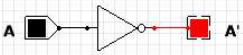
LOGIC GATES

BASIC GATES

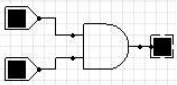
NOT / INVERTER



TRUTH TABLE

INPUT	OUTPUT
A	Y = A'
0	1
1	0

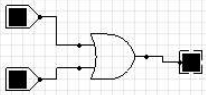
AND



TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	0

OR

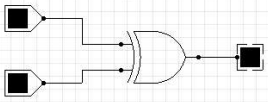


TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

SPECIAL PURPOSE LOGIC GATES

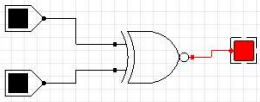
X-OR



TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

X-NOR

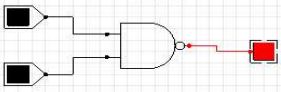


TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

UNIVERSAL LOGIC GATES

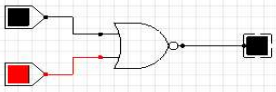
NAND



TRUTH TABLE

INPUT		OUTPUT
A	B	Y =(A . B)'
0	0	1
0	1	1
1	0	1
1	1	0

NOR



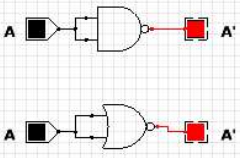
TRUTH TABLE

INPUT		OUTPUT
A	B	Y = (A + B)'
0	0	1
0	1	0
1	0	0
1	1	0

IMPLEMENTATION OF LOGIC GATES USING UNIVERSAL LOGIC GATES

BASIC GATES

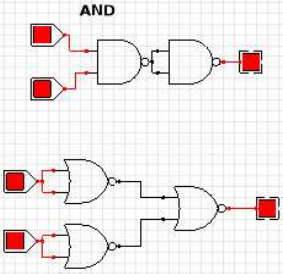
NOT / INVERTER



TRUTH TABLE

INPUT	OUTPUT
A	Y = A'
0	1
1	0

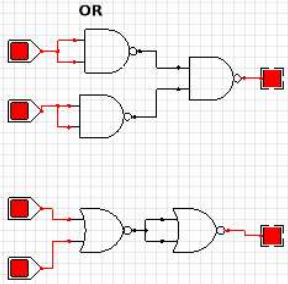
AND



TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

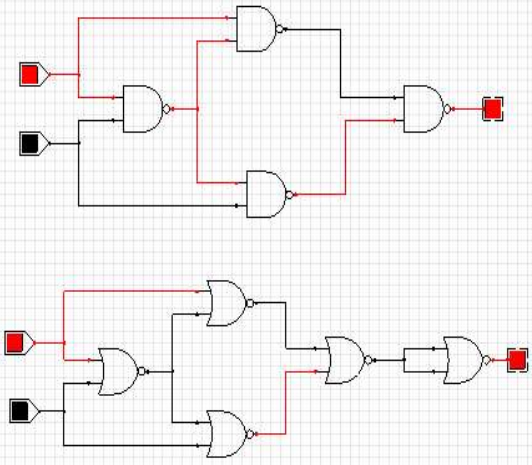
OR



TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

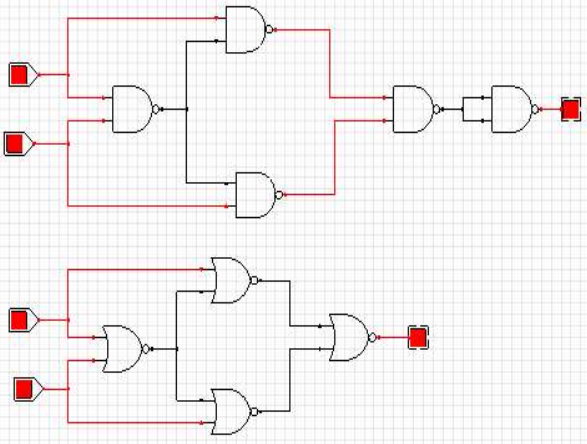
X-OR



TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

X-NOR

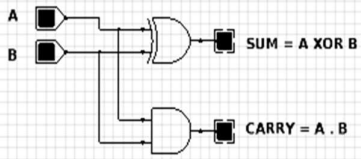


TRUTH TABLE

INPUT		OUTPUT
A	B	Y = A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

EX. 3) Design and verification of the truth tables of Half and Full adder circuits.

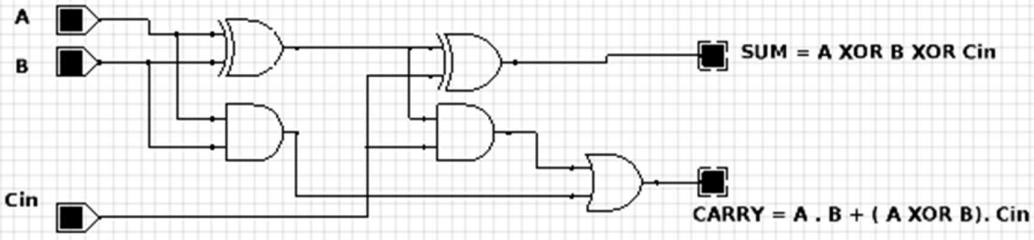
1. Hall adder



TRUTH TABLE

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

1. FULL adder

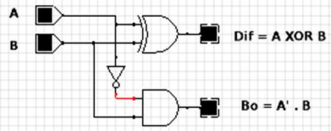


TRUTH TABLE

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

EX 4). Design and verification of the truth tables of Half and Full subtractor circuits

