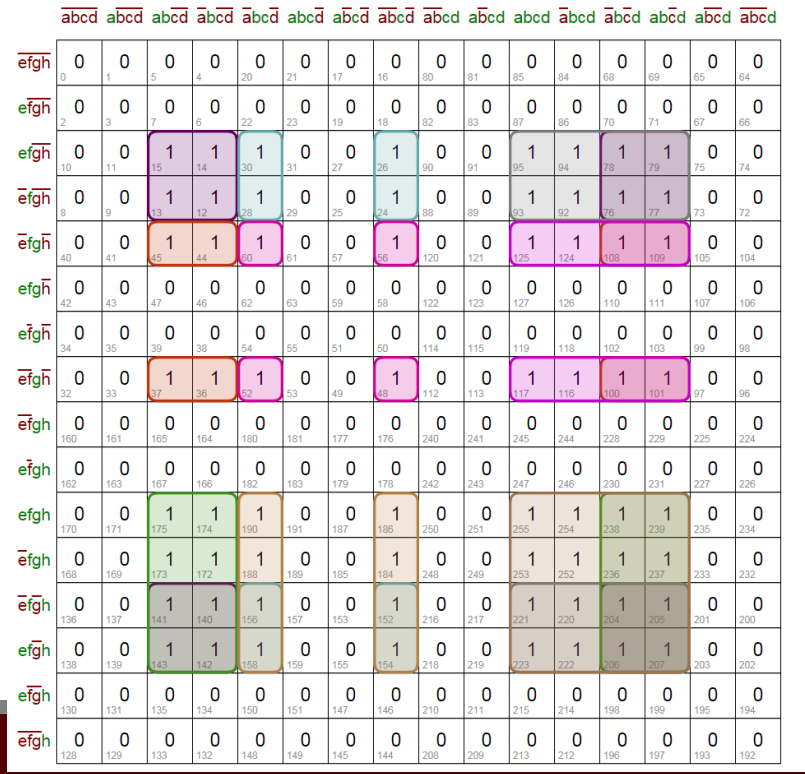
One Hot Encoding

Sequential Design So Far

* Designing circuits from state diagrams has been achieved with binary encoding so far
  - Each state is given a binary number
  - Use state table and k-maps to get simplified logic circuits for next states and outputs

* Limited by the number of total states and inputs
  - K-map becomes too large to handle by hand



One Hot Encoding

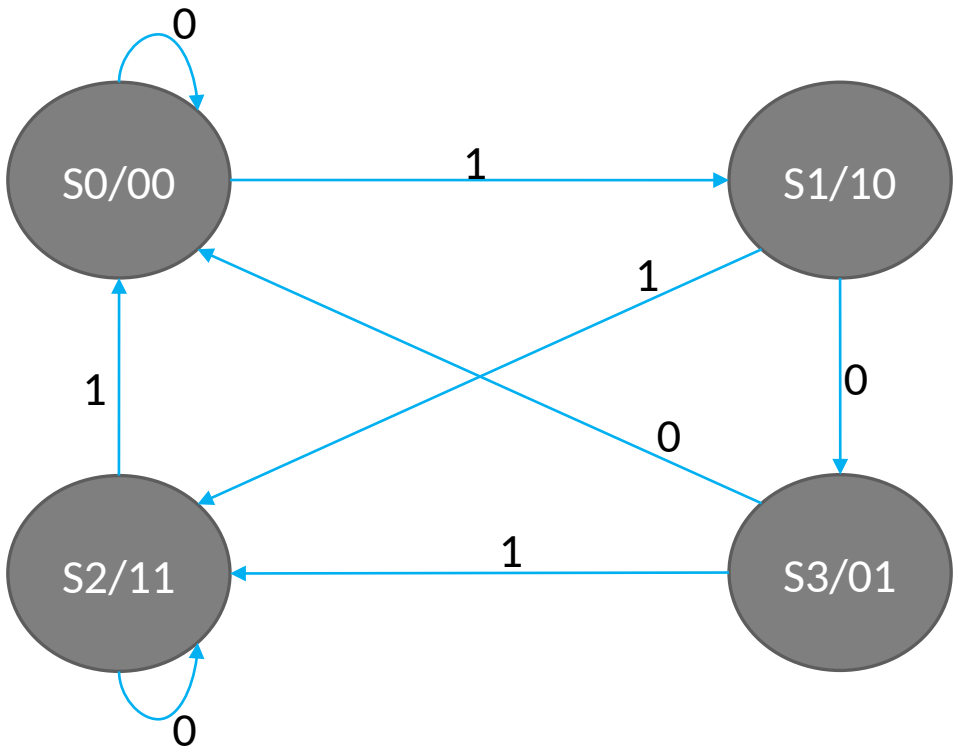
* Instead of assigning each state a binary number, each state is represented as a single binary value.

* Number of bits in the value is the number of states
  - Only one bit can be 1 for each value
  - 1 flip flop for each state
  - One hot encoding uses more flip flops than binary encoding
  - No state table necessary

