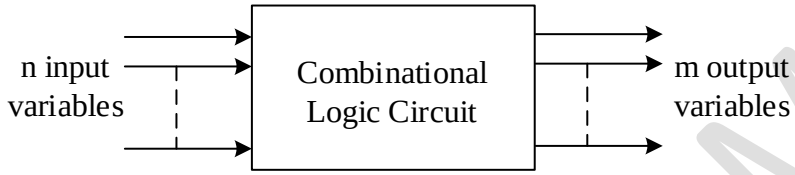
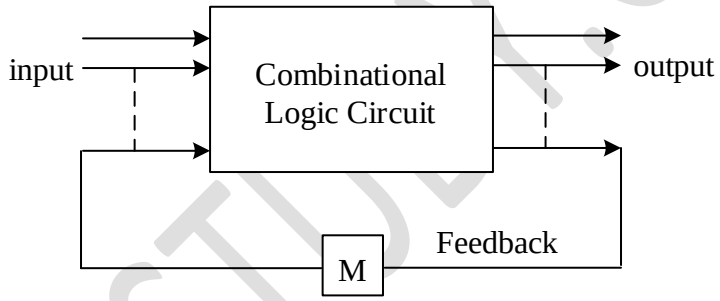
Combinational Circuits DC-IV (Part I) Notes

Digital Circuits have been classified as:

(a) **Combinational Circuits:** In these circuits output at any instant of time depends on inputs present at that instant of time. This means that output is dependent at all times on the combination of its inputs .These circuits have no memory or feedback loops. Given Fig gives Block Diagram representation for combinational circuits.



(b) **Sequential Circuits:** In these circuits output at any instant of time depends upon the present inputs as well as past inputs/outputs. These have memory.
Given Figure gives block diagram representation for Sequential Circuits. There can be more than one feedback paths.



This part of note discusses combinational circuits which are classified as Arithmetic Circuits and comparators, other combinational circuits will be considered in Part II.

Combinational circuits: They are made up from basic logic gates and are building blocks of combinational circuits .They are classified as

(i) **Arithmetic and logical functions:** Like, adders, subtractors, comparators, etc.
(ii) **Data transmission:** Like Multiplexers, De Multiplexers, Encoders, Decoders, etc.
(iii) **Code converters:** Like Gray to Binary, Binary to BCD, BCD to binary, etc.

These circuits can be designed using gates and can be implemented using small scale integrated circuits (SSIs).

Design procedure: Any combinational circuit can be designed as per following procedure (classical approach). They can be designed using gates and can be implemented using small scale integrated circuit (SSIs)

(i) From the word statement identify and draw the block diagram.
(ii) Write the truth table.
(iii) Write down the Boolean Expression

(iv) Simplify the Boolean Expression using theorems or K-Map.
(v) Implement the expression using logic gates.

Above classical approach of design says that if the two circuits perform the same function then the one with less number of gates is preferable since it will cost less. This is not true when ICs are used. Since several gates are included in a single IC, so it is economical to use as many Gates as possible from an already used package .Also in some ICs the interconnection of gates are internal, so it is always better to use connections which are internal for better reliability. So with ICs it is not the number of gates that determine the cost but the number and type of ICs. So the design should consider that whether the function is available in IC package. The selection of MSI components in preference to SSI gates is extremely important since it would result in considerable reductionof IC packages and interconnecting wires.

Many combinational circuits are available in MSI (Medium scale Integrated Circuits).Components such as adders, subtractors,comparators,decoders and encoders**,**multiplexers de multiplexers etc.

Part I of notes discusses Arithmetic circuit such as adders/subtractors, multipliers and comparators.

Part II will discuss other combinational circuits such as decoders, encoders, multiplexers, demultiplexers etc.

Arithmetic Circuits: Computers and calculators perform arithmetic operations. They are performed in arithmetic logic unit (ALU) of computer. They need to be performed at a high speed.

The basic unit of arithmetic circuit is adder.

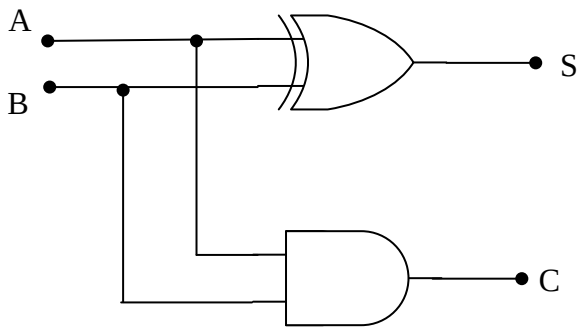
Half adder: It adds two 1 bit values and produce a sum and carry output. The half part of the name comes from the lack of carry input.