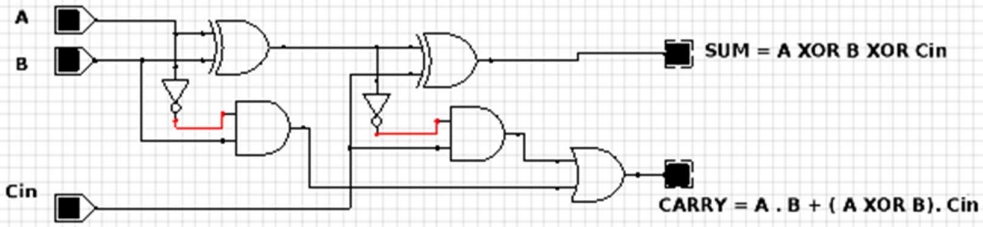
1. Hall SUBTRACTOR



TRUTH TABLE

A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

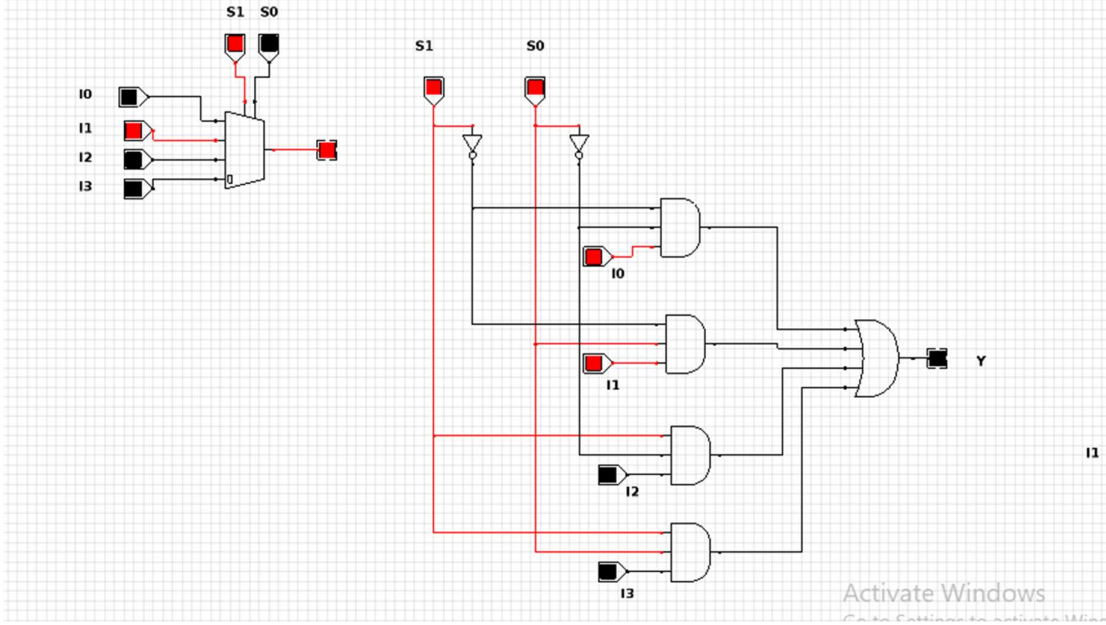
1. FULL adder



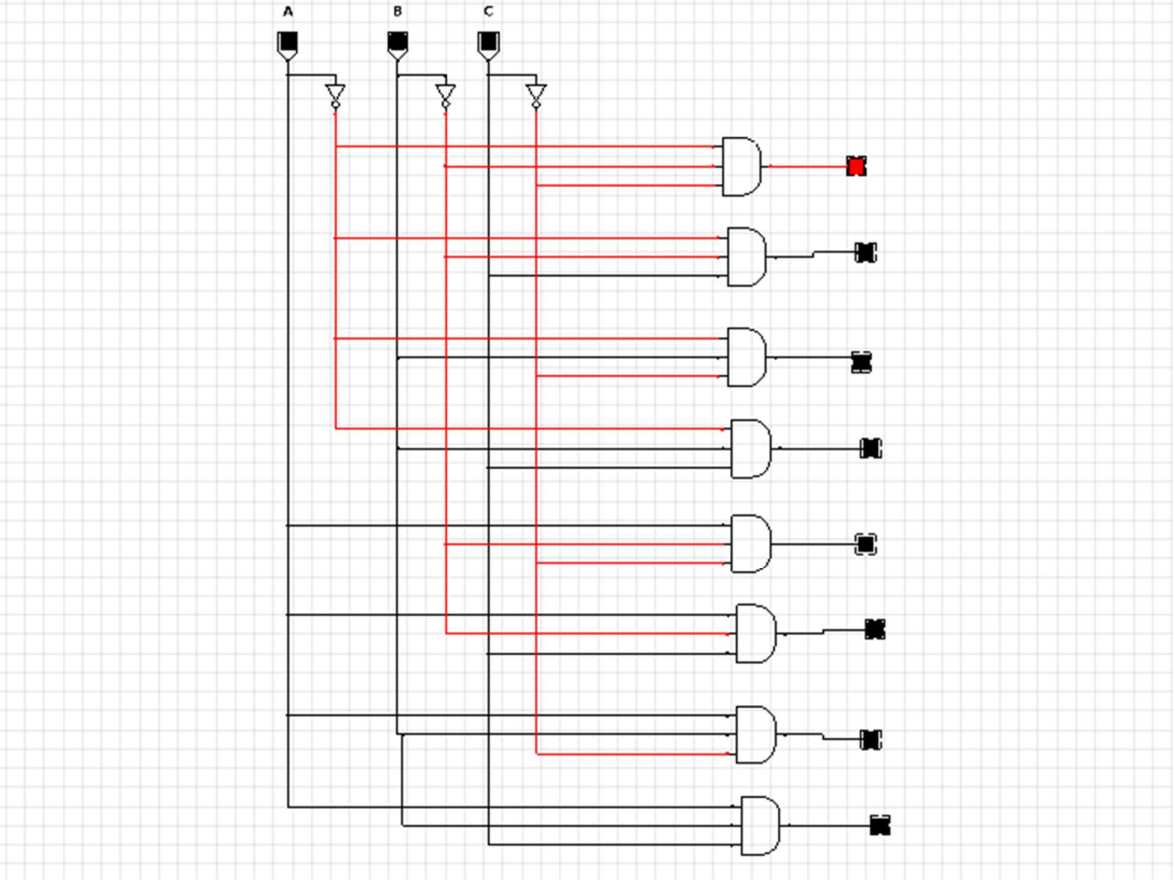
TRUTH TABLE

A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

EX. 5) Design and implementation of 4:1 Multiplexer using logic gates

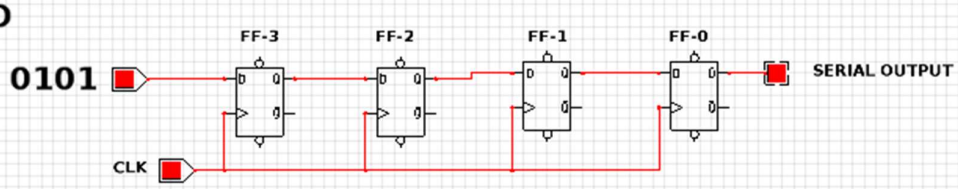


ex.6 Design and implementation of 3:8 decoder using logic gates



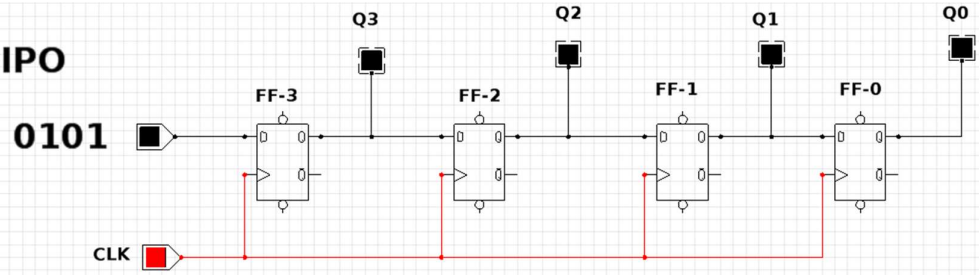
PRACTICAL 8 : DESIGN SHIFT REGISTER USING FLIP FLOPS

1. SISO



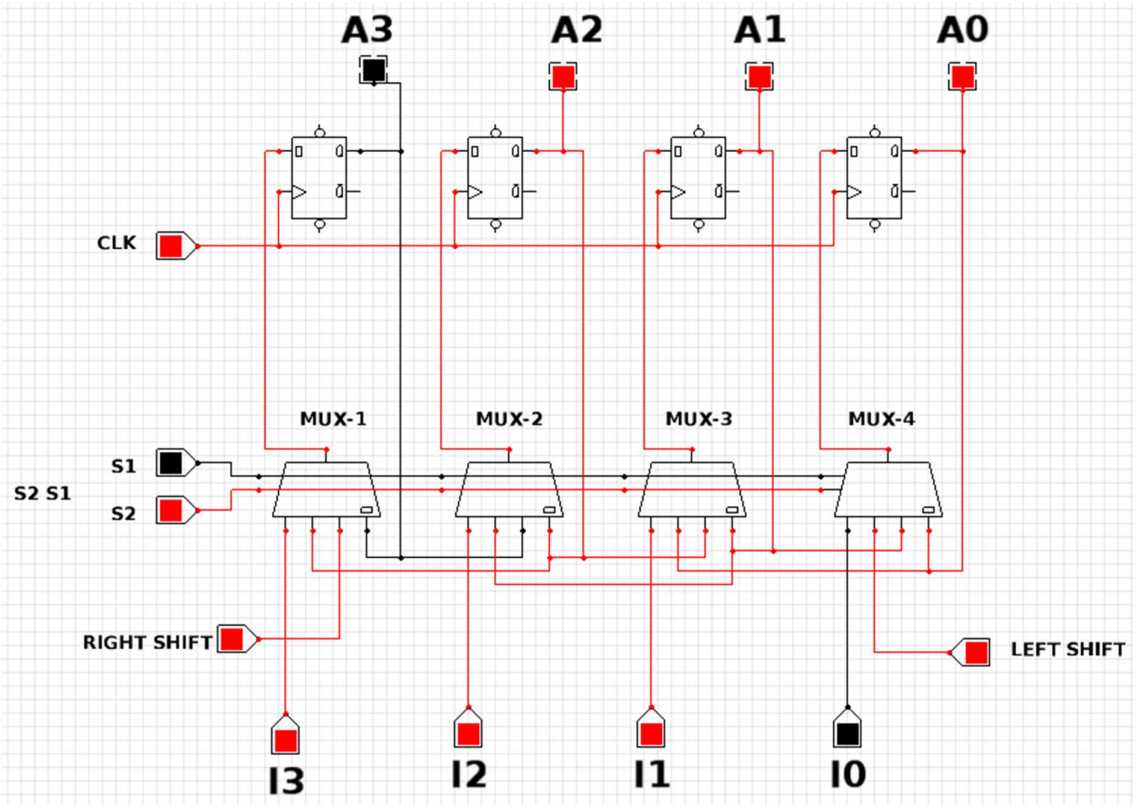
CLK	SERIAL I/P	Q3	Q2	Q1	Q0
0	1	0	0	0	0
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	1	0
4	-	0	1	0	1

