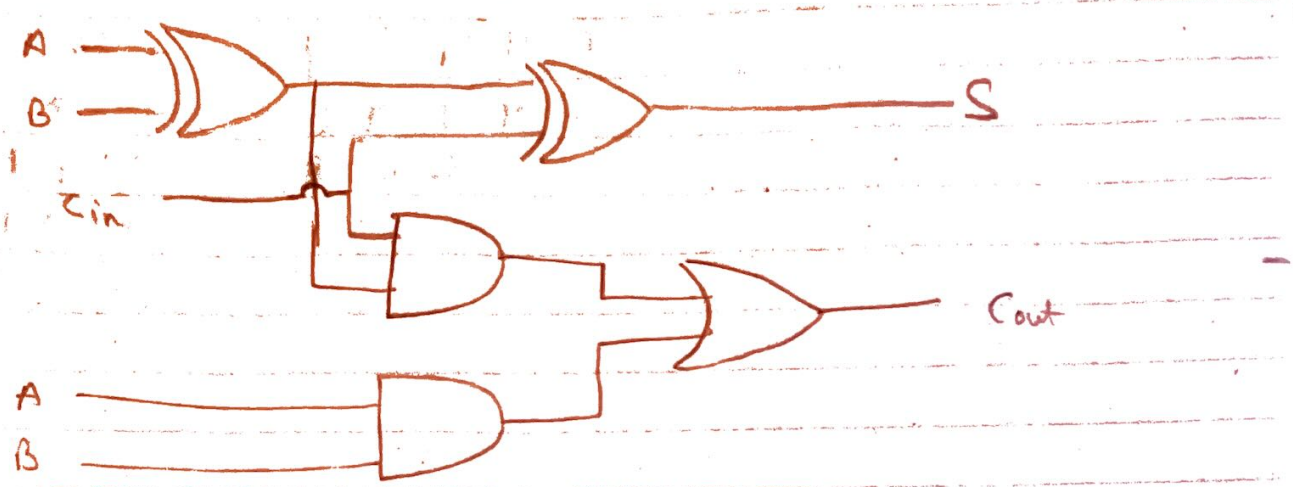
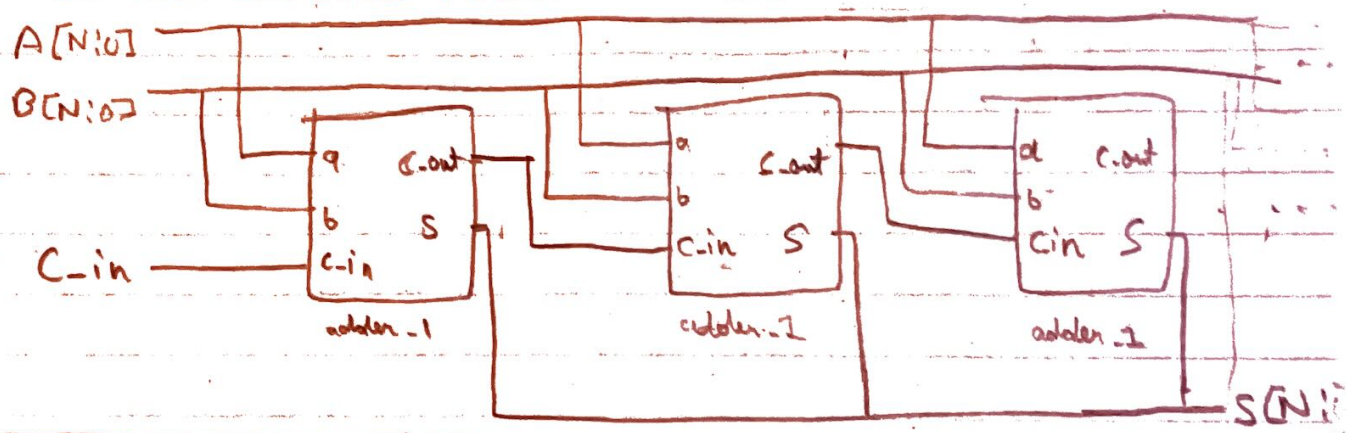
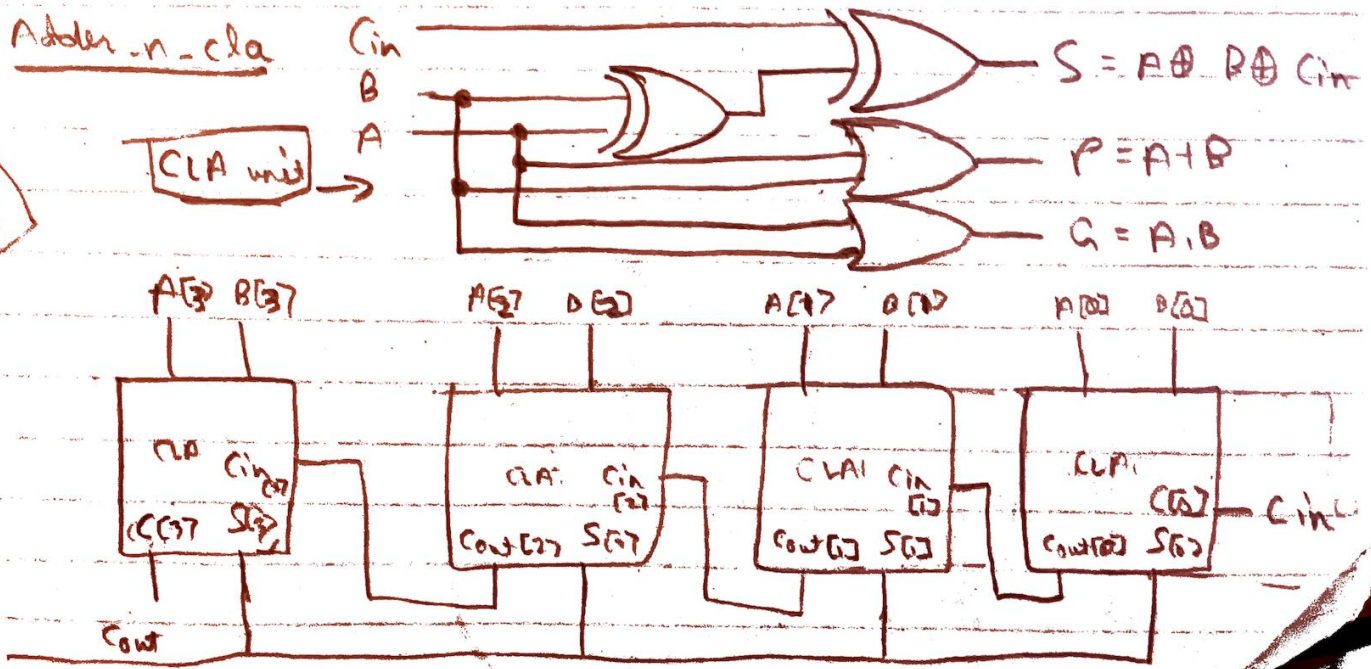
Adder_1



Adder_n (RCA)



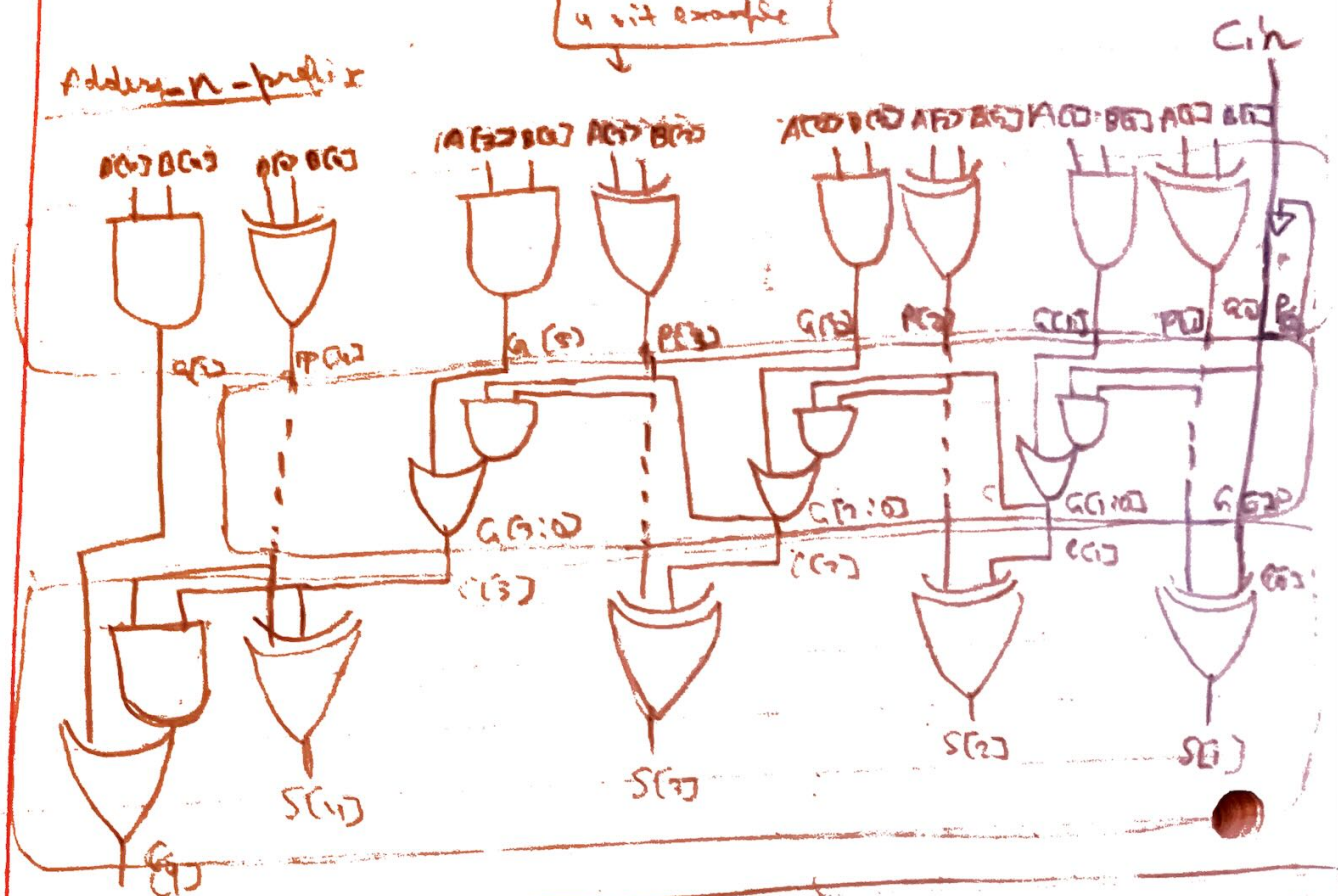
Adder_n_cla



4-bit example unit

4 bit example

Adder_N_prefix



Compare_eq

decoder 1_2

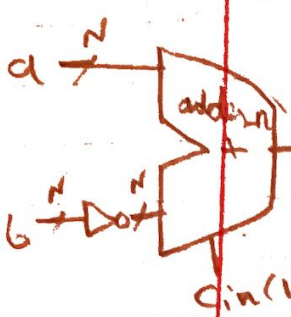
en	in	out[1]	out[0]
0	x	0	0
1	0	0	1
1	1	1	0

