Aayush Shah
19BCE245
22 September 2020

Practical 5

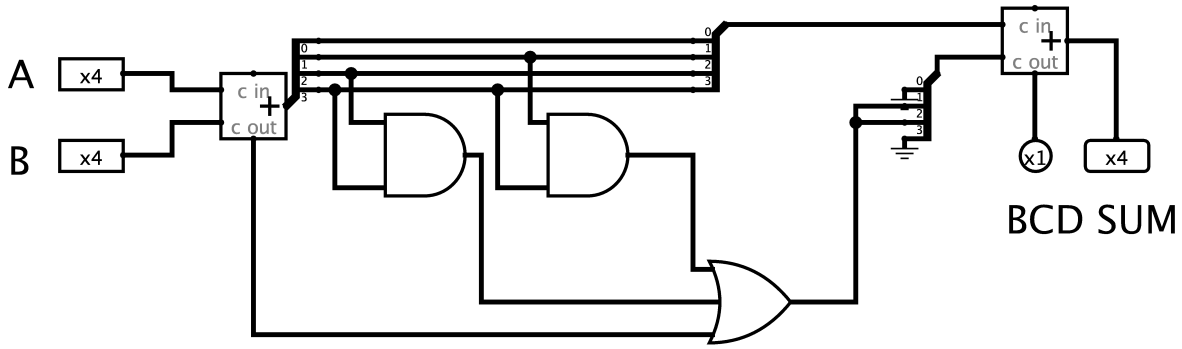
Practical 5 A&B

Design and Implementation of BCD and excess 3 parallel adder using two binary adders .

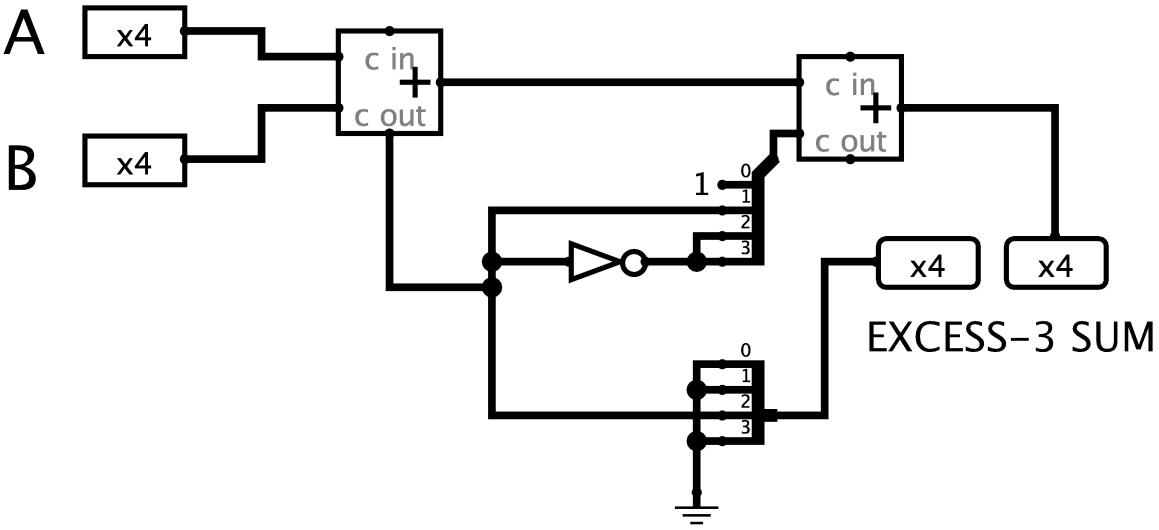
OUTLINE :

- Snapshots of logisim
- Introduction and Conclusion [Design and implementation]
- Truth-Tables

BINARY SUM TO BCD SUM



EXCESS-3 PARALLEL ADDER



Introduction 5A

The key purpose of this practice was the insertion of two binary numbers and give the output of their BCD sum. The numbers are from range 1 to 9 [binary : 0000 to 1001].
The BCD sum will be same as binary sum if the sum is less than or equal to 9 otherwise it will differ by 6.

Conclusion 5A

From this practical , We learnt the usage of splitter and 4 bit adder. This experience gave us an insight into how to think critically according to our criteria for setting up a circuit. We learned to work out the conditions that would fulfil our output according to the inputs, we set up 3 conditions, which even one of them was valid than we would add 6 to the output of initial sums of input. This way, we conducted our practical 5(A) of obtaining BCD sum from binary sum along with their Design and Implementation.

Introduction 5B

The main aim of this practical to add two 4 bit excess-3 parallel adder using two binary adders. The numbers can be from 0 to 9 [Binary : 0000 to 1001]. We have to use only two binary adders and give the output as the sum of given two 4 bit binary numbers in excess 3.

Conclusion 5B

In this practical , We want the output of this excess 3 sum in 8 bit. So first we added two binary 4-bit numbers through adder. Then we divided the problem in two parts :
1.When carry is 0
2. When carry is 1
In First case, When Carry is 0 then the input binary number's sum is definitely from range 0 to 9. Then we simply have to subtract 3 from the sum, which can be done by using 1's complement method. Here we used another binary adder and it's result will give us addition of this two excess-3 numbers.
In Second case, When Carry is 1 then we have 8-bit output. Which can be obtained by : First 4 bits will sum and others four bits be 0100. This will give us the addition of this two excess -3 numbers.The first 4 bits showed the sum of output of first 4 bit adders with 3. And last 4 bits (i.e. leftmost 4 bits) showed 0100 value.

Truth Table : Binary sum to BCD sum

Decimal	Binary Sum					BCD Sum				
	C'	A'	B'	C'	D'	C	E	F	G	H
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

Truth Table : Excess 3 parallel adder

Original (A+B)	Excess 3 of A + Excess 3 of B	Excess 6 of (A+B)					Excess 3 of(A+B)							
		E	A	B	C	D	E	F	G	H	A	B	C	D
0	6	0	0	1	1	0	0	0	0	0	0	0	1	1
1	7	0	0	1	1	1	0	0	0	0	0	1	0	0
2	8	0	1	0	0	0	0	0	0	0	0	1	0	1
3	9	0	1	0	0	1	0	0	0	0	0	1	1	0
4	10	0	1	0	1	0	0	0	0	0	0	1	1	1
5	11	0	1	0	1	1	0	0	0	0	1	0	0	0
6	12	0	1	1	0	0	0	0	0	0	1	0	0	1
7	13	0	1	1	0	1	0	0	0	0	1	0	1	0
8	14	0	1	1	1	0	0	0	0	0	1	0	1	1
9	15	0	1	1	1	1	0	0	0	0	1	1	0	0
10	16	1	0	0	0	0	0	1	0	0	0	0	1	1
11	17	1	0	0	0	1	0	1	0	0	0	1	0	0
12	18	1	0	0	1	0	0	1	0	0	0	1	0	1
13	19	1	0	0	1	1	0	1	0	0	0	1	1	0
14	20	1	0	1	0	0	0	1	0	0	0	1	1	1