2. SIPO



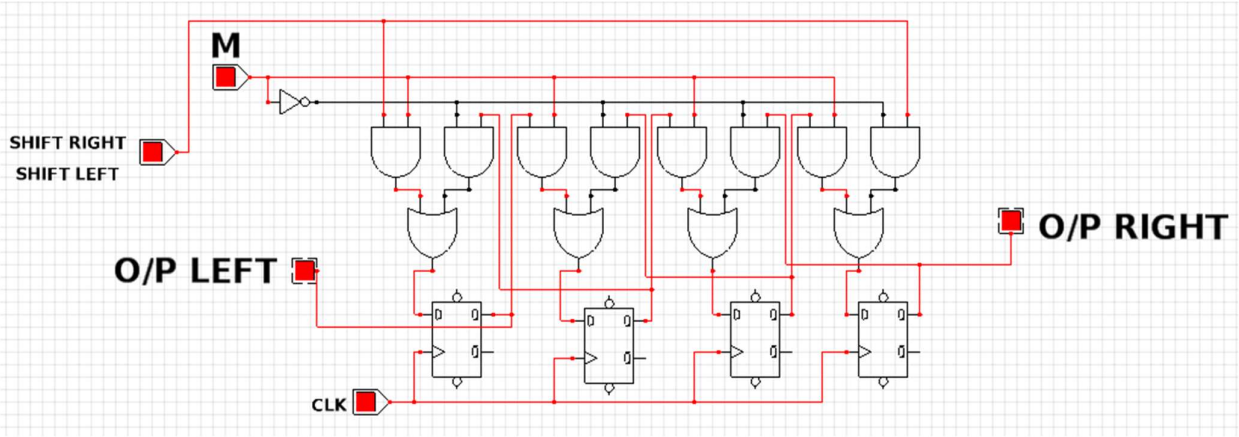
CLK	SERIAL I/P	Q3	Q2	Q1	Q0
0	1	0	0	0	0
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	1	0
4	-	0	1	0	1

