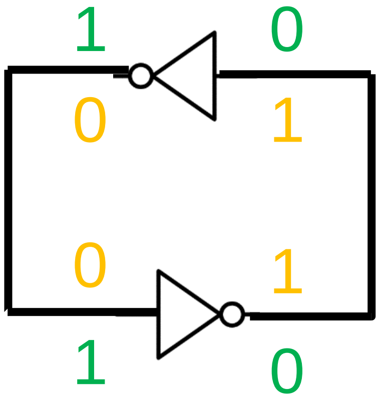
SEQUENTIAL CIRCUITS

- Principles
- Storage mechanism

A FIRST CIRCUIT WITH MEMORY CAPABILITY (1)

Two inverters connected in a loop.
ATTENTION: This is not a WF circuit, this not a combinational circuit

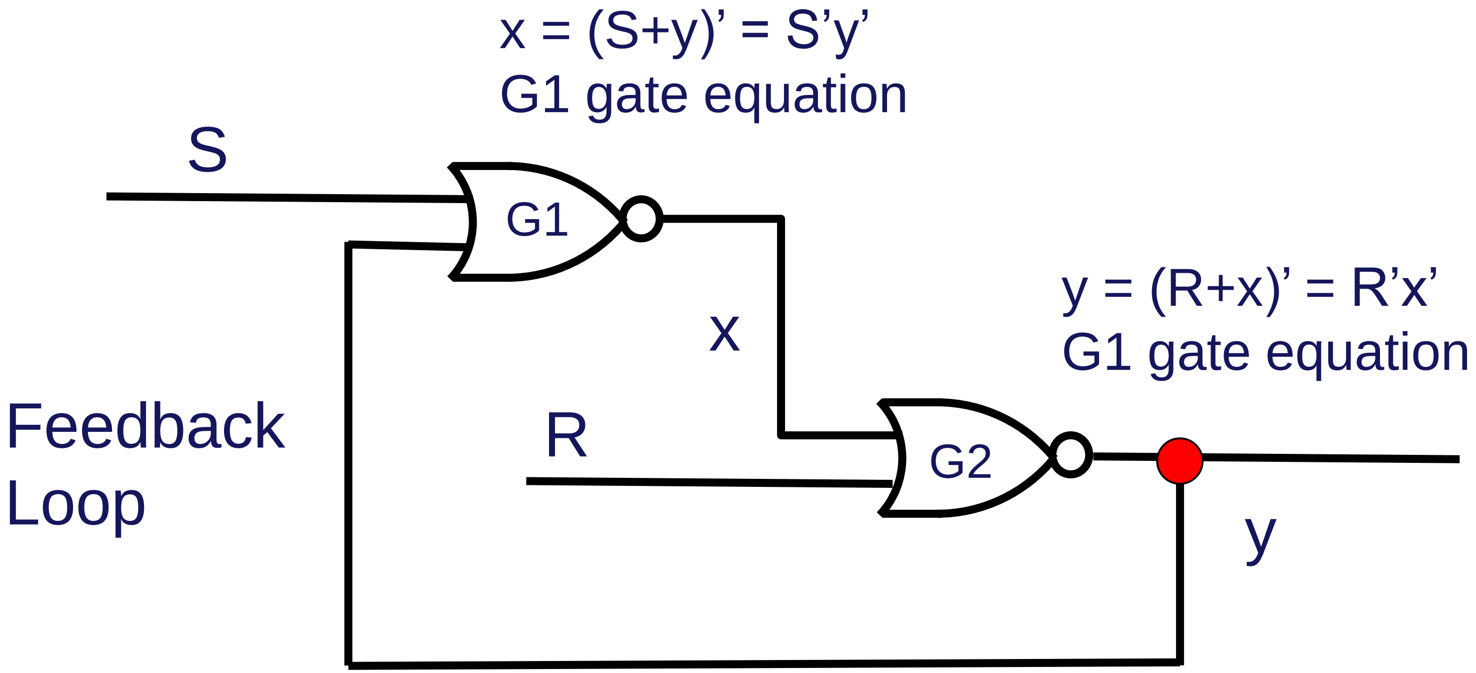
First Configuration: valid according to boolean algebra



