

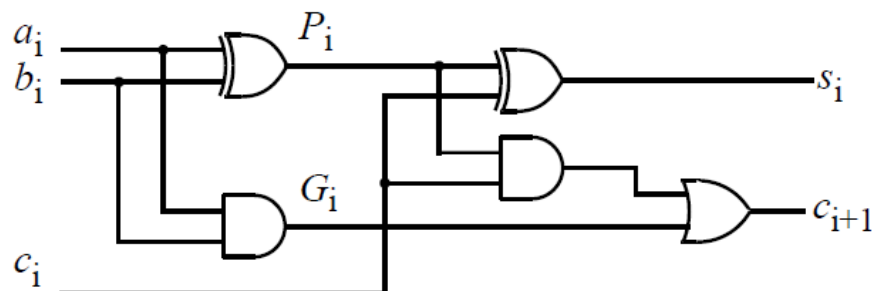
PRACTICAL - 4**Institute of Computer And Technology****B.Tech – CSE(BDA)****Name:- Dwij Vatsal Desai****Sem:- 2****Sub: - CO****Enrollment No.:- 23162121027****Prac:- 4****Date:- 5/3/2024****AIM:** To design Carry look-ahead adder.**THEORY:**

The ripple-carry adder, its limiting factor is the time it takes to propagate the carry. The carry look-ahead adder solves this problem by calculating the carry signals in advance, based on the input signals. The result is a reduced carry propagation time.

To be able to understand how the carry look-ahead adder works, we have to manipulate the Boolean expression dealing with the full adder. The Propagate P and generate G in a full-adder, is given as:

$$P_i = a_i \oplus b_i \quad \text{Carry propagate}$$

$$G_i = a_i b_i \quad \text{Carry generate}$$



The new expressions for the output sum and the carryout are given by:

$$S_i = P_i \oplus C_{i-1}$$

$$C_{i+1} = G_i + P_i C_i$$

These equations show that a carry signal will be generated in two cases:

- 1) if both bits a_i and b_i are 1
- 2) if either a_i or b_i is 1 and the carry-in C_i is 1.

Notice that both propagate and generate signals depend only on the input bits and thus will be valid after two and one gate delay respectively.

If one uses the above expression to calculate the carry signals, one does not need to wait for the carry to ripple through all the previous stages to find its proper value. Let's apply this to a 4-bit adder to make it clear.

Putting $i = 0, 1, 2, 3$;

