CHAPTER 4

COMBINATORIAL CIRCUITS

COMBINATORIAL CIRCUITS

In this chapter we look at the different combinatorial circuits;

1. Adders
2. Multiplexer and Demultiplexers
3. Comparators
4. Decoders and Encoders
5. ROMs and PLAs.

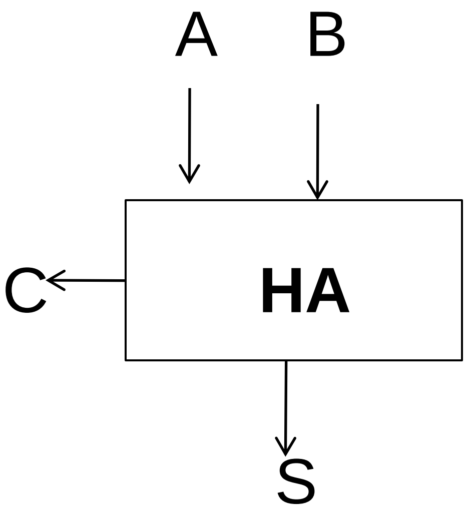
ADDERS

- An adder is a digital circuit in electronics that implements addition of binary numbers.
- Adders are of 3 types:
  - Half Adder
  - Full Adder
  - Ripple carry Adder

HALF ADDER

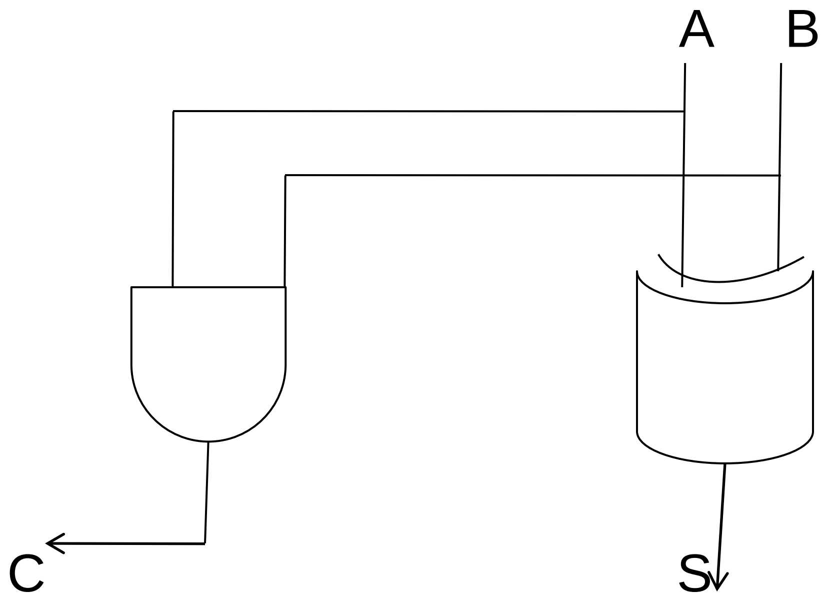
It is a circuit for adding 2, 1 bit quantities.

$$\begin{array}{r} A \\ +\ \underline{B} \\ \text{Carry}\quad \text{Sum} \end{array}$$



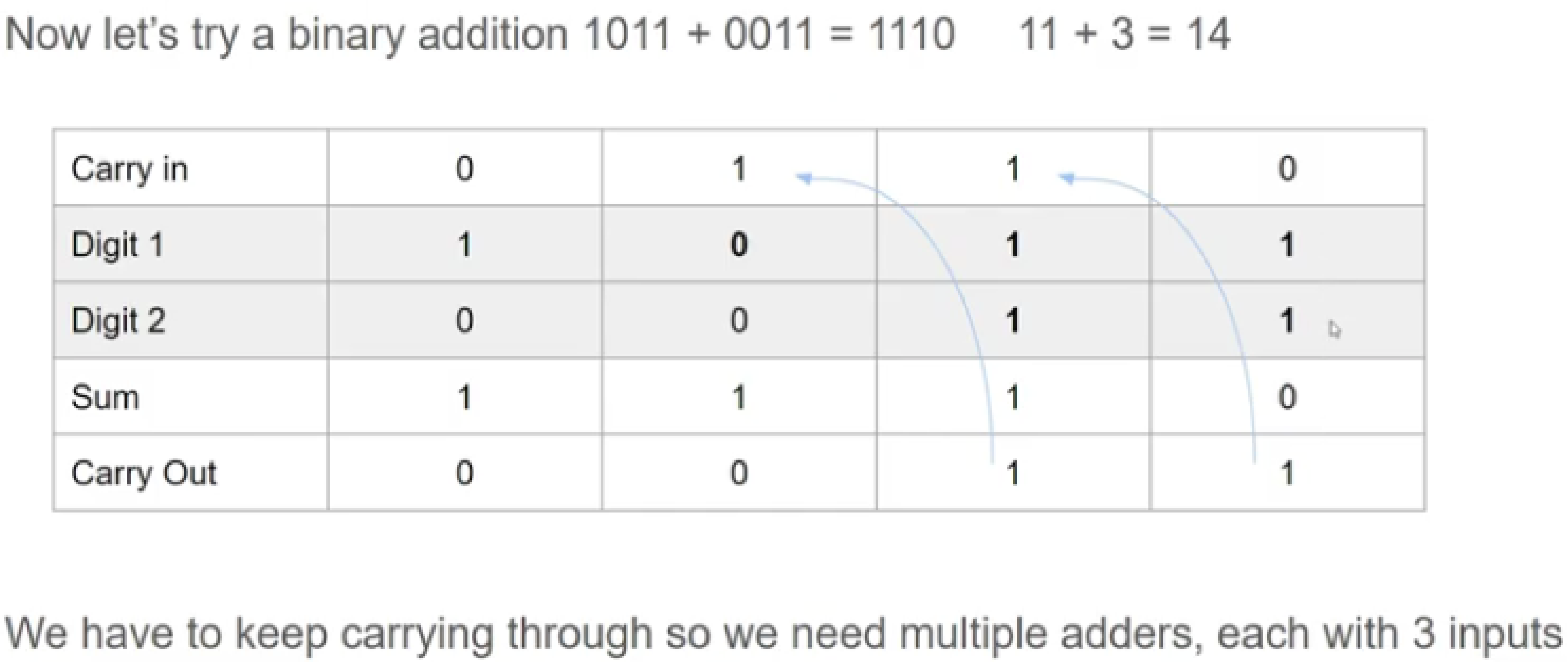
$$C = AB$$

$$S = \bar{A}B + A\bar{B} = A \oplus B$$



A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Example: Adding 2 binary numbers.



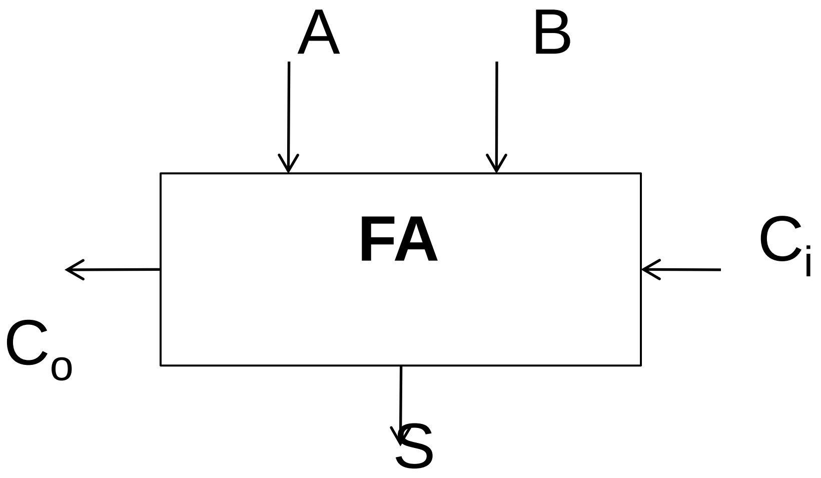
Now let's try a binary addition 1011 + 0011 = 1110   11 + 3 = 14

Carry in	0	1	1	0
Digit 1	1	**0**	**1**	**1**
Digit 2	0	0	**1**	**1**
Sum	1	1	1	0
Carry Out	0	0	1	1

We have to keep carrying through so we need multiple adders, each with 3 inputs

Full Adder

It is an adder that includes a carry input from a lower order sum.



