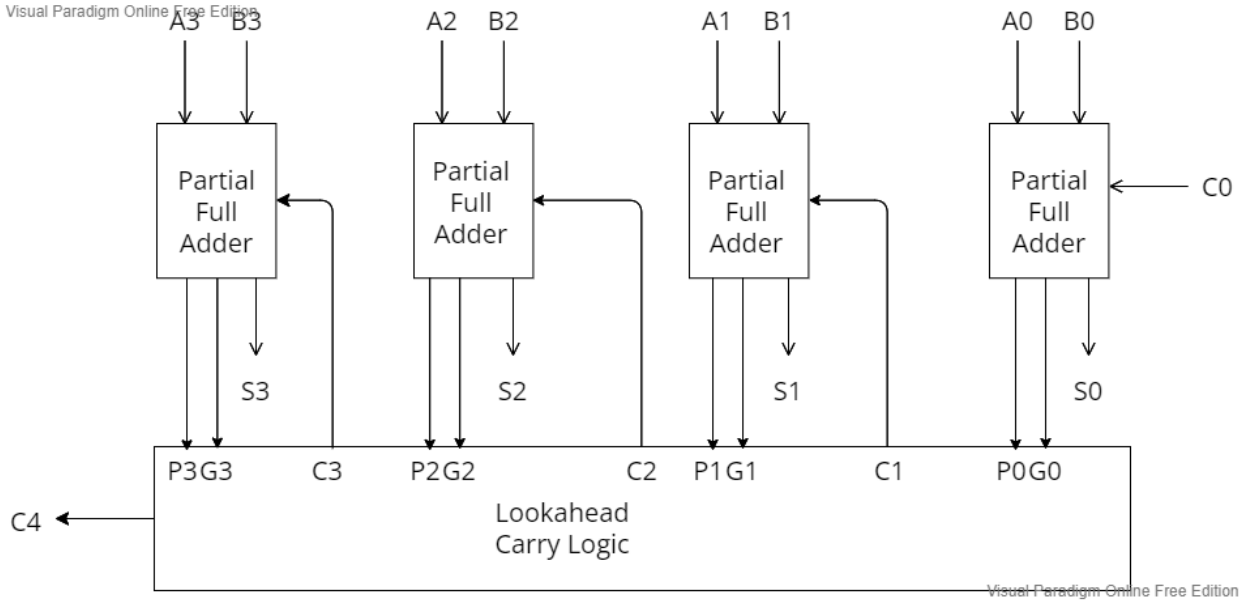
Assignment 2
CS220A : Computer Organization

Shivam Malhotra (190808)
Piyush Kumar (190601)
Shubham Gupta (180748)

1 Lookahead Carry Adder

In standard ripple adders, the carry output of each full adder is given as a carry input to the next higher-order state. Hence, these adders can't produce carry and sum outputs of any state unless a carry input is available for that state. So, for computation to occur, the circuit must wait until the carry bit propagated to all states. This induces carry propagation delay in the circuit. A carry look-ahead adder reduces the propagation delay by introducing more complex hardware, and it generates the carry-in of each full adder simultaneously without causing any delay. We create two signals, P and G, i.e. Carry Propagator and Carry Generator, to reduce the computation time. The carry propagator is propagated to the next level, whereas the carry generator generates the output carry.

The following diagram shows the overall block diagram of a 4-bit look-ahead carry generator:



Carry generate G_i =1 whenever there is a carry C_{i+1} generated. It depends on A_i and B_i inputs. G_i is 1 when both A_i and A_i are 1. Hence, G_i is calculated as $G_i = A_i \cdot B_i$.

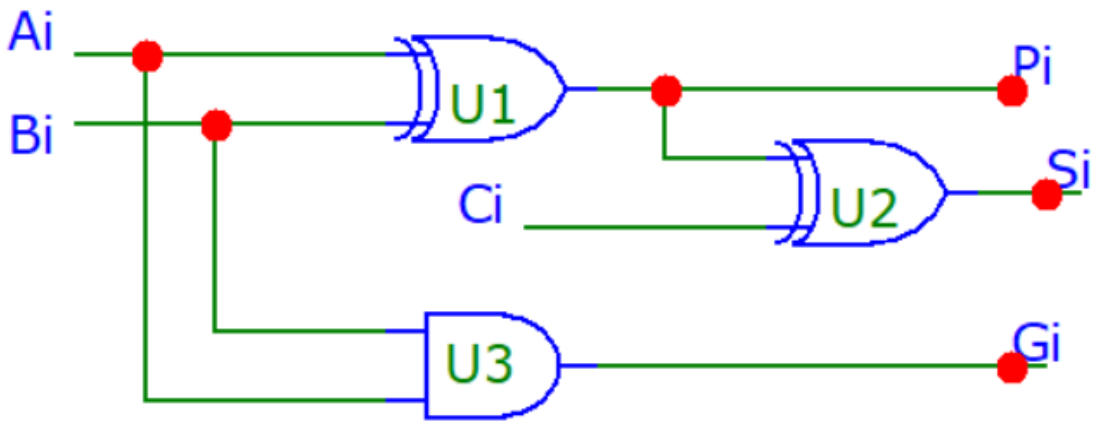
Carry propagated P_i is associated with the propagation of carry from C_i to C_{i+1}. It is calculated as $P_i = A_i \oplus Bi$.