* For 3 states, the encoding is State 0 ->001, State 1 -> 010, State 2 ->100

Example

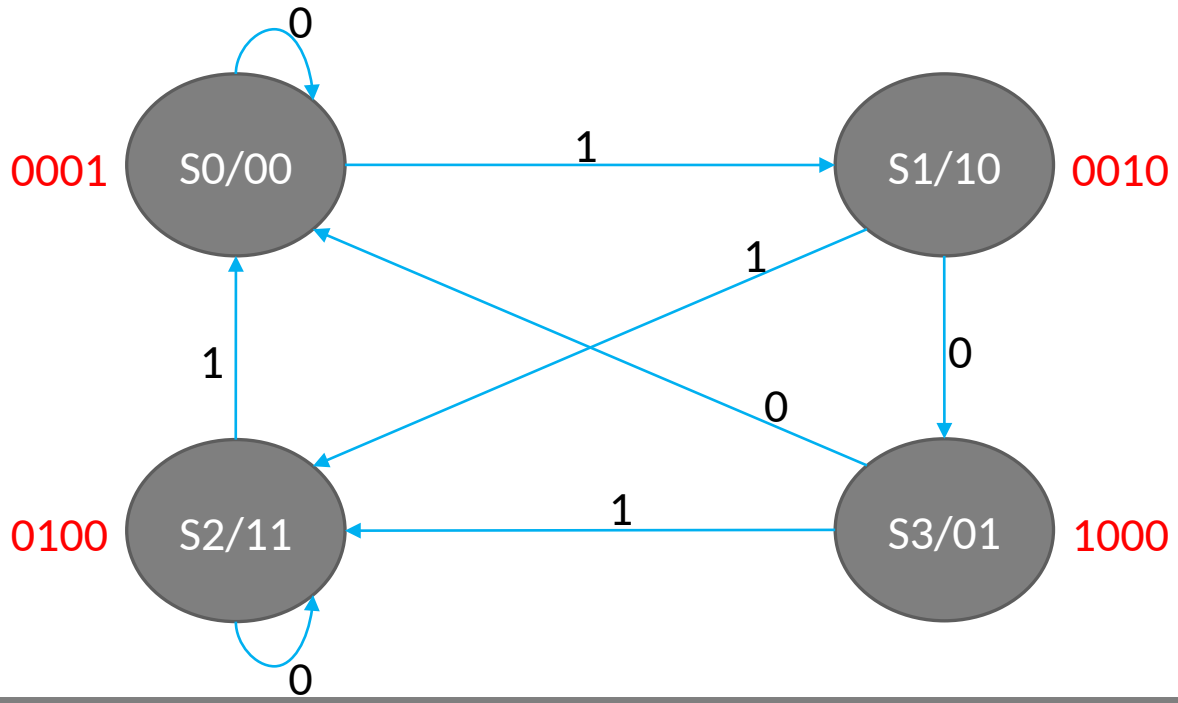
Given the state diagram, encode the states with one hot encoding



Encoding

*Need 4 bits in encoding because 4 states exist

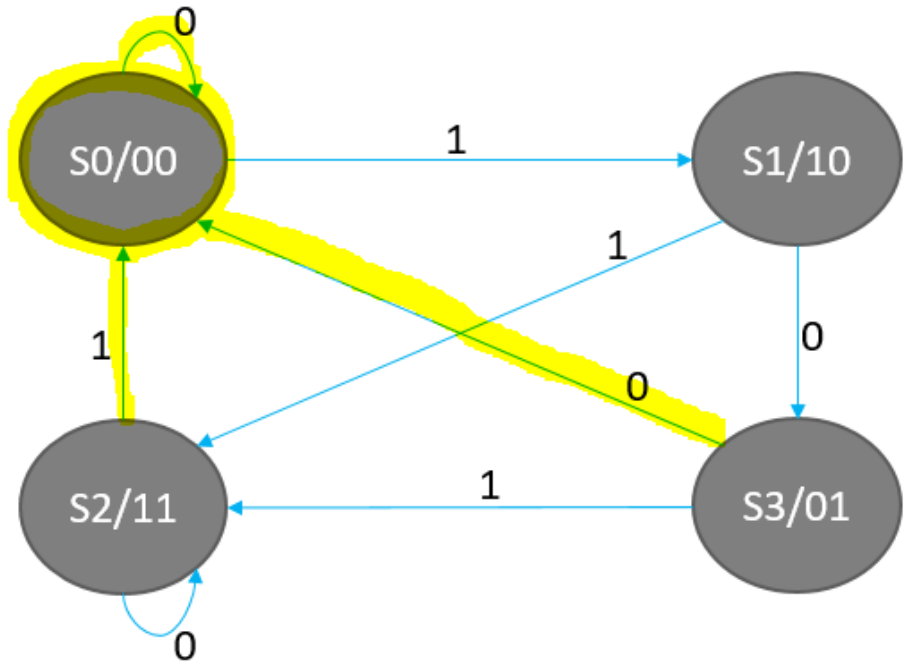
* Each encoded number can only contain a single 1
  - 1 represents output of flip flop. (Will not be using



Next State Logic

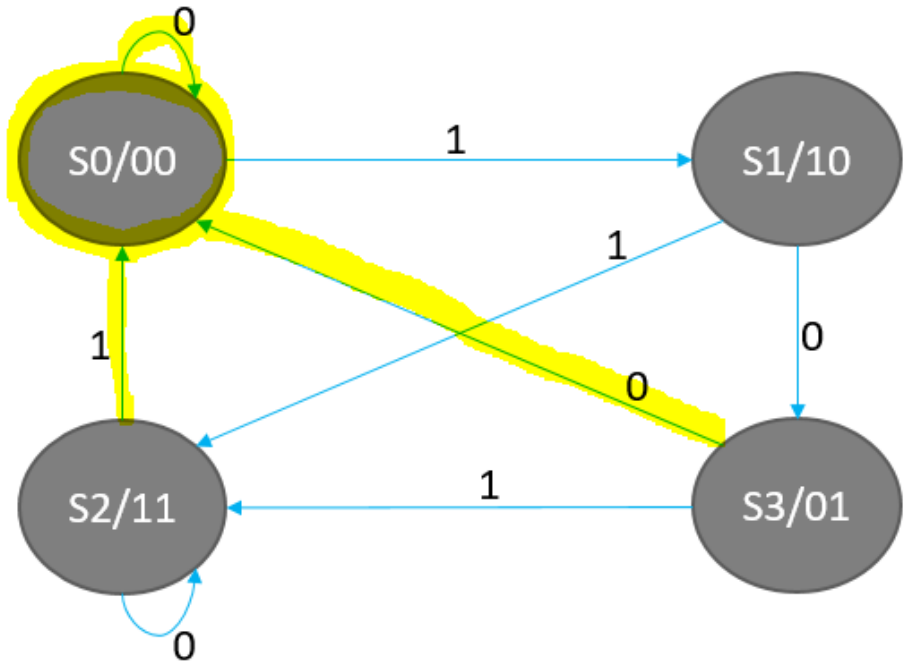
* Do not need state table
* For a given state, look at the transition arrows *into* it, each arrow is an input on an OR gate
  - Connected to the OR gate input is the state coming from AND the input

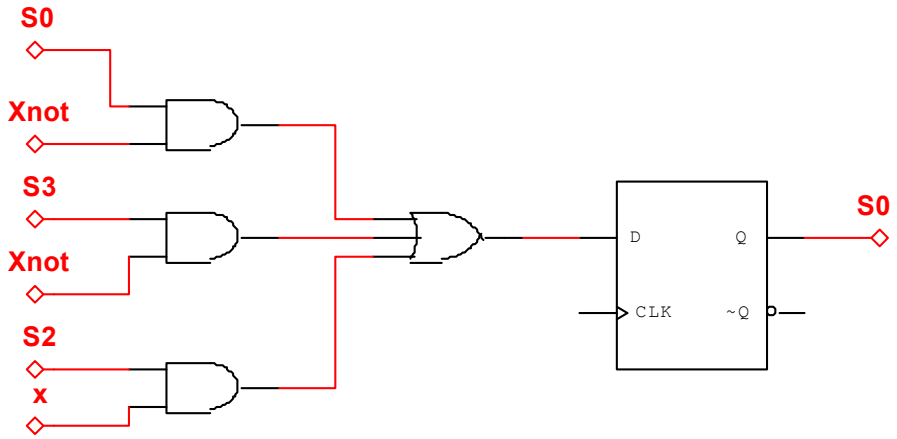
What is the next state logic for S0?



