Control unit design :

the states for the control unit:

S0 : initial state

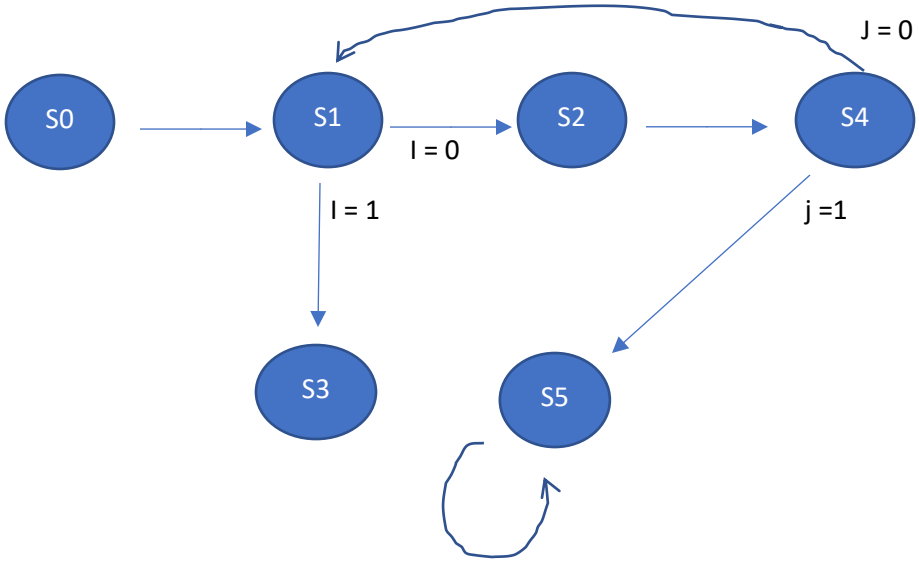
S1 : checks the LSB of the product at each iteration

S2: shifts the product one to the right

S3: adds to the 32 MSB of the product the multiplicand

S4 : checks the counter (if J == 1 then the counter has reached 32)

S5: the finial state(when j = 1)



5 states can be represented with 3 bits

I and j are my inputs and n2 n1 n0 are the states

Current state					Next state		
N2	N1	N0	I	J	N2`	N1`	N0`
0	0	0	X	X	0	0	1
0	0	1	0	X	0	1	0
0	0	1	1	X	0	1	1
0	1	0	X	X	1	0	0
0	1	1	X	X	0	1	0
1	0	0	X	0	0	0	1
1	0	0	X	1	1	0	1
1	0	1	X	X	1	0	1