Full adder: It adds two 1 bit values plus a carry and produces a sum and a carry output. The full part of the name comes from carry input bit.

As an illustration, truth table and logic diagram for half adder are given:

Truth Table

Inputs		Outputs	
A	B	SUM (S)	CARRY (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Logic diagram for half adder

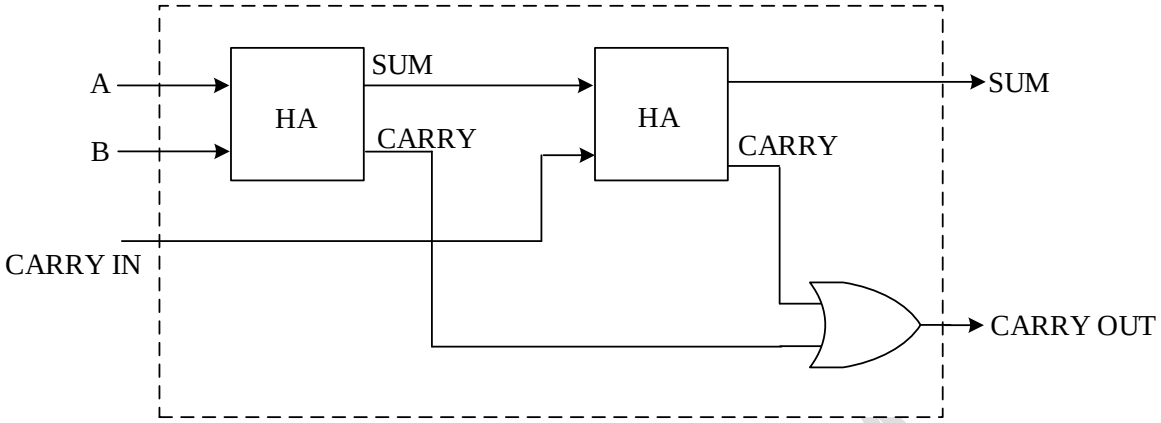
Adders and subtractors have been described in most of the text books. MCQs on combinational circuit on the website have included them. These notes will summaries the related equations for binary adders and subtractors. Key points are highlighted. They may be helpful in problem solving.

Comparative study for adders and subtractors:

Adders	Subtractors
Half $S = A \oplus B$	$D = A \oplus B$
C = A.B	$B_{out} = \bar{A}.B$
Full $S = A \oplus B \oplus C_{in}$	$S = A \oplus B \oplus B_{in}$
$C_{out} = BC_{in} + AC_{in} + AB$ $= (A + B)C_{in} + AB$	$B_{out} = \bar{A}B + \bar{A}B_{in} + B.B_{in}$ $= B_{in}(\bar{A} + B) + \bar{A}B$

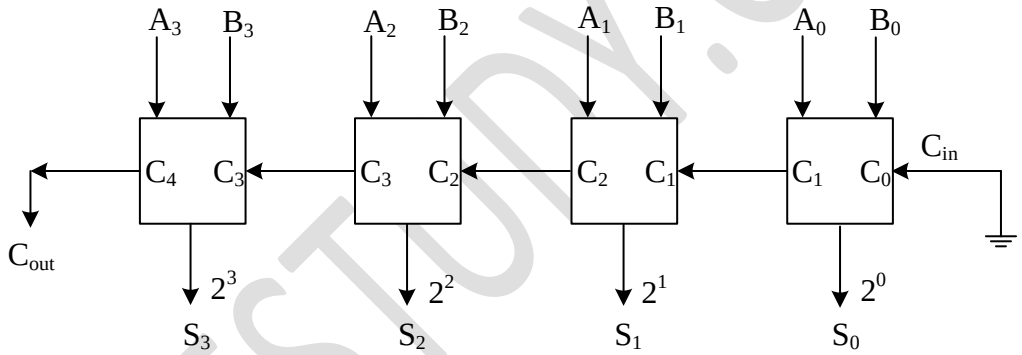
Key Points:

- **Half adders/subtractors: They** are implemented using Ex-OR gate for sum (S) and AND gate for carry/borrow.
- **Full adders/subtractors:** Sum and difference implemented using 3 input EX-OR gates. C_{out} and B_{out} expressions are the same except input variable A is complemented
- n bit adder requires (n-1) full adders and one half adder
- Full adder can be implemented using two half adders.
- An n bit adder can be built by cascading n full adders.
- Full adders can be converted to full subtractors by complementing input A prior to its applications to gates that form the carry output.



Full Adder

Binary Parallel Adder (Or Ripple carry Adder): It has full adders connected in cascade with the output carry from one full adder connected to input carry of next full adders.