out[0] = en.$\overline{in}$

out[1] = en.in

Compare_lt

A - B = A + $\overline{B}$ + 1



decoder 2_4

en	in[1]	in[0]	out[3]	out[2]	out[1]	out[0]
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

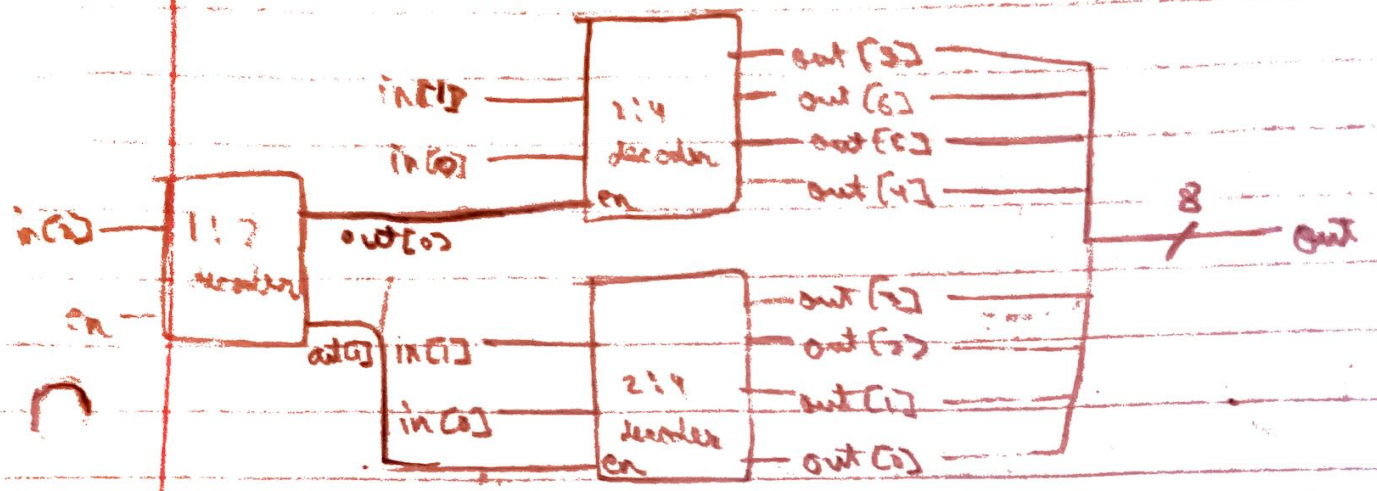
out[3] = en . in[1] . in[0]

out[2] = en . in[1] . $\overline{in[0]}$

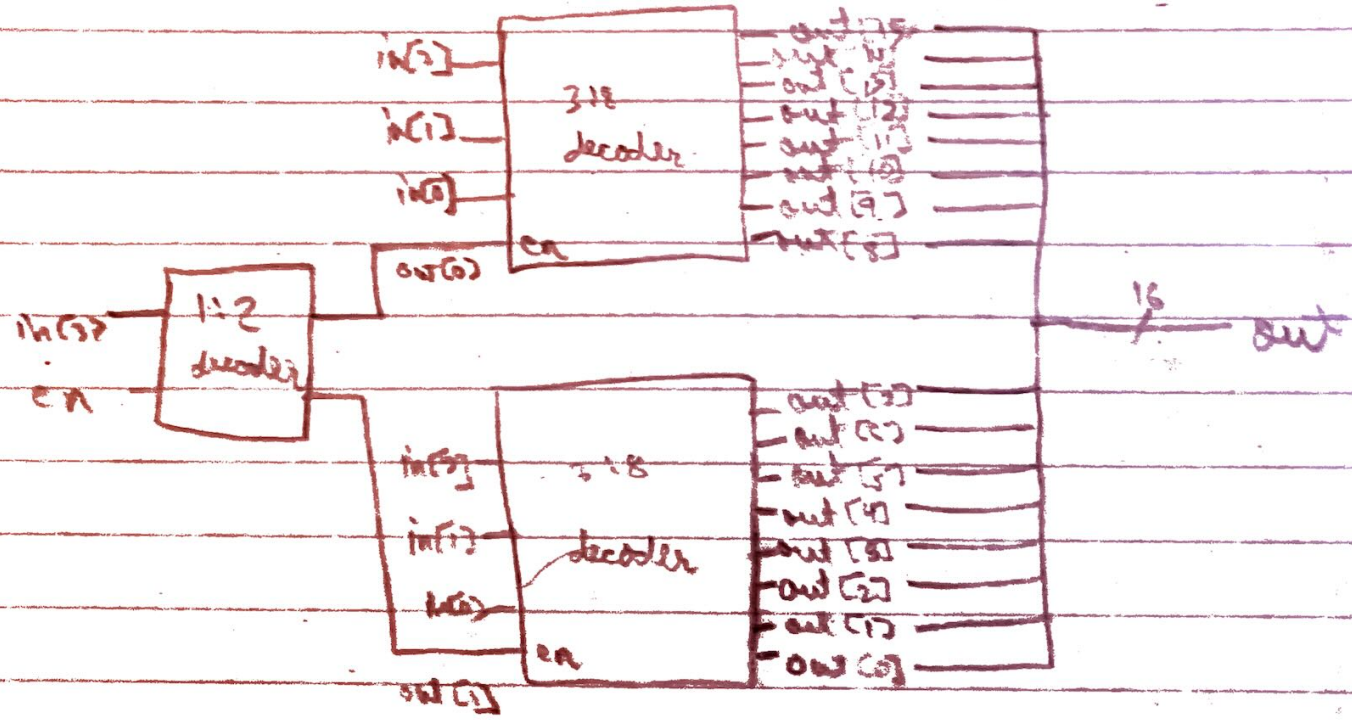
out[1] = en . $\overline{in[1]}$. in[0]

