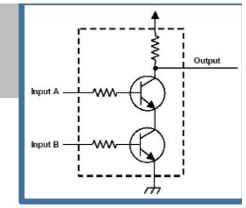
2. Combinational Circuits

Aims

- to show how logic gates are combined on a chip
- to show how logic gates can be combined to produce useful circuits such as :-

multiplexors and ***comparators***

2.0 Introduction



- Gates are made from **TRANSISTOR circuits**
  - Transistors **AMPLIFY** or **SWITCH** electrical signals. Gates use the **SWITCHING** property

- Gates are not manufactured individually but in units called **integrated circuits**

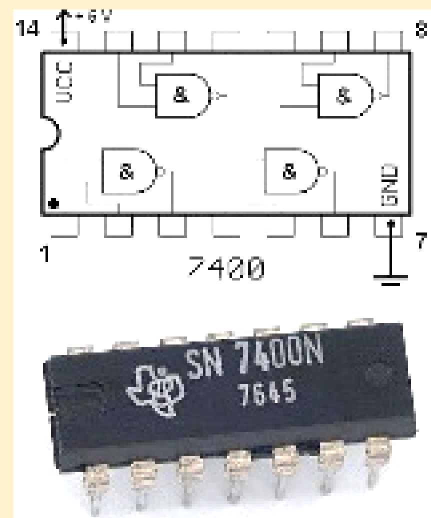
  (***ICs*** or ***chips***)

- Chips can be divided into rough classes based on the *number of gates* they contain :-

SSI (Small Scale Integrated)	circuit	1 to 10	logic gates
MSI (Medium Scale Integrated)	circuit	10 to 100	logic gates
LSI (Large Scale Integrated)	circuit	100 to 100,000	logic gates
VLSI (Very Large Scale Integrated)	circuit	> 100,000	logic gates

In the 1970s, computers were constructed out of large numbers of these chips, but technological advances have allowed an entire CPU to be etched onto a single chip

*The current state of the art allows nearly **10 million gates** on a chip*



Fairchild '74' series chip, 1969

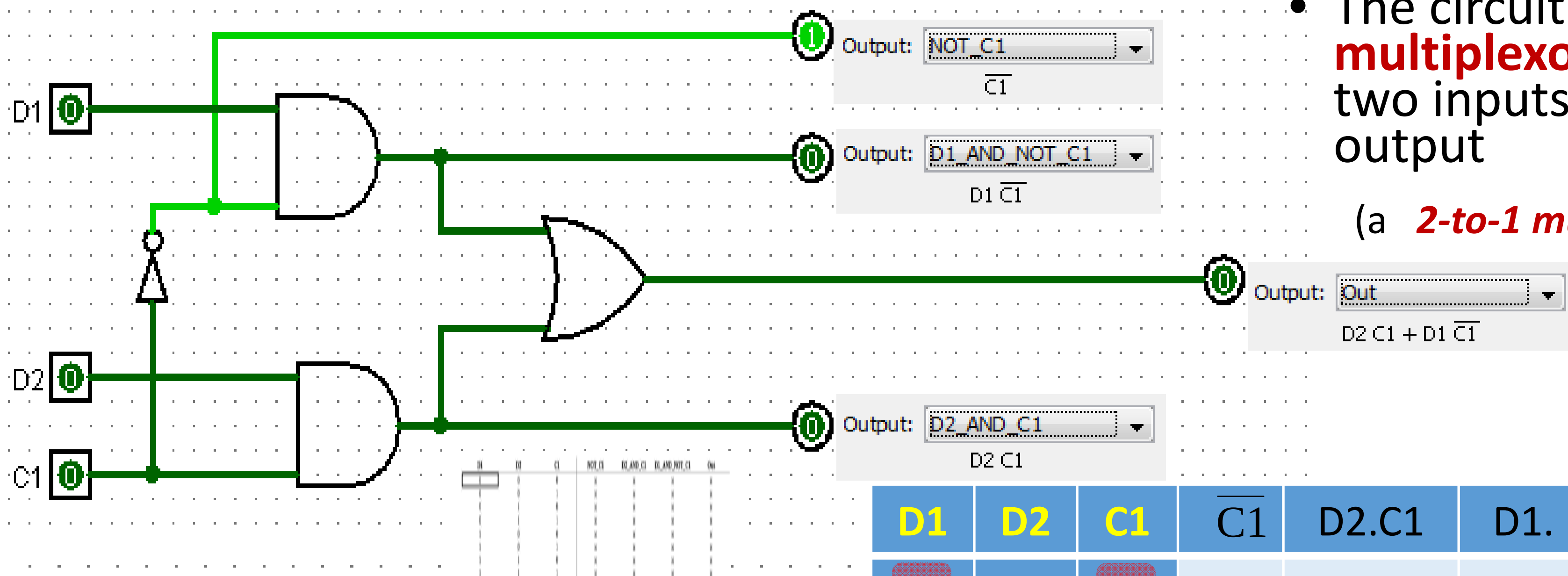
2.1 Combinational Circuits

Many circuits have the property of their outputs being wholly dependent on their inputs