4 bit binary parallel adder

- Speed is limited due to carry propagations delay. 4 bit adder output is generated after 4 propagation delays.
- IC adders such as 74LS83 can be used to construct high speed parallel adders and subtractors.
- Two or more parallel adders can be connected in cascade to perform addition for larger numbers.
- A parallel binary subtractor can also be implemented by cascading several full subtractors.
- By using EX-OR gate as controlled inverter adder/subtractor can be implemented in the same circuit.

Fast Adder: Carry propagation delay can be reduced by a look ahead logic circuit.

Serial Adder: The disadvantage of parallel adder is that it requires large amount of circuitry. An alternation is serial addition, the operation is performed bit by bit, thus simpler circuitry but low speed.

Serial Adder	Parallel Adder
1. Low in speed	High speed
2. Requires less component	More hardware
3. Performed bit by bit	All bits added at the same time

BCD Adder: It adds two BCD digits in parallel and produces sum in BCD form

- Add two BCD numbers using binary addition.
- If 4 bit sum is less than 9 sum is in BCD form.
- If it is greater than 9 a correction of $0110\ \ (6_{10})$ should be added to sum to produce the proper BCD result. This will produce carry to be added to next decimal position.

Multiplication of Binary Numbers:

It can be carried out by the following methods.

(i) Partial product addition and shifting
(ii) Parallel multipliers or carry multipliers

4 bit multiplier using shift method requires 4 cycles of addition and shifting operations, but it requires only a single 4 bit parallel adder. The speed of multiplication process can be increased considerable in parallel multiplier at extra cost of increased hard ware. Parallel multiplier is discussed here with an example of 2 bit numbers.

		a_1	a_0	Multiplicand
		b_1	b_0	Multiplier
		$a_1 b_0$	$a_0 b_0$	Partial product
	$a_1 b_1$	$a_0 b_1$		Partial product
O_3	O_2	O_1	O_0	Product or output

Note that each product bit or output bit (O_x) is formed by adding partial product cols. Each product is formed by AND gates.

Also note that binary multiplication is same as AND truth table. So such multiplier can be implemented using AND gates and adders.

$$O_0 = a_0 b_0$$

$$O_1 = a_1 b_0 + a_0 b_1 + c_0$$

$O_2 = a_1 b_1 +$ c1

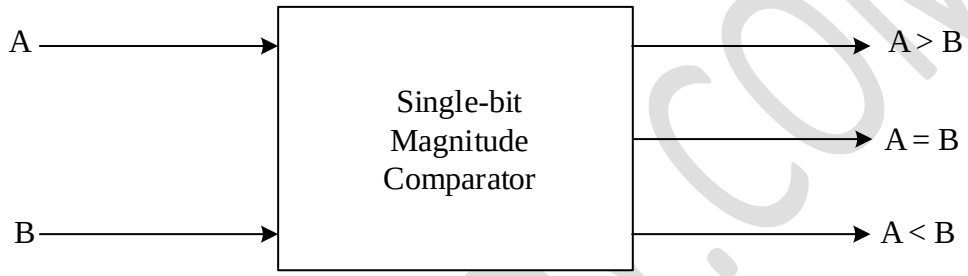
$O_3 = c_2$

Also note that binary multiplication is same as AND truth table. So such multiplier can be implemented using AND gates and adders.

Magnitude Comparators:

A binary magnitude comparator is a logic circuit that provides output information indicating relative magnitude of two inputs. These output conditions exist as a result of comparison of two inputs.

Block diagram for single bit comparator and logical implementation is given in the figures



Block Diagram

Truth Table

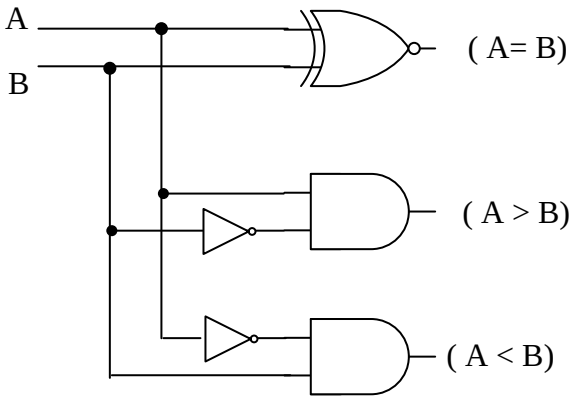
A	B	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

From the truth table the equation for each output is

A = B Logic equation $\bar{A}\bar{B} + AB = \overline{(A \oplus B)}$

A > B Logic equation $A\bar{B}$

A < B Logic equation $\bar{A}B$



A truth table can be generated for 2 bit comparator. Beyond 2 bit size of truth table becomes unwieldy