out[0] = en . $\overline{in[1]}$. $\overline{in[0]}$

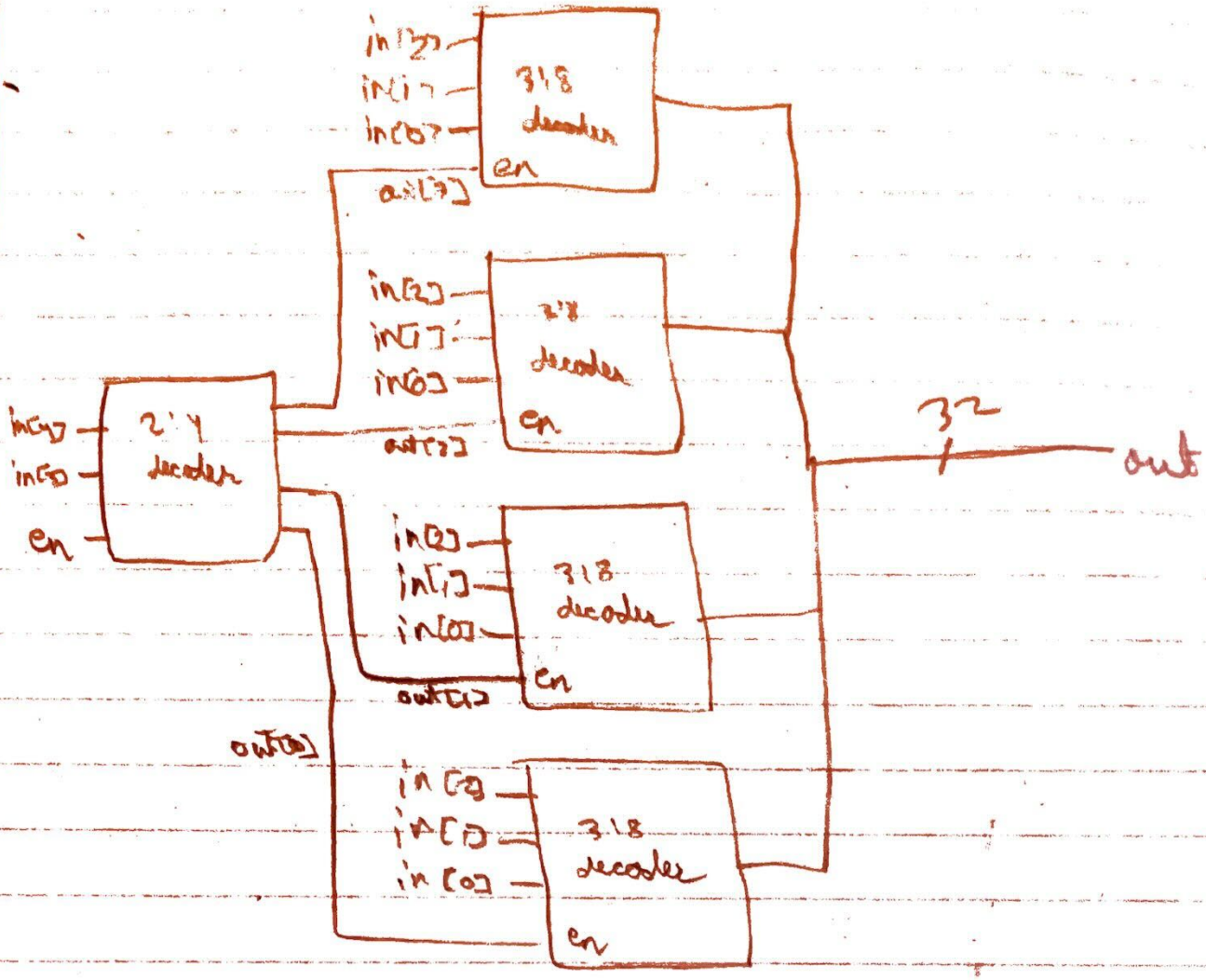
decoder_3 to 8



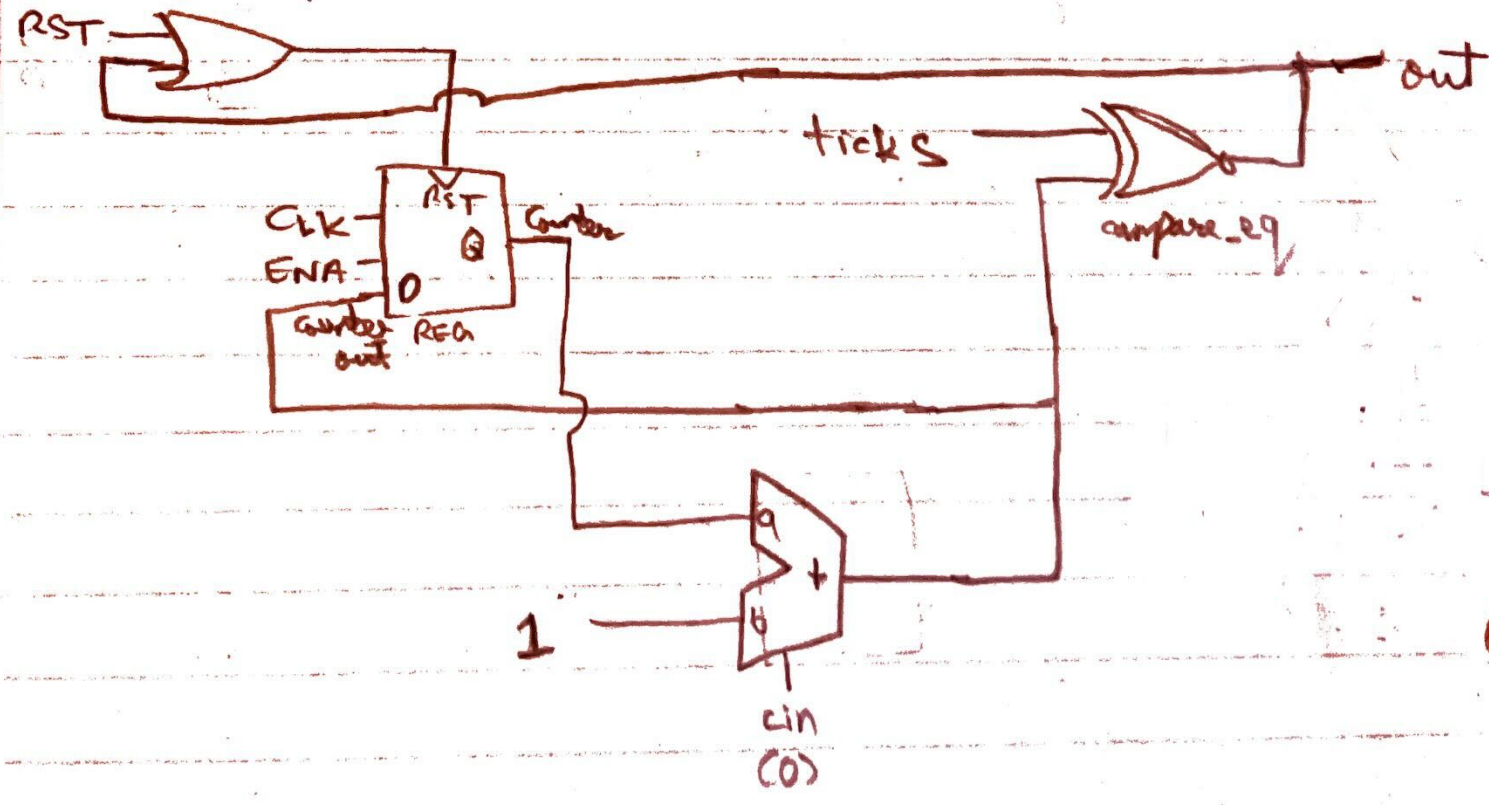
decoder_4 to 16