A	B	C_i	S	C_O
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$\mathbf{C_o} = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$= C(\overline{A}B + A\overline{B}) + AB(\overline{C} + C)$$

$$= \mathbf{C(A \oplus B) + AB}$$

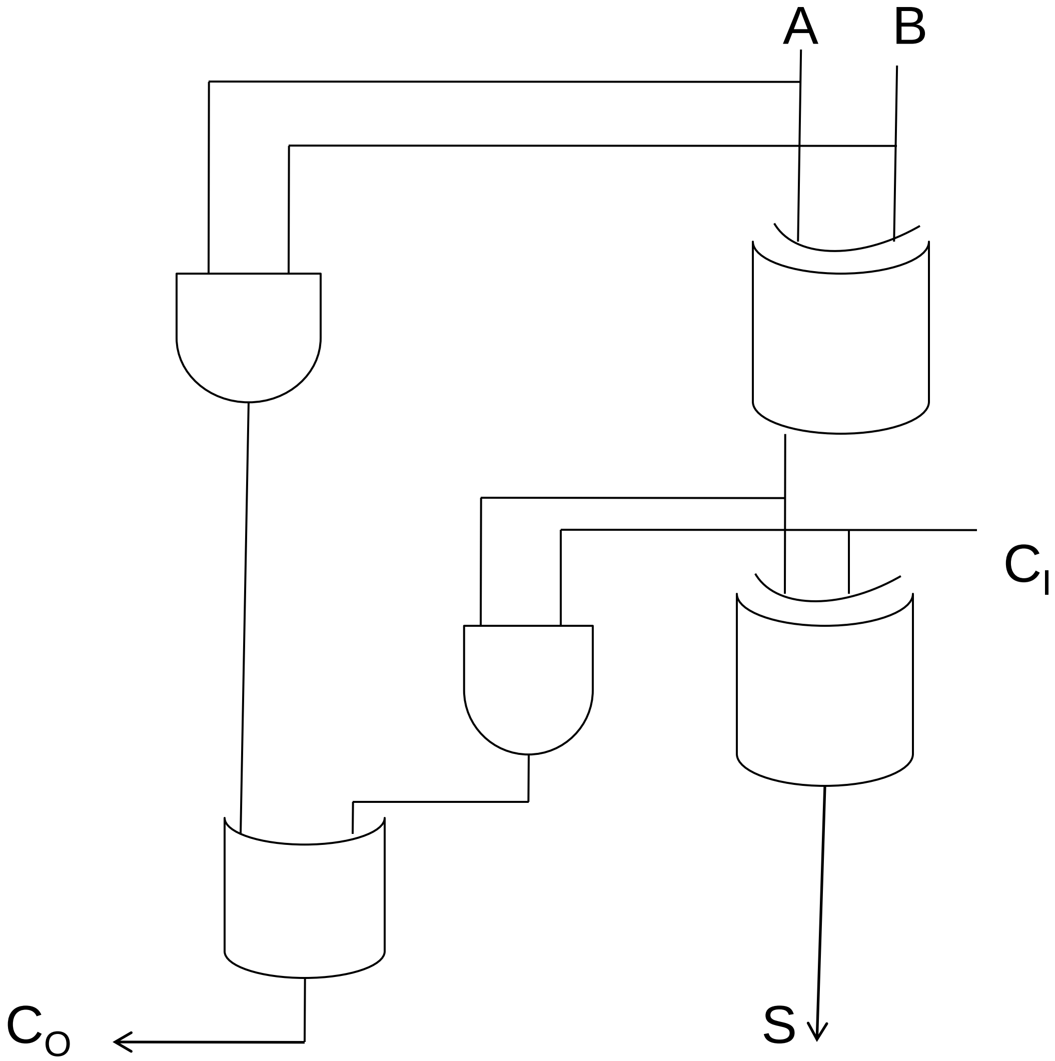
$$\mathbf{S} = \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC$$

$$= \overline{A}(\overline{B}C + B\overline{C}) + A(\overline{B}\,\overline{C} + BC)$$

$$= \overline{A}(B \oplus C) + A(\overline{B \oplus C})$$

$$= \mathbf{A \oplus B \oplus C}$$

FULL ADDER

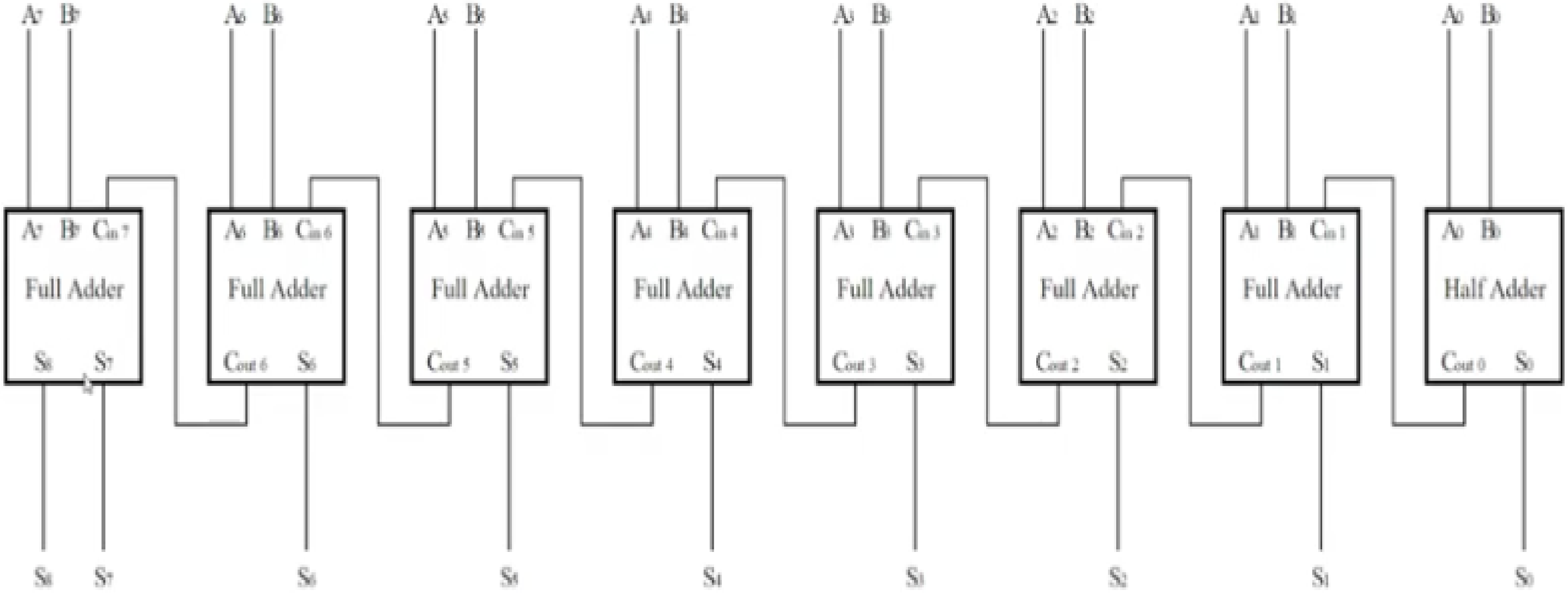


Ripple Carry Adder

It is a circuit built up of many full adders consisting of the required number of bits. **The carry out bit is used as a carry in into its left neighbour**. The carry into the right most bit is set to 0.

