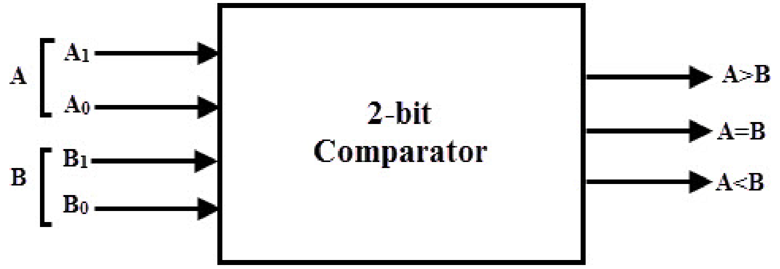
Boolean Algebra Exercise 3

- **Simplify the following Boolean expression**

$$AB + \overline{AB}CD + \overline{CD}EF$$

2-bit Comparator

- Here we'll be designing circuits to compare 2-bit binary numbers.
- Suppose we have two 2-bit numbers A & B at the inputs, and three outputs as A>B, A==B, A<B
- Only one of the three outputs would be true accordingly if A is greater than or equal to or less than B.
- We'll practice the circuit design for f(A==B), try to work on f(A>B) and f(A<B) by yourself



Solution: 2-bit Comparator Truth Table

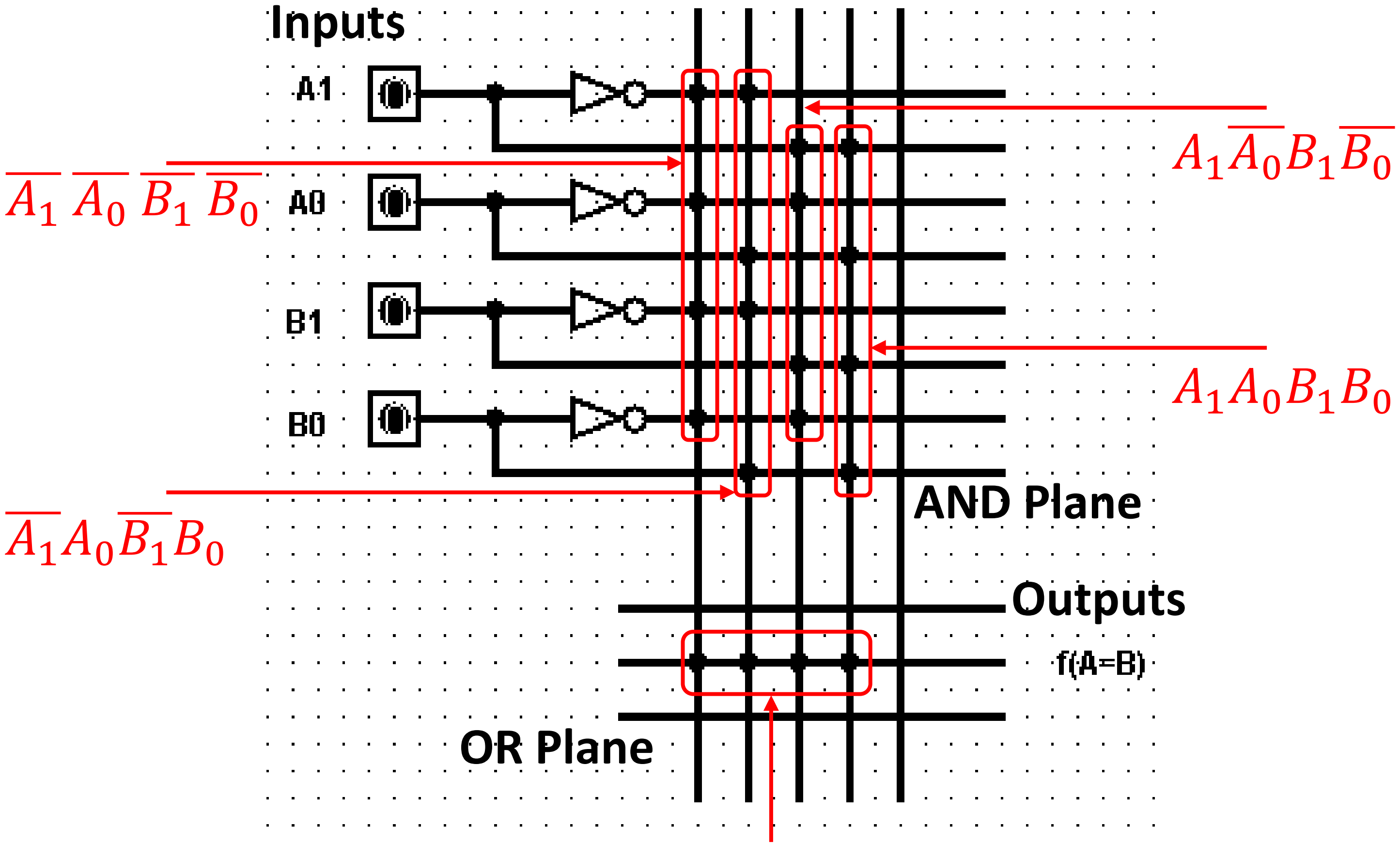
$A(A_1A_0)$	$B(B_1B_0)$	**f (A>B)**	**f (A==B)**	**f (A<B)**

Solution: Logic Function for f(A==B)

Solution: Circuit for f(A==B)

Solution: PLA Implementation

- The same circuit can be equivalently represented by a **programmable logic array** (PLA) circuit.