Second Configuration: valid according to boolean algebra

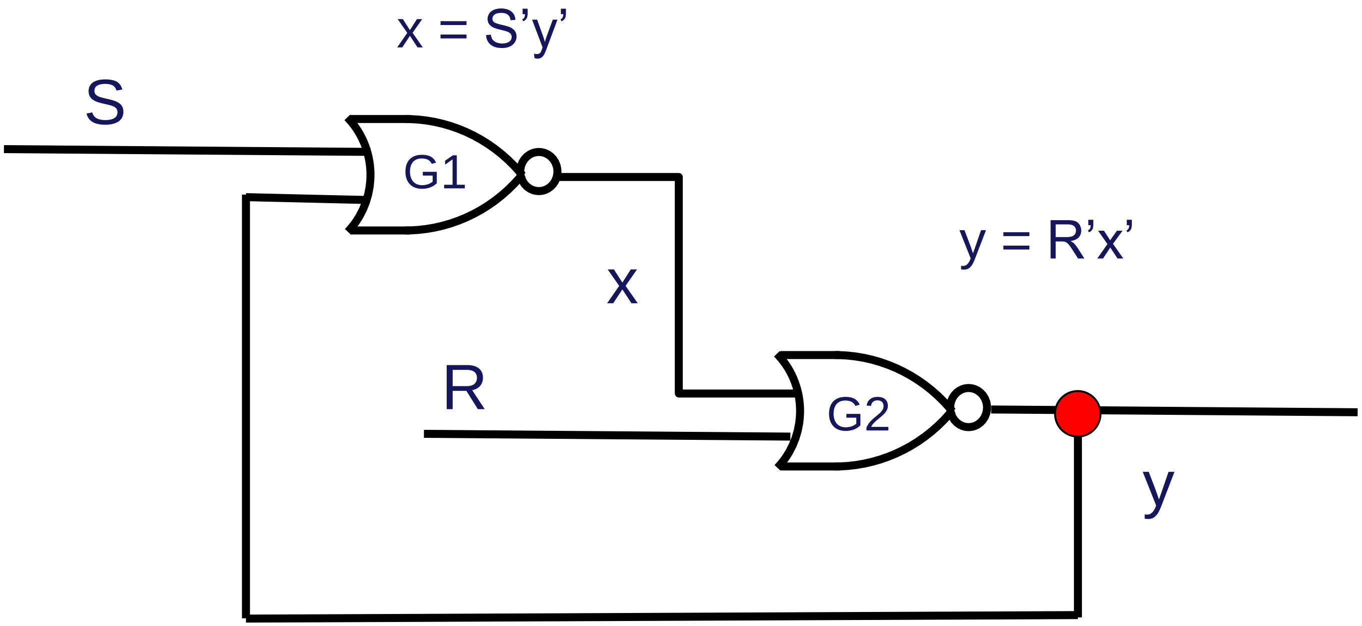
Two stable configurations: Two stable STATES
Irrealistic circuit: Input ?? Output ??
FUNDAMENTAL QUESTION: How does the circuit end up in one state or the other one ??

A STRANGE GATE ORGANIZATION



ATTENTION:
This is not a WF circuit.
This is not a combinational circuit

QUICK SUMMARY OF A STRANGE CIRCUIT



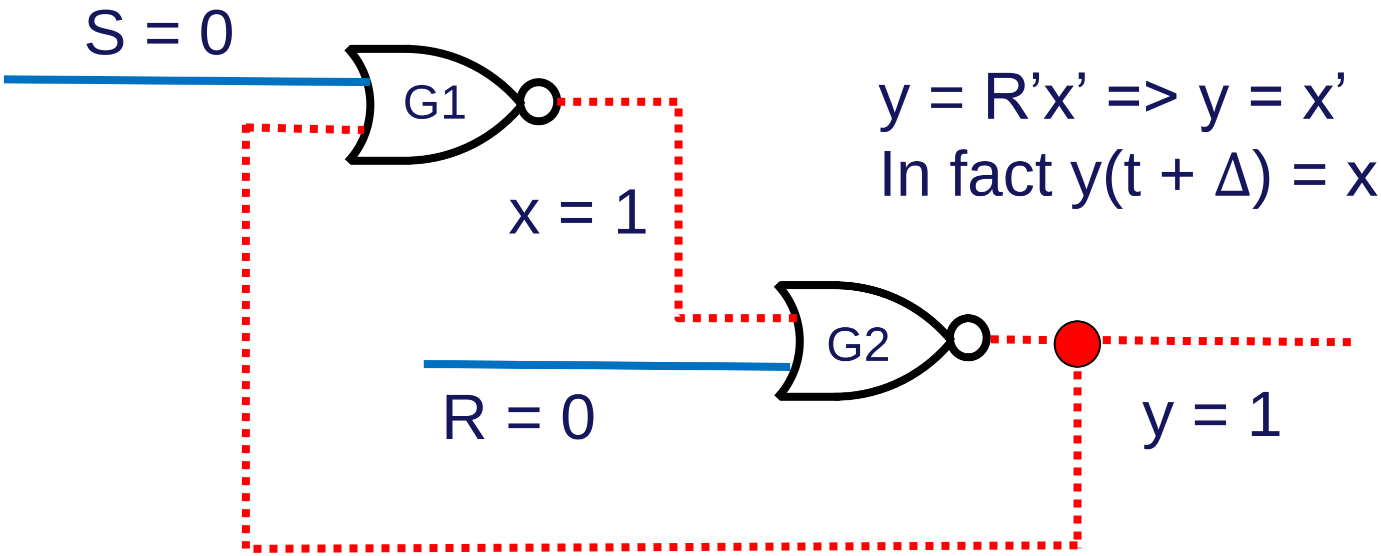
S	R	x	y
1	0	0	1
0	1	1	0
1	1	0	0
0	0	0	1
0	0	1	0