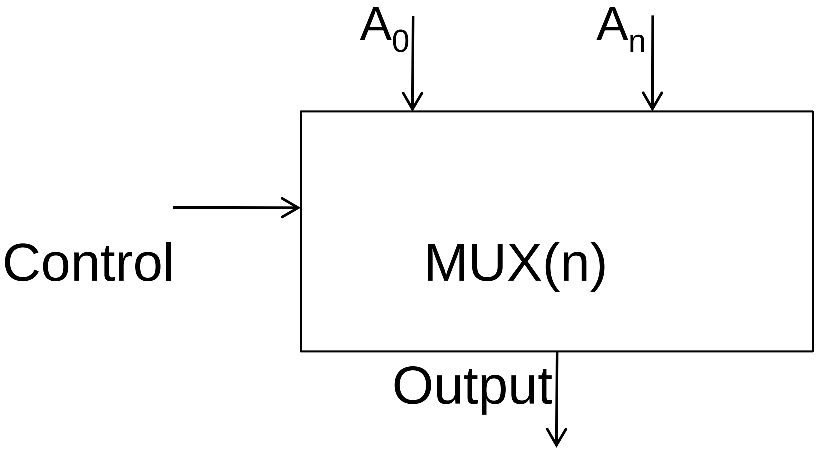
Ripple Carry Adder

8 bit adder



A Multiplexer (Data Selector)

- It is a logical network capable of selecting a single set of data inputs from a number of sets of inputs and it passes the selected inputs to the outputs.
- A multiplexer has 2 kinds of inputs, the ***control inputs*** and the ***data inputs***.
- The control inputs are used to select which of the inputs in the data is to be passed through to the outputs.



2-1 Multiplexer

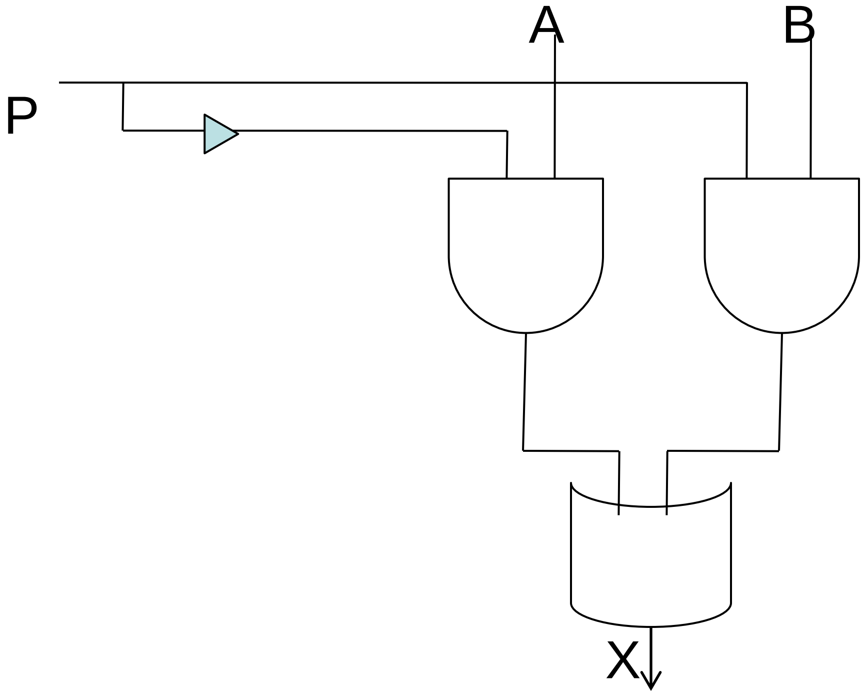
It has 2 data inputs A and B, one control input , P, and one output, X. When the control input P = 0 the output X is A and when P = 1 X = B

P	A	B	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$X = \overline{P}A\overline{B} + \overline{P}AB + P\overline{A}B + PAB$$

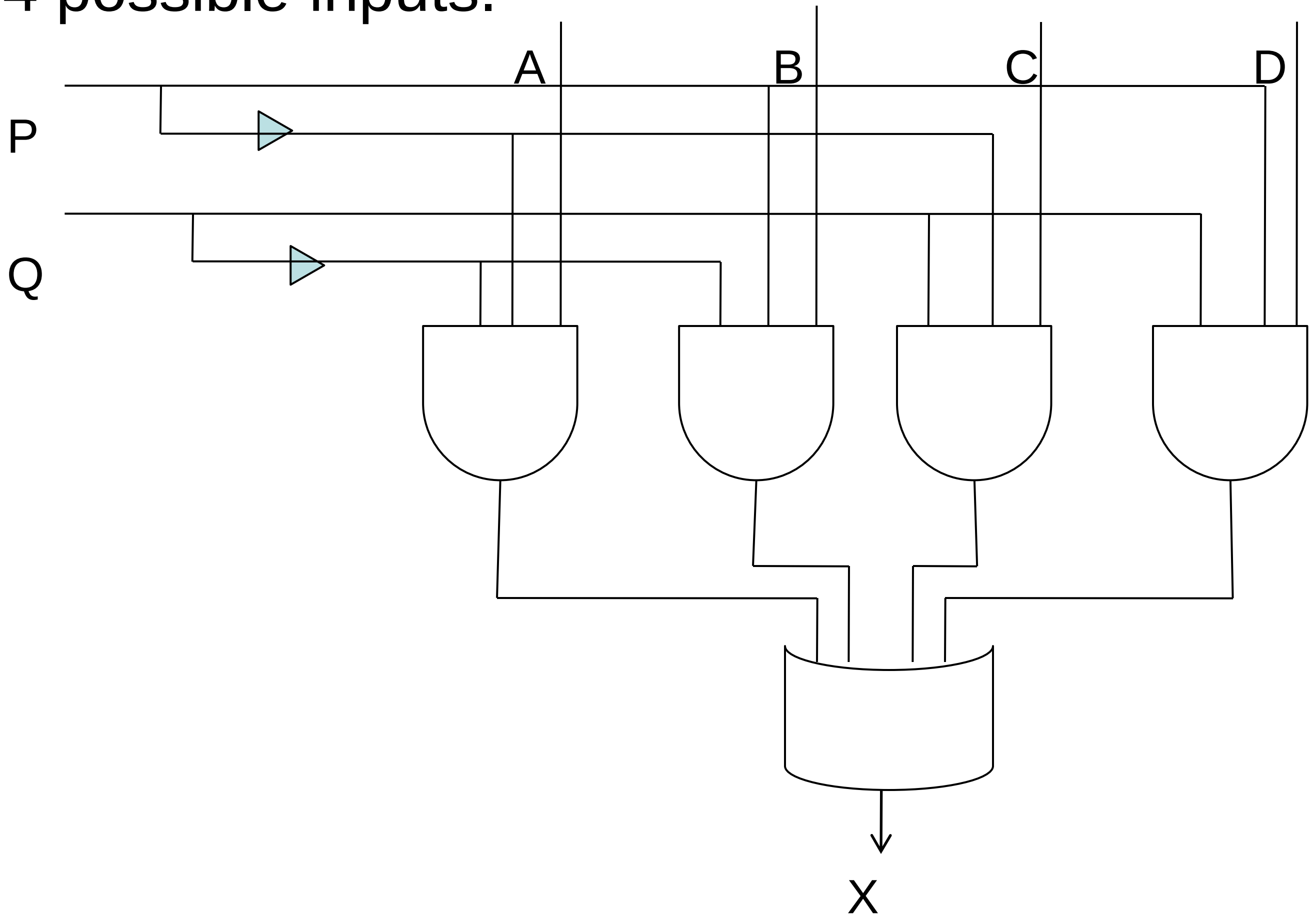
$$= \overline{P}A(\overline{B} + B) \ \ + PB(\overline{A} + A)$$

