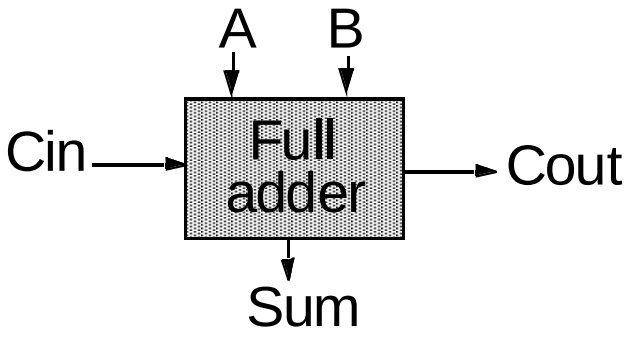
Dwarkadas. J. Sanghvi College of Engineering
Department of Electronics Engineering

Experiment no: 09

Aim: Implementation of Full adder using
A) Static CMOS logic
B) Mirror CMOS logic
C) Transmission Gate Based adder
D) Manchester Carry adder

Apparatus: PC loaded with Microwind software, DSCH3 tool.

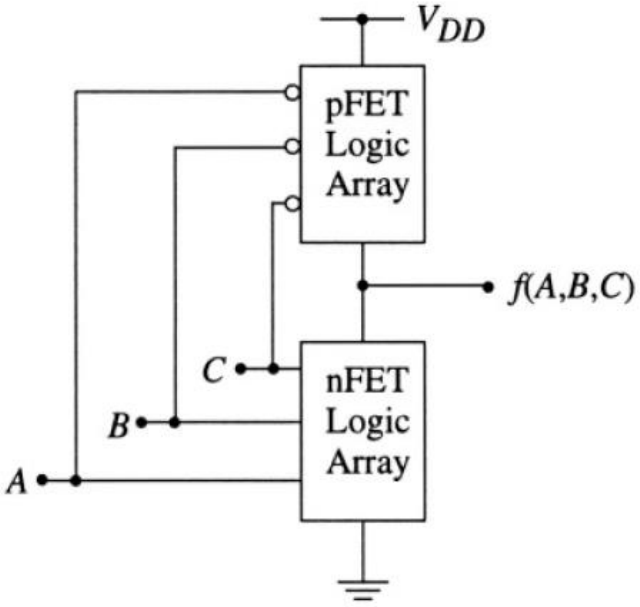
Circuit Diagram: **[a] Block diagram of Full Adder:**



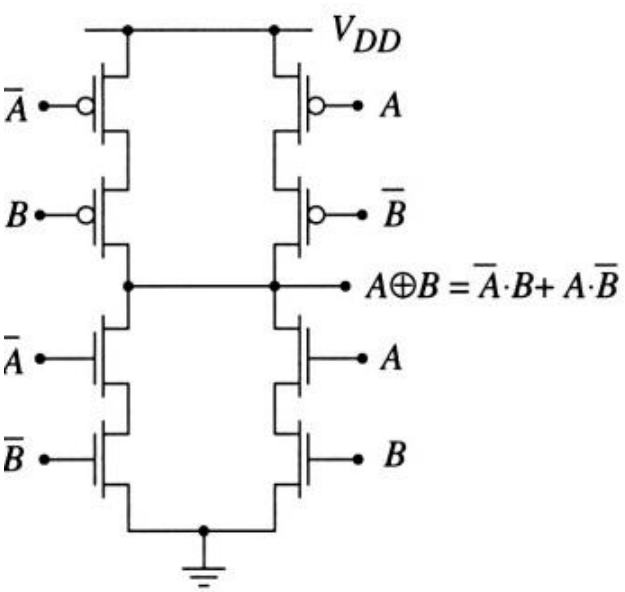
[b] Full Adder : Truth table

A	*B*	C_i	*S*	C_o	*Carry status*
0	0	0	0	0	delete
0	0	1	1	0	delete
0	1	0	1	0	propagate
0	1	1	0	1	propagate
1	0	0	1	0	propagate
1	0	1	0	1	propagate
1	1	0	0	1	generate
1	1	1	1	1	generate

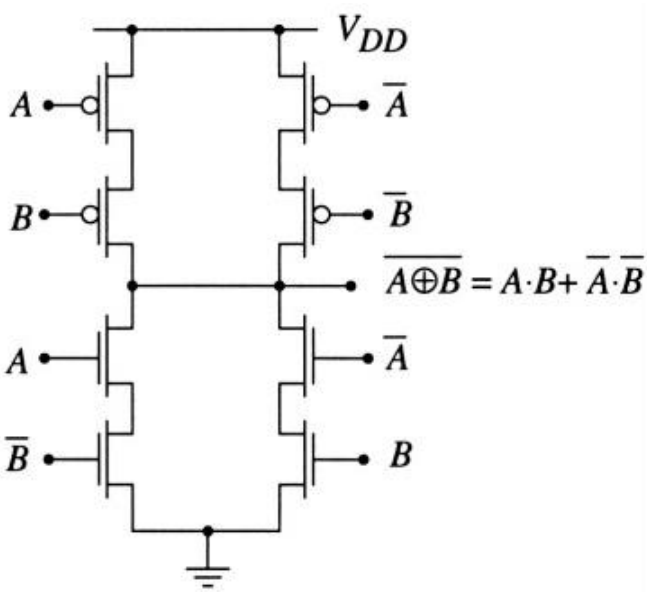
[c] General structure of a static CMOS logic



[d] Mirror Circuits :

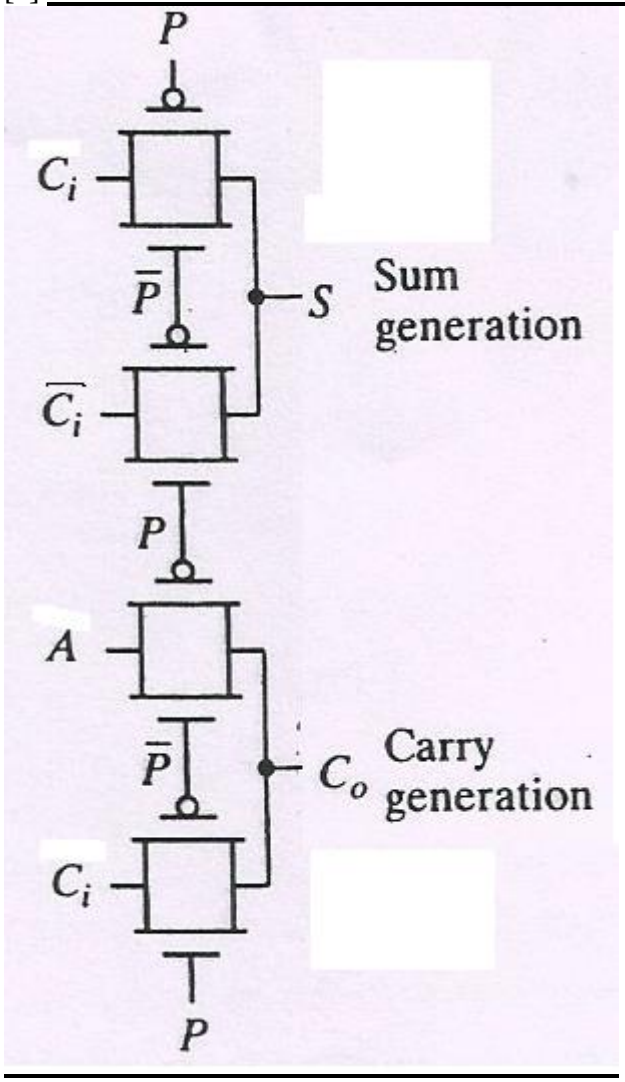


(a) Exclusive-OR circuit

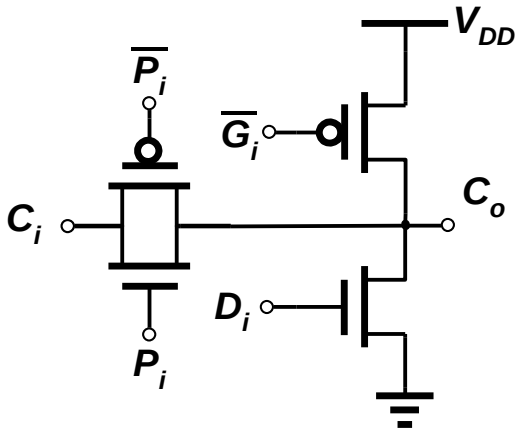


(b) Equivalence circuit

[e] **Transmission Gate / MUX based Full Adder:**



[f] **Manchester Carry adder :**



Theory:

Static CMOS Logic : This logic is constructed using the static CMOS inverter as a basis, where in the nFET and pFET are placed to act as pass transistors. In order to construct a complex logic gate, let us replace the single inverter nFET by an array of nFETs that are connected to operate as a large switch. Similarly, we will substitute an array of pFETs for the single pFET used in the inverter, and view the pFET array as a "giant" switch. In order to insure proper electrical operation, however, we must exercise care so that operation of the nFET array complements the operation of the pFET array. This means that if one array is a closed switch, the other must be open.

The general structure of a complex logic gate can be created by the following steps:

- Provide a complementary pair (an nFET and a pFET with a common gate) for each input.
- Replace the single nFET with an **array** of nFETs that connects the output to ground VSS.
- Replace the single pFET with an **array** of pFETs that connects the output to VDD.
- Design the nFET and pFET switching network so that only one network acts as a closed switch for any given input combination.

This results in the general network shown in Figure [c]. With m >1 inputs, the nFET and pFET arrays are viewed as large "composite" switches, with each array containing m MOSFETs. For a given input combination, only one composite switch can be closed. If the pFET switching array is closed, then the output voltage is giving a logic high (VDD). Conversely, if the nFET array is closed, then the output is a logic 0. Note that, for proper operation, the arrays must be designed so that the two cases where either (i) both switching arrays are closed, or, (ii) both switching arrays are open, cannot occur, since both situations give an undefined output.

Mirror Circuits: In thin the nFET and pFET arrays have the exactly the same structure. These do not use series-parallel logic formation, but have many of the same characteristics. The origin of the mirror circuits can be seen by the XOR and XNOR truth tables. With 2 inputs *A* and *B,* each given gate has 2 outputs that are high and two outputs that are low (0*v*). This implies that we can provide 2 paths from the output to the power supply, and 2 paths from the output to ground with each path consisting of two series-connected FETs. To understand the philosophy, consider a 2 variable gate. The possible input combinations are :

$$A \cdot B,\ \overline{A} \cdot B,\ A \cdot \overline{B},\ \overline{A} \cdot \overline{B}$$

Since the exclusive-OR function has the form

$$A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$$

this means that the combinations A'B and AB' should provide connections from the output to the power supply, while and should connect the output to ground. A circuit constructed with these characteristics is shown in Figure [d]. and constitutes an XOR gate. Similarly, since the equivalence function is given by

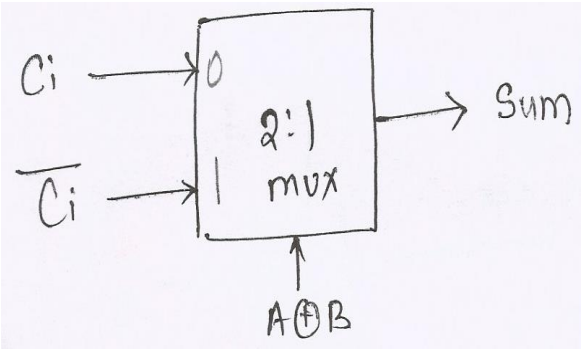
$$\overline{A \oplus B} = A \cdot B + \overline{A} \cdot \overline{B}$$

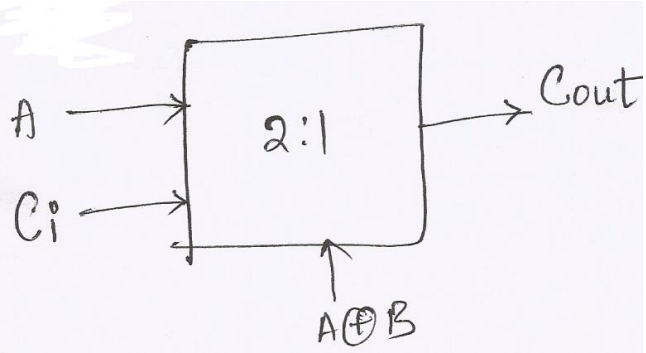
the circuit shown in Figure [d] acts as an XNOR gate. These are called "mirror circuits" since due to the symmetry above and below the output, i.e., if you place a mirror along the output node, then you will see the other half of the gate as the reflection. Obviously these do not have a series parallel structuring. Aside from their departure away from the more general approach to logic formation, these circuits are of interest because they may have shorter switching times and symmetric layout.

Transmission Gate / MUX based Full Adder:

A full adder can be designed to use multiplexers and XORs. While this is impractical in a complementary CMOS implementation, it becomes attractive when the multiplexers and XORs are implemented as transmission gates.

The propagate signal, which is the XOR of inputs *A* and *B*, is used to select the true or complementary value of the input carry as the new sum output. Based on the propagate signal, the output carry is either set to the input carry, or either one of inputs *A* or *B*. One of interesting features of such an adder is that it has similar delays for both sum and carry outputs.





Manchester Carry Chain: This is switch based approach used in design of Full Adders. Here Carry generated is defined in terms of control signals as G (generate), P(propagate) and K/D (kill/delete). Only one control (p, g, k) is active at given time which is implement in switch-logic as shown in Figure [f].

generate	$g_i = a_i \cdot b_i$
propagate	$p_i = a_i \oplus b_i$
carry-kill	$k_i = \overline{a_i + b_i}$

if $g = 1 \rightarrow c_{i+1} = 1$
if $p = 1 \rightarrow c_{i+1} = c_i$
if $k = 1 \rightarrow c_{i+1} = 0$

a_i	b_i	c_i	c_{i+1}	p_i	g_i	k_i
0	0	0	0	0	0	1
0	1	0	0	1	0	0
1	0	0	0	1	0	0
1	1	0	1	0	1	0
0	0	1	0	0	0	1
0	1	1	1	1	0	0
1	0	1	1	1	0	0
1	1	1	1	0	1	0

Procedure:

1. Draw the circuit using DSCH3 tool.
2. Apply clock signal for all combinations of inputs A, B and C_{in}.
3. Connect LED's at observing nodes (Sum, Carry).
4. Run the simulation and observe the output waveforms for all four implementations.

Conclusion: