Course: CSC258F

Professor: Steve Engels

Experimenter: Yuhao Yang

Student ID: 1005808057

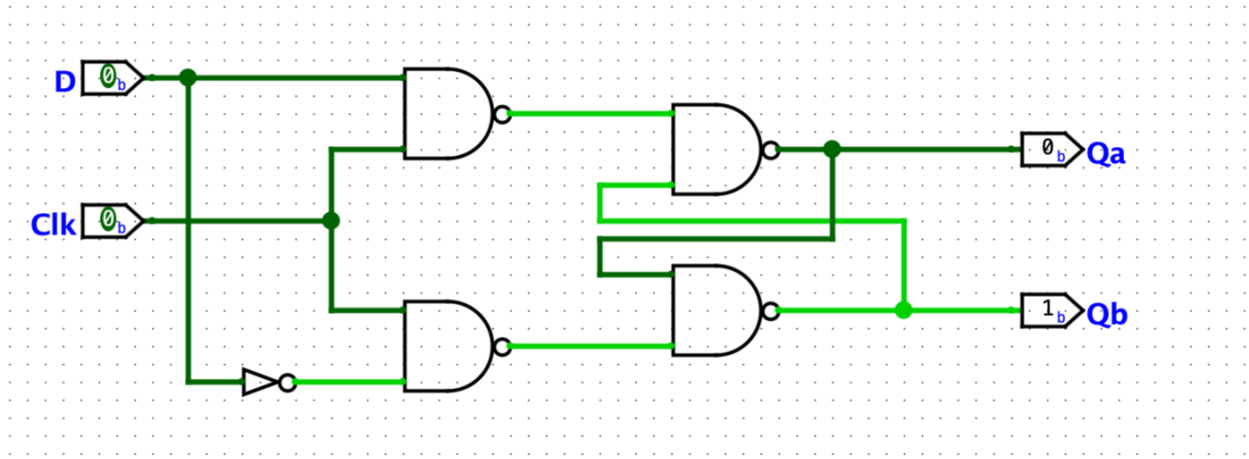
Lab4

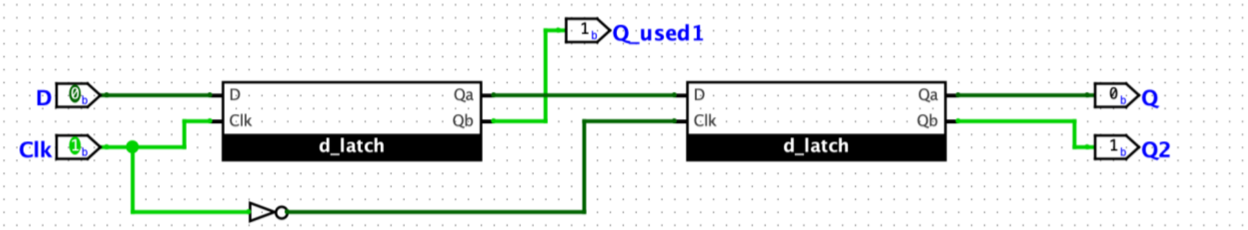
Pre-Lab Report

Part1:

1. Build the D latch and master slave flip-flop in Logisim in different modules.

D latch circuit





2. Show the circuit is correct by using Poke.

Showed on Logisim.

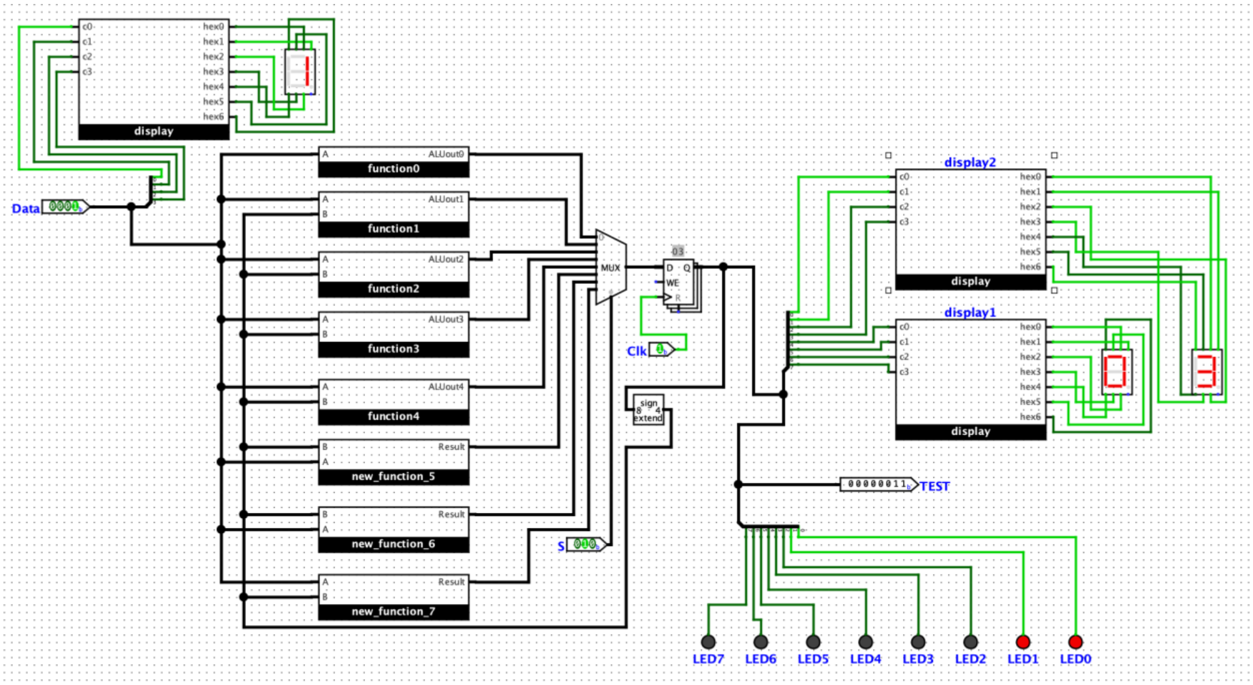
3. For the D latch and the flip flop, are there any input combinations of Clk and D that should NOT be the first you test?

From my perspective, all combinations for the D latch should not be the first test, because no matter Clk takes value of zero of one for the first test, one of the D latches will have zero as input of Clk since invertor. As we know, when Clk is zero, then the D latch will keep the previous state, but there is no previous state for both D latches when I first test, then it will raise an error. Same to flip flop, one of D latch

will have zero of Clk, but there is no previous state for that D latch, so the flip flop will raise an error for all combinations of Clk and D.

Part2:

1. Build the circuit in Logisim.



2.

a) What would happen if we remove register in the circuit?

By observing the circuit, we know that if we remove the register, then the function 1 to function 7 will not have input signal B, and it will cause errors if we select function 1 to 7 in MUX.

b) How many bits we need to store the result if we multiple two n-bit binary number?

If we multiple two n-bit binary number, then we need 2n bits to store the result.

3. Answer the questions below.

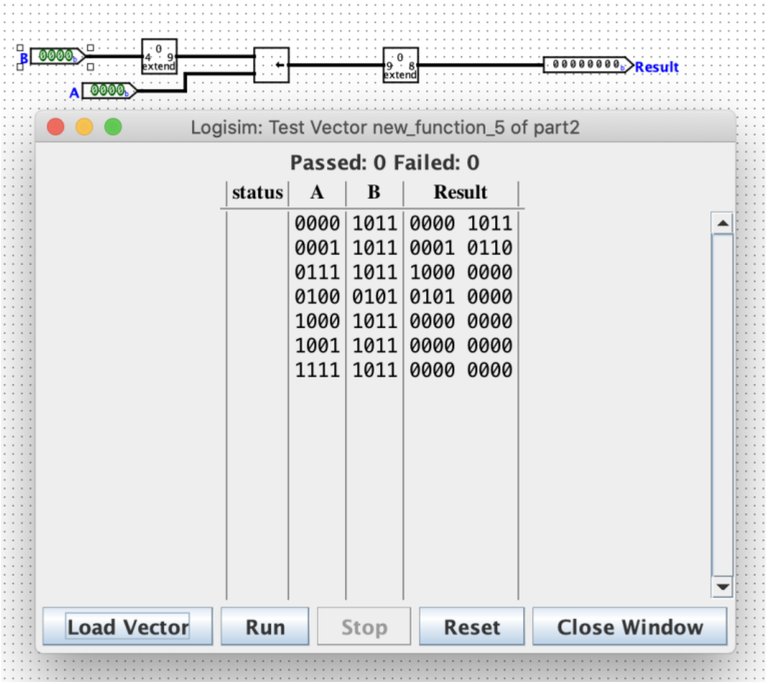
a) Let data to be zero than we set clk to be one, then set clk to be zero and change s to be 001 from 000 which means we changed to the next function and set clk to be one again, we just need to follow this process and we can get a correct and net sequence.

