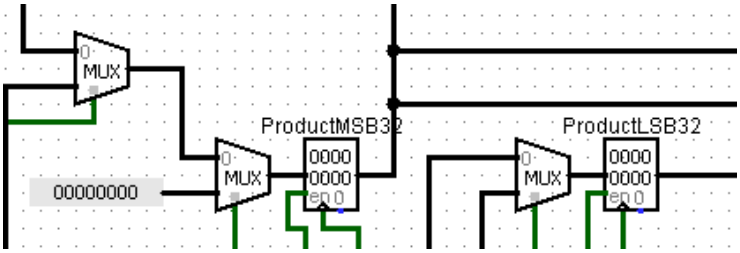
CSE 331/503
Computer Organization
Homework 3 Report

Berk PEKGÖZ
171044041

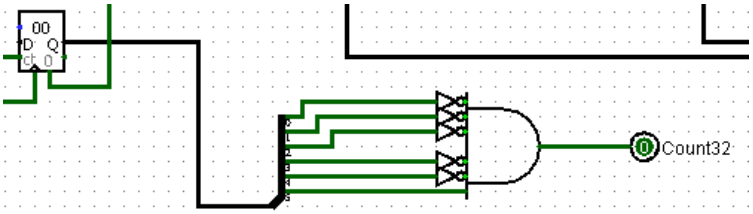
DATAPATH (datapath.circ)

64-Bit Product:



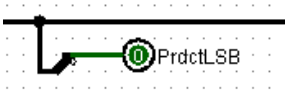
64-Bit product is stored in 2x32-bit registers. And their new contents decided by control unit with muxes (adder output - WAdd signal / shift output - WShift signal / initialization-Init signal). There are two muxes for ProductMSB32 because Most significant 32 bit can be written by adder so we need to choose one from three (add/shift/init).

6-Bit Counter and Count32 Signal:



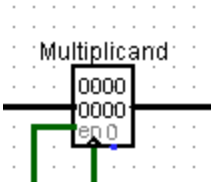
Datapath needs counter to create Count32 signal (when loop executed 32 times, gives 1). To make this possible 6-Bit counter used and count signal comes from control unit (Cnt signal). A and gate check the counter if “32” reached.

PrdctLSB Signal:



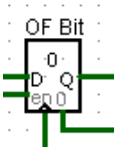
This signal gets the least significant bit of ProductLSB32 (with splitter) and deliver to control unit.

Multiplicand Register:



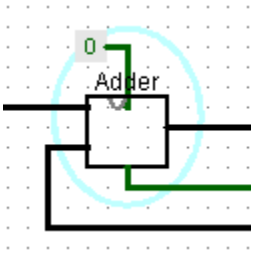
Keeps multiplicand number. Can be written when init signal is 1. Value of multiplicand is one of the outputs of adder.

Overflow Bit (OF Bit) Register:



This 1-bit register keeps the overflow value of the adder. Can be written when WAdd signal is 1. Resets when ResOF signal comes from control unit.

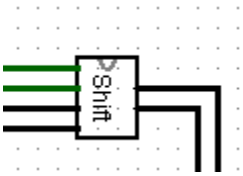
32-Bit Adder (BONUS):



Takes Multiplicand and Product32MSB as input. Cin is constant 0 as default because there is no Cin.

This adder designed by me. First, I built 1-Bit full adder and built 4-bit adder (4x 1bit adder). Then, 32-Bit adder (8 x 4bit adder) built.

64-Bit Shifter (BONUS):