$$= \quad \mathbf{\overline{P}A + PB}$$



4-1 Multiplexer

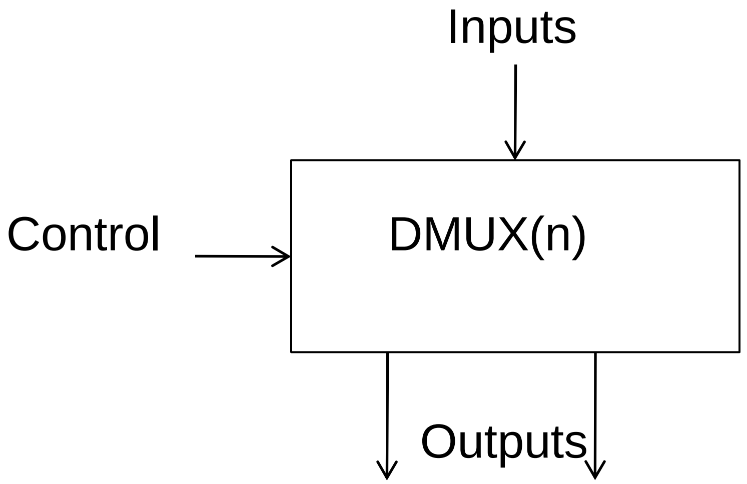
It selects only one of the 4 inputs. It has 2 control lines to choose one of the 4 possible inputs.



In general for an **n to 1** multiplexer, the inequality **2^K = n** must be satisfied where **k = number of control lines**.

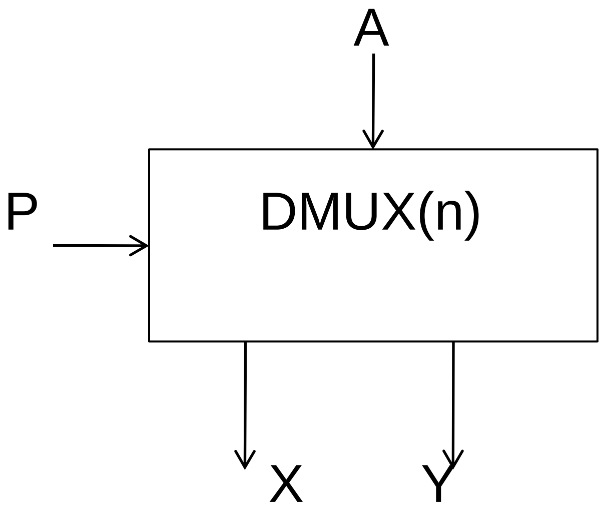
A Demultiplexer

It has 1 set of data inputs and two or more sets of outputs and a set of control inputs whose purpose is to select the set of outputs to transmit. The other outputs are 0.

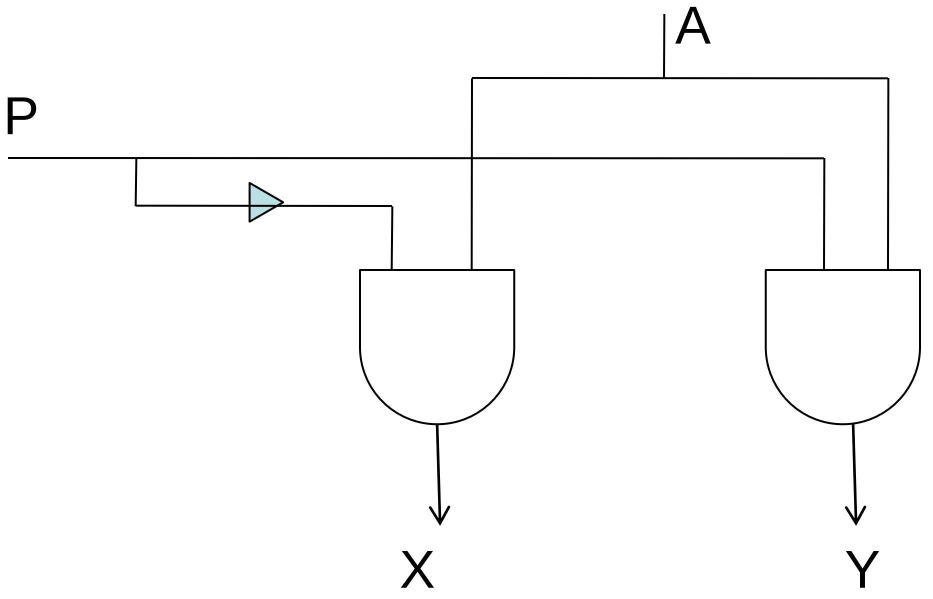


A 1- 2 Demultiplexer

It has 1 data input (A) , two outputs(X, Y) and one control input (P) .



$X = \overline{P}A;$ $Y = PA$

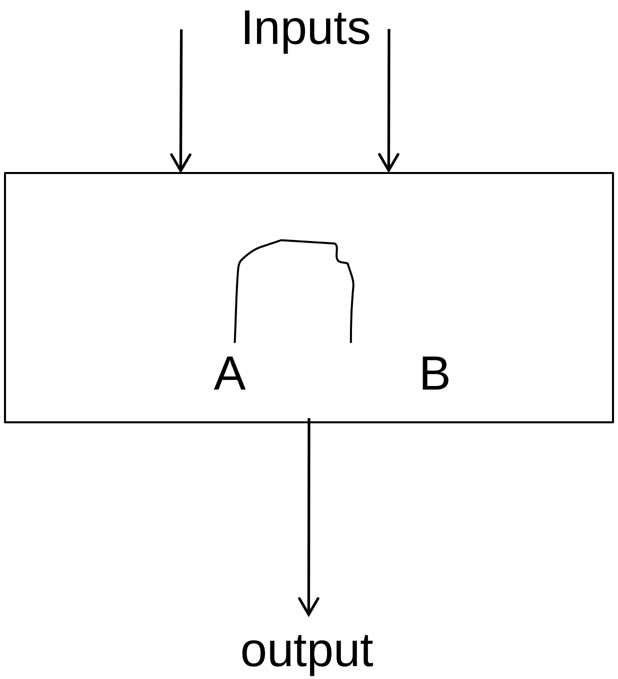


P	A	X	Y
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Comparators

A comparator compares two sets of inputs and outputs a 1 if the comparison is satisfied.

The comparisons are =,!=, >, >=, <, <=



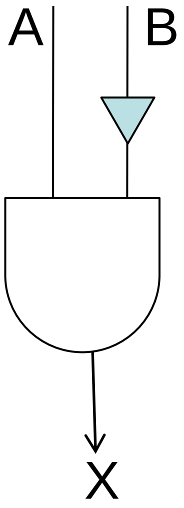
For equality comparator, the design is based upon the XNOR gate which outputs a 1 if its inputs are the same and a 0 if they are not equal.