Such circuits are called ***combinational circuits***

Over the next few weeks we will examine some frequently used combinational circuits

2.1.1 Multiplexors



- The circuit (left) is a **multiplexor** with two inputs and one output

  (a ***2-to-1 multiplexor***)

D1	D2	C1	$\overline{C1}$	D2.C1	D1. $\overline{C1}$	Out
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	0	1	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	1	0	1

D1	D2	C1	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- If you carefully examine the truth table you can see that the **MULTIPLEXER** output is
- either *D1* or *D2*, depending on the value of *C1*

D1	D2	C1	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

The **MULTIPLEXER** is acting as a **DATA SWITCH**

- The truth table reduces to : -

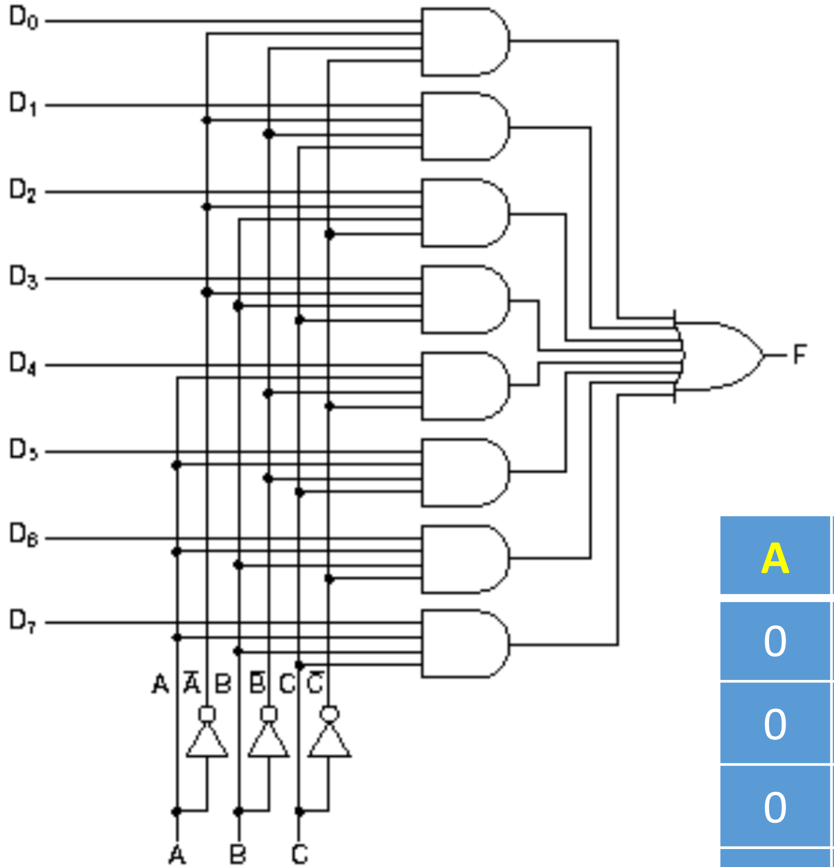
C1	Out
0	D1
1	D2

- The function of a multiplexor is to

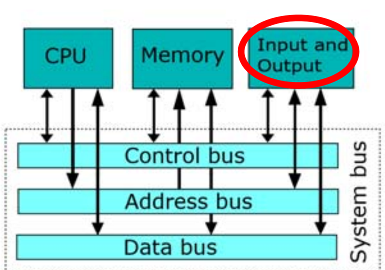
route *one* of many data inputs to a single output

The control input (***C1***) determines which of the data inputs is passed through to the output

A 2^n – to - 1 multiplexor has 2^n data inputs and n control inputs.