$$c_1 = G_0 + P_0.c_0$$

$$c_2 = G_1 + P_1.G_0 + P_1.P_0.c_0$$

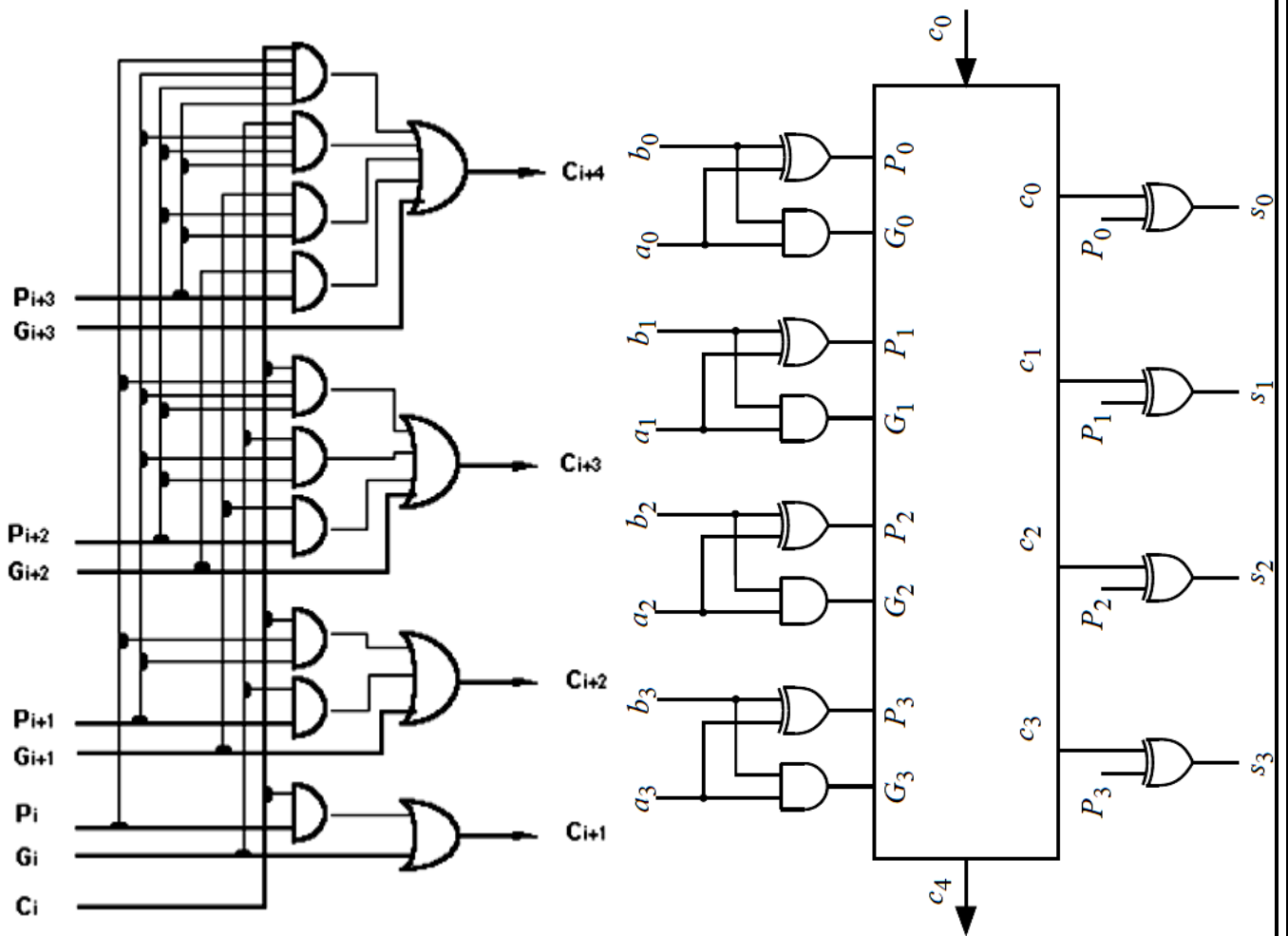
$$c_3 = G_2 + P_2.G_1 + P_2.P_1.G_0 + P_2.P_1.P_0.c_0$$

$$c_4 = G_3 + P_3.G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.G_0 + P_3.P_2.P_1.P_0.c_0$$

These expressions show that C_2 , C_3 and C_4 do not depend on its previous carry-in. Therefore C_4 does not need to wait for C_3 to propagate. As soon as C_0 is computed, C_4 can reach steady state. The same is also true for C_2 and C_3 .

The disadvantage of CLA is that the carry logic block gets very complicated for more than 4-bits. For that reason, CLAs are usually implemented as 4-bit modules and are used in a hierarchical structure to realize adders that have multiples of 4-bits.

-bits.