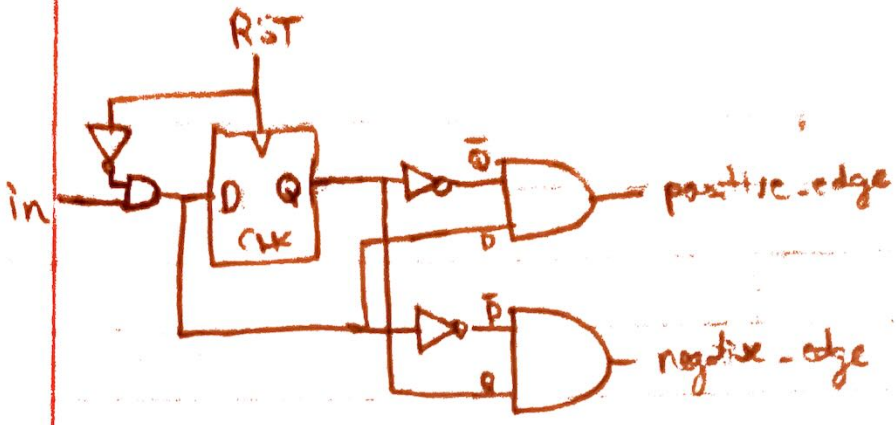
decoder_5 to 32



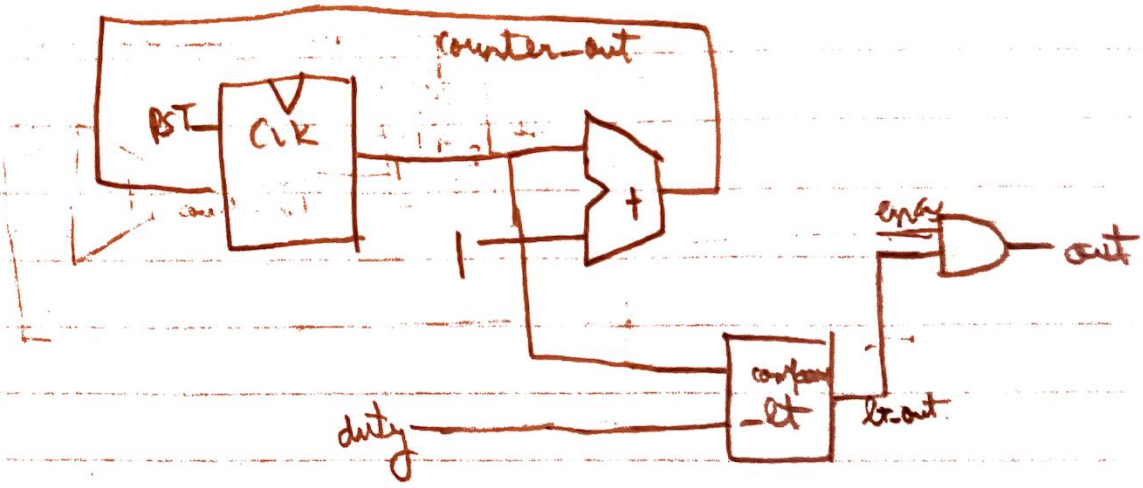
Pulse Generator



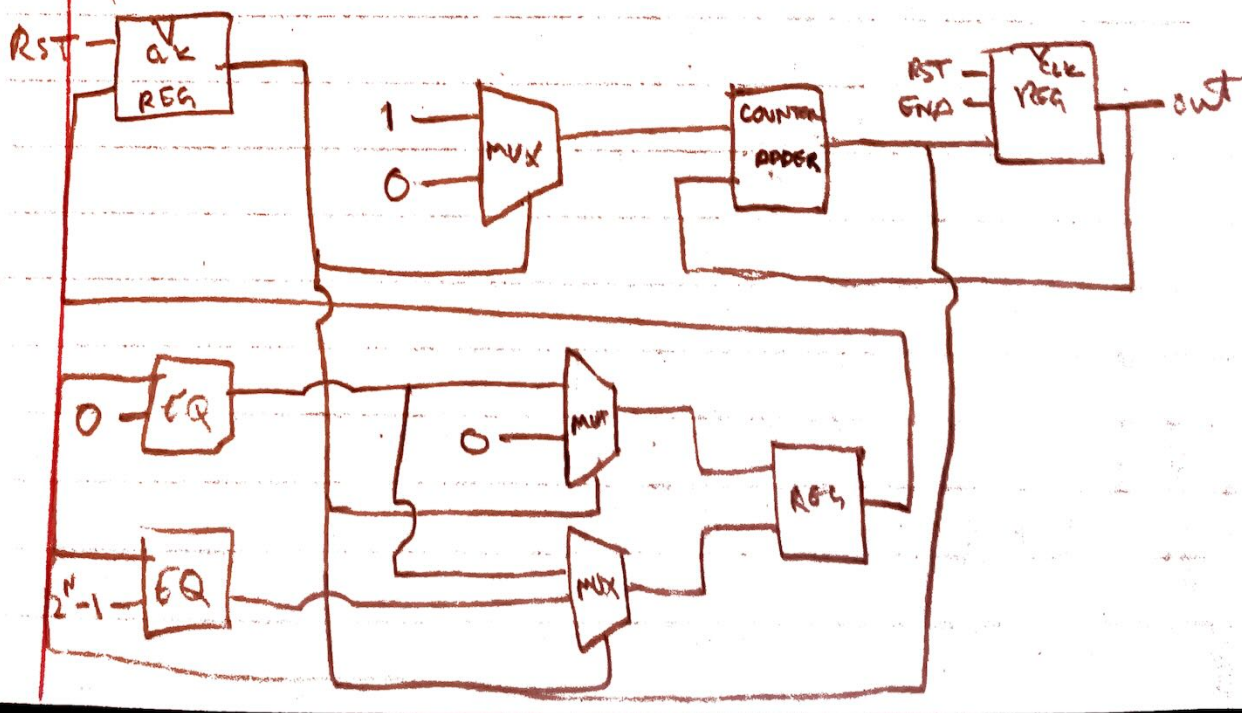
EDGE DETECTOR