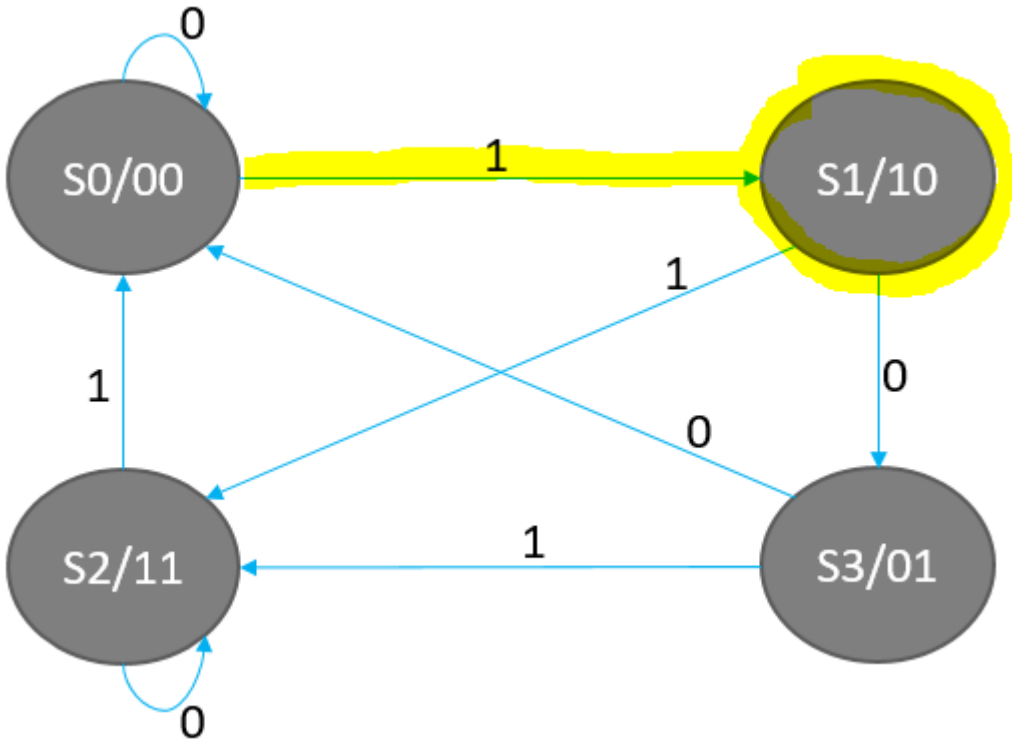
Next State Logic

$$S0 = S0\bar{x} + S3\bar{x} + S2x$$

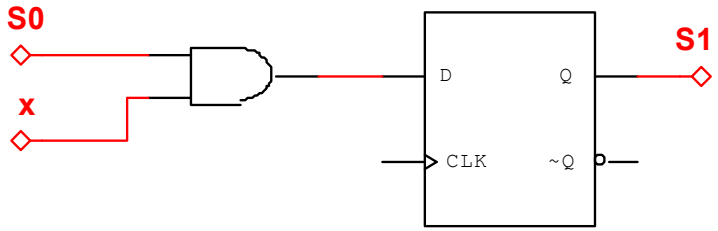




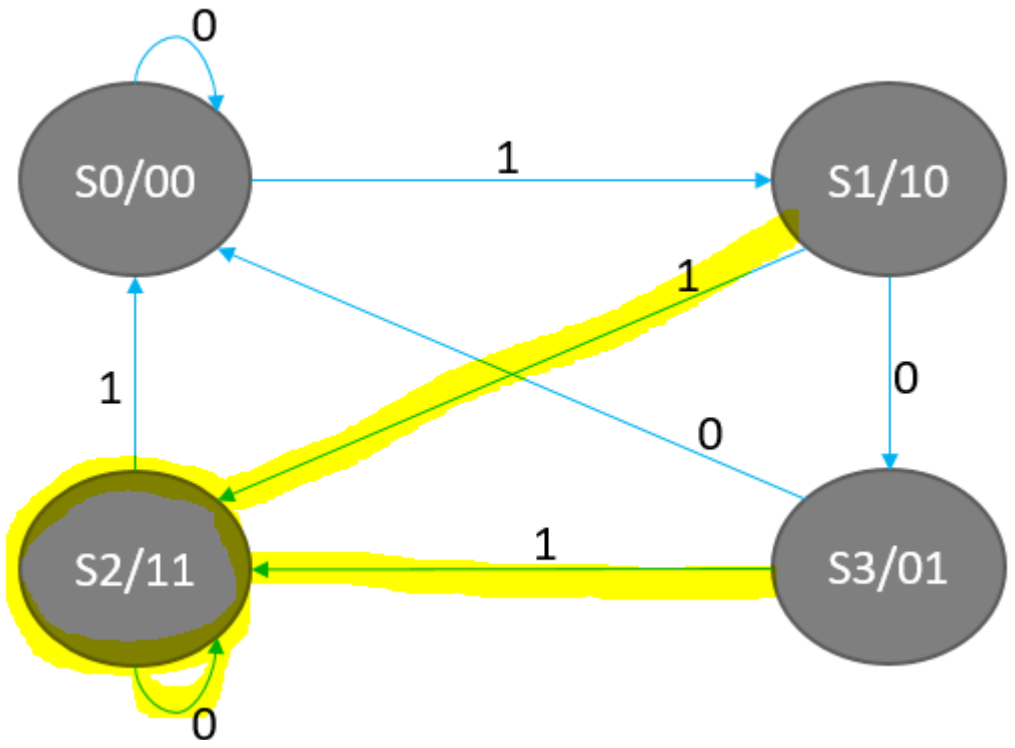
Next State Logic



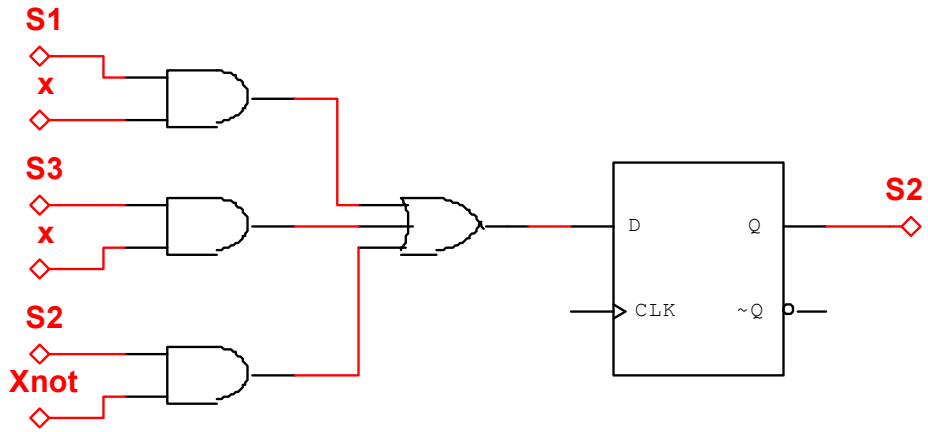
$$S1 = S0\,x$$



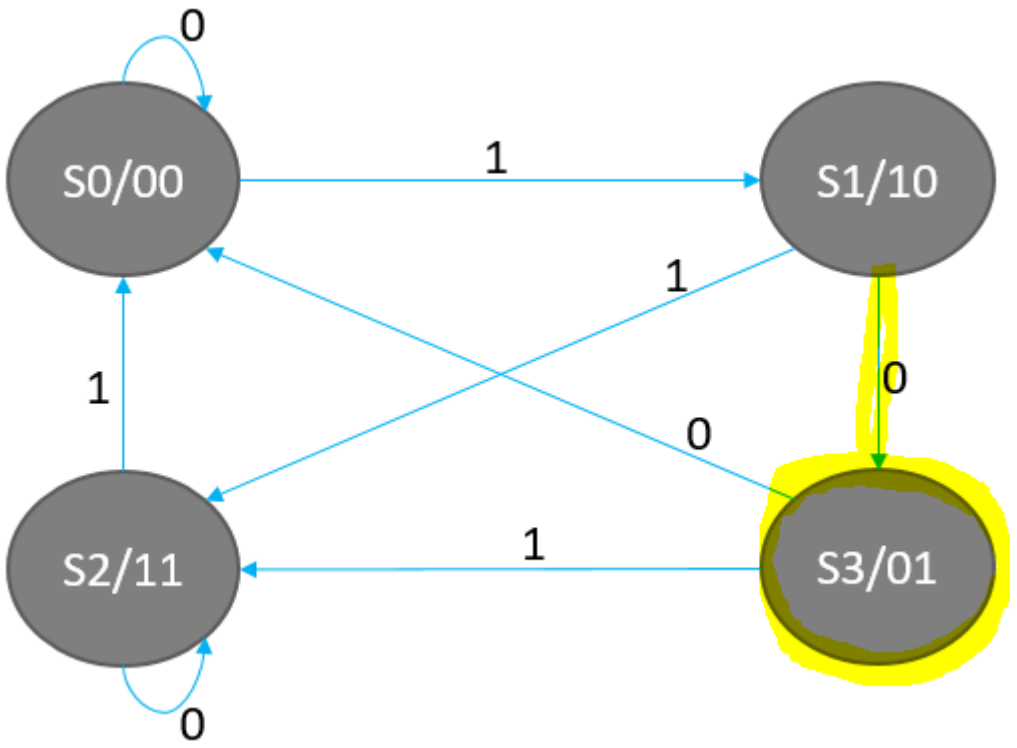