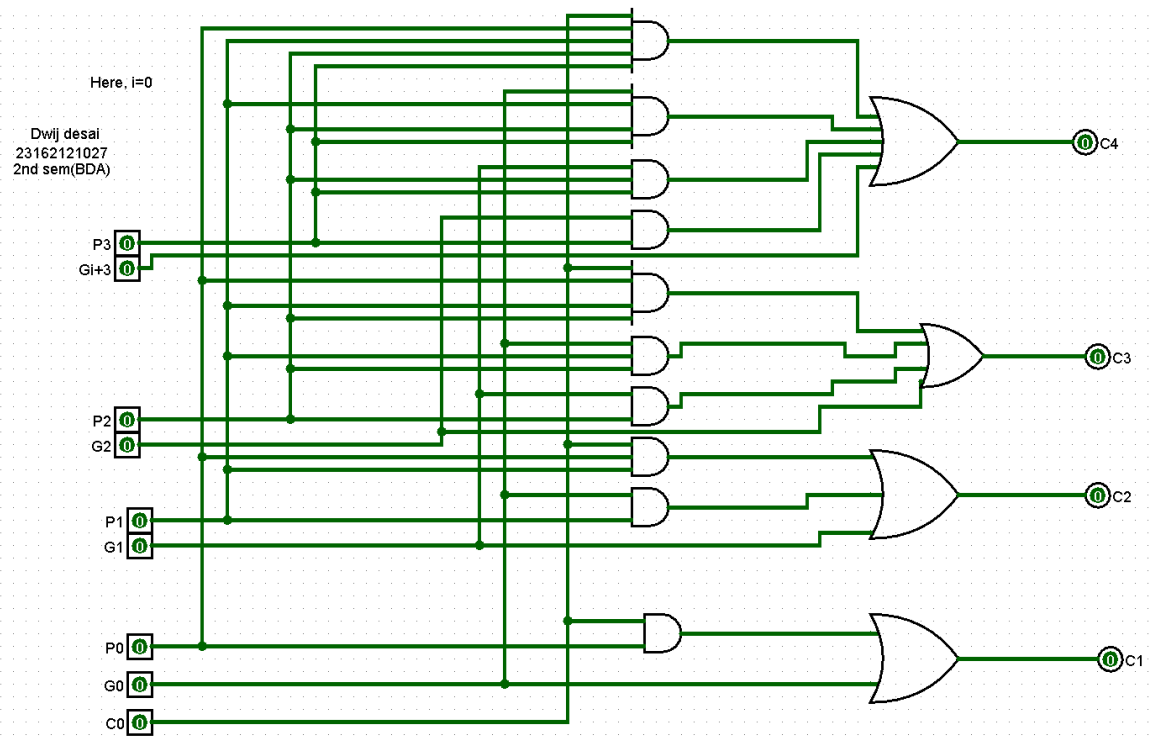


Look ahead carry generator

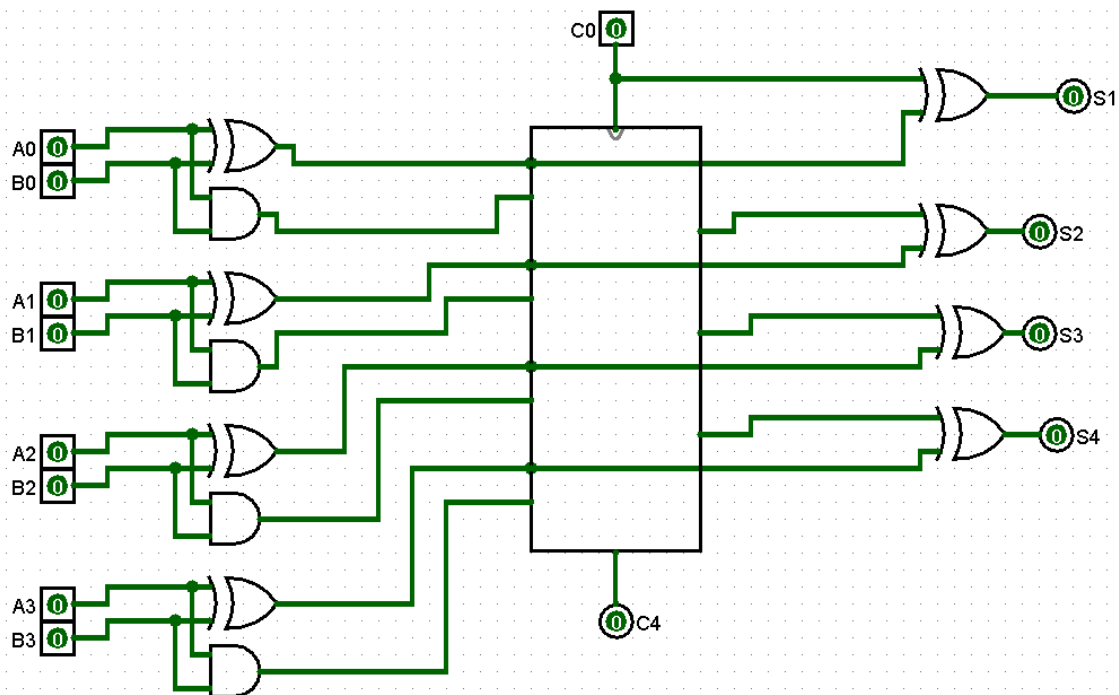
Carry Look Ahead adder

LABWORK: add extra pages

1. Look ahead carry generator



2. Carry Look Ahead adder



Dwij desai 23162121027 2nd sem(BDA)

CONCLUSION:

The carry look-ahead adder improves addition performance by precomputing carry signals, reducing propagation delay. However, its complexity grows significantly for larger bit widths, often implemented in 4-bit modules for efficiency.