Two issues:

1. Command (0,0) gives unknown outputs
2. For all commands except (1,1), y = x'

(S = 1 AND R = 1) THEN (S = 0 AND R =0) (3)

x = S’y’ => x = y’ In fact x(t + Δ) = y’(t)



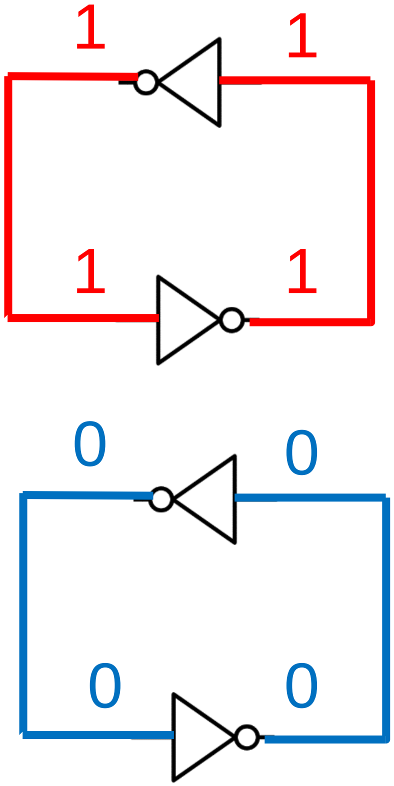
y = R’x’ => y = x’
In fact y(t + Δ) = x’(t)

T = 2 Δ after R value change
Snapshot right before T = 2 Δ
Outputs of gates G1 and G2 are represented with dotted lines because these values are due to gate delays

(S = 1 AND R = 1) THEN (S = 0 AND R = 0) (4)

The two NOR gates with inputs S = 0 and R = 0 behave finally like inverters .
Two inverters connected in a loop.

Unstable : it is going to oscillate periodically between the two « configurations » on the right….



SEQUENCE OF COMMANDS CONCLUSION

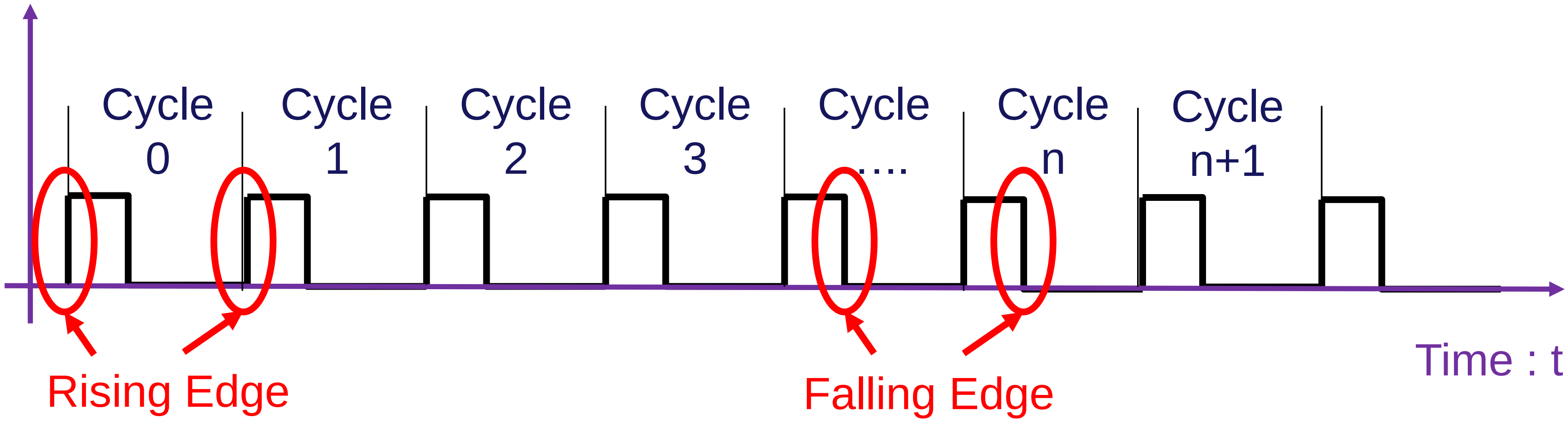
- Any command S, R followed by S=1, R=0 will result in y=1 and x=0
- Any command S, R followed by S=0, R =1 will result in y=0 and x=1
- Any command S, R followed by S=1, R =1 will result in y=0 and x=0