UNIVERSAL SHIFT REGISTER

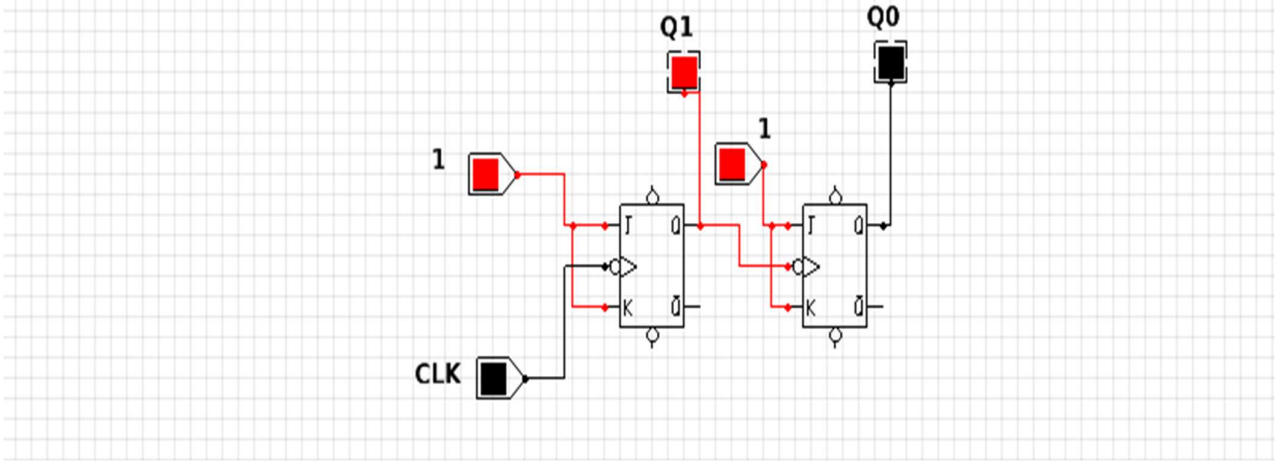


S_2 S_1	OPERATION
0 0	NO CHANGE
0 1	SHIFT RIGHT
1 0	SHIFT LEFT
1 1	PARALLEL LOAD

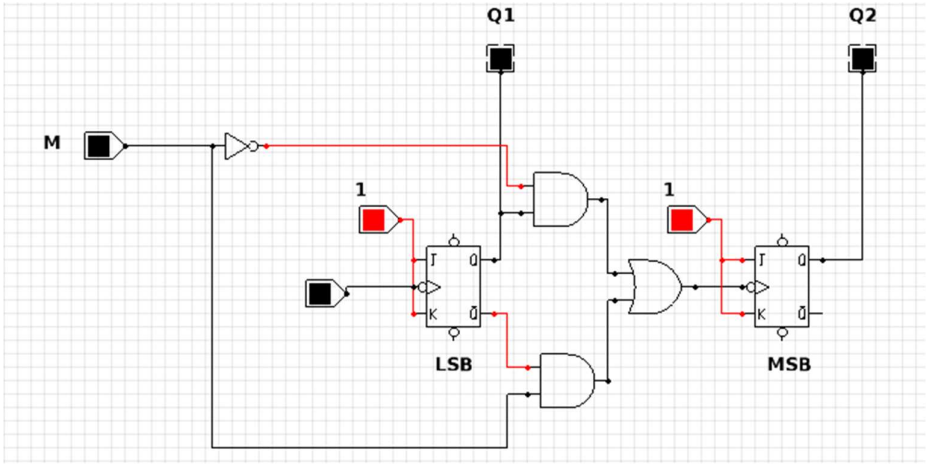
BIDIRECTIONAL SHIFT REGISTER

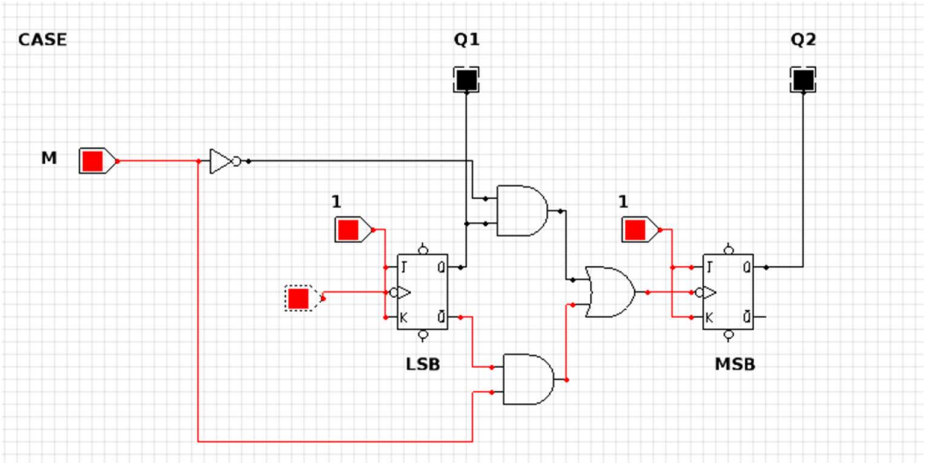


PRACTICAL 9 : DESIGN RIPPLE UP AND RIPPLE DOWN COUNTER AND MOD-N COUNTER USING FF



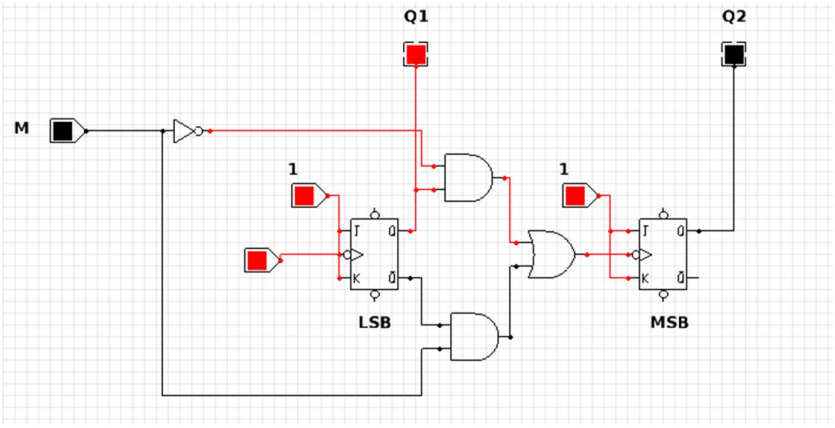