N2` = 010xx + 100x1 + 101xx

N1` = 001xx+011xx = 0x1xx

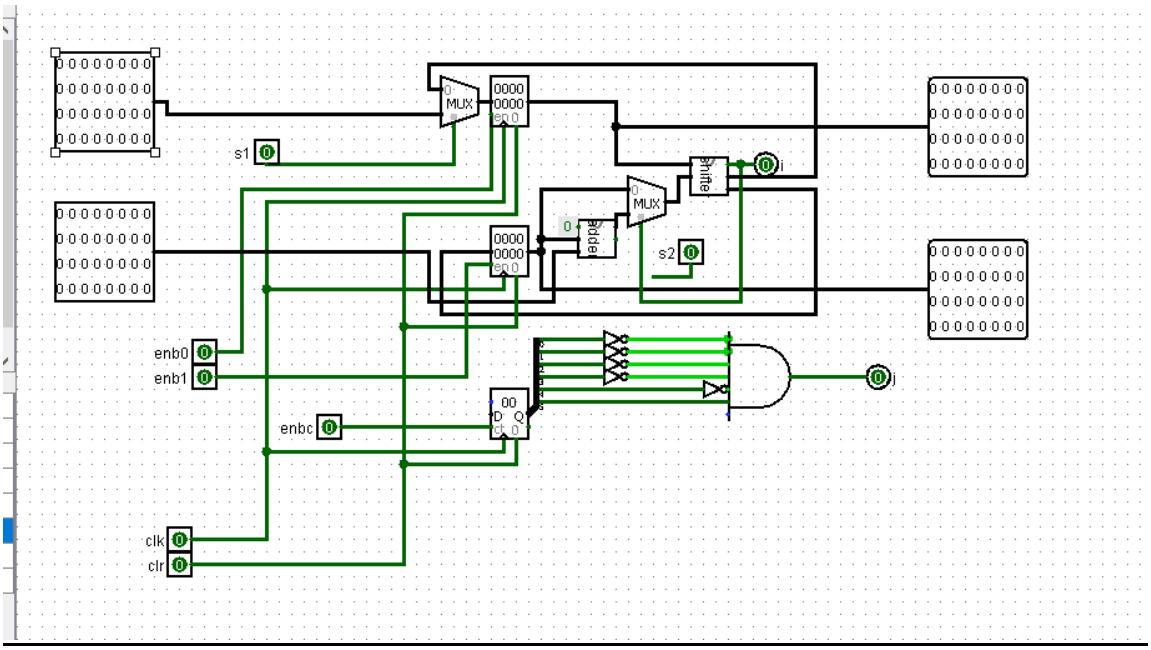
N0` = 000xx + 0011x + 100xx + 101xx

S1 = 000xx

Enb0 = 100xx

Enb1 = 100xx + 000xx

Datapath design :



S1 is 1 at the initial state only so it can put the multiplier into the 32 LSB of the product

Here i (the LSB of the product) is the selector of the mux :

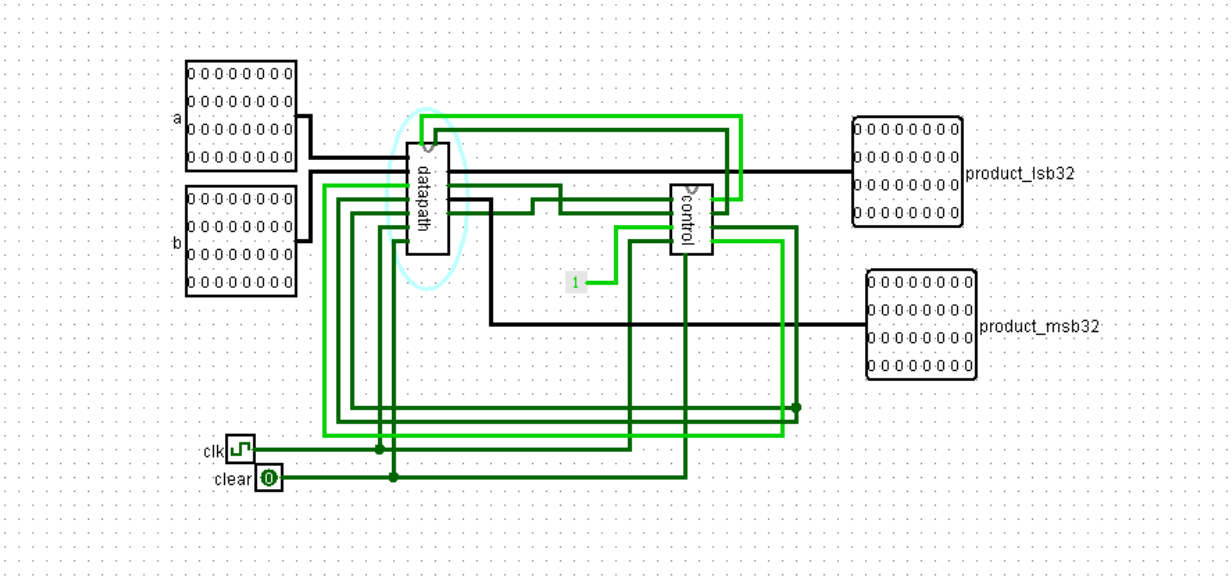
When I = 1 it adds the multiplicand with the 32 MSB of the product

When I = 0 it selects the 32 MSB of the product

The counter starts from 0 t0 32 and when it is 32 j becomes 1 and its clock is connected to enb0 when it becomes 1 at the 4^{th} state

32bit full adder design : designed one bit full adder then I used it to design 4 bit full adder then 16 bit then 32 bit full adder using two 16 bit full adder

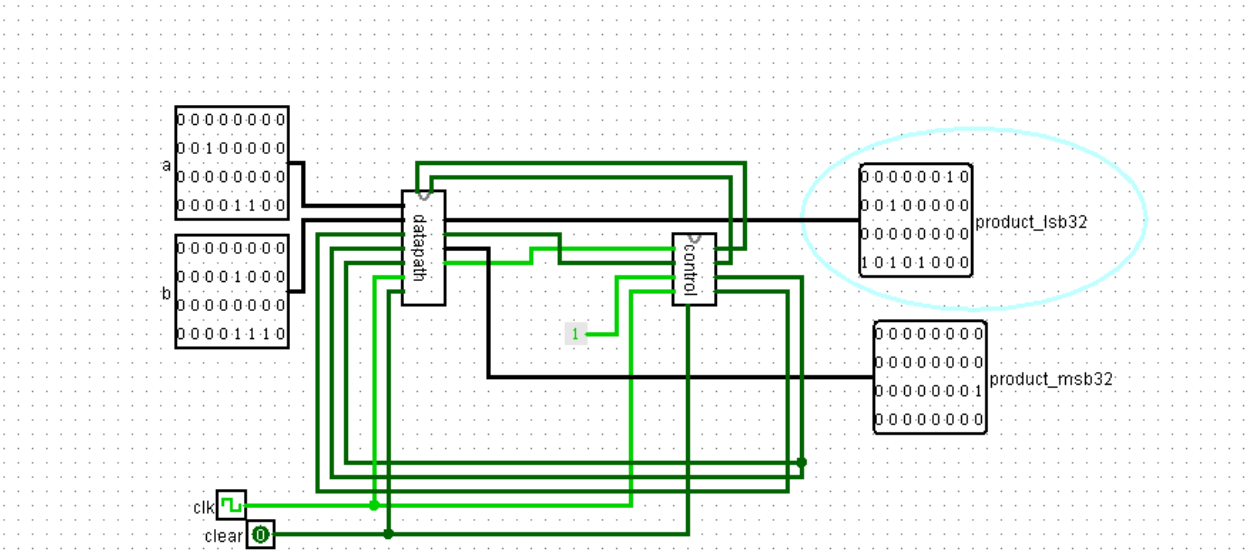
Mult32 design :