This is due to the boolean equations of the NOR gates (depending in fact upon a single input)

- Now let us have a look at the case where the second command is S = 0, R = 0. 4 subcases
  - S=1, R=0 followed by S=0, R=0 => y=1 and x=0
  - S=0, R=1 followed by S=0, R=0 => y=0 and x=1
  - S=0, R=0 followed by S=0, R=0 => prolongation
  - S=1, R=1 followed by S=0, R=0 => UNSTABLE

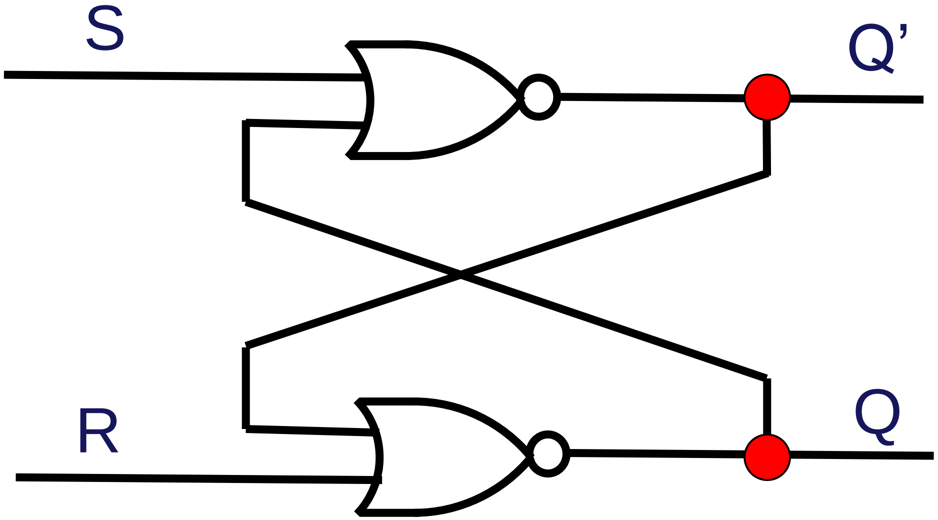
S = 0 and R = 0 allows to memorize the last command used

CLOCK AND CYCLES (2)



The time interval separating two consecutive rising edges of the clock is Cycle.
The inverse of the Cycle duration is Frequency (usually expressed in Mhz or Ghz).
Rising (or alternatively falling) Edge of the clock can be used as sequence of reference events.