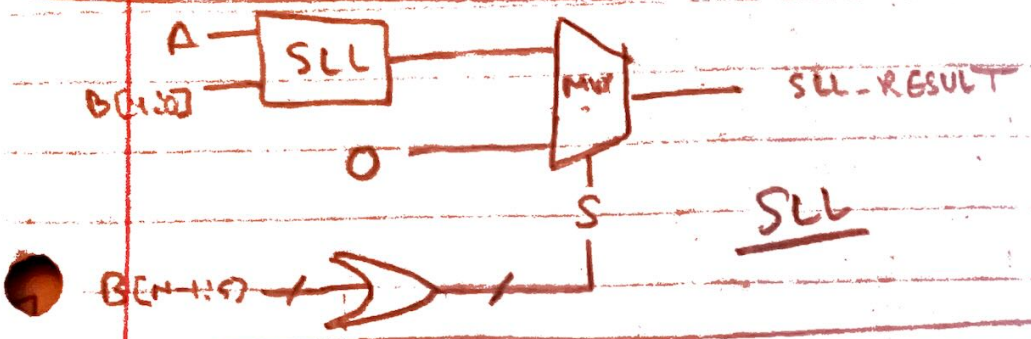
PWM

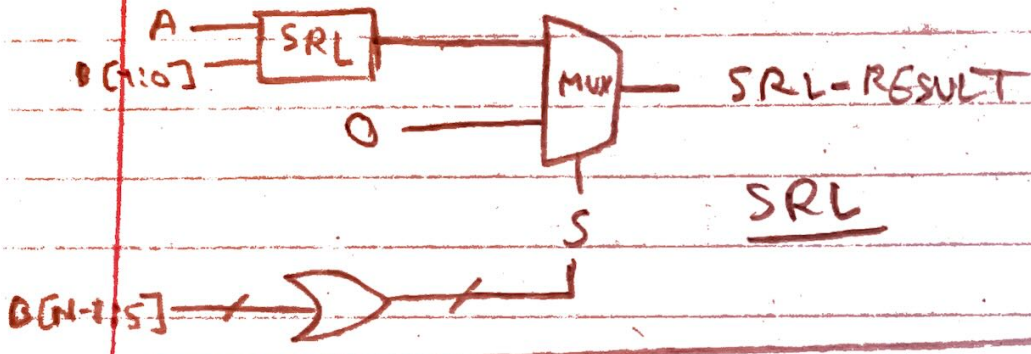


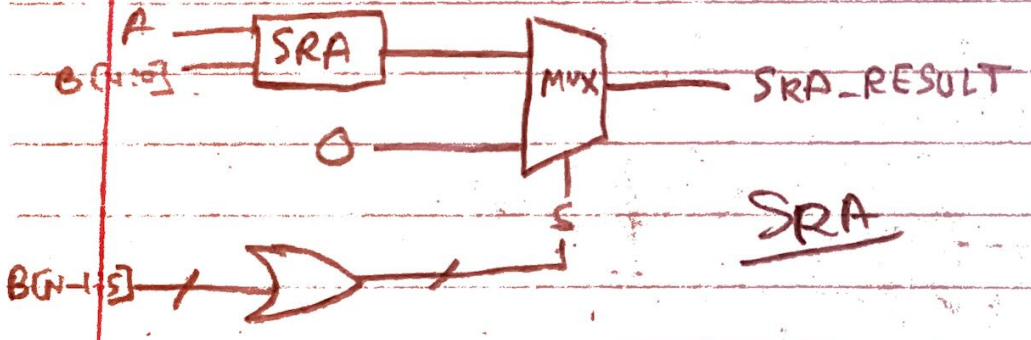
TRIANGLE_GEN



ALU

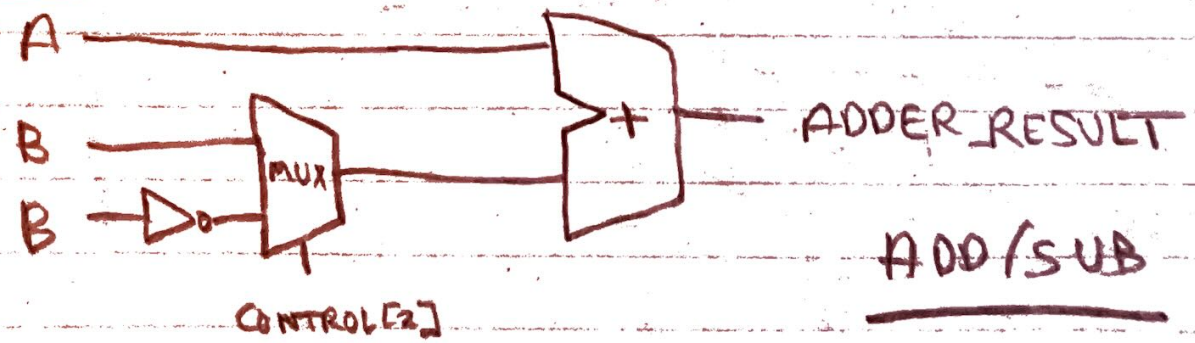