Data	Function	S	A	B	Clk	TEST
0001	0	000	0001	X X	1 0	00000001
0001	1	001	0001	0010 0011	1 0	00001111
0001	2	010	0001	0011 0100	1 0	00000100
0001	3	011	0001	0100 0101	1 0	01010101
0001	4	100	0001	0101 0101	1 0	00000001
0001	5	101	0001	0001 0001	1 0	00000010
0001	6	110	0001	0001 0001	1 0	00000001
0001	7	111	0001	0001 0001	1 0	00000001

The order from top to bottom

b) Use test vector to show new operations are correct.

Function5

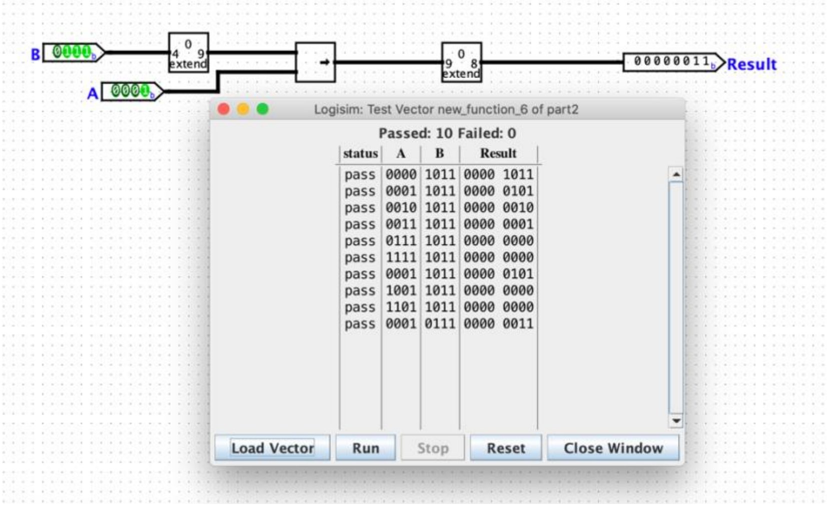


Logisim: Test Vector new_function_5 of part2

Passed: 0 Failed: 0

status	A	B	Result
	0000	1011	0000 1011
	0001	1011	0001 0110
	0111	1011	1000 0000
	0100	0101	0101 0000
	1000	1011	0000 0000
	1001	1011	0000 0000
	1111	1011	0000 0000

Function6

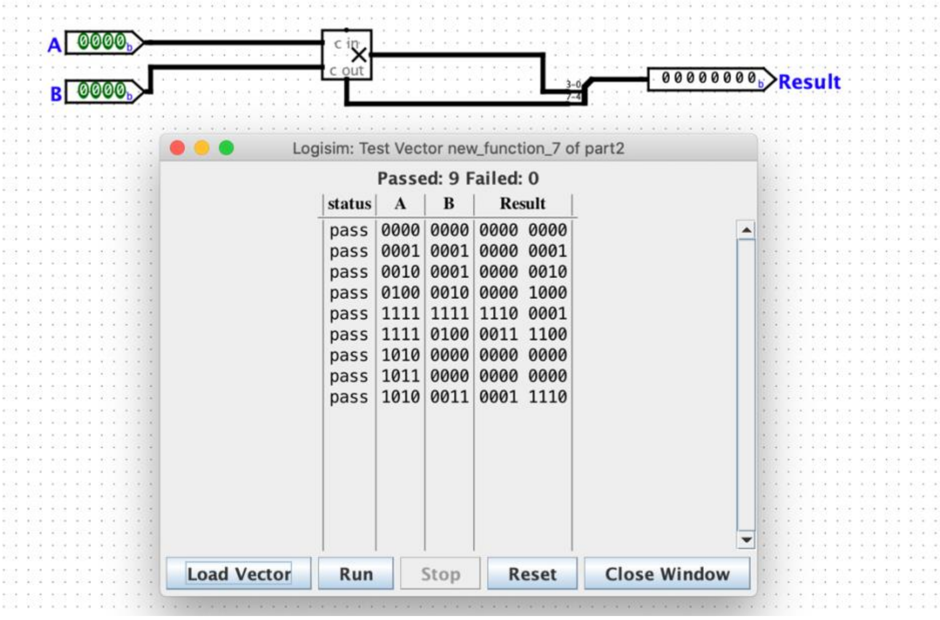


Logisim: Test Vector new_function_6 of part2

Passed: 10 Failed: 0

status	A	B	Result
pass	0000	1011	0000 1011
pass	0001	1011	0000 0101
pass	0010	1011	0000 0010
pass	0011	1011	0000 0001
pass	0111	1011	0000 0000
pass	1111	1011	0000 0000
pass	0001	1011	0000 0101
pass	1001	1011	0000 0000
pass	1101	1011	0000 0000
pass	0001	0111	0000 0011

Function7



Logisim: Test Vector new_function_7 of part2

Passed: 9 Failed: 0

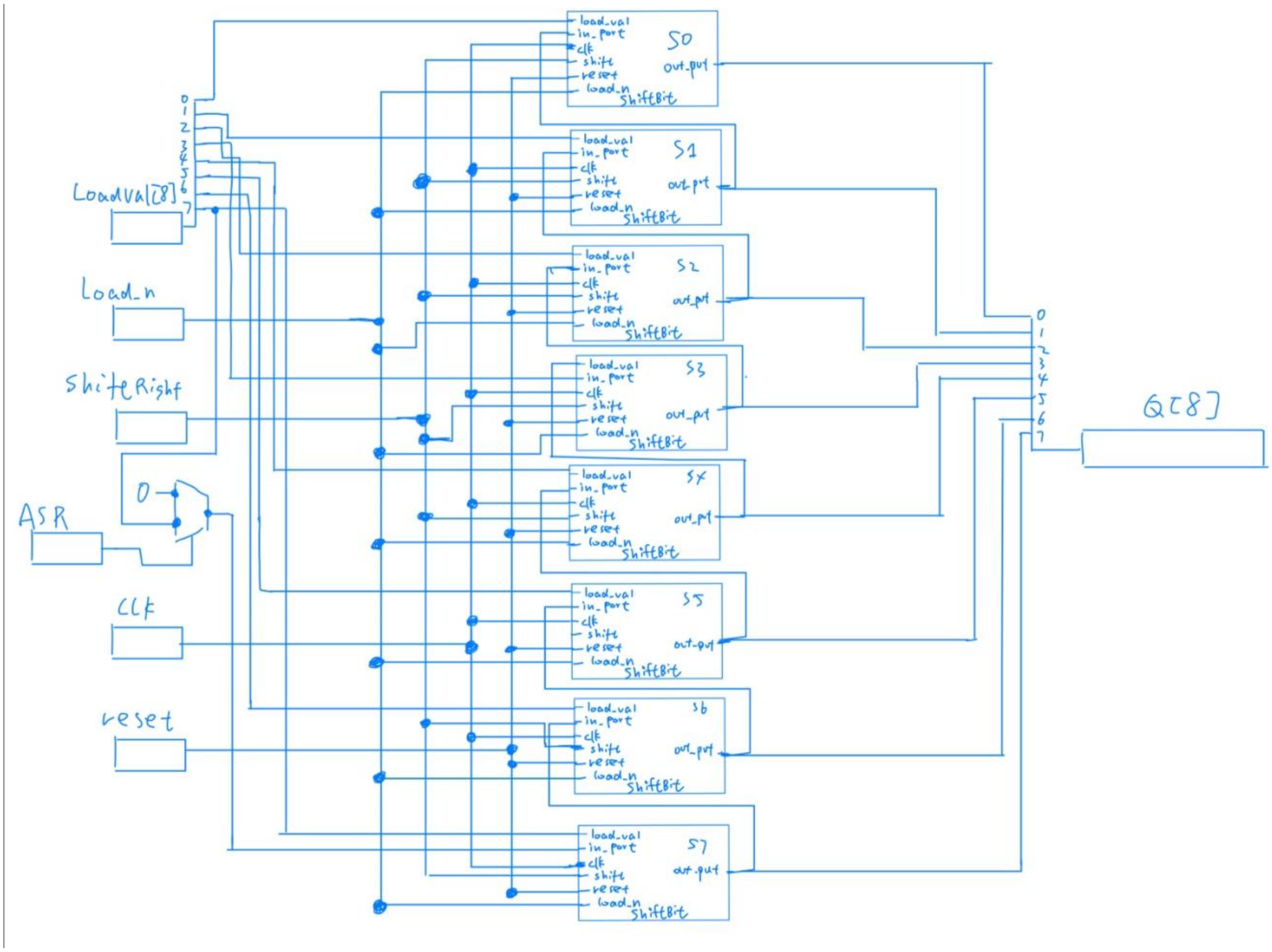
status	A	B	Result
pass	0000	0000	0000 0000
pass	0001	0001	0000 0001
pass	0010	0001	0000 0010
pass	0100	0010	0000 1000
pass	1111	1111	1110 0001
pass	1111	0100	0011 1100
pass	1010	0000	0000 0000
pass	1011	0000	0000 0000
pass	1010	0011	0001 1110

Part3:

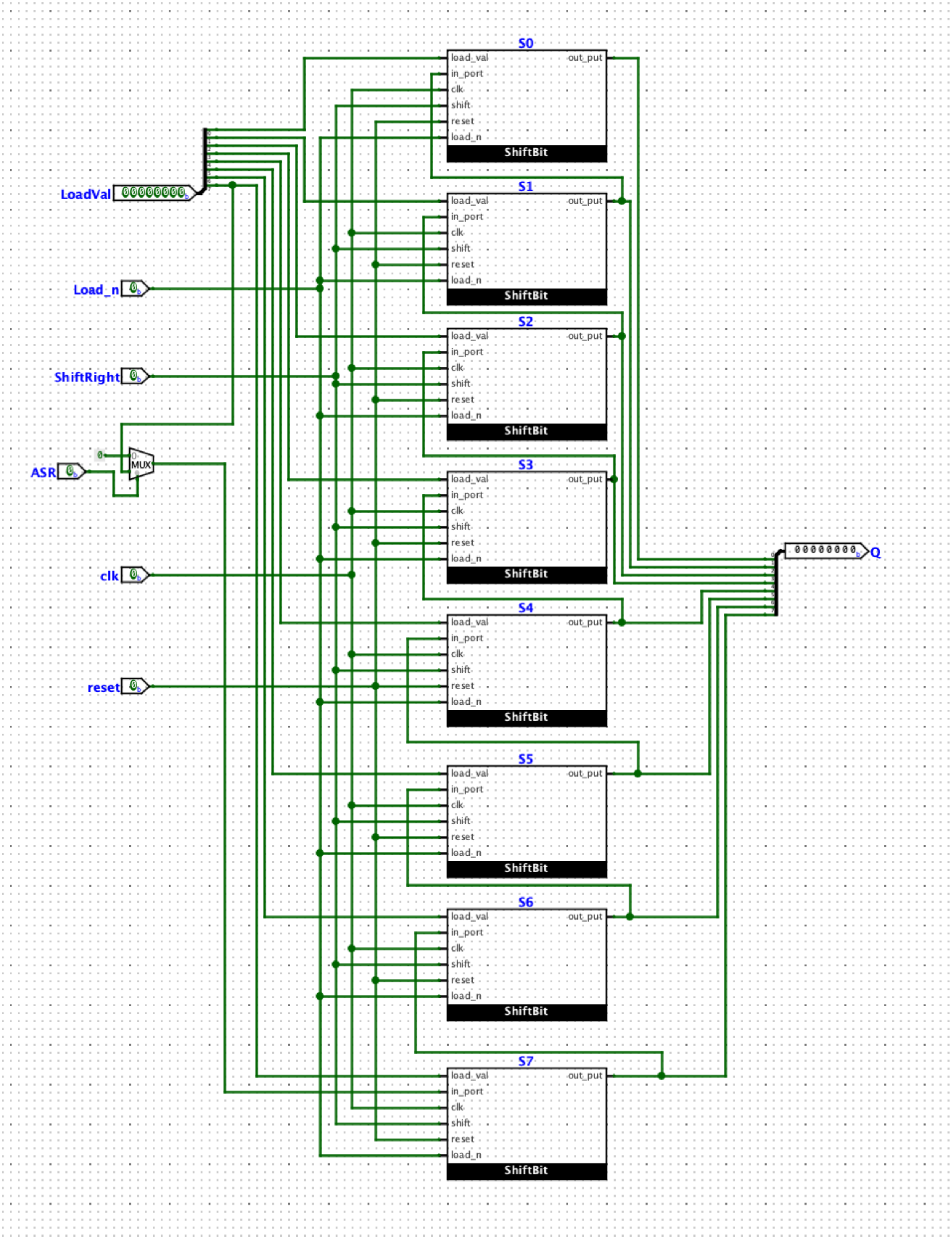
1. What is the behaviour of the 8-bit shift register shown in Figure 5 when Load n = 1 and ShiftRight = 0?

When Load n = 1 and ShiftRight = 0, the output will remain the previous state because load_val cannot pass the second MUX in ShiftBit.

2. Draw the schematic



4. Show the circuit in Logisim.


S0
S1
S2
S3
S4
S5
S6
S7
load_val
in_port
clk
shift
reset
load_n
out_put
ShiftBit
LoadVal
00000000
Load_n
ShiftRight
ASR
MUX
0
clk
reset
00000000
Q