SR LATCH

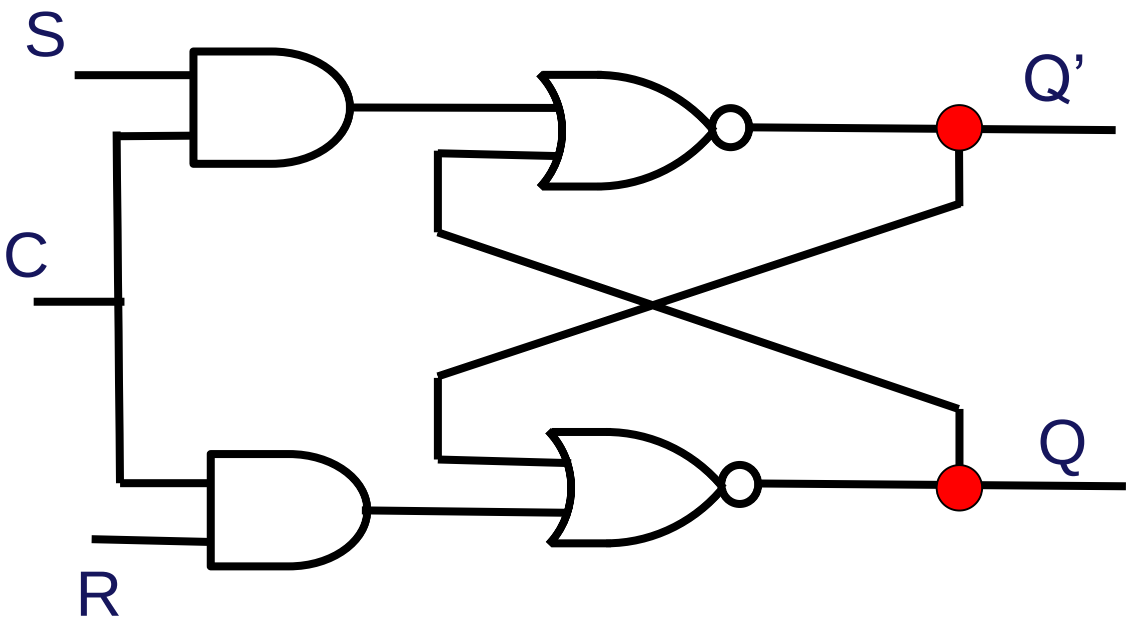


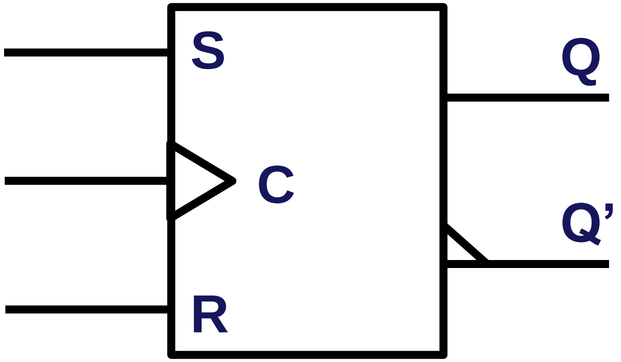
$Q^{+} = S + R' Q$

S	R	Q^{+}	
0	0	Q	Hold State
1	0	1	Set
0	1	0	Reset
1	1	0	Forbidden

SR CLOCKED LATCH

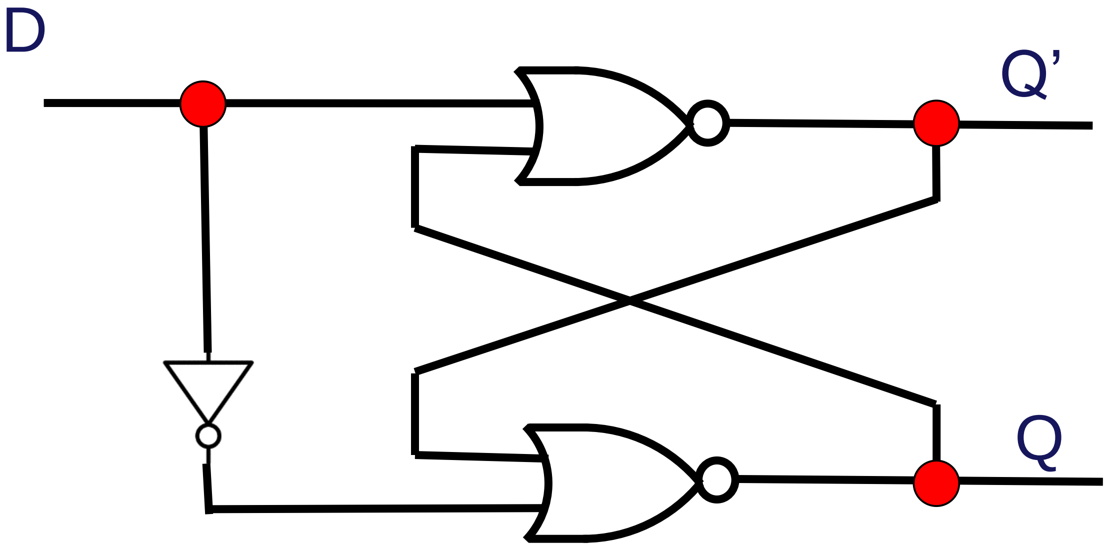
EQUATION $Q_{n+1} = C_n' Q_n + C_n(S_n + R_n' Q_n)$



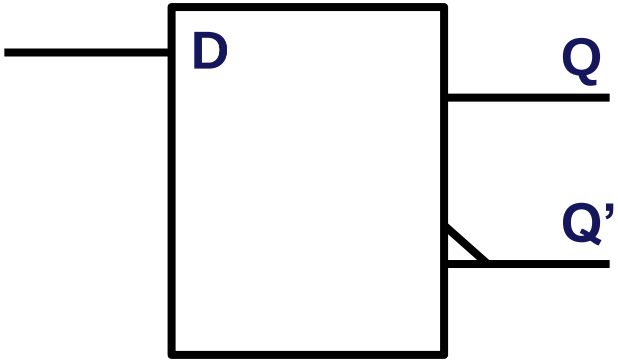


IEC Symbol

D LATCH (1)



NO MORE 1 1 at the entry of NOR network (SR Latch)