Next State Logic



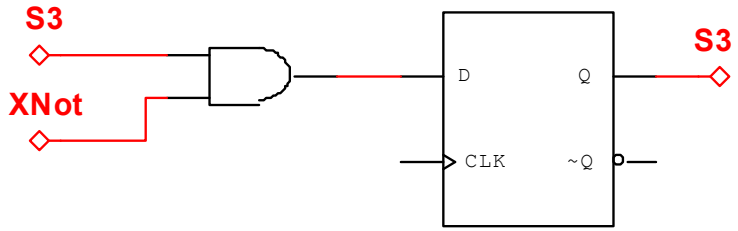
$$S2 = S1x + S3x + S2\bar{x}$$



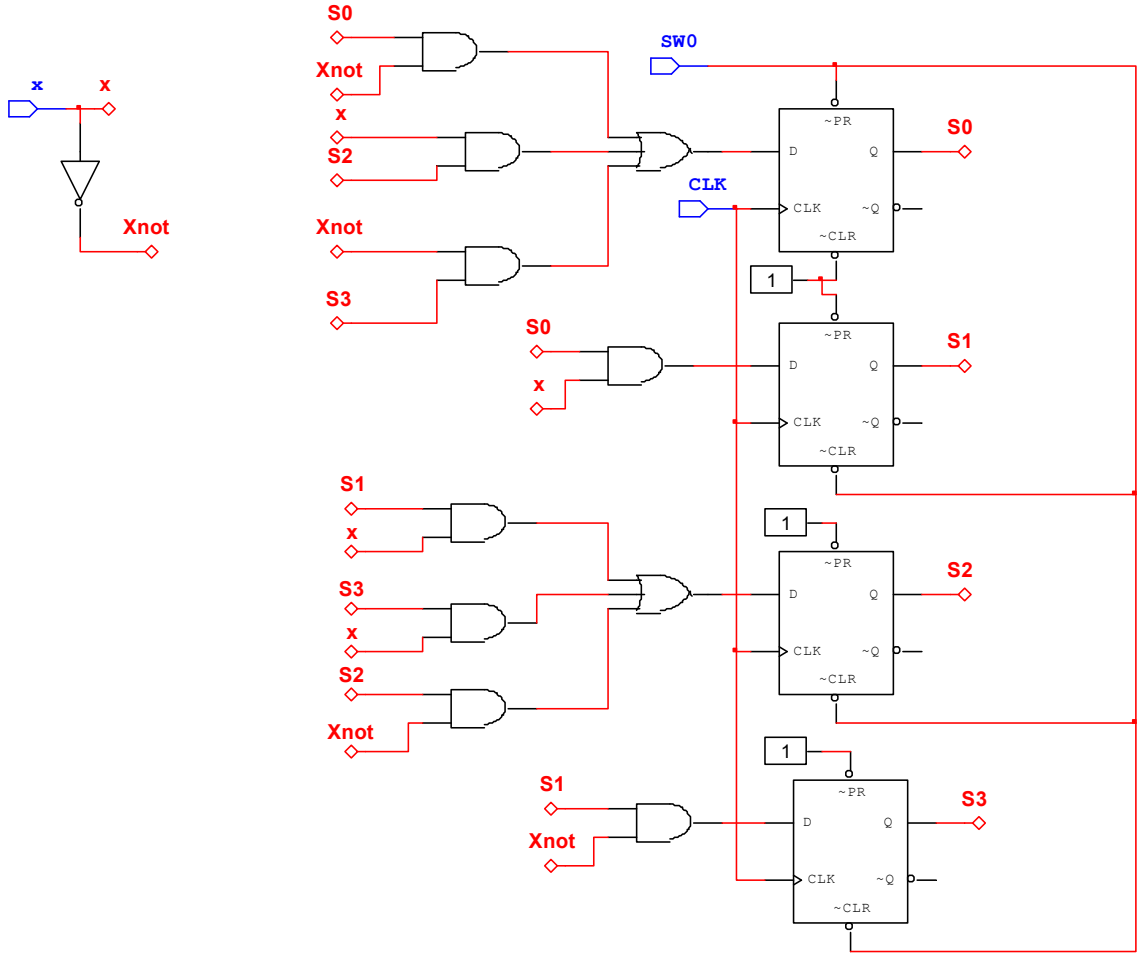
Next State Logic



$$S3 = S1\bar{x}$$

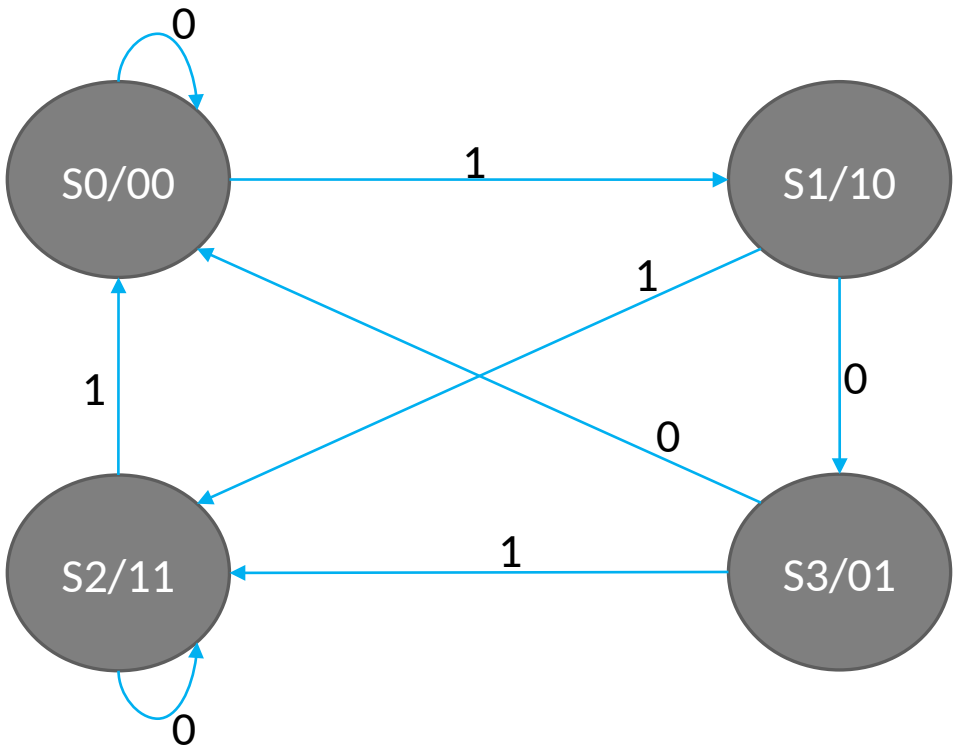


Next State Logic



Output Logic

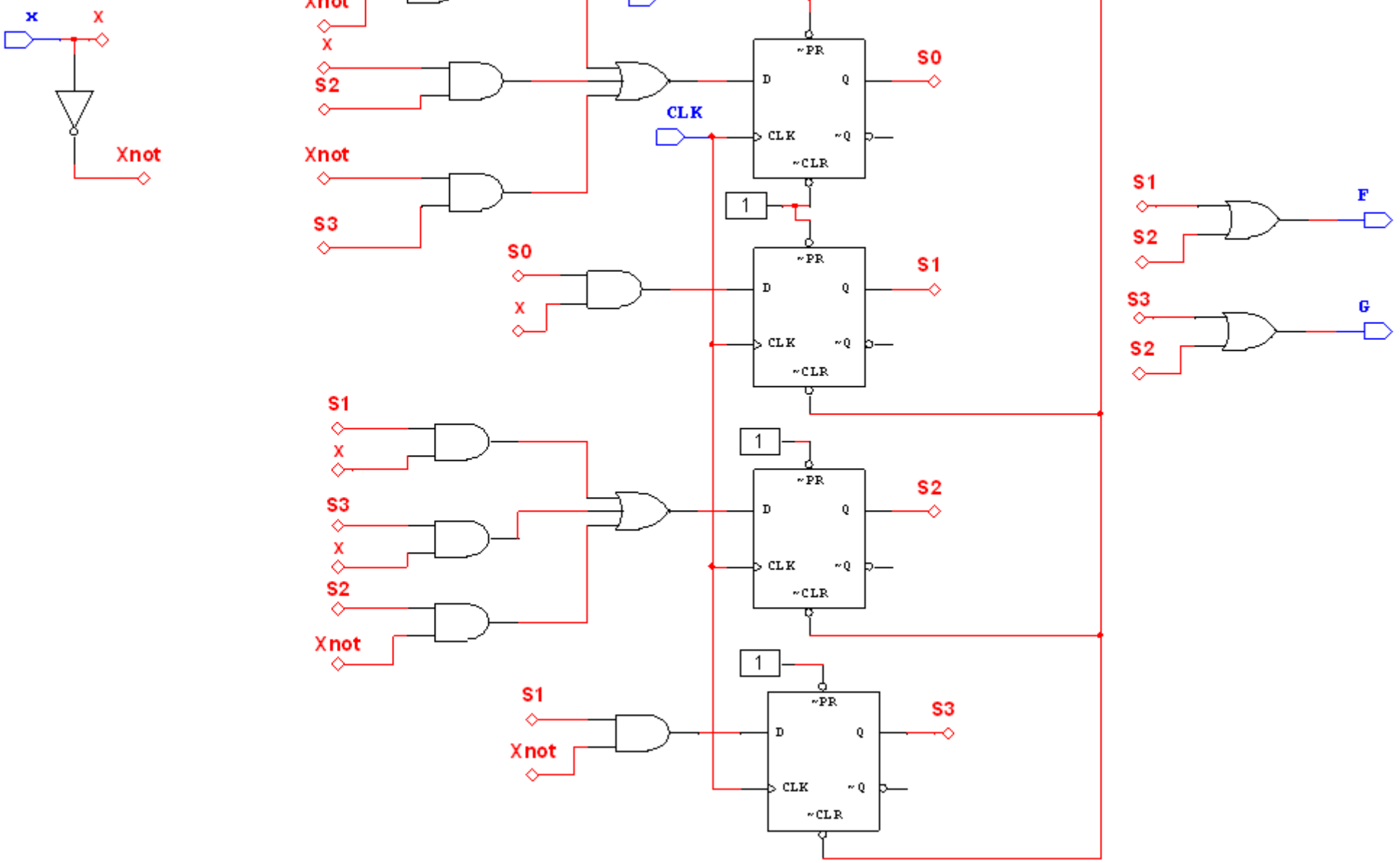