1 bit > comparator

The output is 1 if A > B

$$X = A\overline{B}$$

A	B	X
0	0	0
0	1	0
1	0	1
1	1	0



Decoder

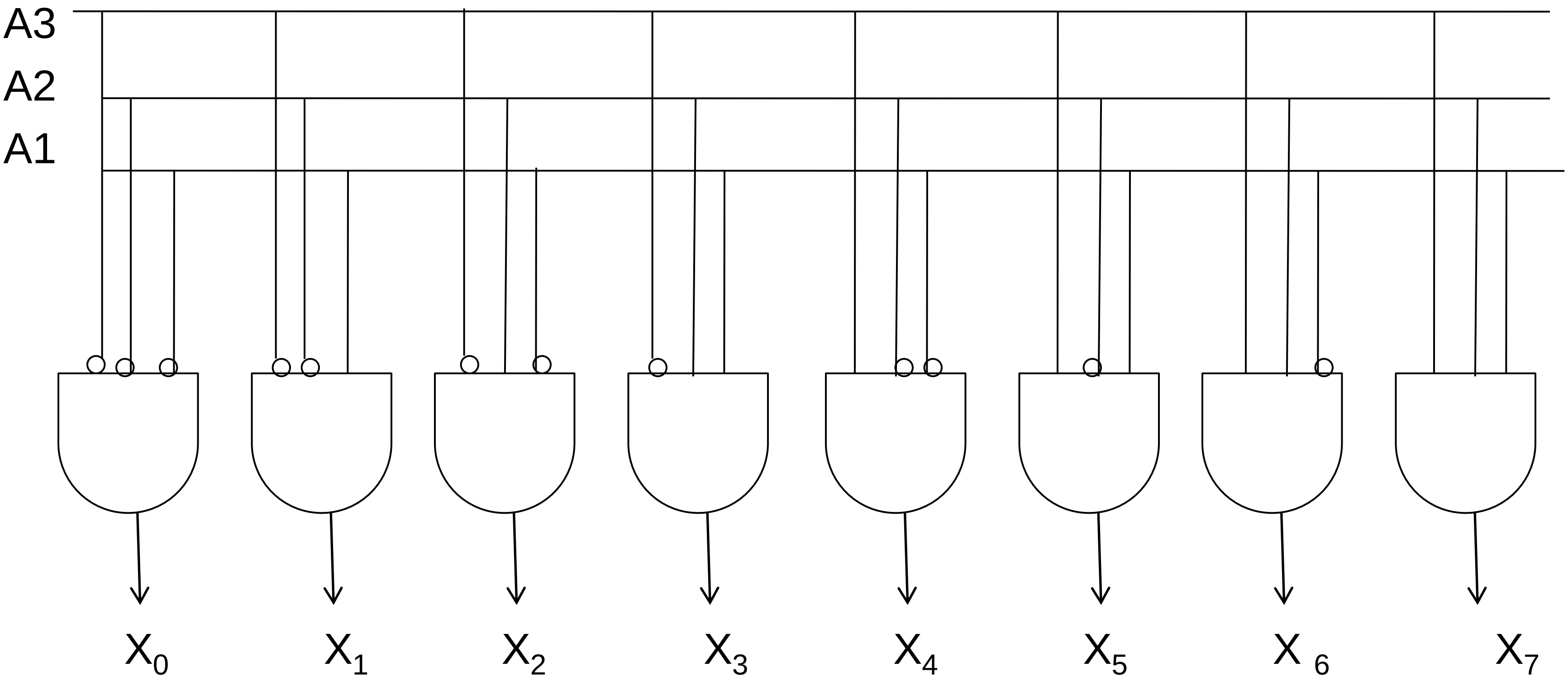
Circuit **whose outputs are miniterms of the inputs**. Exactly only one output is a 1 at any given time.

If **n** is the number of inputs and **m** the number of outputs then **2^n >= m**

e.g. if the binary number on the input lines is **k** then output line k will be 1 and all the others will be 0's.

A_1	A_2	A_3	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Decoder



The Encoder

The opposite of a decoder. Only one input can be a 1 (activated) at a time. Its number in binary is presented as the output.

It has 2^n inputs and **n** outputs.

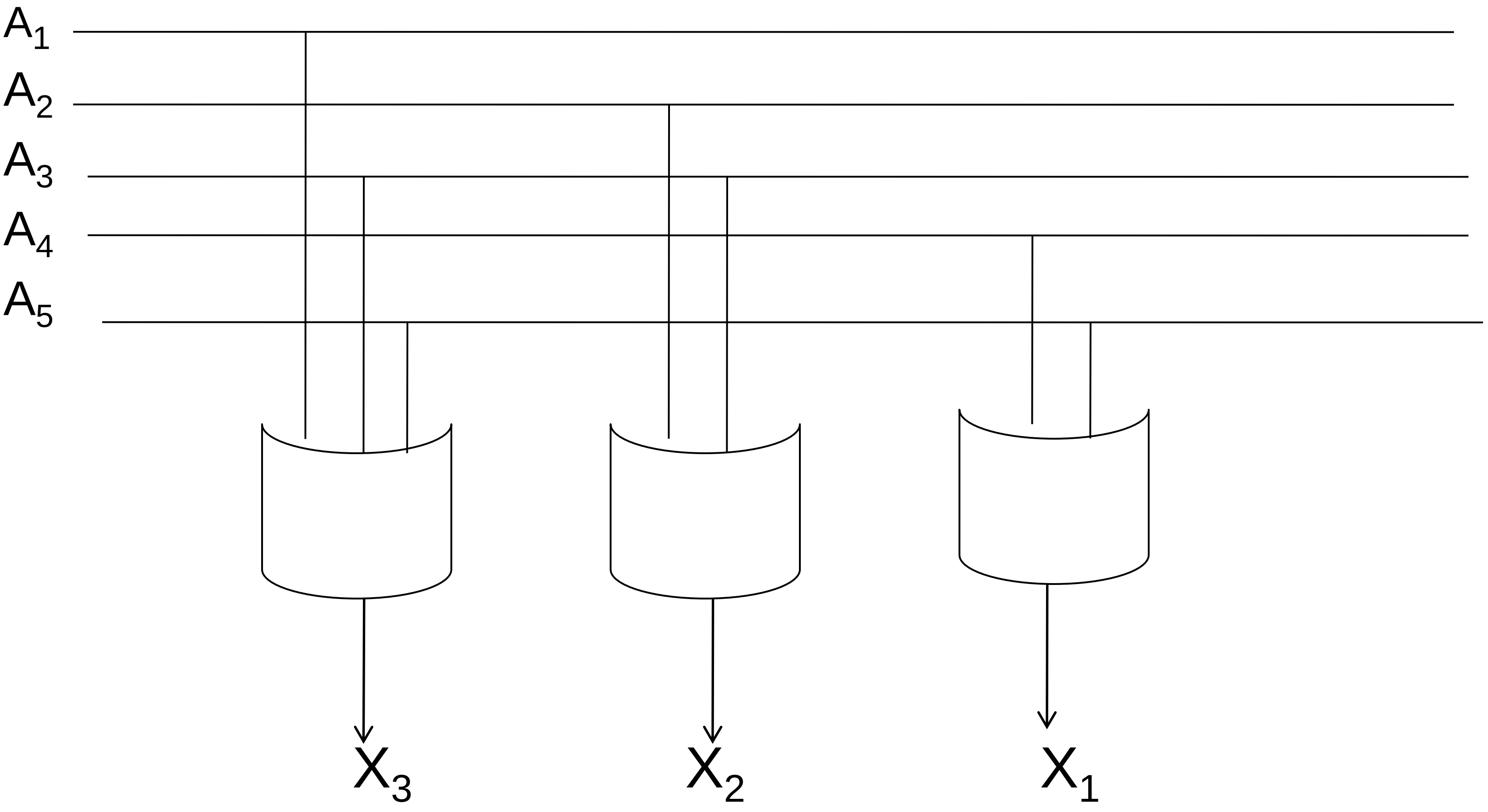
A_0	**A_1**	**A_2**	**A_3**	**X_1**	**X_2**
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Inputs					**Outputs**		
A_1	A_2	A_3	A_4	A_5	X_1	X_2	X_3
1	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	0	0	1	1	0	1

$X_3 = A_1 + A_3 + A_5$ **$X_2 = A_2 + A_3$**   **$X_1 = A_4 + A_5$**

Encoder

$X_3 = A_1 + A_3 + A_5$ $X_2 = A_2 + A_3$ $X_1 = A_4 + A_5$



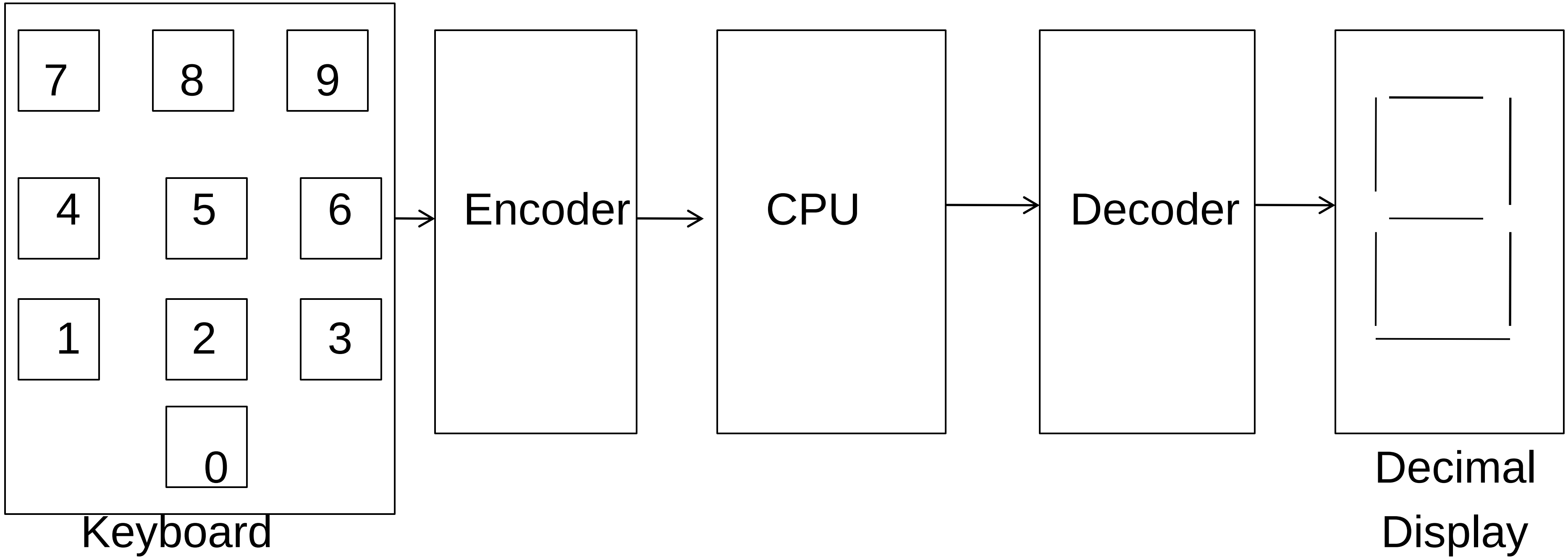
Code Converters

They are electronic circuits whose purpose is to convert data from one format to another.

Data in a computer system may take on several different forms as it changes from one format to another.

e.g. the decimal input from a keyboard calculator must be converted into BCD using an encoder.

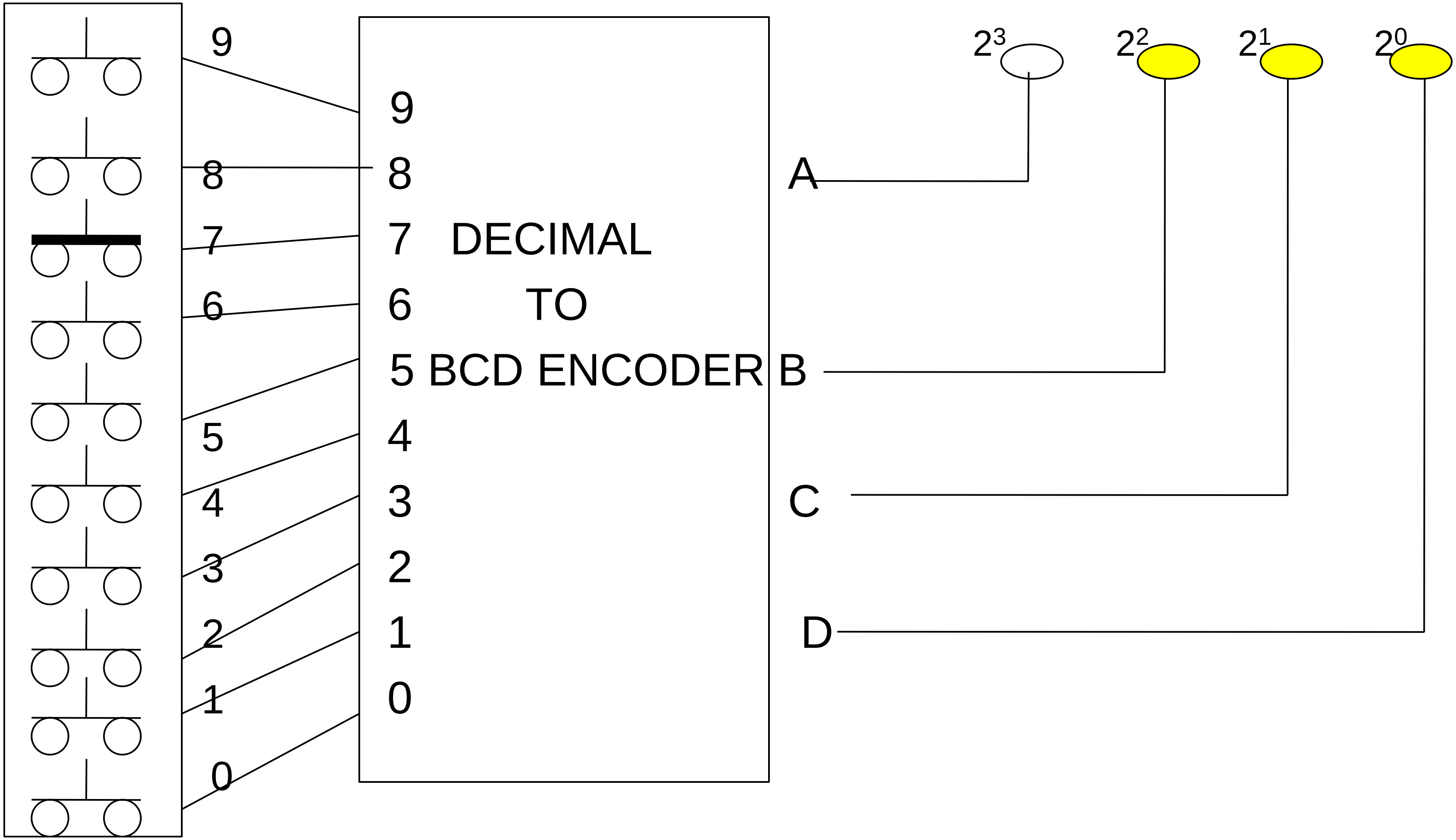
The CPU's output is in BCD and the decoder translates the BCD to a special 7 segment display code.



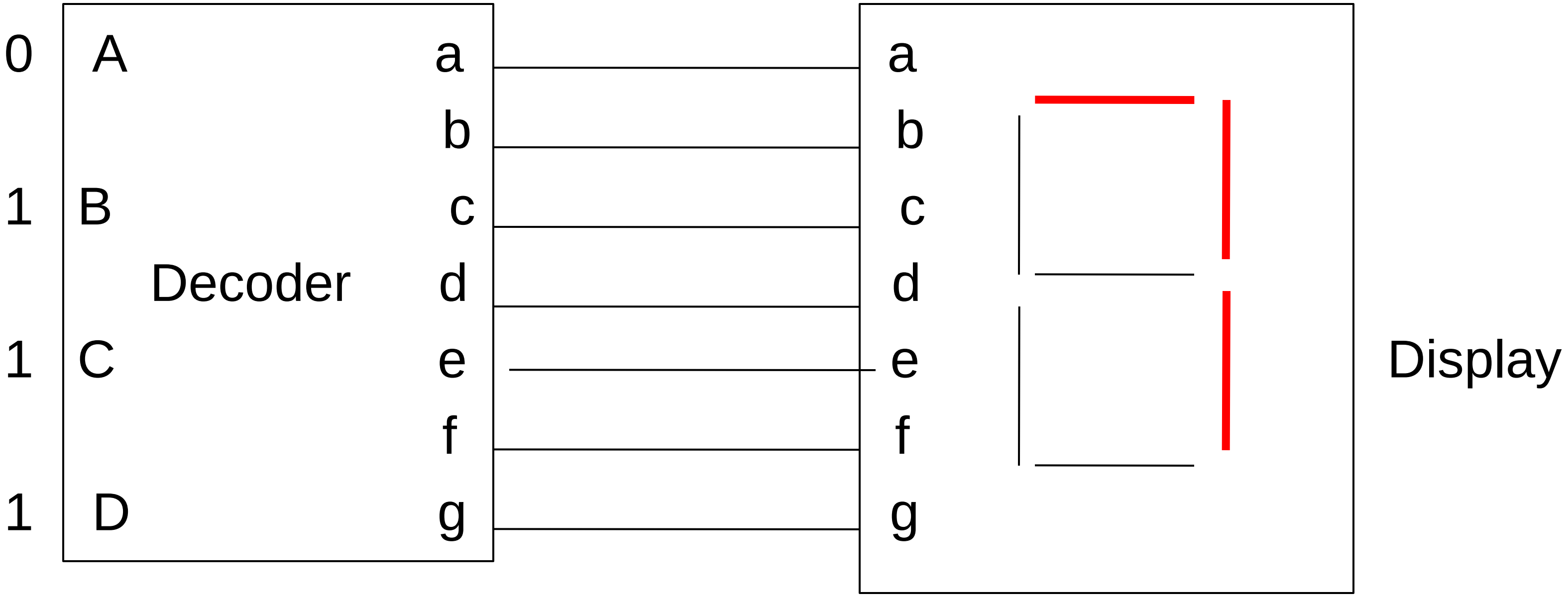
The encoder circuit

The encoder has 10 active inputs and 4 outputs connected to input lamps.

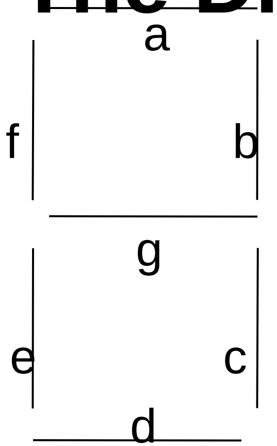
e.g. The input 7 causes a BCD output of 0111.



The Decoder circuit



The Display



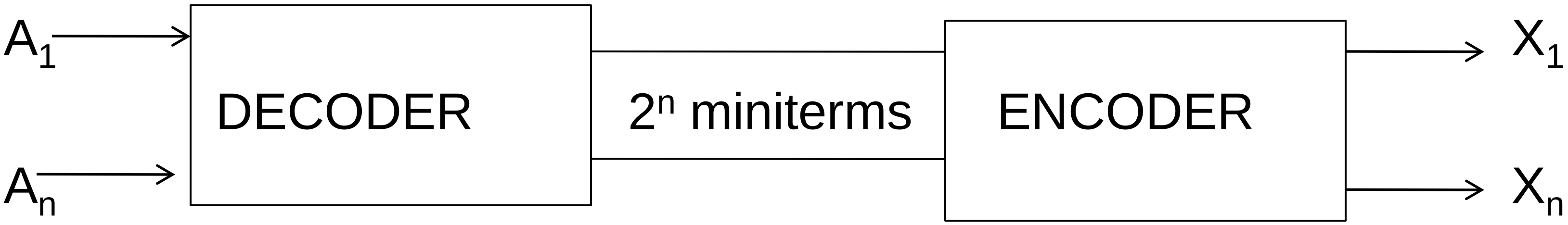
A	**B**	**C**	**D**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1

ROMS AND PLA'S

ROM (Read Only Memory)

- Its circuit is equivalent to a decoder. It outputs all possible miniterms of the inputs followed by an encoder.
- The output combinations are permanently embedded in its circuitry and the inputs serve to select one of these combinations. Each output is obtained by disconnecting the OR inputs from the AND gates whose miniterms are not to be included in the output.
- Because a ROM must produce all the possible miniterms its decoder portion is fixed by n (the number of inputs). The encoder portion depends on both the outputs and the way in which all the outputs of a decoder are used to generate the final ROM outputs.

ROMS



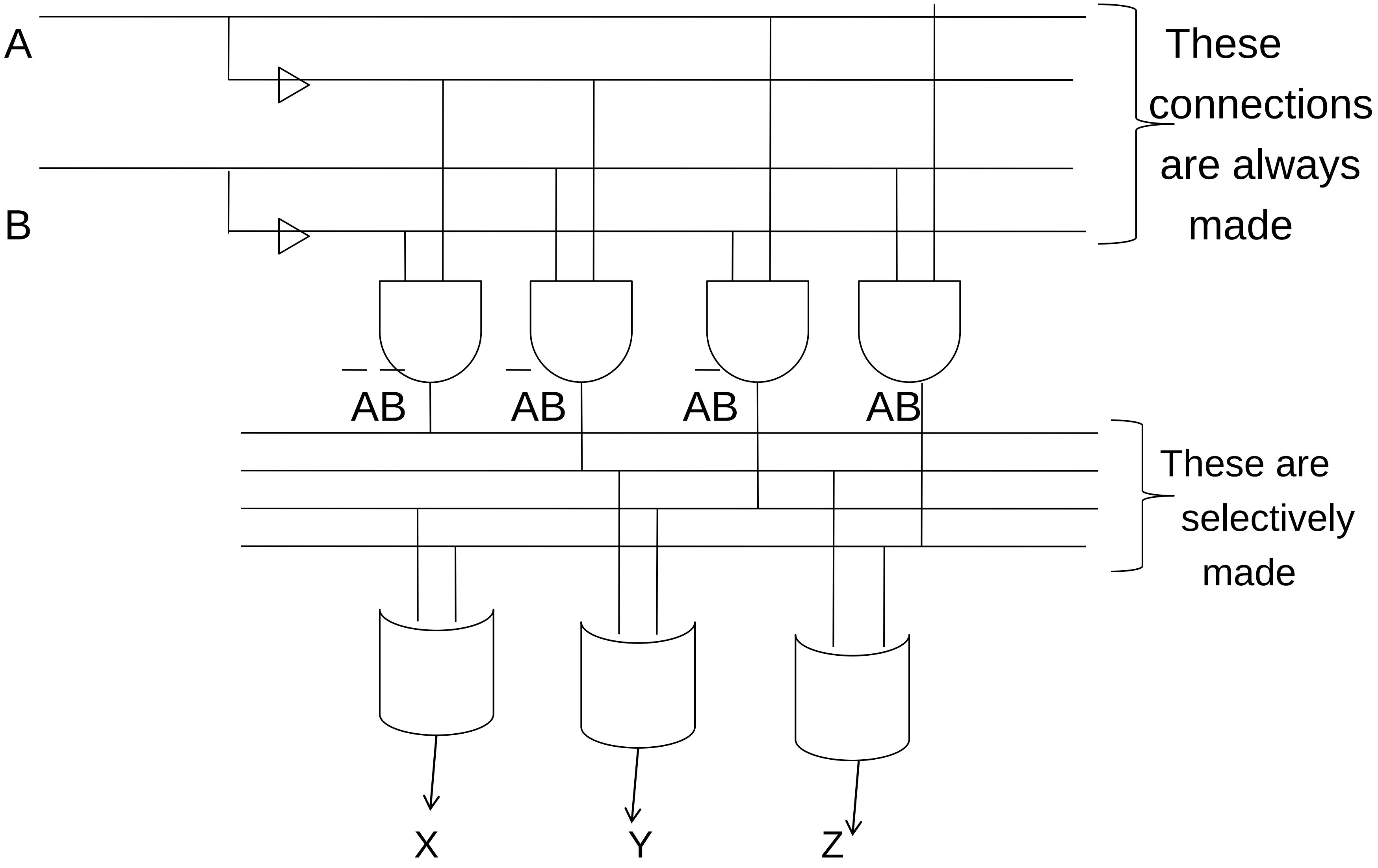
Once the disconnections are made they cannot be changed. (ROM nature)

EXAMPLE

Construct a 2 input (A,B) and 3 output (X,Y,Z) ROM such that:

$$X = A\overline{B} + AB; \qquad Y = \overline{A}B + A\overline{B}; \quad Z = \overline{A}B + AB$$

ROM EXAMPLE

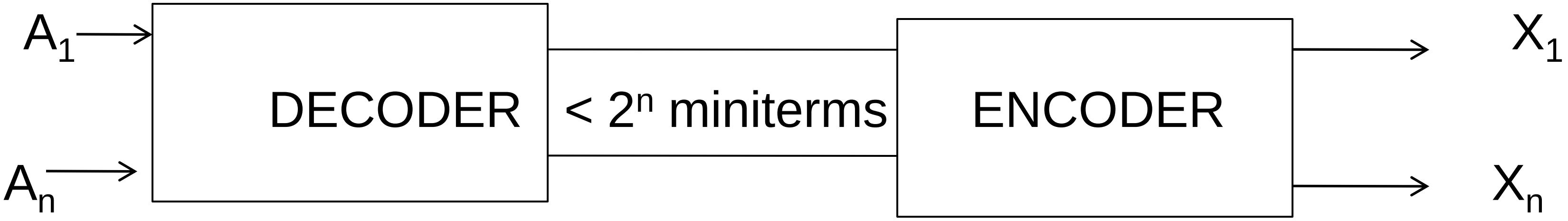


These connections are always made

These are selectively made

PLA (Programmed Logic Array)

- Similar to a ROM but does not output miniterms that will not be needed in any of the outputs. i.e. the decoder does not necessarily produce all the miniterms. For an n input network, we have <= 2^n AND gates



A_1

A_n

DECODER

< 2^n miniterms

ENCODER

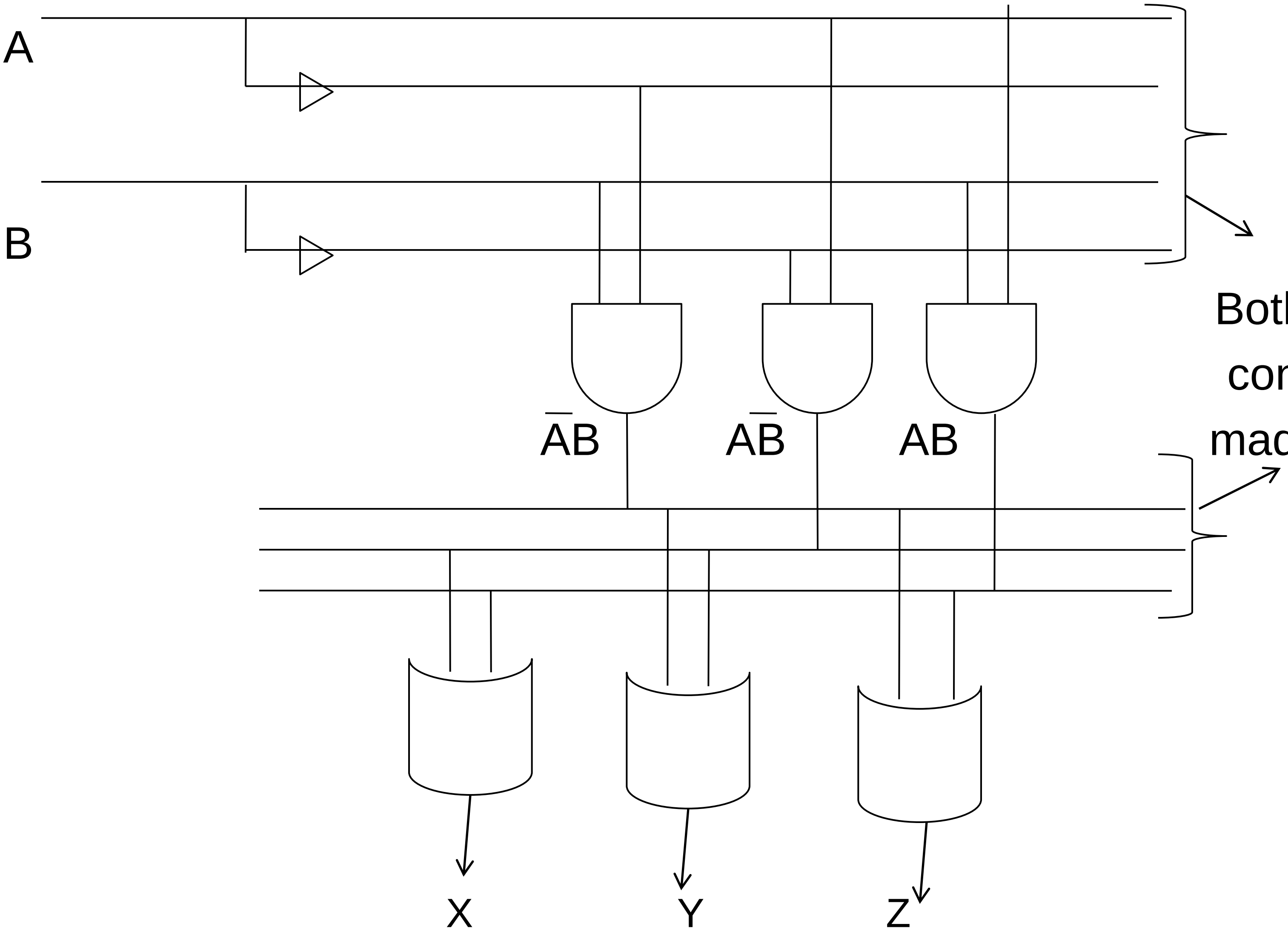
X_1

X_n

EXAMPLE

Implementation of the above ROM as a PLA

PLA example



Both sets of connections are made selectively

Reading Assignment 4

1. Read and make notes on the following circuits:

   a) Half Subtractor

   b) Full Subtractor

   Note:

   – Examine the 4 steps you go through to design the circuit.