IEC Symbol

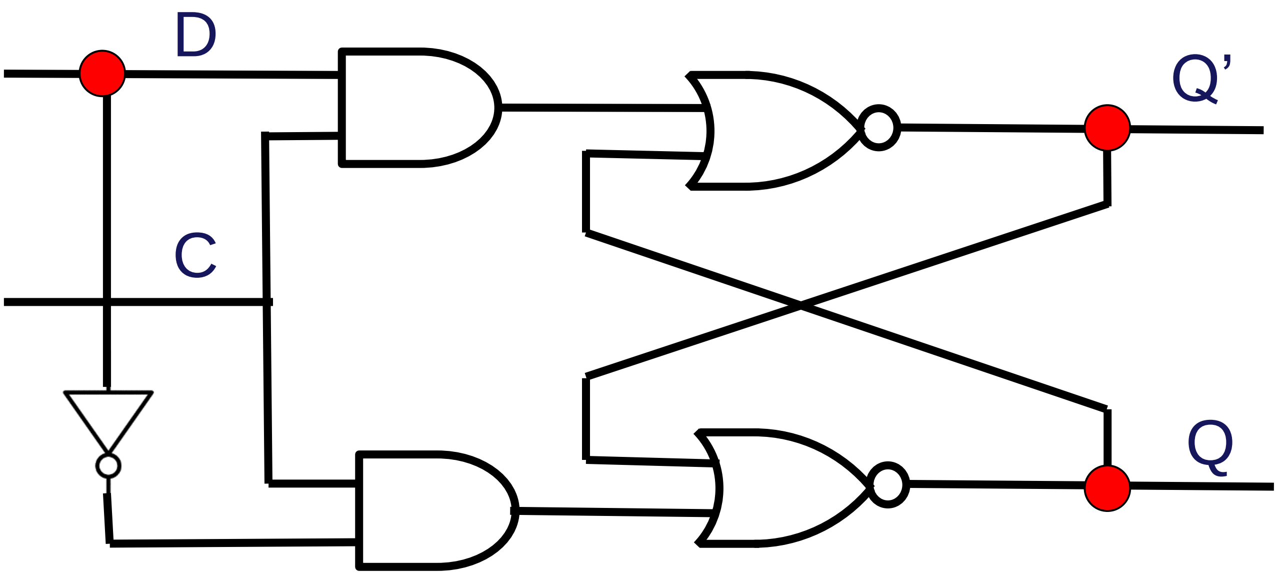
D LATCH (2)

EQUATION $Q^{+} = D$

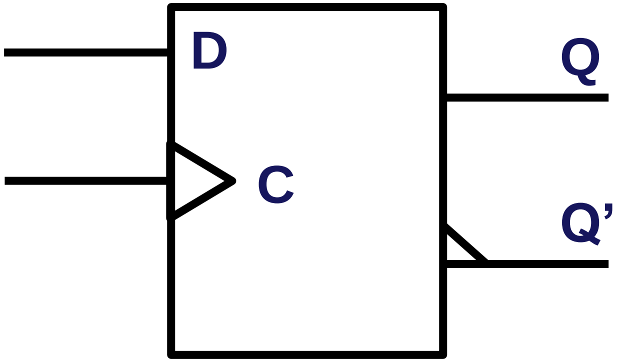
D	Q^{+}	
0	0	Reset
1	1	Set

D as delay.
The output propagates the input with one cycle delay

D CLOCKED FLIP FLOP (1)



NO MORE 1 1 at the entry of NOR network (SR Latch)



IEC Symbol

D CLOCKED FLIP FLOP (2)

EQUATION $Q_{n+1} = C_n' \, Q_n + C_n \, D_n$

C_n	D_n	Q_{n+1}	
0	0	Q_n	Hold State
0	1	Q_n	Hold State
1	0	0	Reset
1	1	1	Set

D as delay.

The output propagates the input with one cycle delay

JK FLIP FLOP

EQUATION $Q^{+} = J\,Q' + K'\,Q$

J	K	Q^{+}	
0	0	Q	Hold State
0	1	0	Reset
1	0	1	Set
1	1	Q’	Toggle

JK Flop Flop also exists in Clocked version:
EQUATION: $Q_{n+1} = C_n'\,Q_n + C_n(J_n\,Q_n' + K_n'\,Q_n)$

MEALY MACHINE

Le us assume a clocked system:

S_{n+1} (resp. S_n) is the state at cycle n+1 (resp. n)

