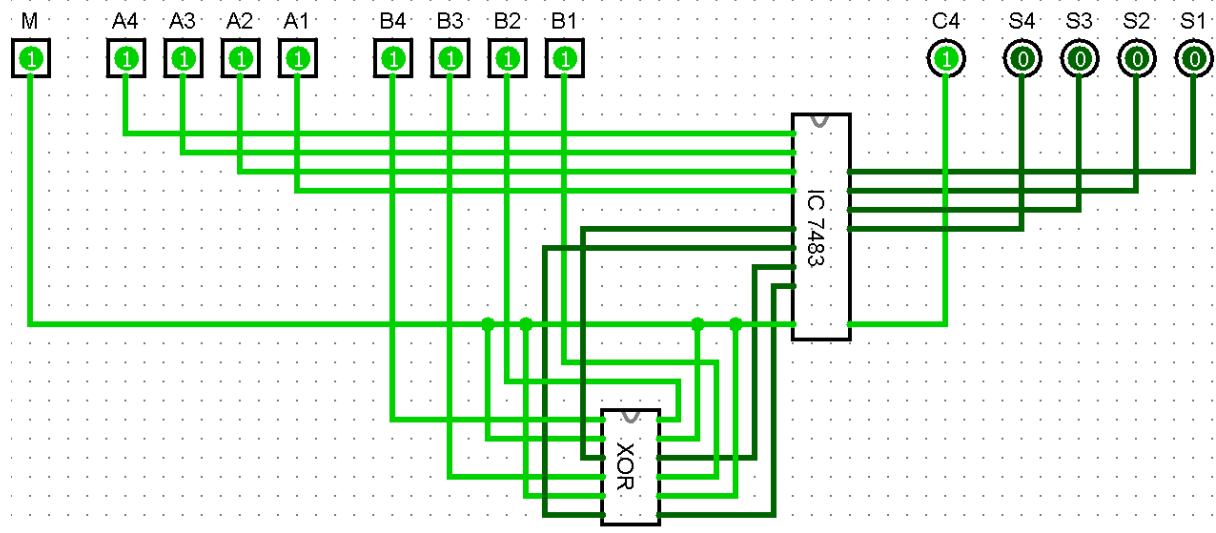


Md. Ishtiaq Ahamed Fahim  
2012518642  
Lab-5

### C Binary Adder-Subtractor:

Logisum:

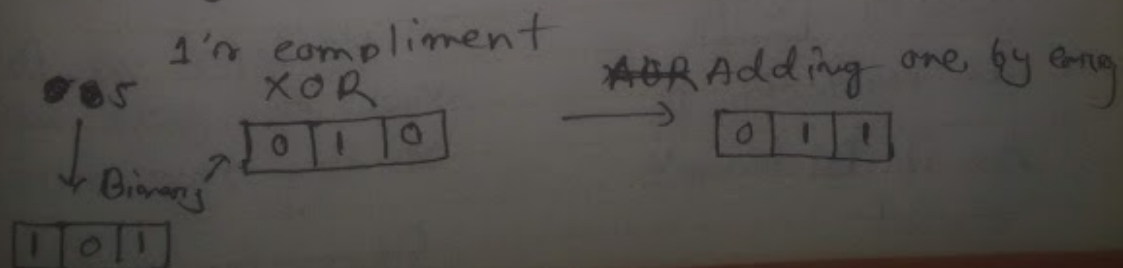


Proof and Discussion:

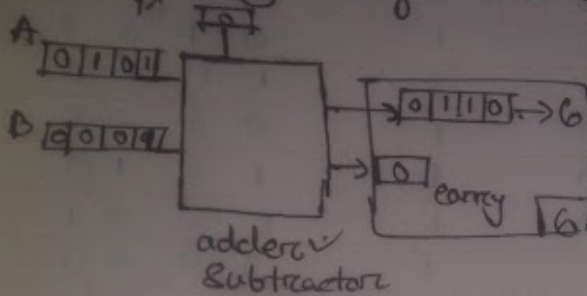
			8	4	2	1				
			0	0	0	0				
operation	M	A	B	C4	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>		
7+5	0	0111	0101	0	1	1	0	0		
9+6	0	0100	0110	0	1	0	1	0		
9+11	0	1001	1011	1	0	1	0	0		
15+15	0	1111	1111	1	1	1	1	0		
7-5	1	0111	0101	1	0	0	1	0		
4-6	1	0100	0110	0	1	1	1	0		
11-2	1	1011	0010	1	1	0	0	1		
15-15	1	1111	1111	1	0	0	0	0		

M is Responsible for creating ~~an~~ 1's complement and cin which also input by M we get added one carry as input which makes 1's complement to 2's.

So at end M is Responsible to creating 2's complement just by plugging 1 on M



Let's see 3 picture And we add every thing in last picture.



Nothing Interesting  
Happens because  
B doesn't change

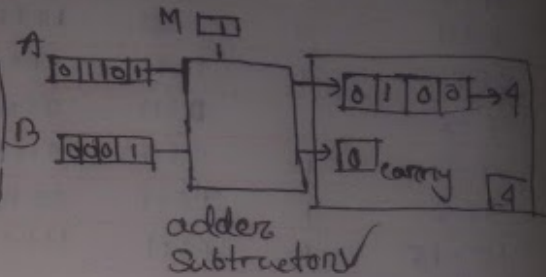
Because

$B \oplus M$		$B \oplus M$	
B	M		
0	0	0	✓ M=0
0	1	1	
1	0	1	✓ M=0
1	1	0	

on  $M=0$  There  
is No change in

$B \oplus M$  ✓

It reflects B  
As if B doesn't  
exist



Something Interesting  
Happens because  
B changes

Because

$B \oplus M$		$B \oplus M$	
B	M		
0	0	0	
0	1	1	✓ M=1
1	0	1	
1	1	0	✓ M=1

on  $M=1$  There is  
change in  $B \oplus M$  ✓

It Reflects on B  
and A gets added  
on B by the following

Step in ③ pto

③ OK let's think now  
tell you want to add 1 to a number

You have 5 pen  
your friend have 5 pen

and you want to add them but  
first you want to add another pen

So Answer is

You  $\rightarrow A \rightarrow 5 \rightarrow +1 \rightarrow 11$  pen  
friend  $\rightarrow B \rightarrow 5 \rightarrow$

It doesn't matter who you add  
1 with

You  $\rightarrow A \rightarrow 6$   
friend  $\rightarrow B \rightarrow 5$

You  $\rightarrow A \rightarrow 5$   
friend  $\rightarrow B \rightarrow 6$

Both  
are  
Same

So for Subtracting operation we put 1 in M and This

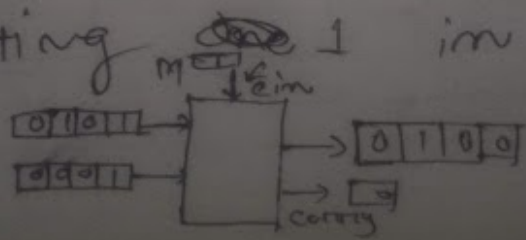
Adds 1 in both of its ~~adding~~ <sup>addition</sup>

but why?

because XOR creates 1's complement of a number if M is ~~(1)~~ (1)

$B \oplus M$	$B'$ 1's complement
$0 \oplus 1 \rightarrow$	1
$1 \oplus 1 \rightarrow$	0

But To add a negative number we need to add 1 in ones complement which is done by putting ~~1~~ 1 in cin



Date: / / Page:

Now let's see  $A = 5 - 1 \rightarrow$  in the circuit

A	B	
$5 \rightarrow 0101$	$1 \rightarrow 0001$	
	$-1 \rightarrow 1110$	1's complement because of
	$-1 \rightarrow 1111$	$[B \oplus M] [M=1]$
		2's complement because of
		$cin \rightarrow 1 [M=1]$

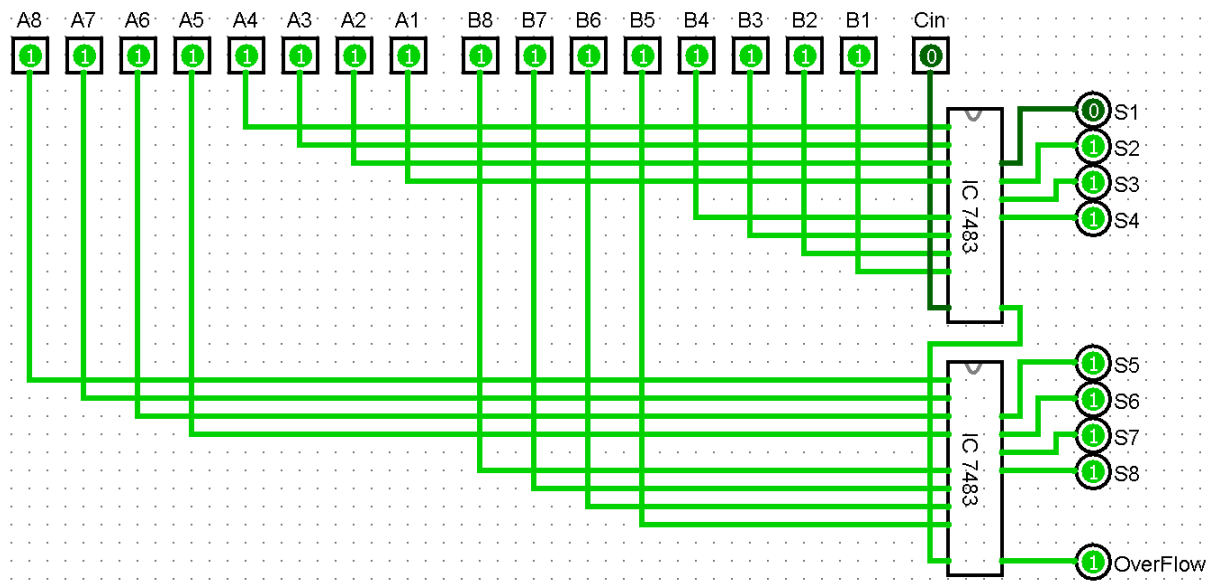
Adding 5 and 2's complement of -1

$$\begin{array}{r}
 0101 \\
 1111 \\
 \hline
 \cancel{1}0100 \rightarrow 4
 \end{array}$$

And this is the answer

D 8-Bit-Ripple-Through-Carry Adder:

Logisum:



Proof and Discussion:



Date:                      Page:                       
 Title:                     

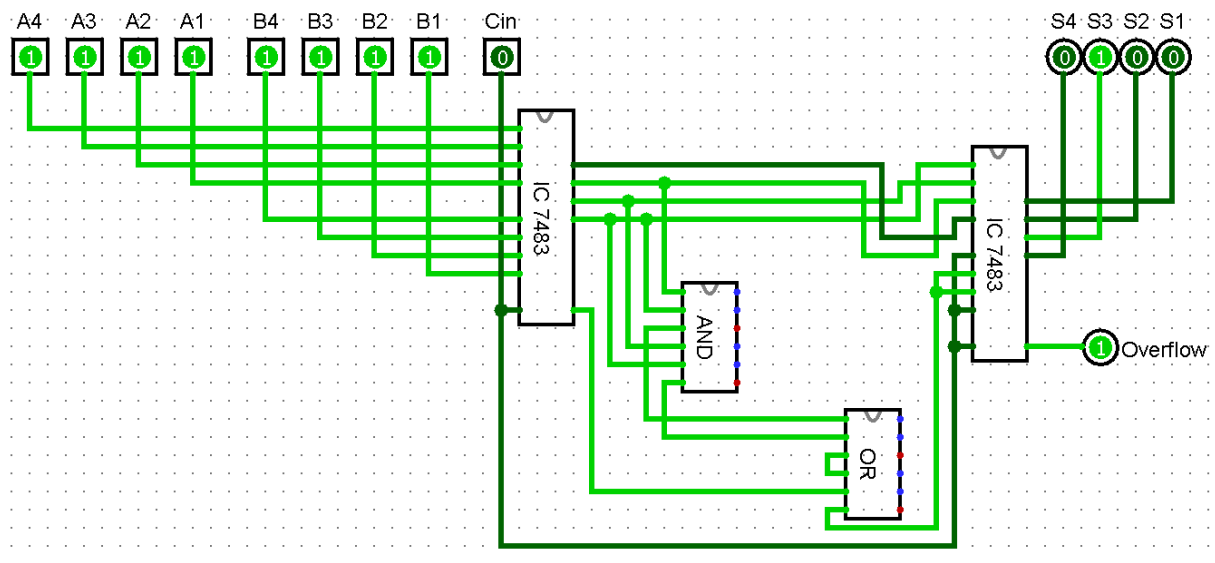
operation	A	B	Over Flowing	Sum
7 + 5	0000	0111	0	0000 0101
18 + 19	0001	0010	0	0001 0011
72 + 83	0100	1000	0	0100 1001
129 + 255	1000	0001	1	1001 0000