2-BIT UP-DOWN ASYNCHRONOUS COUNTER



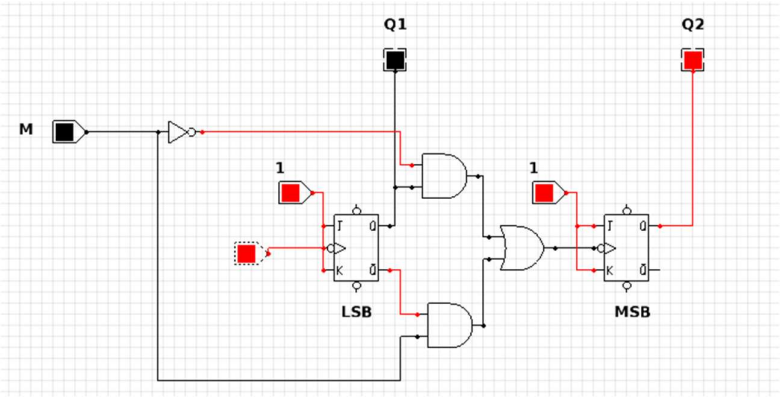


CASE-1 WHEN M=0

M=0 : 1ST CLOCK PULSE :→ Q_2Q_1= 00

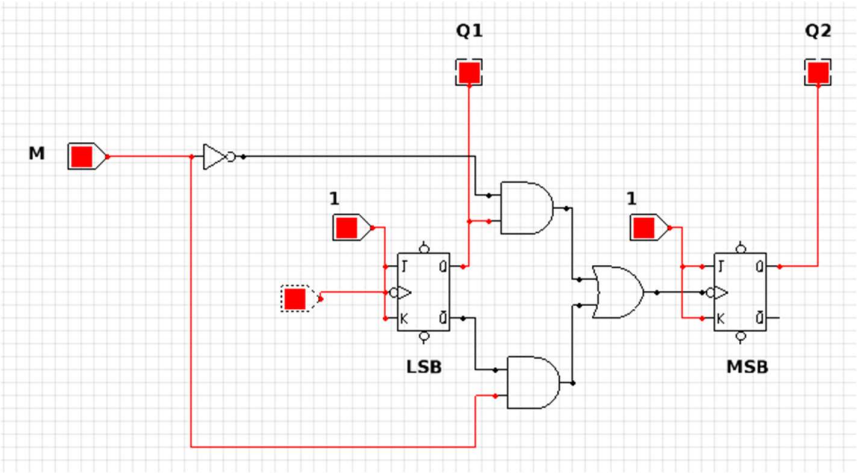


M=0 : 2ND CLOCK PULSE :→ Q_2Q_1= 01



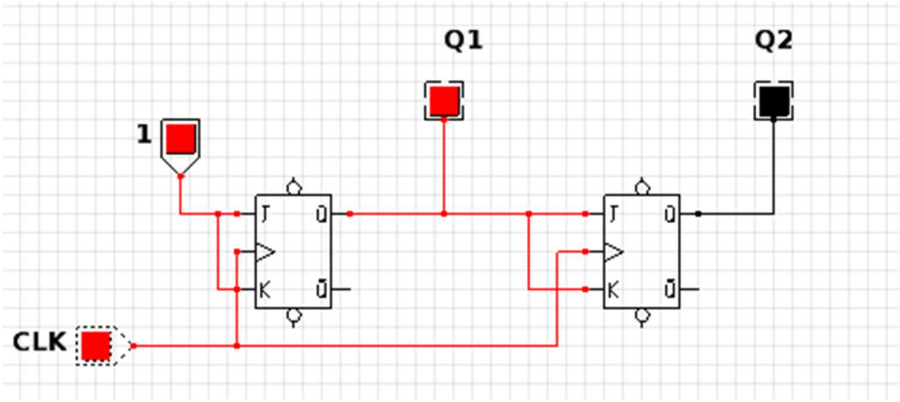
M=0 : 3RD CLOCK PULSE :→ Q_2Q_1= 10

CASE -2 : M=1 : CLOCK PULSE 1: Q_2Q_1= 11

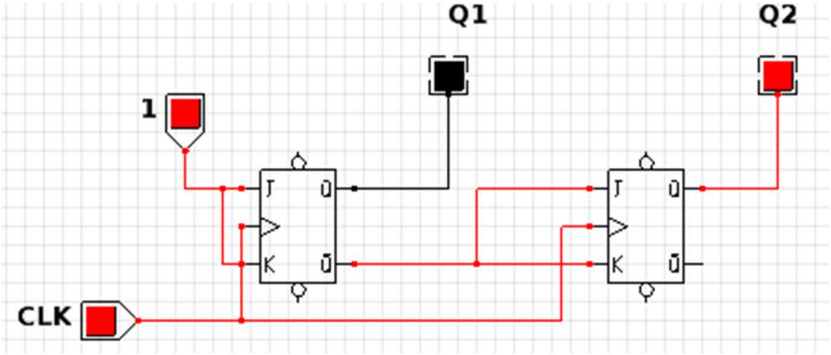