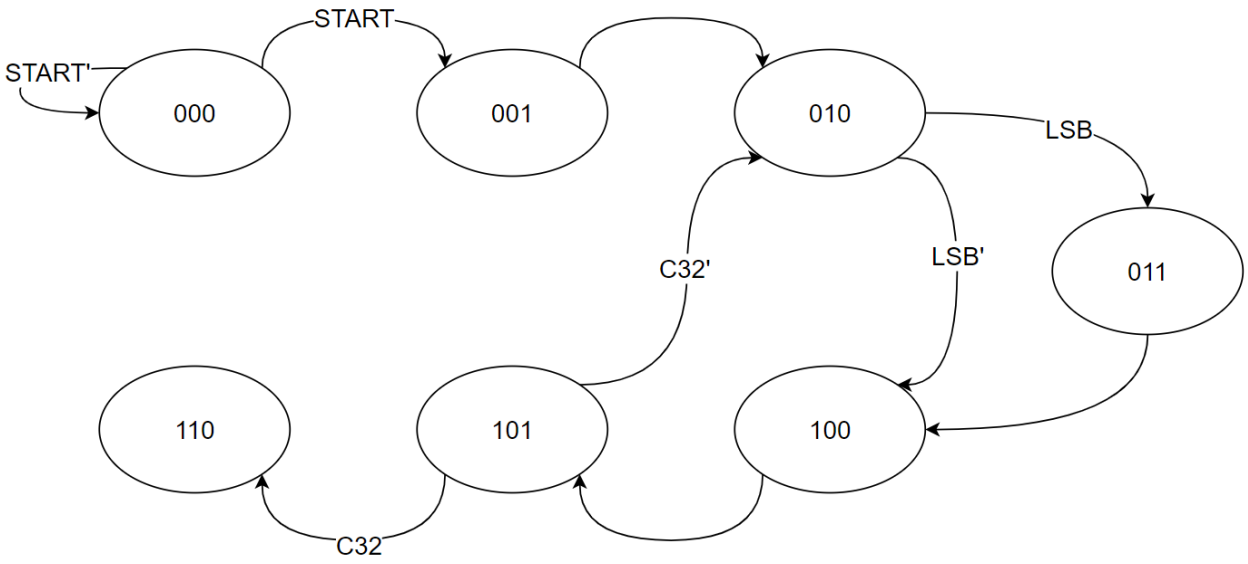


Takes 64-Bit as input (ProductMSB32 and ProductLSB32) and gives 1-Bit right shifted output if WShift signal is 1. Also takes OFBit as input and Most significant bit replaced by it.

This shifter designed by me. This shifter is only shift 1-bit right.

CONTROL UNIT (control.circ)

FSM for Control Unit:



Input signals:

START = Starts the FSM
LSB= Least Significant Bit of Product in Datapath
C32= If counter in the Datapath reach 32 then this signal become 1.

Reset= This signal resets the state registers directly.

States:

000 = Empty state. When start signal comes pass to state 001.
001 = Initialization state. Sets the registers in Datapath properly. Init = 1.
010 = Condition state. If LSB is 1 then pass to add state. If not, pass to shift state.
011 = Add state. WAdd = 1.
100 = Shift state. WShift = 1, Cnt = 1
101 = Condition state. If C32 is 1 then pass to 110 state. If not, pass to state 010.
110 = End state. FSM ends.

Truth table for FSM:

S2	S1	S0	Start	LSB	C32		Z2	Z1	Z0
0	0	0	0	x	x		0	0	0
0	0	0	1	x	x		0	0	0
0	0	1	x	x	x		0	1	0
0	1	0	x	0	x		1	0	0
0	1	0	x	1	x		0	1	1
0	1	1	x	x	x		1	0	0
1	0	0	x	x	x		1	0	1
1	0	1	x	x	0		0	1	0
1	0	1	x	x	1		1	1	0

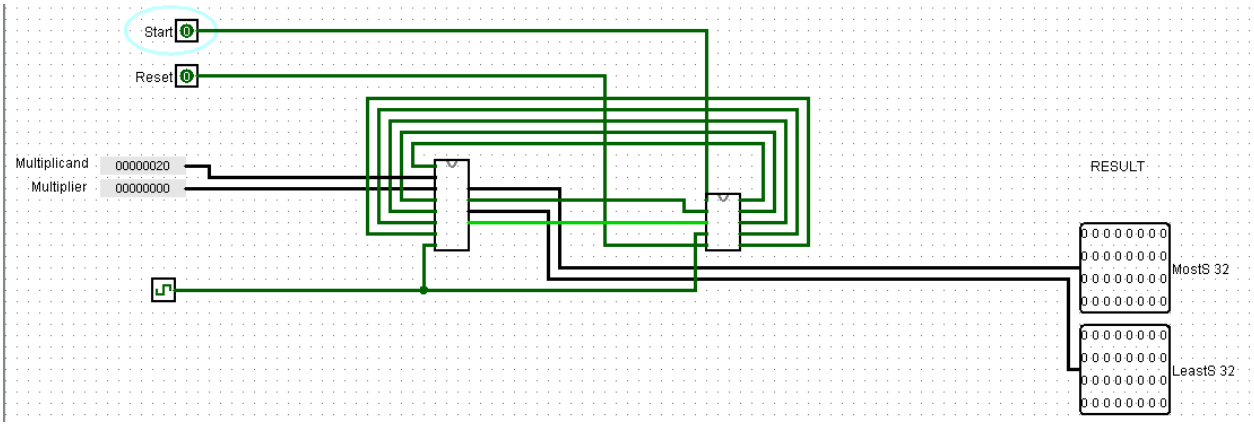
1	**1**	**0**	**x**	**x**	**x**		**1**	**1**	**0**