* Look at only the states where the output is 1
  - OR all these states together



F = S1 + S2

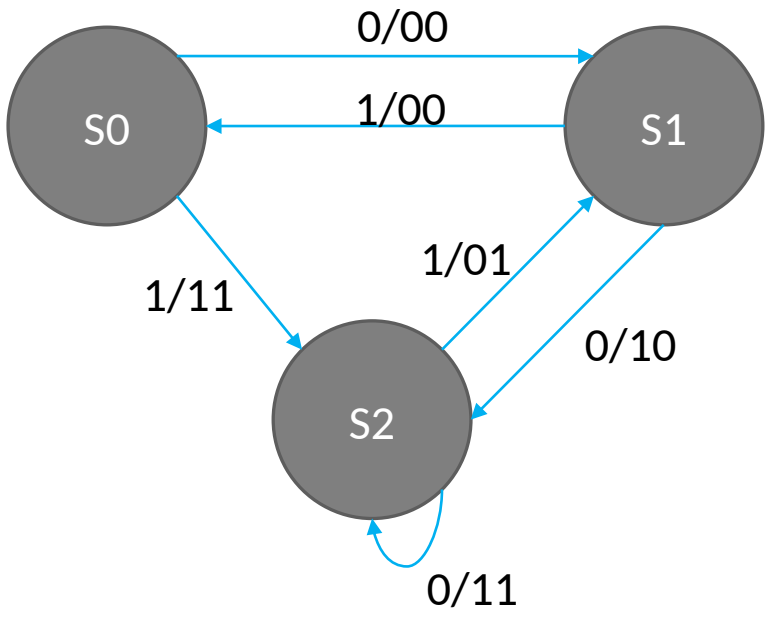
G = S2 + S3

F is MSB output


x
X
Xnot
S0
Xnot
X
S2
Xnot
S3
SW0
CLK
~PR
D
Q
CLK
~Q
~CLR
S0
1
S0
X
S1
S1
X
S3
X
S2
Xnot
S2
1
S1
Xnot
S3
1
S1
S2
F
S3
S2
G