EX. 10 : Design of synchronous up and down counter using flip-flops

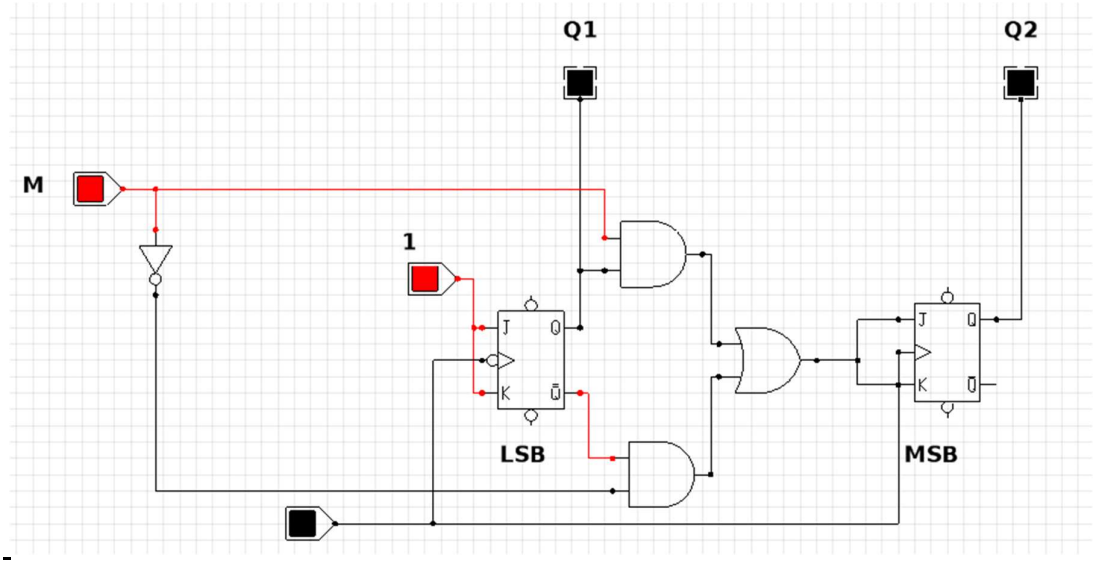
2 -BIT UP SYNCHRONOUS COUNTER



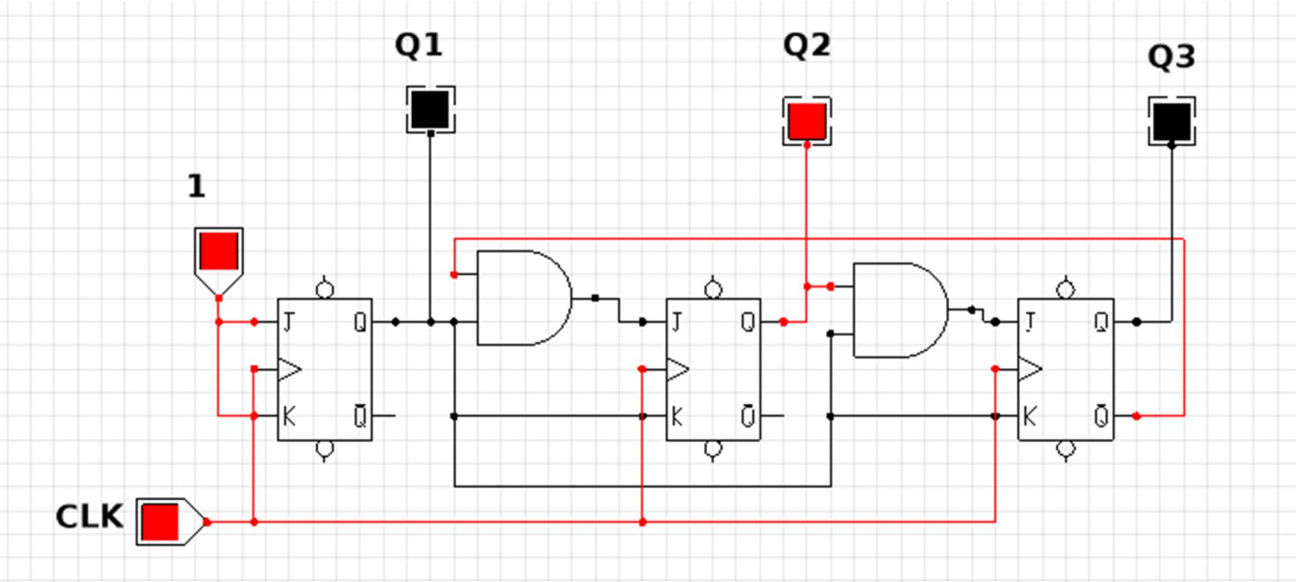
2-BIT DOWN SYNCHRONOUS COUNTER



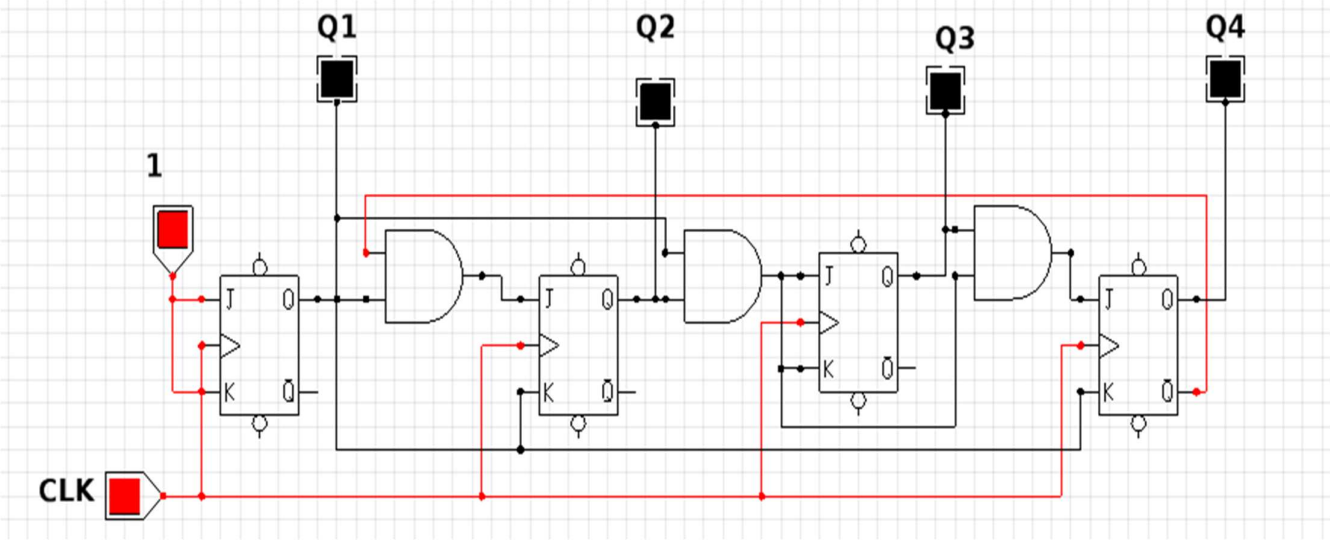
2-BIT UP-DOWN SYNCHRONOUS COUNTER



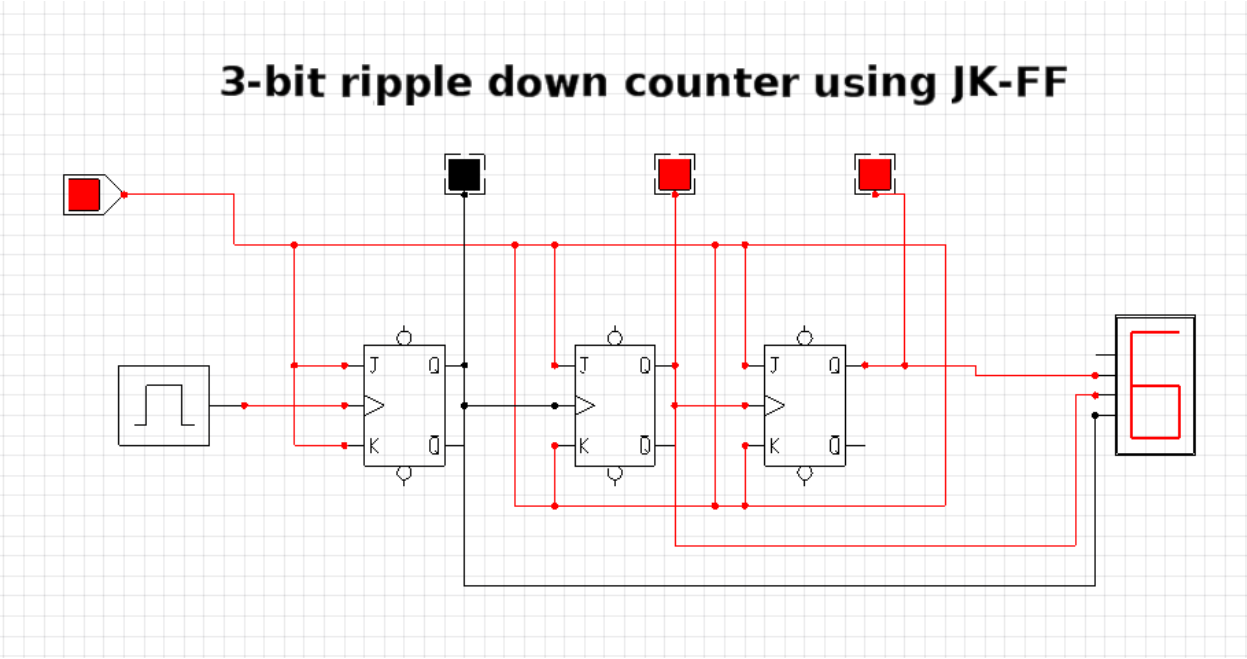
MOD-6 SYNCHRONOUS UP COUNTER



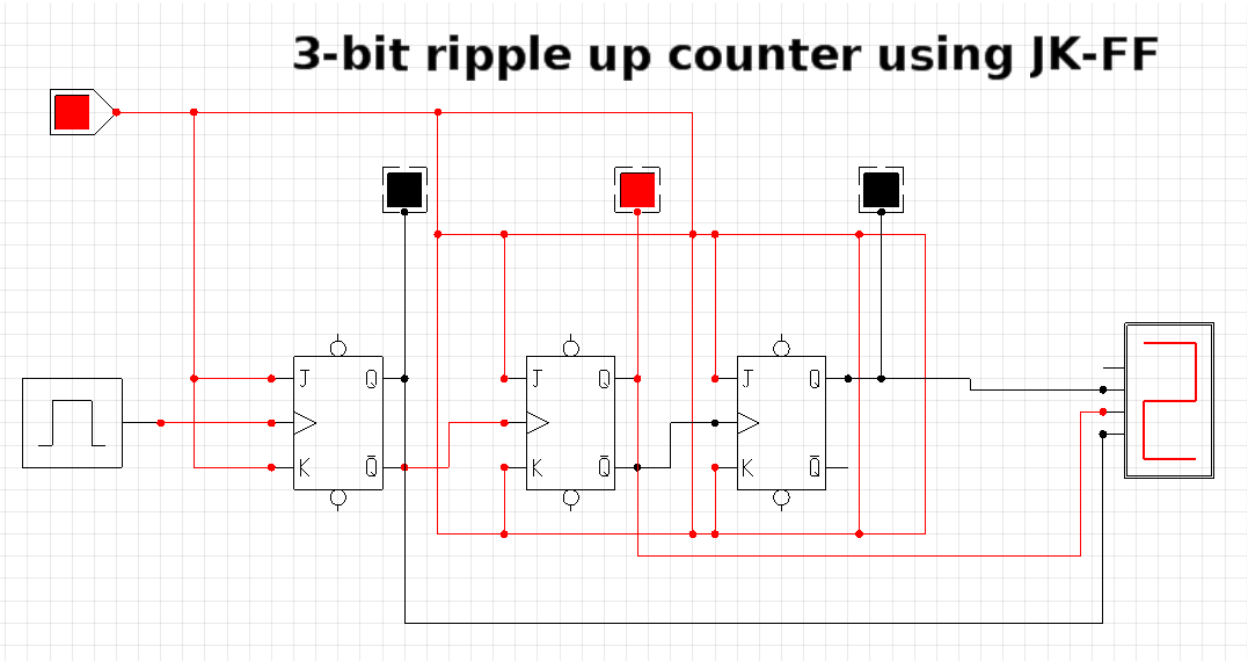
MOD-10 SYNCHRONOUS UP COUNTER



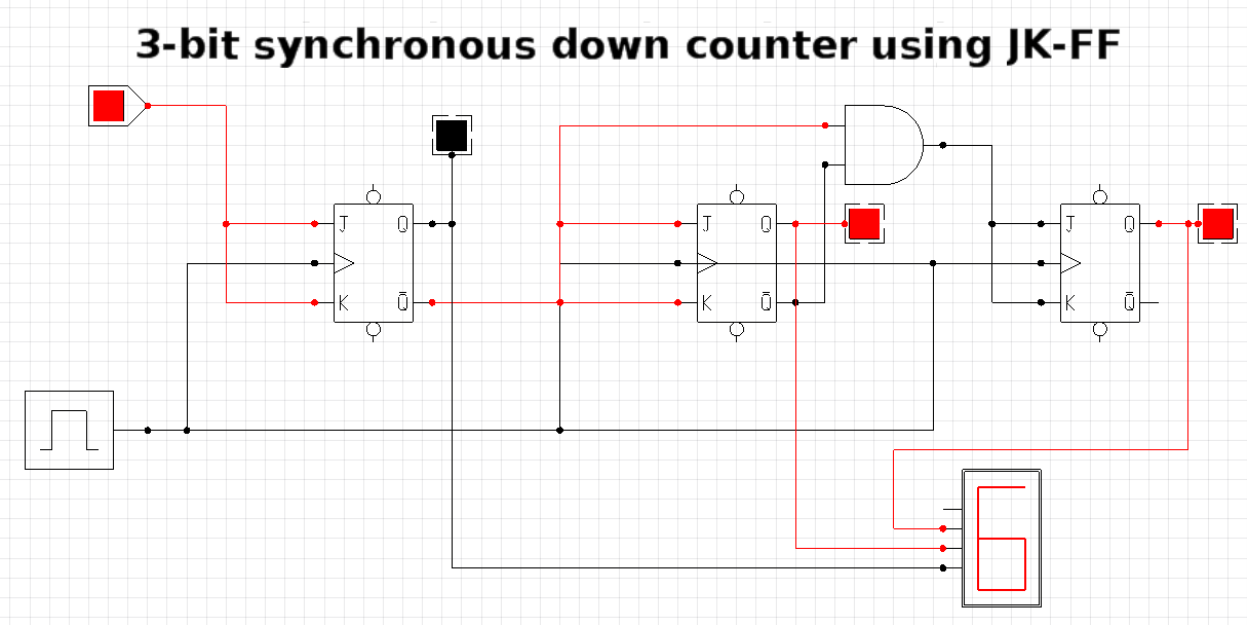
3-bit ripple down counter using JK-FF



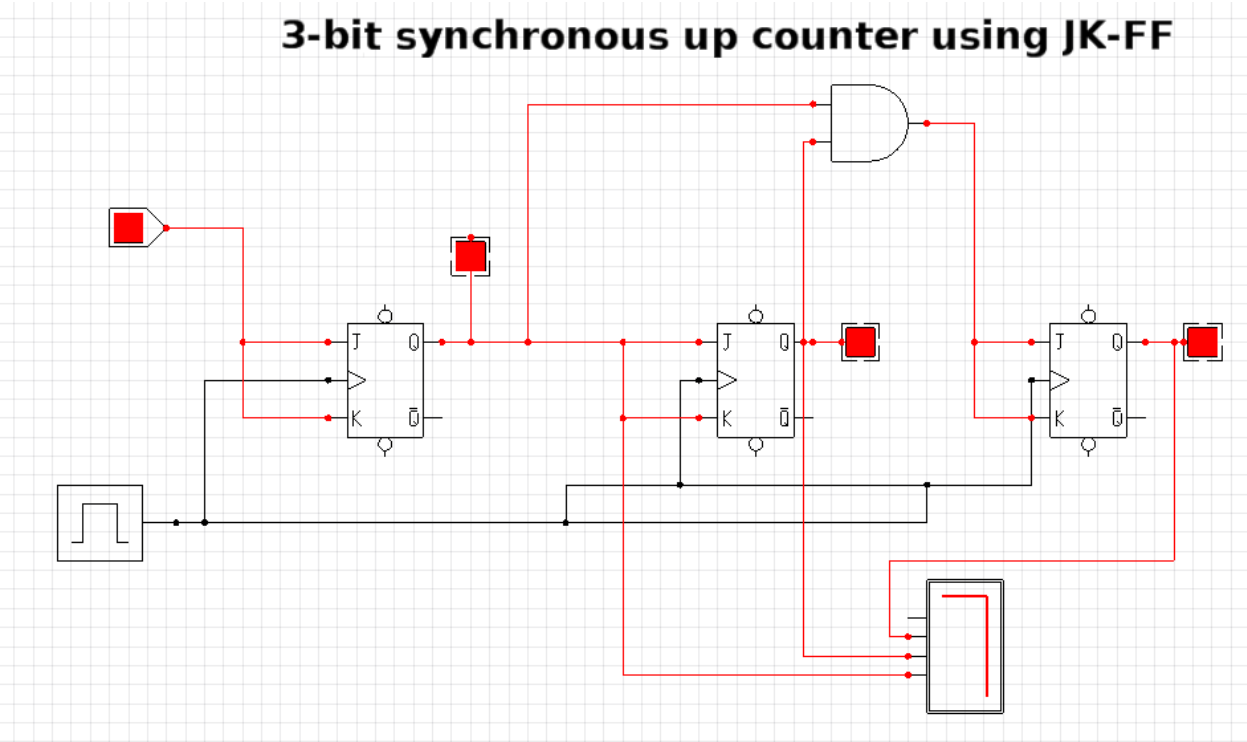
3-bit ripple up counter using JK-FF



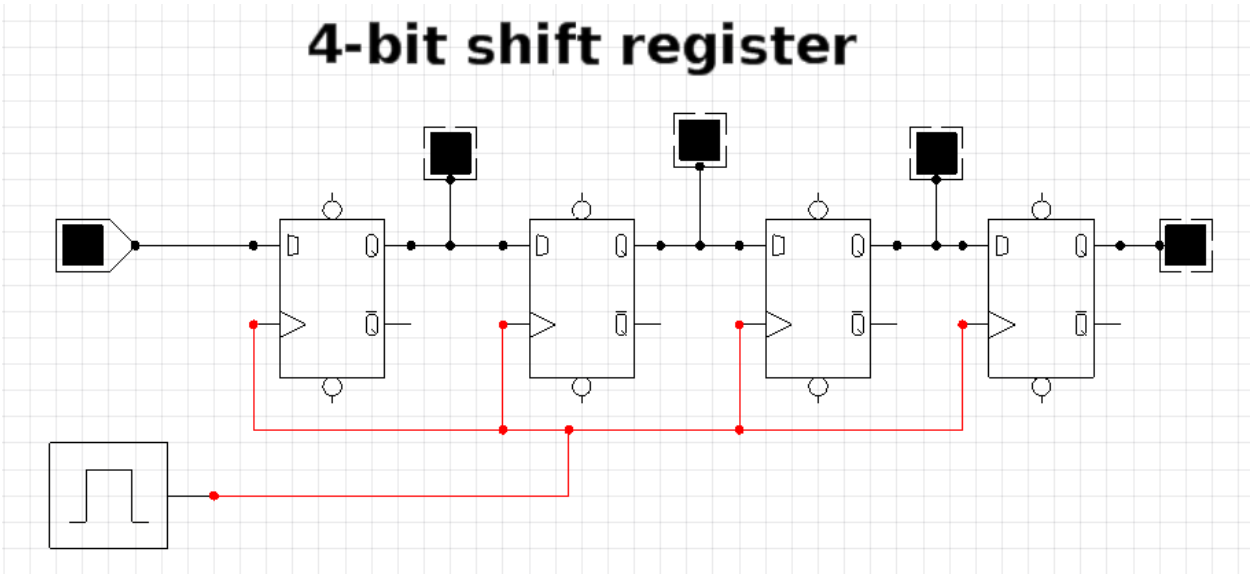
3-bit synchronous down counter using JK-FF



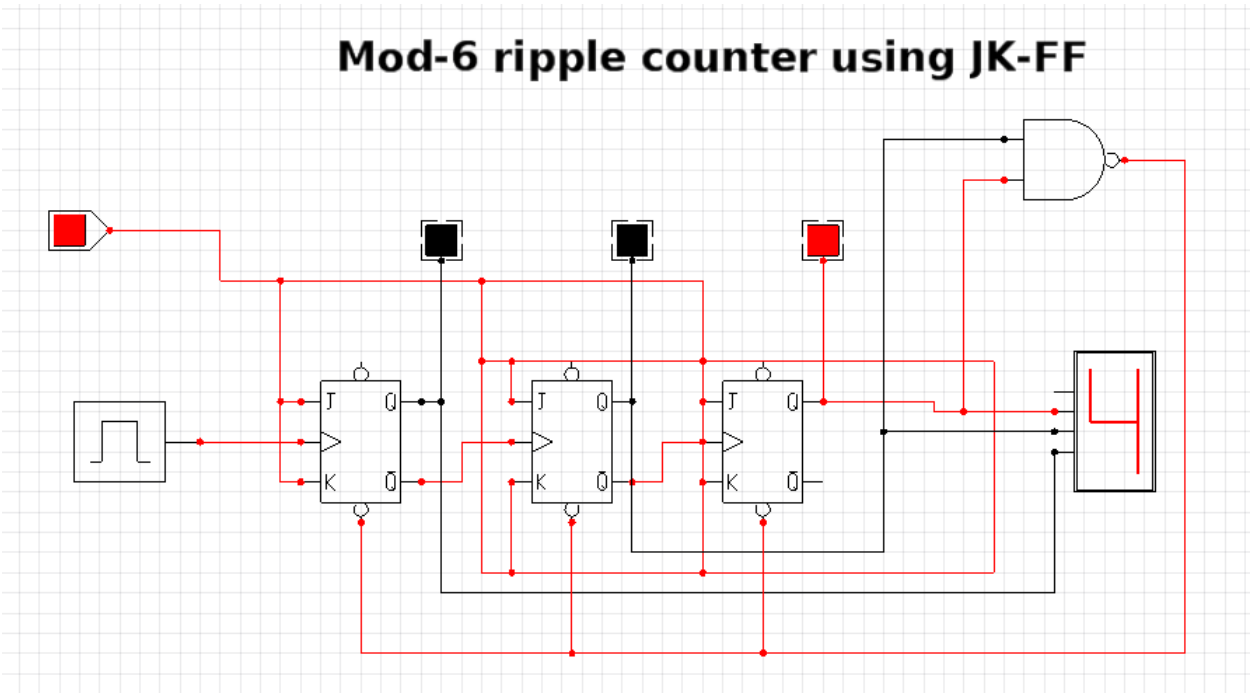
3-bit synchronous up counter using JK-FF

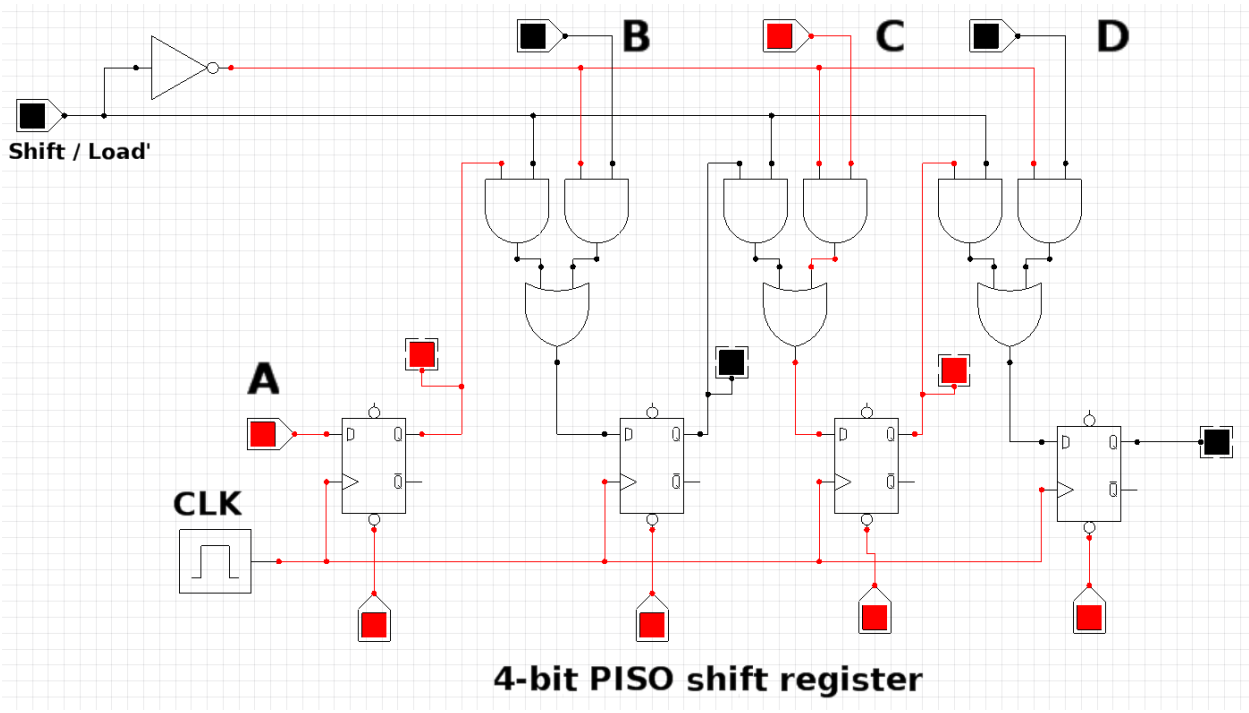


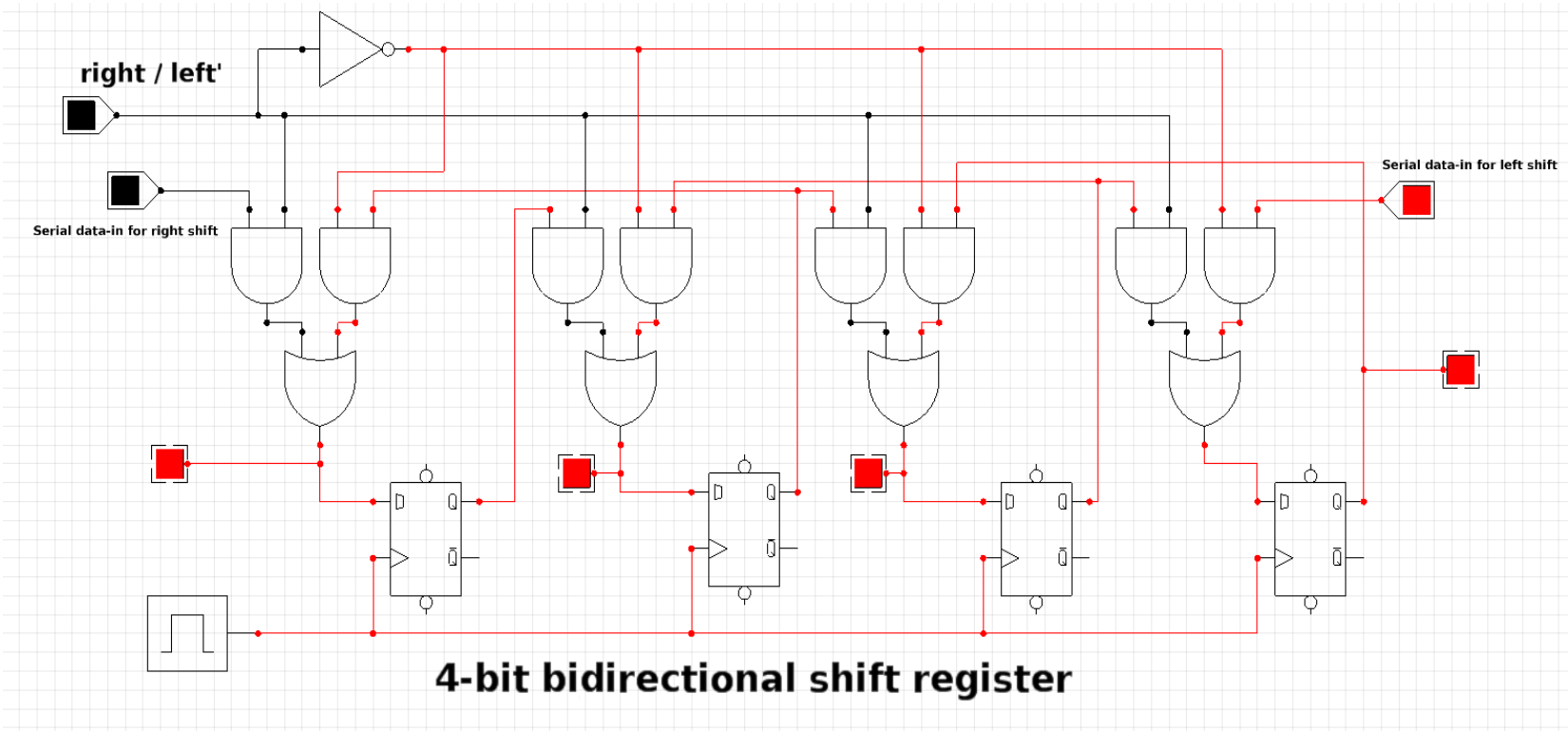
4-bit shift register

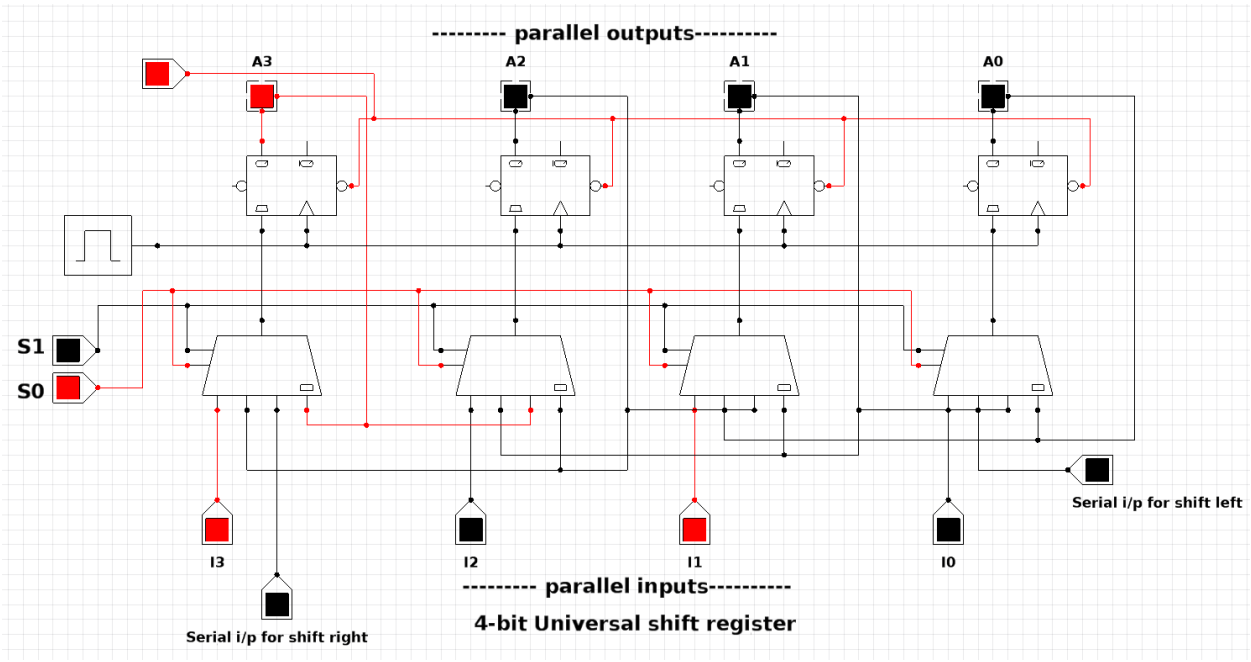


Mod-6 ripple counter using JK-FF




B
C
D
Shift / Load'
A
CLK
4-bit PISO shift register


right / left'
Serial data-in for right shift
Serial data-in for left shift
4-bit bidirectional shift register


--------- parallel outputs----------
A3
A2
A1
A0
S1
S0
Serial i/p for shift left
I3
I2
I1
I0
--------- parallel inputs----------
Serial i/p for shift right
4-bit Universal shift register