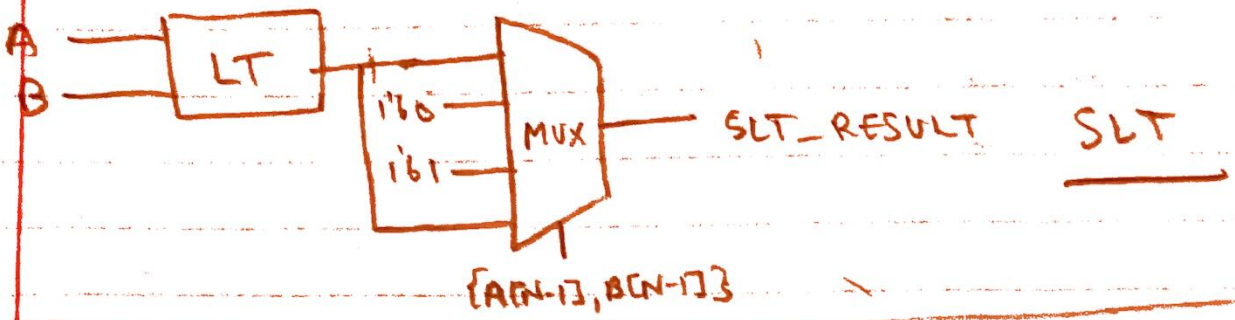
CONTROL [3:0] {

AND = 0001
OR = 0010
XOR = 0011
SLL = 0101
SRL = 0110
SRA = 0111
ADD = 1000
SUB = 1100
SLT = 1101
SLTU = 1111 }

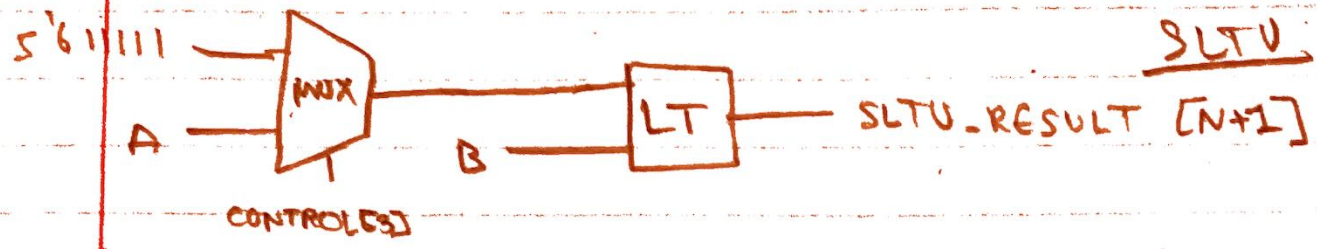