TEST CASES

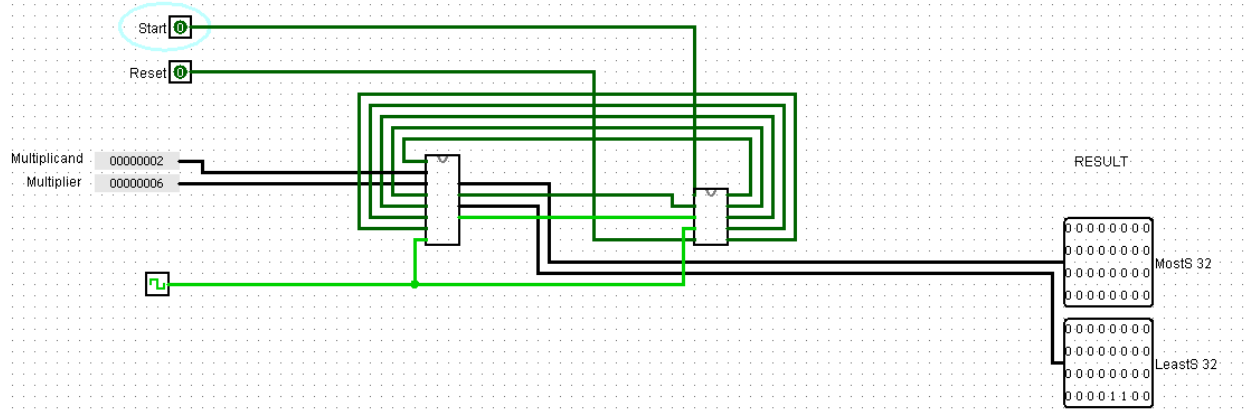
CASE	**SCENERIO**	***Expected***	***Actual***	***Pass***
1	32 x 0	00000000000000000000000000000000	00000000000000000000000000000000	Yes
2	2 x 6	00000000000000000000000000001100	00000000000000000000000000001100	Yes
3	2045 x 1999	00000000000000000000011111001100	00000000000000000000011111001100	Yes
4	234920 x 674	00000000000000000010010111000000	00000000000000000010010111000000	Yes
5	4294967295 x 2 (Overflow test)	00000000000000001111111111111110	00000000000000001111111111111110	Yes
6	4294967295 x 4294967295 (Overflow test)	11111111111111111111111111111000	11111111111111111111111111111000	Yes

Note: In mult32.circ file 32-bit constants are used for input numbers because it is easier to test and giving numbers in logisim by changing their value on the sidebar. They can be replaced by 32-bit inputs

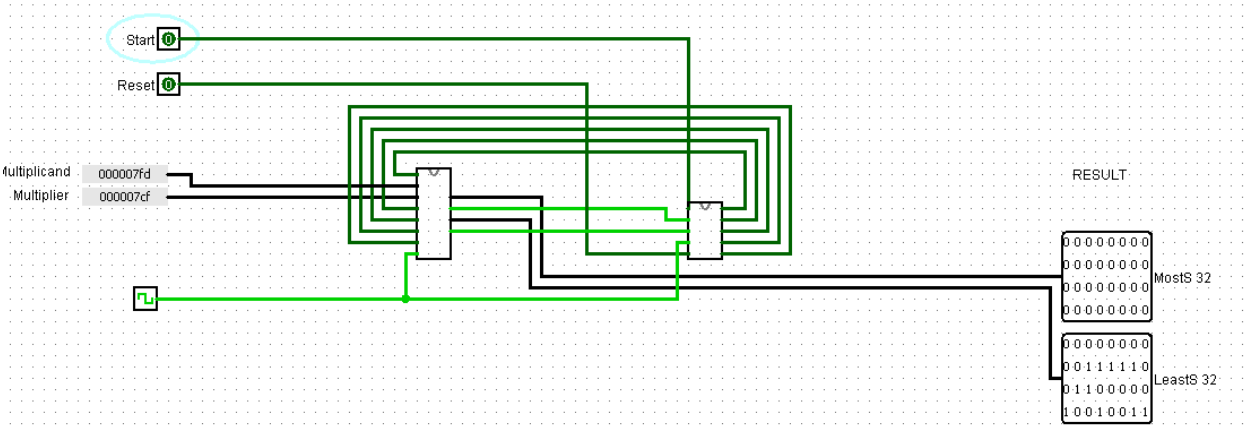
Case 1:



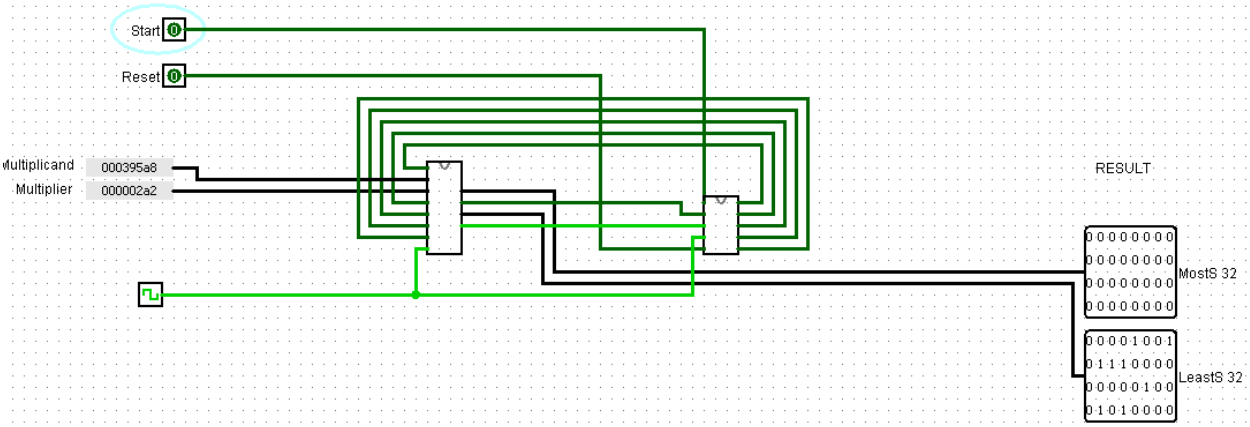
Case 2:



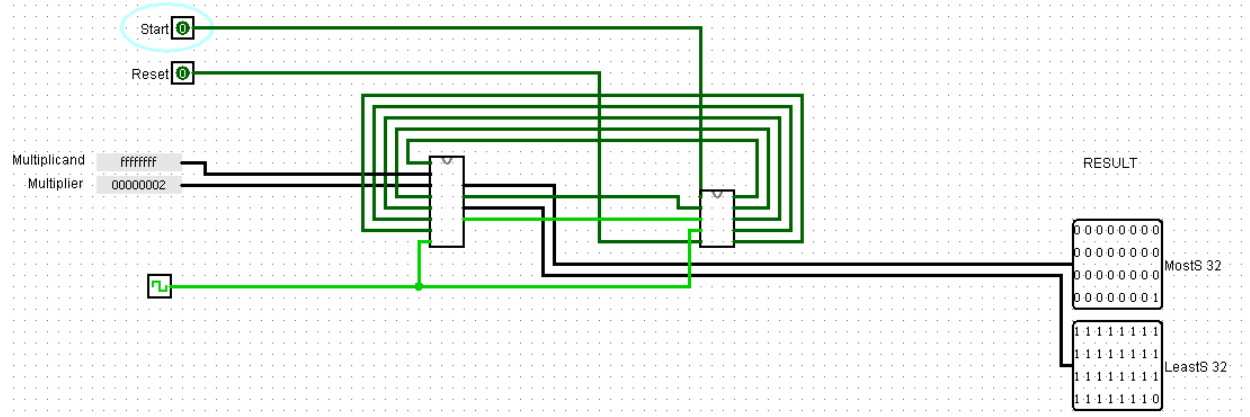
Case 3:



Case 4:



Case 5:



Case 6: