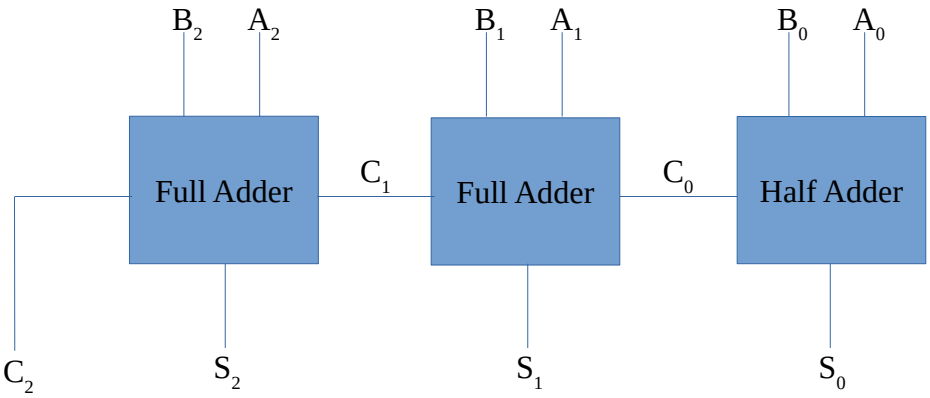
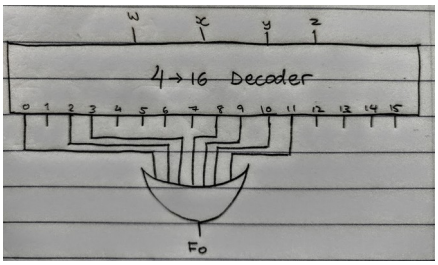
Digital Systems – HW6

1. Design a Ripple Adder for two numbers (A and B) consisting of three bits each.

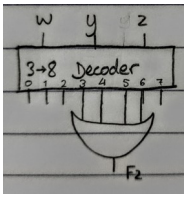


2. Design the following functions using a decoder and external logic gates:

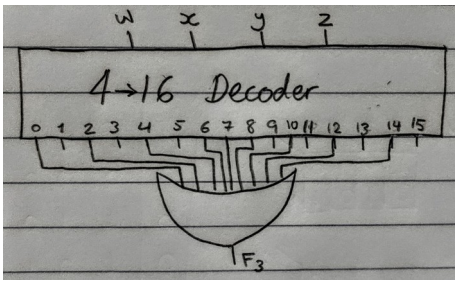
   a) $F_1=wx'+x'y+x'z'$



   b) $F_2=w'yz+wy'+wz'$

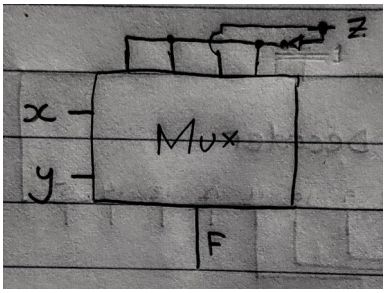


   c) F3=z'+w'xy

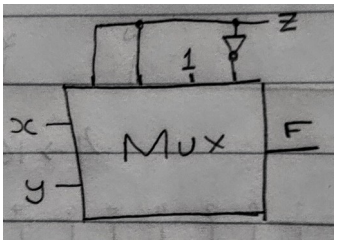


3. Realize the following functions using multiplexers (as efficiently as possible)

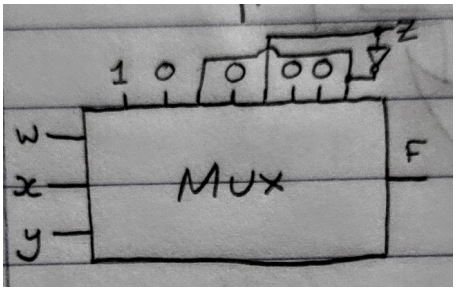
a) F(x,y,z)= Σ(0,2,5,6)



b) F(x,y,z)= Σ(1,3,4,5,6)



c) F(w,x,y,z)= Σ(0,1,4,8,9,14)

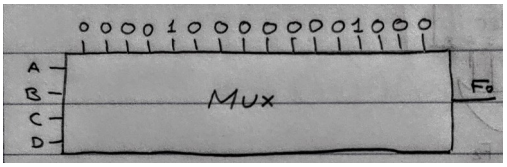


4. You are given a system which receives a 4 digit binary number at its input. The system then looks at the ratio between the number 300 and the input. If the number 300 is exactly divisible by the input – the output is equal to the ratio. Otherwise the output is 0. (The input 0000 may be considered illegal and ignored). You may use fixed (constant) inputs (of zeros and ones).

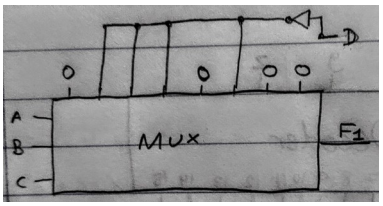
   You do not need to design all 8 outputs just the last three least significant bits

n	A	B	C	D	300/n	F2	F1	F0
0	0	0	0	0	Φ	Φ	Φ	Φ
1	0	0	0	1	300	1	0	0
2	0	0	1	0	150	1	1	0
3	0	0	1	1	100	1	0	0
4	0	1	0	0	75	0	1	1
5	0	1	0	1	60	1	0	0
6	0	1	1	0	50	0	1	0
7	0	1	1	1	-	0	0	0
8	1	0	0	0	-	0	0	0
9	1	0	0	1	-	0	0	0
10	1	0	1	0	30	1	1	0
11	1	0	1	1	-	0	0	0
12	1	1	0	0	25	0	0	1
13	1	1	0	1	-	0	0	0
14	1	1	1	0	-	0	0	0
15	1	1	1	1	20	1	0	0

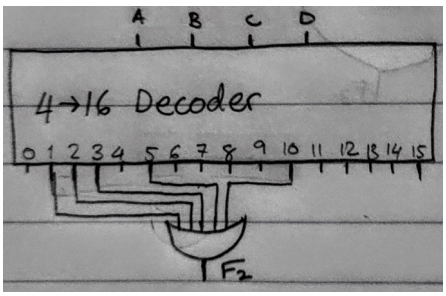
a) Design the LSB (least significant bit) using a 16->1 Mux



b) Design the next bit (lefthand side of LSB) using an 8->1 Mux.



c) Design the next bit using a 4->16 decoder.



5. Using a 3->8 decoder, design a full adder

a	b	cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1