Mealy One Hot

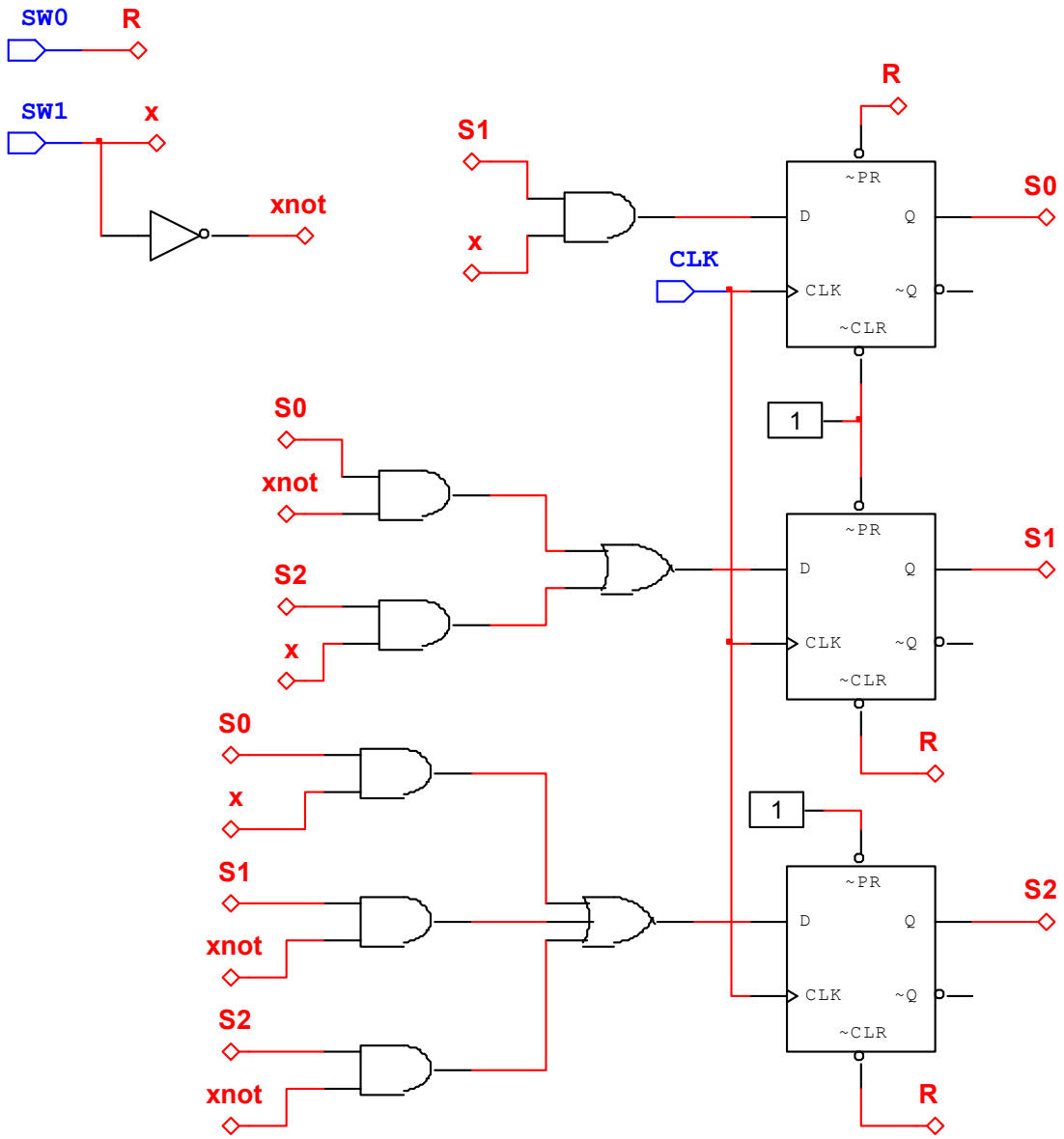
* Next transition logic is the same approach regardless of if state machine is Moore or Mealy



$$S0 = S1x$$

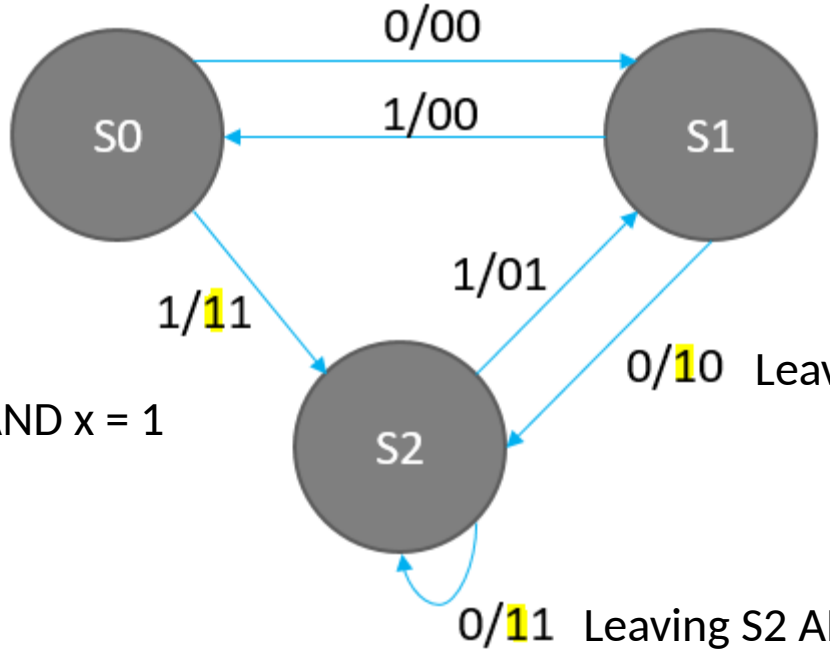
$$S1 = S0\overline{x} + S2x$$

$$S2 = S2\overline{x} + S0x + S1\overline{x}$$


SW0
R
SW1
x
xnot
S1
x
R
~PR
D
Q
S0
CLK
CLK
~Q
~CLR
1
S0
xnot
S2
x
~PR
D
Q
S1
CLK
~Q
~CLR
R
S0
x
1
S1
xnot
S2
xnot
~PR
D
Q
S2
CLK
~Q
~CLR
R

Mealy Output

* Find where output is 1 on transition arrow
  - Output = (state A leaving AND input) + (state B leaving AND input)….