Using the G_i and P_i terms the Sum S_i and Carry C_{i+1} are given as below :

$$S_i = P_i \oplus C_i \tag{1}$$

$$C_{i+1} = C_i \cdot P_i + G_i \tag{2}$$

This gives us the circuit diagram of the partial full adder:



Therefore, the carry bits C_1, C_2, C_3, and C_4 can be calculated as:

$$C_1 = C_0 \cdot P_0 + G_0 \tag{3}$$

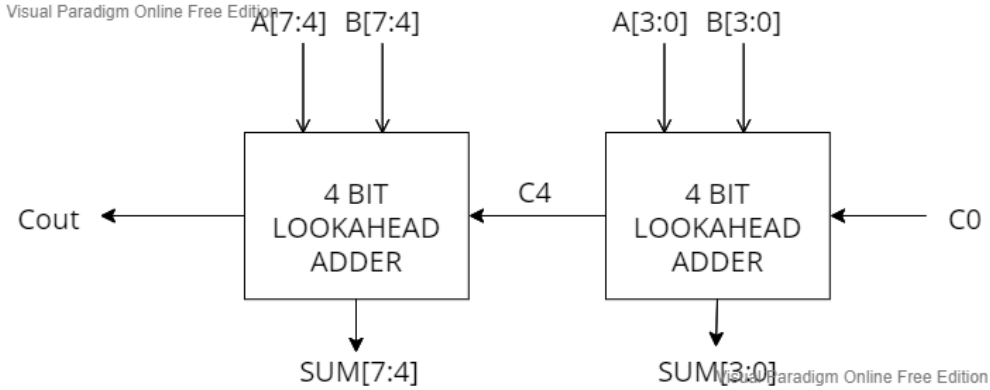
$$C_2 = C_1 \cdot P_1 + G_1 = (C_0 \cdot P_0 + G_0) \cdot P_1 + G_1 \tag{4}$$

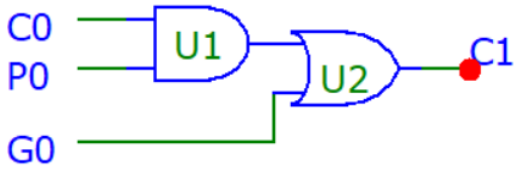
$$C_3 = C_2 \cdot P_2 + G_2 = (C_1 \cdot P_1 + G_1) \cdot P_2 + G_2 \cdot \tag{5}$$

$$C_4 = C_3 \cdot P_3 + G_3 = C0 \cdot P_0 \cdot P_1 \cdot P_2 \cdot P_3 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot G_1 + G_2 \cdot P_3 + G_3 \tag{6}$$

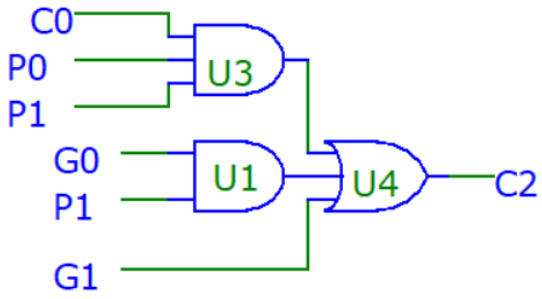
It can be observed from the equations that carry C_{i+1} can be easily extracted from only carry C_0. We can now easily draw the circuit diagram of Lookahead Carry Logic.

Extending this logic to a 8 bit carry ahead adder is easy and is simply done by cascading two 4 bit adders and connecting the Carry Output of first with the Carry Input of second and connect the corresponding 4 bits of both A and B to each adder.