ADD/SUB

ALU

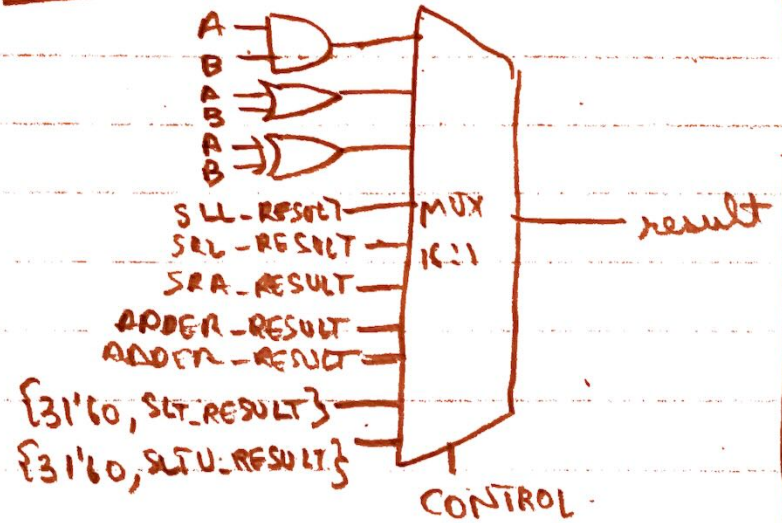


SLT



SLTU

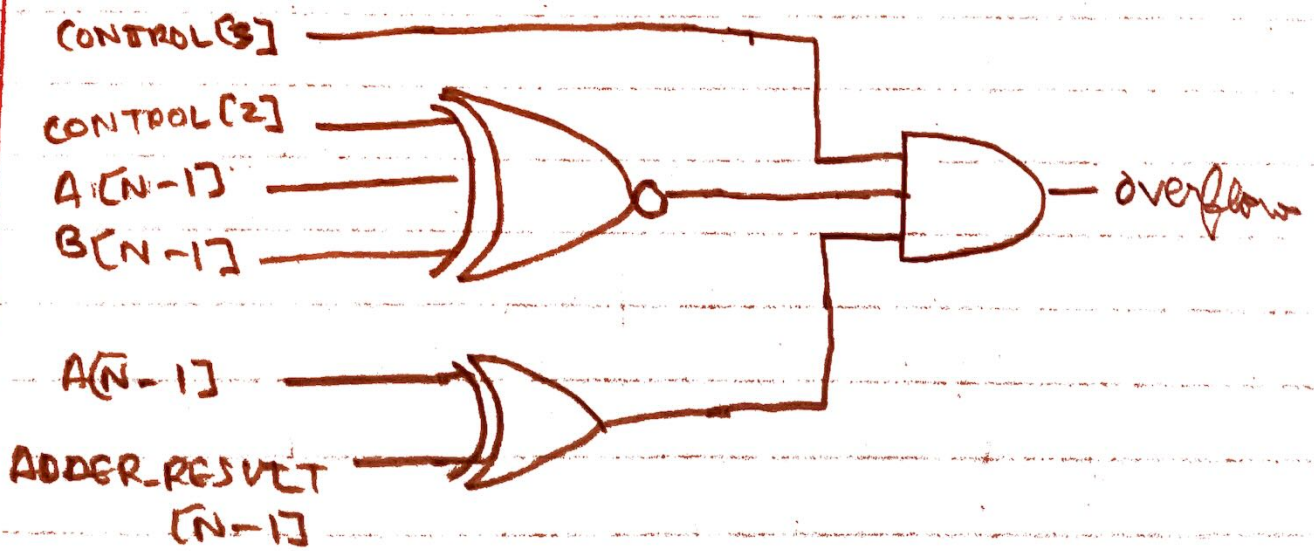
ALU MUX



zero

equal

OVERFLOW



RISC-V CPU
(RV32i)