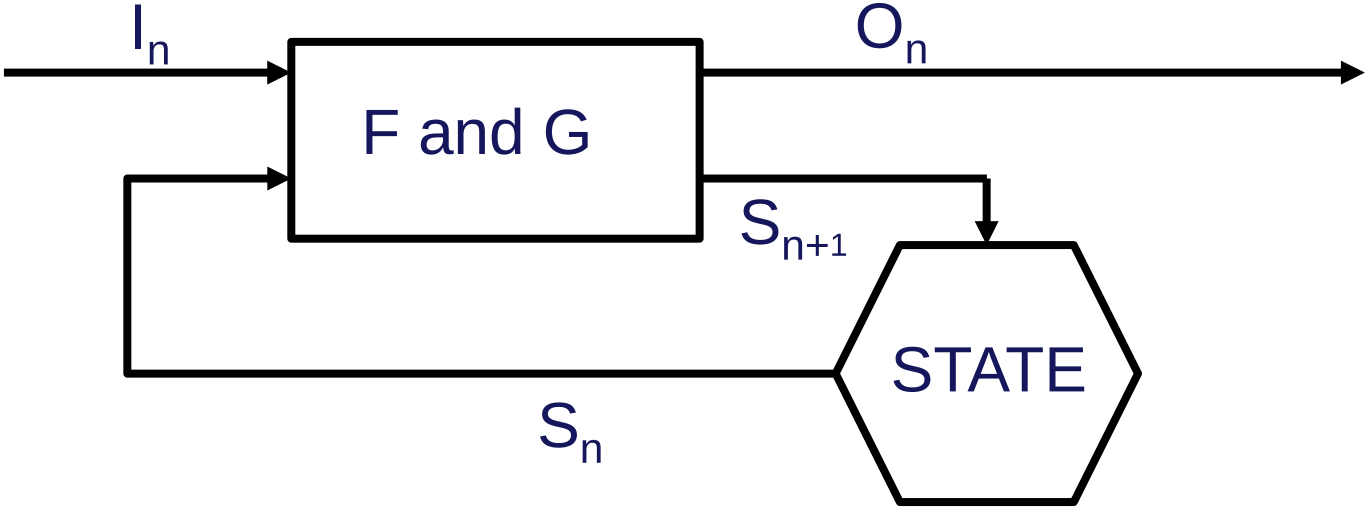
O_n is the output values at cycle n

I_n is the input values at cycle n

F and G are Boolean functions

$S_{n+1} = F(I_n , S_n)$

$O_n = G(I_n , S_n)$



MOORE MACHINE

Le us assume a clocked system:

S_{n+1} (resp. S_n) is the state at cycle n+1 (resp. n)

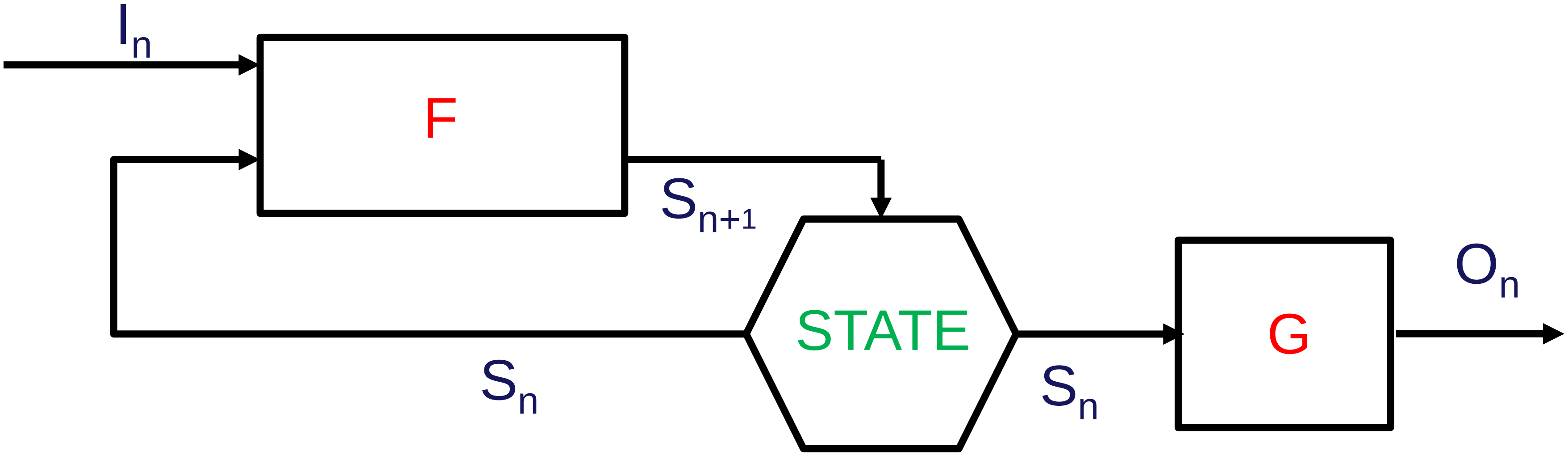
O_n is the output values at cycle n

I_n is the input values at cycle n

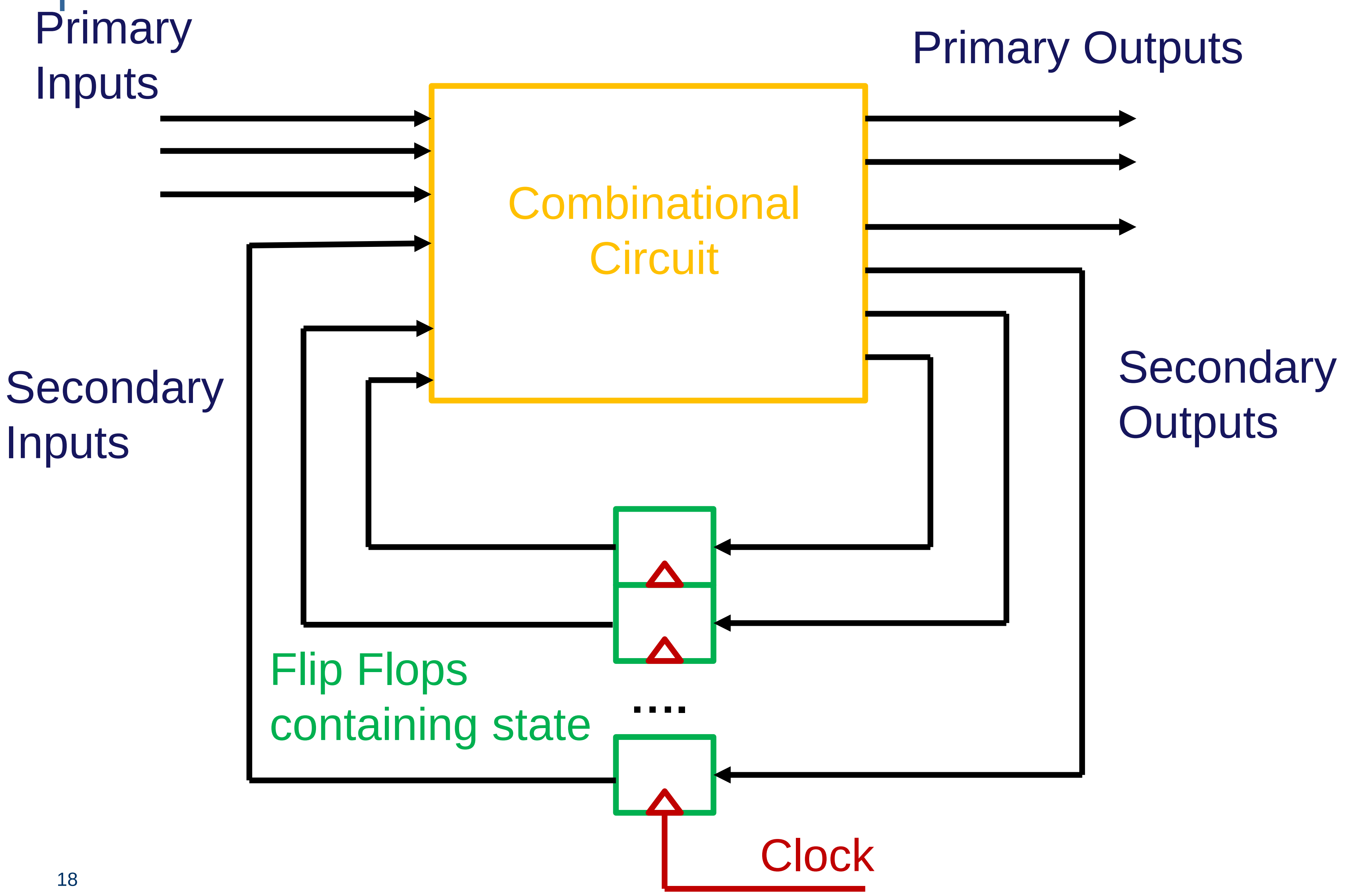
F and G are Boolean functions

$S_{n+1} = F(I_n , S_n)$

$O_n = G(S_n)$



A PRACTICAL MODEL: HUFFMAN MODEL



MORE INFO

Flip Flop see: https://en.wikipedia.org/wiki/Flip-flop_(electronics)

Asynchronous circuits: https://en.wikipedia.org/wiki/Asynchronous_circuit

Acknowledgements

A lot of material was found in Wikipedia.

Some of these slides were inspired by slides developed by:

- University of Washington Computer Science & Engineering (CSE 370)
- Y.N. Patt (Univ of Texas Austin)
- S. J. Patel (Univ of Illinois Urbana Champaign)
- Walid A. Najjar (Univ California Riverside)
- Brian Linard (Univ California Riverside)
- G.T. Byrd (Univ North Carolina)