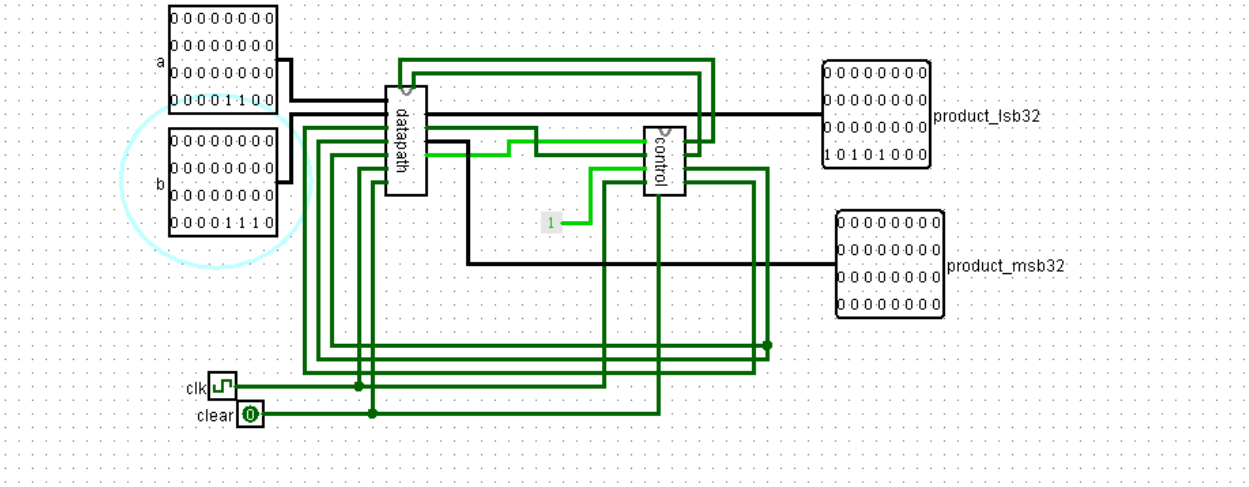


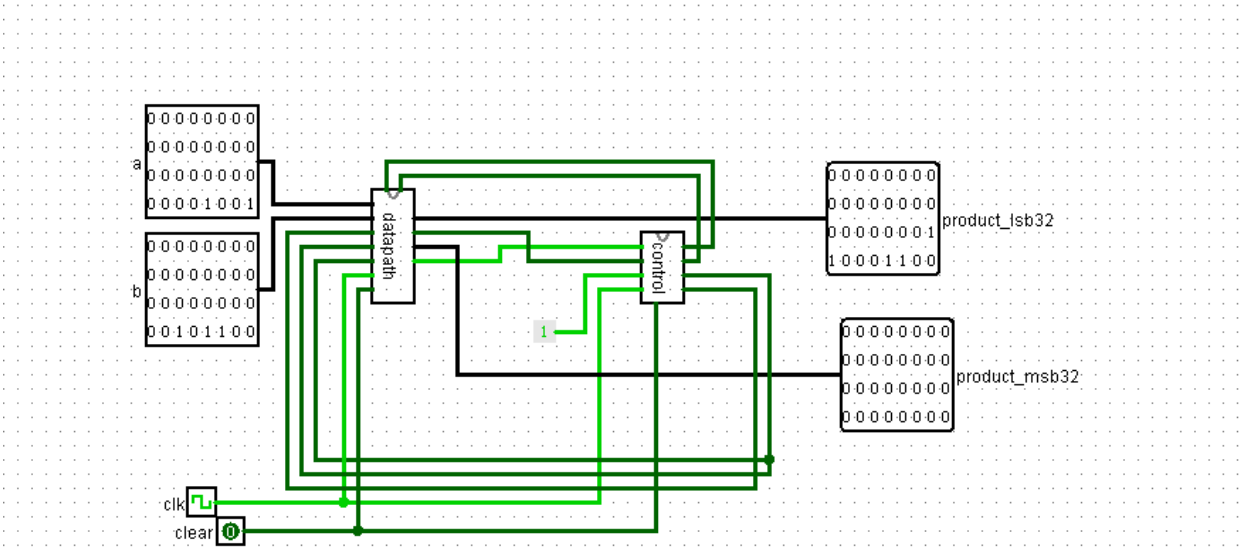
Contains the datapath circuit and the control unit

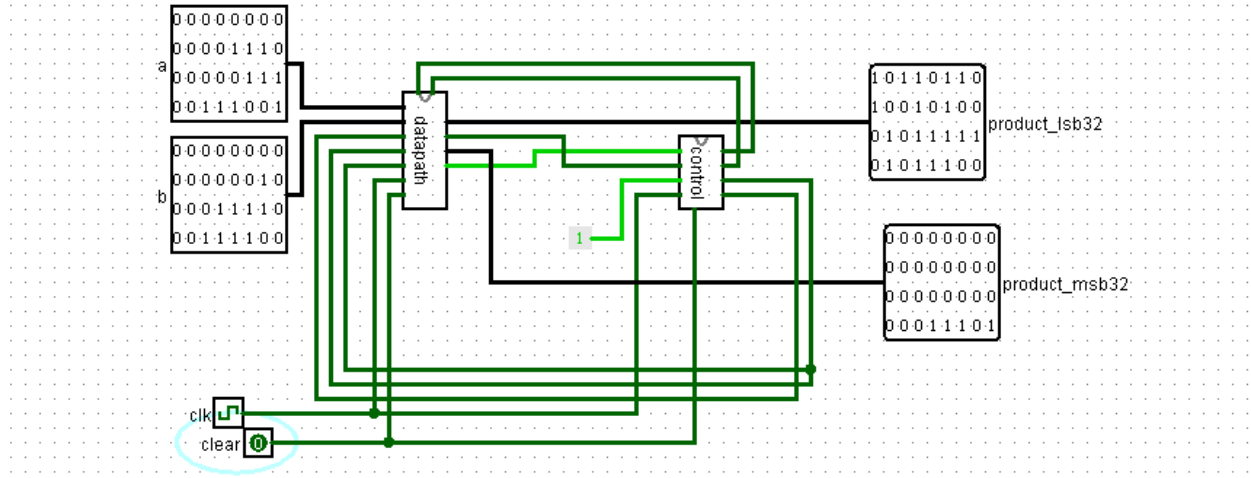
The clear pin to reset everything

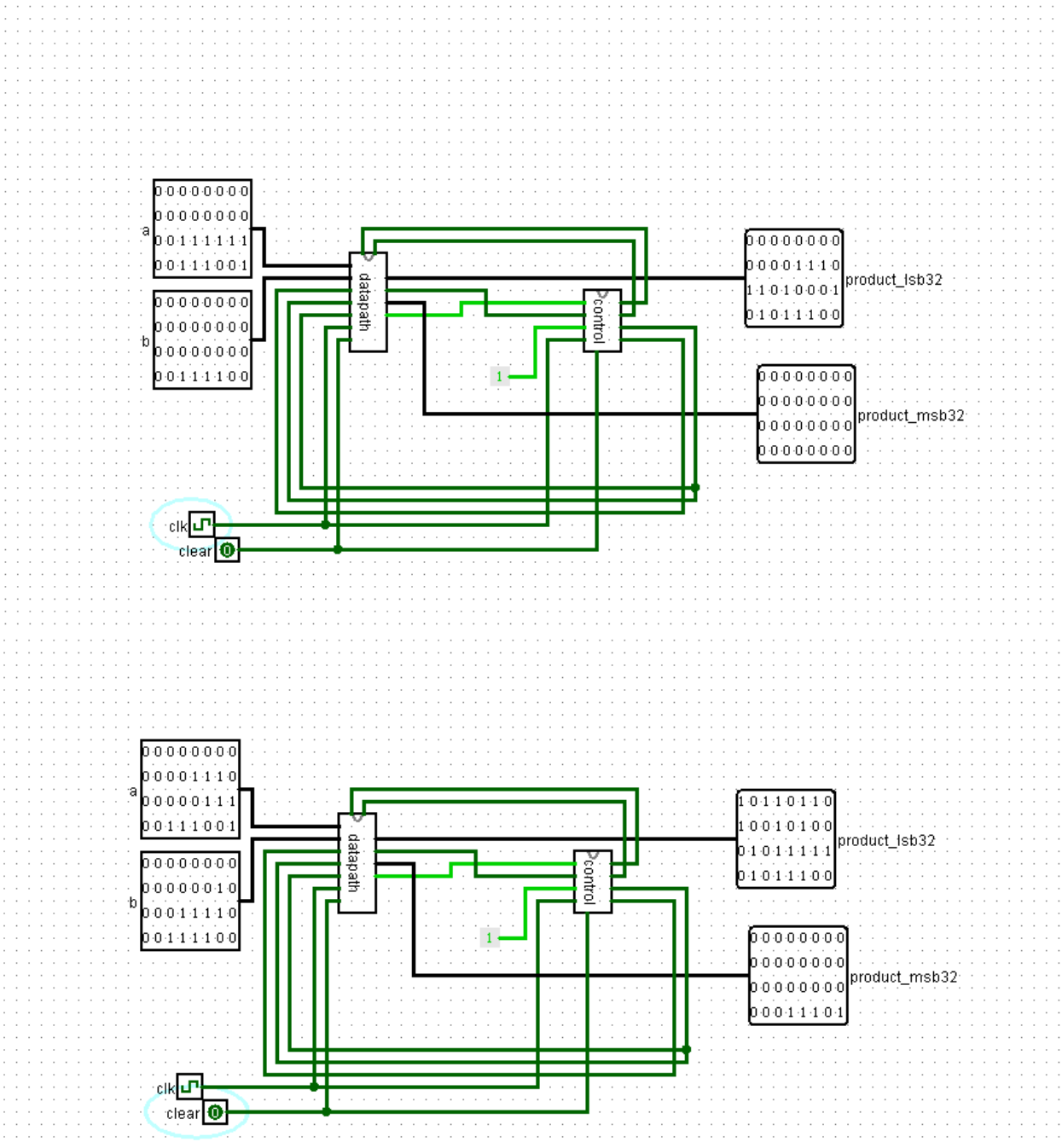
Test cases


a
b
datapath
control
1
clk
clear
product_lsb32
product_msb32


a
b
datapath
control
1
clk
clear
product_lsb32
product_msb32

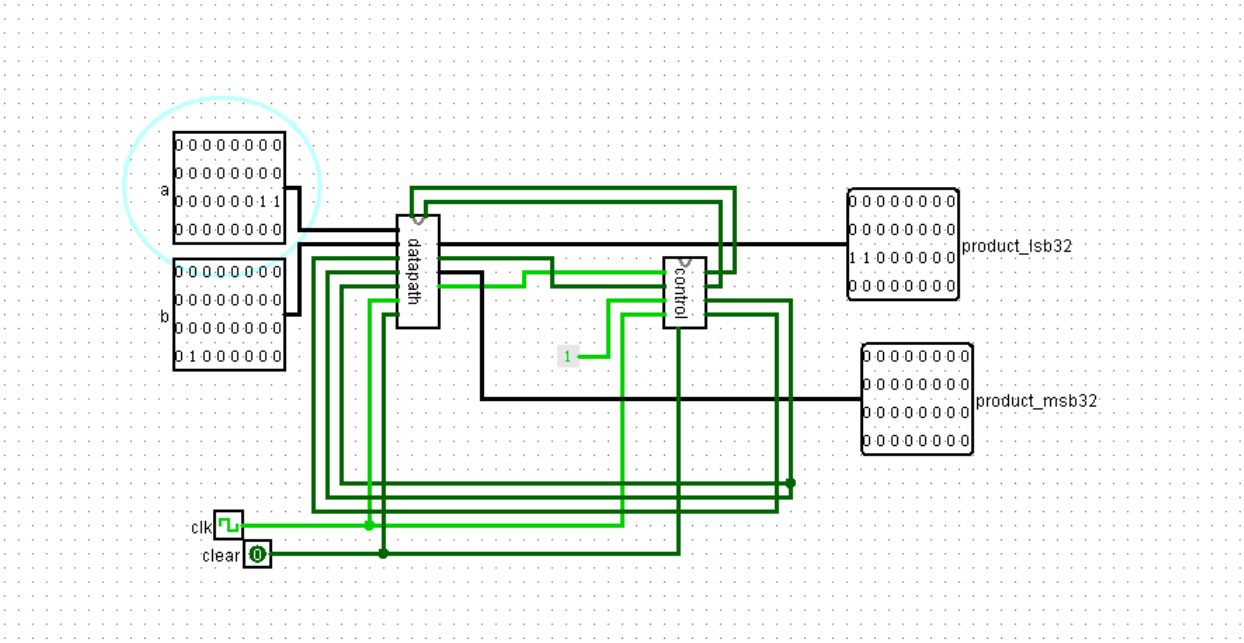

a
b
datapath
control
1
clk
clear
product_lsb32
product_msb32


a
b
datapath
control
1
