Assignment–2

200856-Sandeep Kumar Bijarinia
200562-Rishwan Reddy
200530-Kuldeep Singh Chouhan

Question 1

A carry-look-ahead adder improves speed by reducing the amount of time required to determine carry bits. The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger-value bits of the adder, in the ripple carry adder the next computation of the sum bit has to wait for the previous bit to complete which increases the duration of computation.
All 1 bit adders calculate their results and simultaneously the next carry is calculated
Analyzing the below truth table we can deduce the formula of the carry of bit i^th^

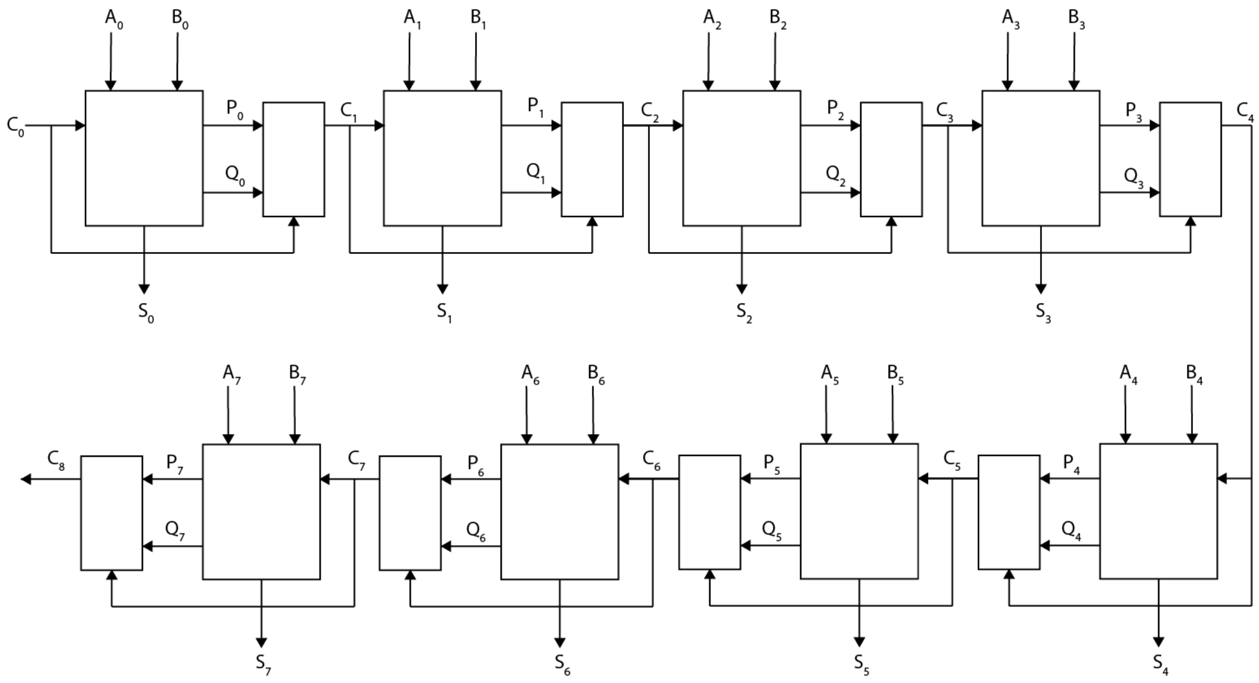
$C_i = A_i \wedge B_i + ((A_i \oplus B_i) \wedge C_{i-1})$

Here, A_i denotes ith bit of A

Using this table we can find the carry of all bits simultaneously

A	B	C_{i-1}	C_i
0	0	x	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	x	1

We, will use $P_i = A_i \wedge B_i$ and $Q_i = A_i \oplus B_i$
and $C_i = P_i + Q_i.C_{i-1}$
We, will recursively call above formula to get thye value of Carry after each stage.
The implementation can be understood by the help of an abstract circuit diagram given below.


A_0
B_0
C_0
P_0
Q_0
S_0
C_1
A_1
B_1
P_1
Q_1
S_1
C_2
A_2
B_2
P_2
Q_2
S_2
C_3
A_3
B_3
P_3
Q_3
S_3
C_4
A_7
B_7
C_8
P_7
Q_7
S_7
C_7
A_6
B_6
P_6
Q_6
S_6
C_6
A_5
B_5
P_5
Q_5
S_5
C_5
A_4
B_4
P_4
Q_4
S_4

Question 2

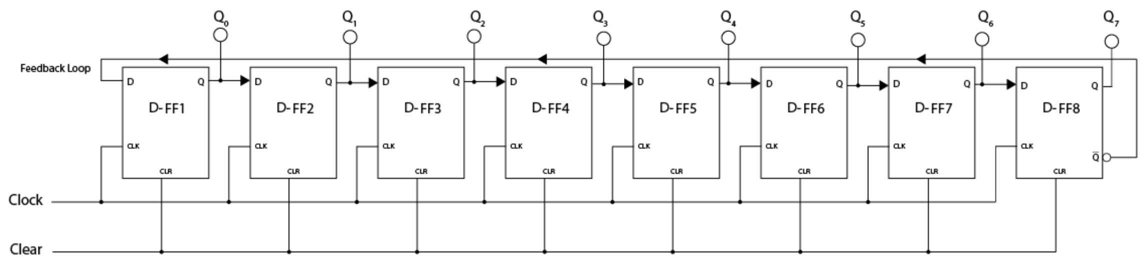
8-bitJohnson Counter, is a counter which uses D-Flip Flops (DFFs) one after another where output from previous DFF is used as input for next DFF, the only change being at the last DFF where negation of last output is fed as input for the first DFF, and the loop continues.

So, there would be a pattern as 00000000, 10000000, 11000000, 11100000, 11110000, etc.

TRUTH TABLE

State	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	1	1
16	0	0	0	0	0	0	0	1

This can be illustrated using DFFs as shown below.


Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Feedback Loop
D-FF1
D-FF2
D-FF3
D-FF4
D-FF5
D-FF6
D-FF7
D-FF8
D
Q
CLK
CLR
Q̄
Clock
Clear