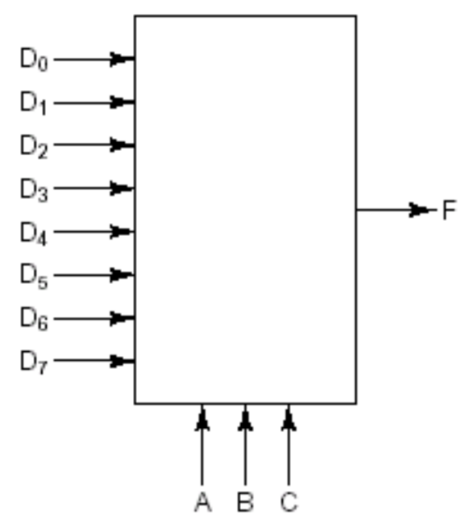


- The bit combination on the control inputs determines which data input is selected for output



A	B	C	Output
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

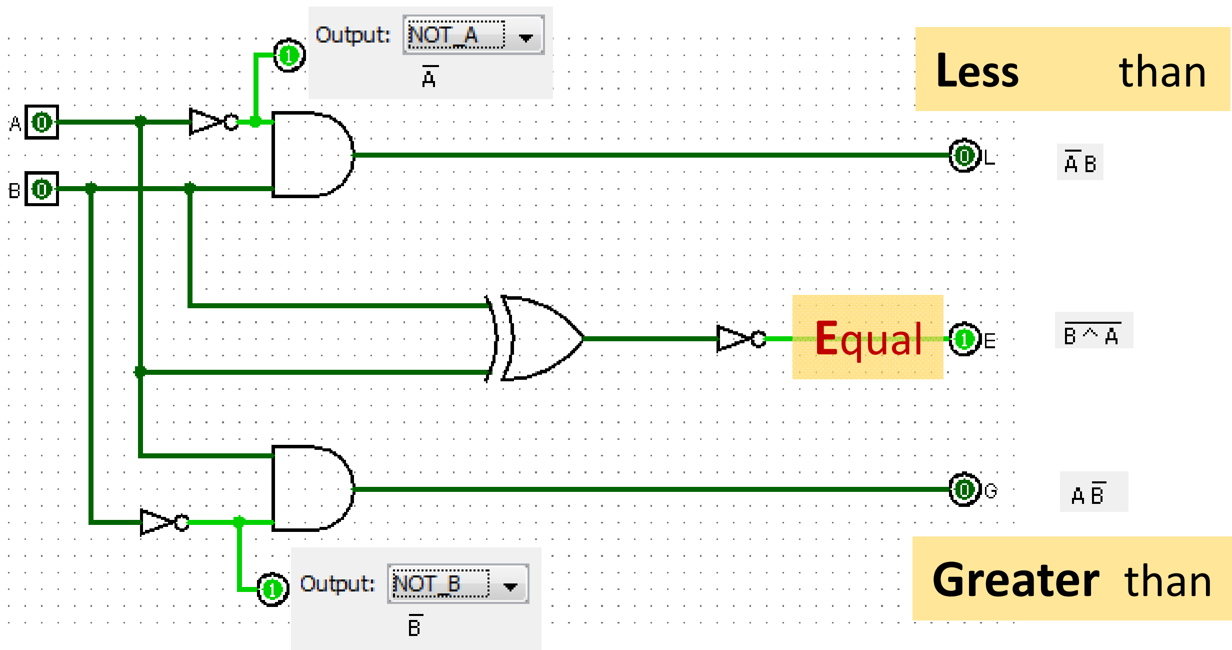
A simpler way to diagrammatically represent an ***eight-input multiplexor*** is shown below.



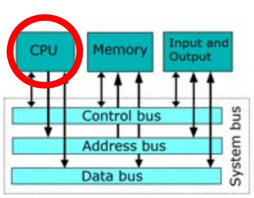
2.2 Comparators

- A ***comparator*** compares *two* inputs

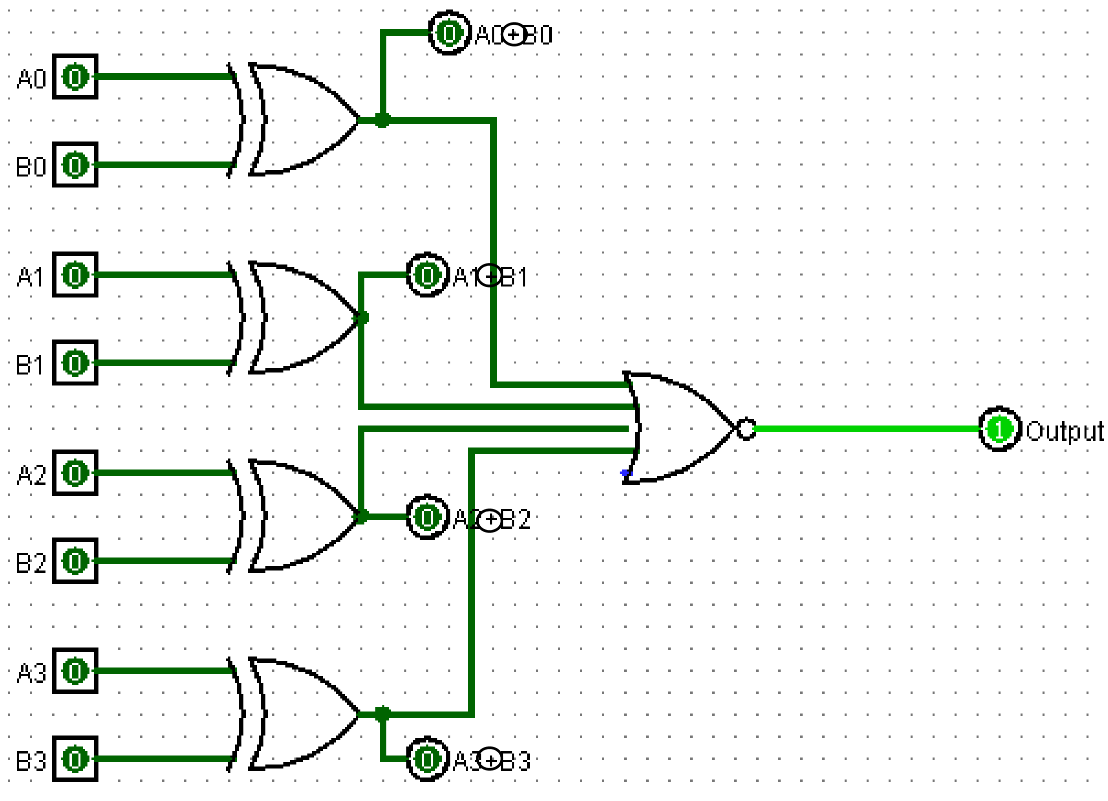
and outputs a **1** on the **E** output if the inputs are **E**qual



A	B	$\overline{A}$	$\overline{B}$	L	E	G
0	0	1	1	0	1	0
0	1	1	0	1	0	0
1	0	0	1	0	0	1
1	1	0	0	0	1	0



Multiple (parallel) Bit EQUAL function



A0	B0	A1	B1	A2	B2	A3	B3	A0B0	A1B1	Output	A2B2	A3B3
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	1	1	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	1	0	1	0	0	0	1	1
0	0	0	0	0	1	1	0	0	0	0	1	1
0	0	0	0	0	1	1	1	0	0	1	1	0
0	0	0	0	1	0	0	0	0	0	1	1	0
0	0	0	0	1	0	0	1	0	0	0	1	1
0	0	0	0	1	0	1	0	0	0	0	1	1
0	0	0	0	1	0	1	1	0	0	1	1	0
0	0	0	0	1	1	0	0	0	0	1	0	0
0	0	0	0	1	1	0	1	0	0	0	0	1
0	0	0	0	1	1	1	0	0	0	0	0	1
0	0	0	0	1	1	1	1	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1	0	1	0	0	1
0	0	0	1	0	0	1	0	0	1	0	0	1
0	0	0	1	0	0	1	1	0	1	0	0	0
0	0	0	1	0	1	0	0	0	1	0	1	0
0	0	0	1	0	1	0	1	0	1	0	1	1
0	0	0	1	0	1	1	0	0	1	0	1	1
0	0	0	1	0	1	1	1	0	1	0	1	0
0	0	0	1	1	0	0	0	0	1	0	1	0
0	0	0	1	1	0	0	1	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1	0	1	1
0	0	0	1	1	0	1	1	0	1	0	1	0
0	0	0	1	1	1	0	0	0	1	0	0	0
0	0	0	1	1	1	0	1	0	1	0	0	1
0	0	0	1	1	1	1	0	0	1	0	0	1
0	0	0	1	1	1	1	1	0	1	0	0	0
0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	1	0	0	1
0	0	1	0	0	0	1	0	0	1	0	0	1
0	0	1	0	0	0	1	1	0	1	0	0	0
0	0	1	0	0	1	0	0	0	1	0	1	0
0	0	1	0	0	1	0	1	0	1	0	1	1
0	0	1	0	0	1	1	0	0	1	0	1	1
0	0	1	0	0	1	1	1	0	1	0	1	0
0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	1	0	1	0	0	1	0	1	0	1	1
0	0	1	0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	0	1	0	1	0
0	0	1	0	1	1	0	0	0	1	0	0	0
0	0	1	0	1	1	0	1	0	1	0	0	1
0	0	1	0	1	1	1	0	0	1	0	0	1

A0	B0	A1	B1	A2	B2	A3	B3	A0⊕B0	A1⊕B1	A2⊕B2	A3⊕B3	NOR
0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	1	0	1	0	1	1	1	1	1	0
1	0	1	0	1	0	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1	0	0	0	0	1

REFERENCES

- http://worldclassprogramme.com/Arithmetic-Circuits.php