Ripple adder takes  $(n-1)$  clock cycle  
 if  $n$  is the bit number  
 to start with with each clock  
 cycle carry is passed through  
 1 bit adder. This can Ripple  
 through whole or many bit possible

E Binary BCD sum:

Logisum:





**E1:**

Proof and Discussion:

Binary Sum					Bed Sum				
K	Z8	Z4	Z2	Z1	C	S8	S4	S2	S1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	0	1	0	1
0	0	1	1	0	0	0	1	1	0
0	0	1	1	1	0	0	1	1	1
0	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0	1
0	1	0	1	0	1	0	0	0	0
0	1	0	1	1	1	0	0	0	1
0	1	1	0	0	1	0	0	1	0
0	1	1	0	1	1	0	0	1	1
0	1	1	1	0	1	0	1	0	0
0	1	1	1	1	1	0	1	0	1
1	0	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1	1	1
1	0	0	1	0	1	1	0	0	0
1	0	0	1	1	1	1	0	0	1

E2:

Proof and Discussion:

	8	4	2	1		
	0	0	0	0		
Operation	A	B	overflow	carry	Sum	
0+0	1001	0000	0		1001	
0+1	1001	0001	1		0000	
0+2	1001	0010	1		<del>0001</del>	
0+3	1001	0011	1		0010	
0+4	1001	0100	1		0011	
0+5	1001	0101	1		0100	
0+6	1001	0110	1		0101	
0+7	1001	0111	1		0110	
0+8	1001	1000	1		0111	
0+9	1001	1001	1		1000	

To understand How BED works  
let's play a game.

convert 80 minute in hh:mm  
format; the naive approach would  
be.

$$\frac{80}{60} = 1.333 \rightarrow 01: \square\square$$

because 60 min in 1h

take the floating unit only

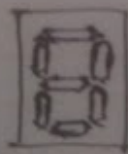
$$.333 \times 60 = 20 \rightarrow 1:20$$

$$\begin{array}{r} 80 \\ + 40 \\ \hline 120 \end{array}$$

Now let's put a ~~semicolon~~ colon after  
1  $\rightarrow$  Making it 1:20

there's the answer right

BED <sup>Address</sup> also works in same  
way. As think of a number  
dial for one digit like this



It can have numbers  
from 0-9

But to represent 9 we need  
4 bit ~~a~~ ~~that~~ binary number  
which has max of 15 ~~and~~  
needing 6 bit.

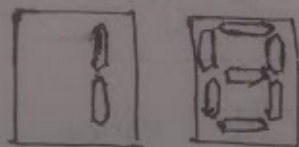
So using old rule

lets represent 12

12

$$\begin{array}{r} 12 \\ + 6 \\ \hline 18 \end{array}$$

and lets put a operator  
as its impossible to put 2 number  
in one digit ~~as~~ led



To construct the circuit for  
6 step ~~run~~ up after 2 we  
don't require  $S_1$  in the  
input so we take  $S_2 S_3 S_4$   
and carry.