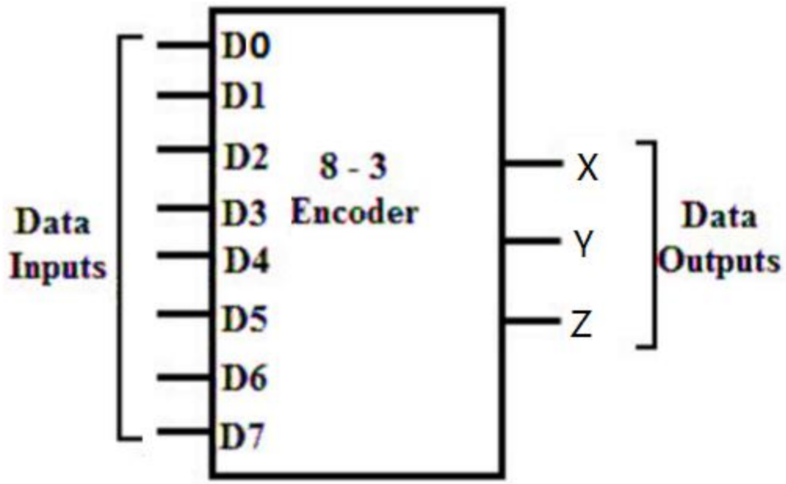


$$f(A == B) = \overline{A_1}\,\overline{A_0}\,\overline{B_1}\,\overline{B_0} + \overline{A_1}A_0\overline{B_1}B_0 + A_1\overline{A_0}B_1\overline{B_0} + A_1A_0B_1B_0$$

8-to-3 Encoder

- An encoder (2^N-to-N encoder) is a logical block with an 2^N-bit input and N 1-bit outputs, which performs the inverse function of a decoder.

- Example (8-to-3 encoder)
  - 8 inputs (D_0, D_1, …, D_7) and 3 outputs (X, Y, Z)



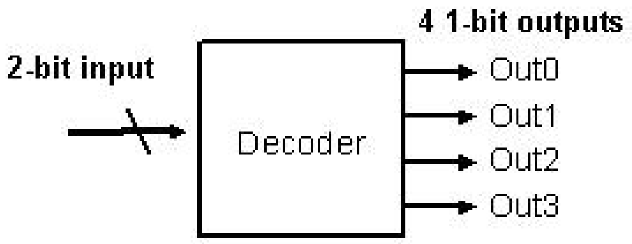
8-to-3 Encoder Truth Table

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	**X**	**Y**	**Z**
1	0	0	0	0	0	0	0	*0*	*0*	*0*
0	1	0	0	0	0	0	0	*0*	*0*	*1*
0	0	1	0	0	0	0	0	*0*	*1*	*0*
0	0	0	1	0	0	0	0	*0*	*1*	*1*
0	0	0	0	1	0	0	0	*1*	*0*	*0*
0	0	0	0	0	1	0	0	*1*	*0*	*1*
0	0	0	0	0	0	1	0	*1*	*1*	*0*
0	0	0	0	0	0	0	1	*1*	*1*	*1*

Solution: Logic Function and Circuit

Extra Exercise: Decoder

- A decoder takes a single N-bit input and outputs 2^N 1-bit signals. The 1-bit output corresponds to the N-bit input bit pattern is true while all other outputs are false.
- The following figure shows a block diagram for a 2-to-4 decoder.



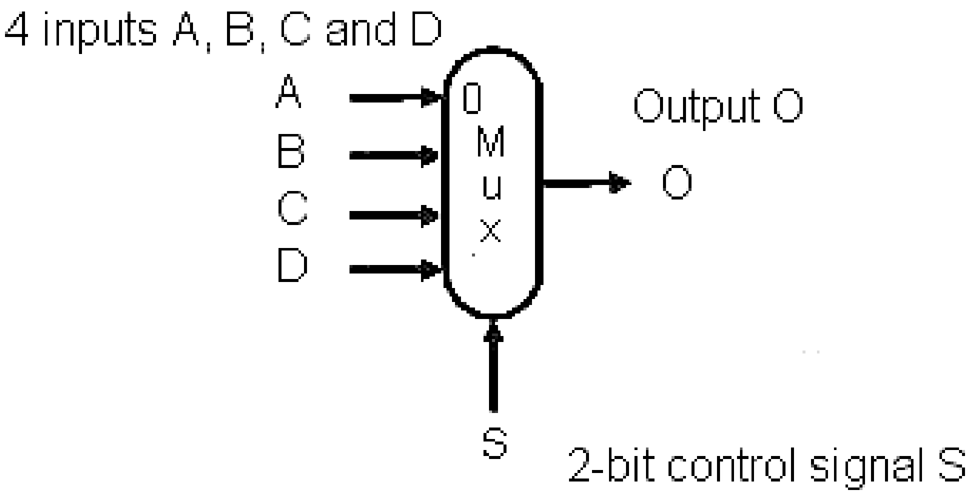
A 2-bit decoder

Questions

- Why a 2-bit input can generate 4 outputs in the decoder?
- If the input bits are 11, what will happen to the outputs of the decoder?
- Is it possible to have more than one outputs asserted?
- Name two potential uses of the decoder.
- Implement the decoder using Logisim.

Extra Exercise: Multiplexor

- A multiplexor is a devices that given the control signal, selects one of the inputs to be forwarded to the output. The following figure shows a 4-input multiplexor.
- 4-to-1 multiplexor



Questions

- If the inputs A/B are 32-bit in width, what is the data width of the Output O?
- What is the maximum number of inputs if the control signal is 10-bit in width?
- What is the bit-width of the control signal for the multiplexor if there are 9 inputs?