clk
clear
product_lsb32
product_msb32



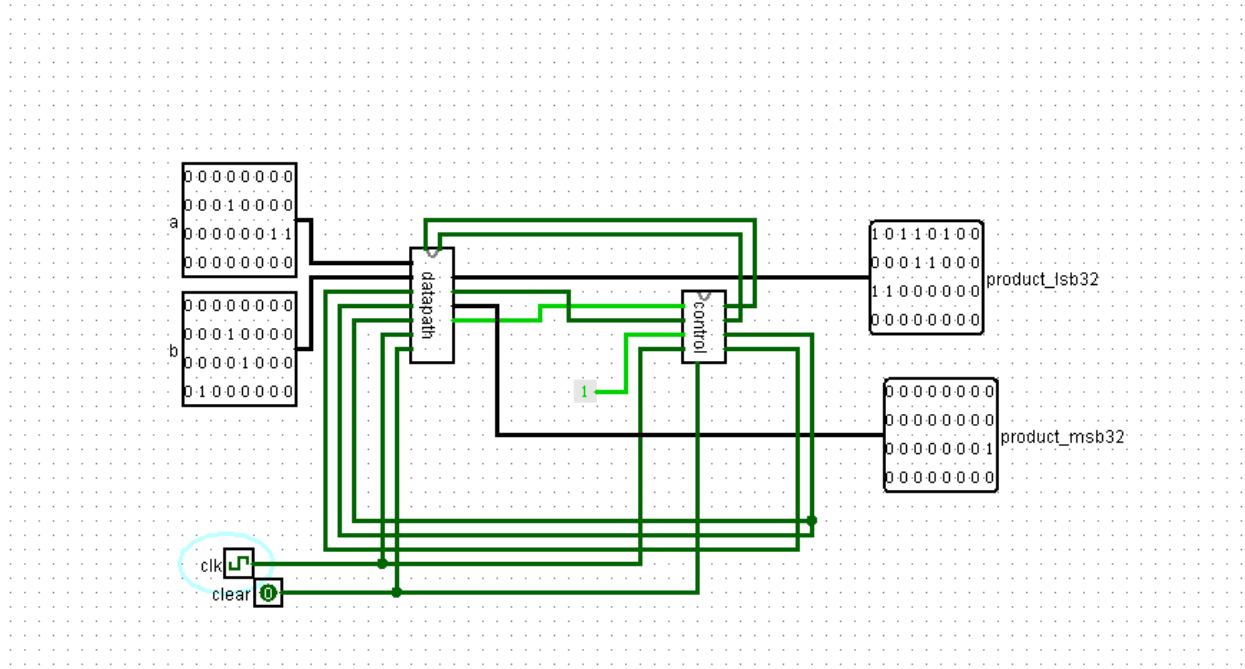
919353 * 138812 = 127617228636

In binary (00000000000000000000000000011101 10110110100101000101111101011100)



768 * 64 = 49152

In binary
(00000000000000000000000000000000000000000000000011000000000000000)



1049344 * 1050688 =1102533148672

In binary = 100000000 10110100000110001100000000000000