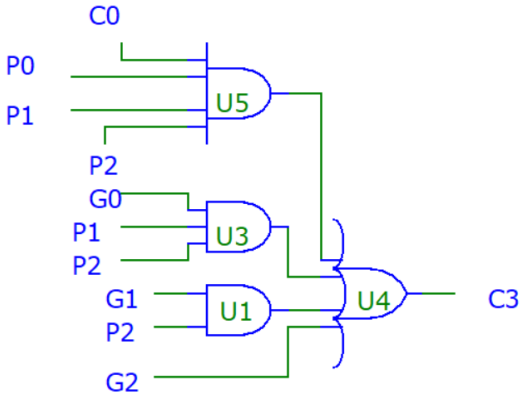




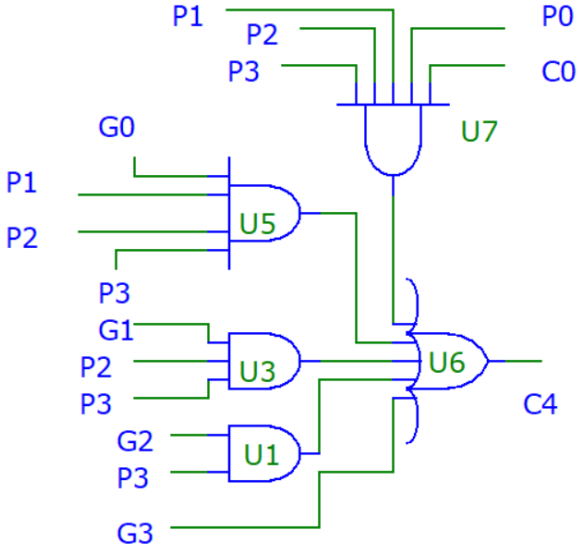
(a) *C1 circuit diagram*



(b) *C2 circuit diagram*



(c) *C3 circuit diagram*



(d) *C4 circuit diagram*

Figure 1: *Circuit Diagrams for C_1, C_2, C_3, C_4*

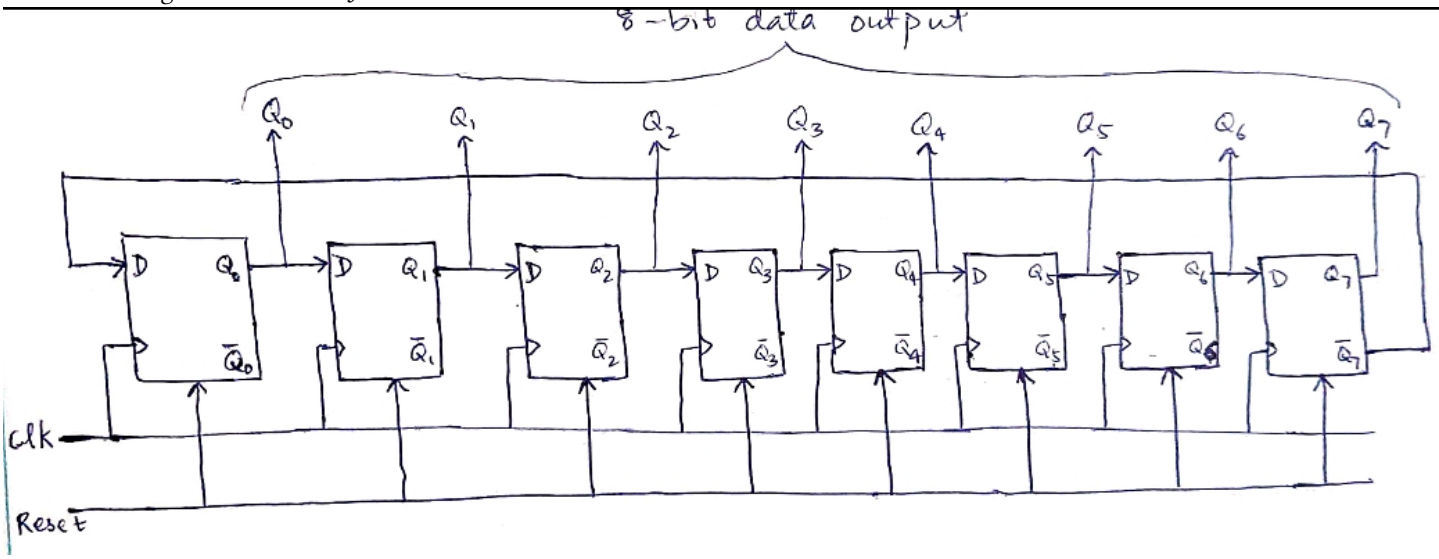
2 Q2 : Solution

Let's first understand what exactly a counter is.
A counter is a special type of sequential circuit which is primarily used to count the pulse. Since it is a sequential circuit hence clock signal is used in its implementation. It is used to store and sometimes display the number of times a certain event has occurred.
Johnson counter is also a type of counter in which a series of flip-flops are connected in feedback manner. In this, the complement output of the last flip-flop is fed back as an input to the first flip-flop. Apart from the last flip-flop, the output of the current flip-flop is fed as an input to the next flip-flop. Apart from the clock signal, we also use reset signal which resets the output to a certain chosen value.

Some of the other properties of Johnson counter is as below :

- If it is an n-bit Johnson counter then it uses n flip-flops. So in our case, it is an 8-bit Johnson counter hence it requires 8 flip-flops.
- If the number of bits is n then it counts 2n different states. So in our case, it counts 16 different states.
- Since there can be 2^n different states in an n-bit number so the number of unused states is (2^n - 2n) as only 2n different states are used in Johnson counter.

The circuit diagram of the 8-bit Johnson counter is as below :



In the above diagram, Q_o correspond to the most significant bit and Q_7 correspond to the least significant bit.

The truth table corresponding to the 8-bit Johnson counter is on the next page :

Clock Pulse No.	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0
5	1	1	1	1	1	0	0	0
6	1	1	1	1	1	1	0	0
7	1	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1	1
9	0	1	1	1	1	1	1	1
10	0	0	1	1	1	1	1	1
11	0	0	0	1	1	1	1	1
12	0	0	0	0	1	1	1	1
13	0	0	0	0	0	1	1	1
14	0	0	0	0	0	0	1	1
15	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0

In the above table, we can see that the outputs corresponding to clock pulse number 0 and the outputs corresponding to clock pulse number 16 are same. This means that we have 16 different states in our 8-bit Johnson counter.