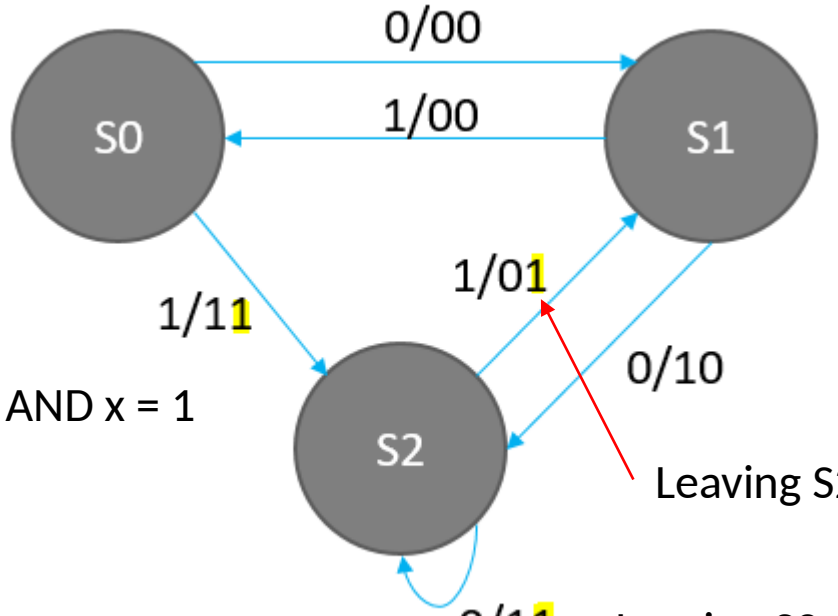
Leaving S0 AND x = 1

Leaving S1 AND x = 0

Leaving S2 AND x = 0

$$F = S0x + S1\bar{x} + S2\bar{x}$$

Mealy Output



Leaving S0 AND x = 1

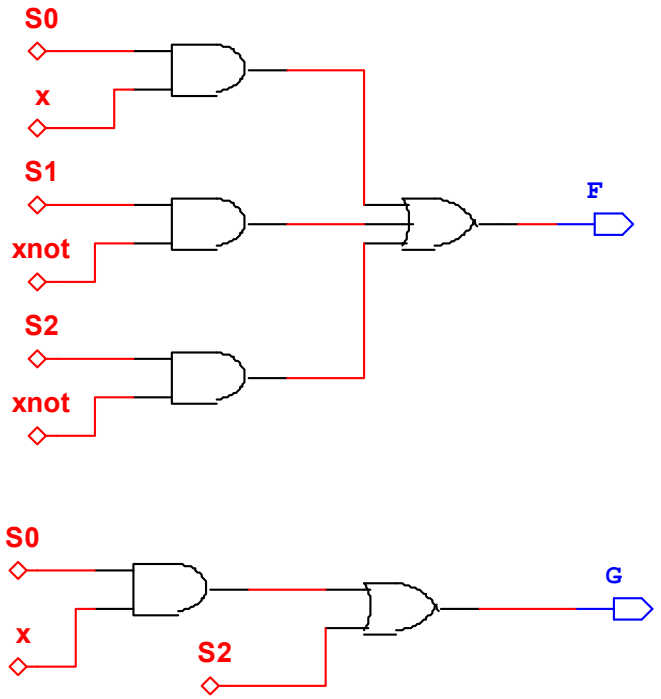
Leaving S2 AND x = 1

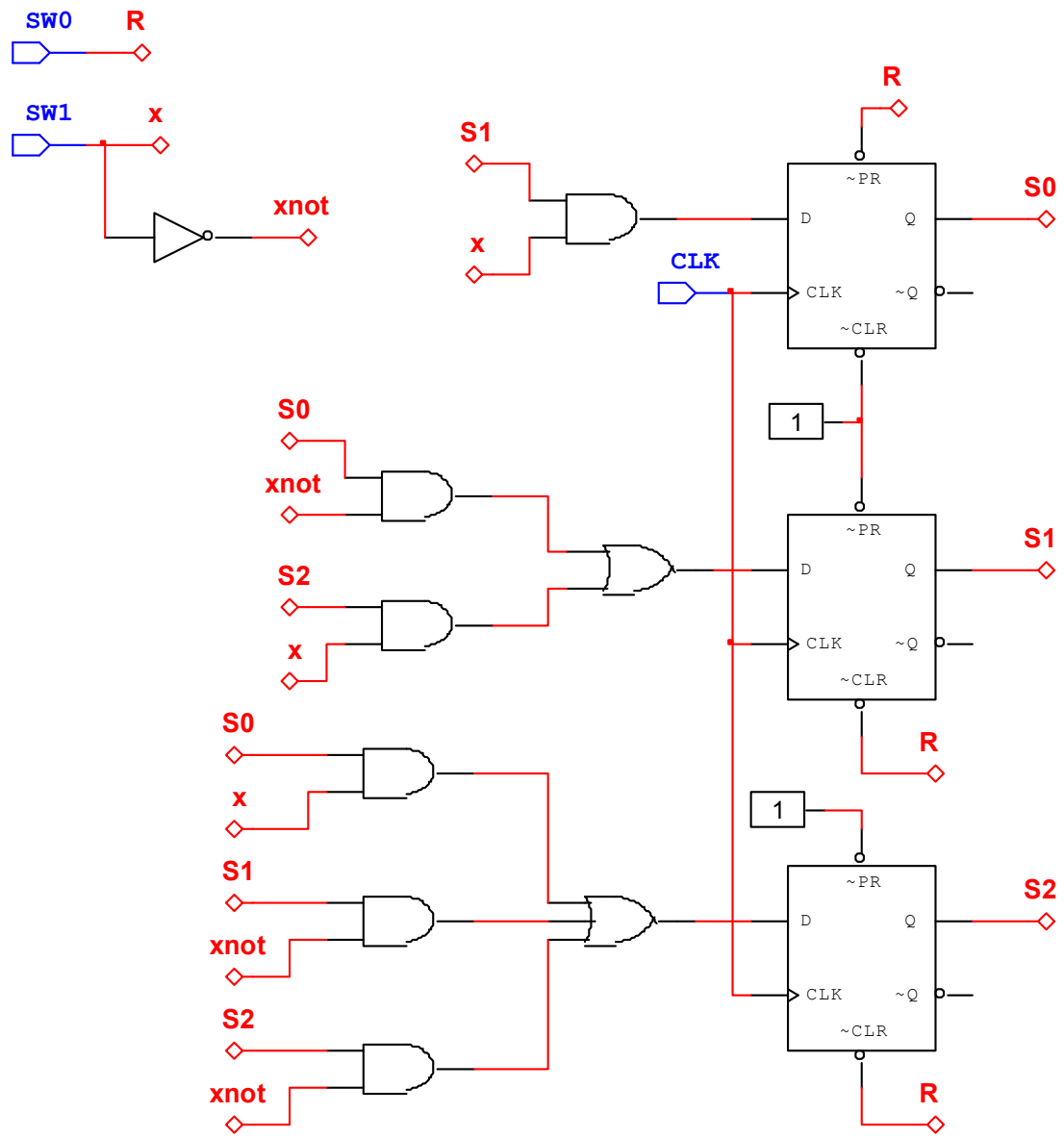
Leaving S2 AND x = 0

$$G = S0\,x + S2\,x + S2\,\bar{x}$$

Output One Hot

* Output logic may be *unsimplified*. Use Boolean algebra to reduce





Why is F not just connected to S2?