

Sequential Logic - aim to break up logical calculations into "clocked" chunks; seek record of past: memory

Set Reset Flip Flop (SRFF) "Reset" = clean

crossed NAND  $\Rightarrow$  active low SRFF

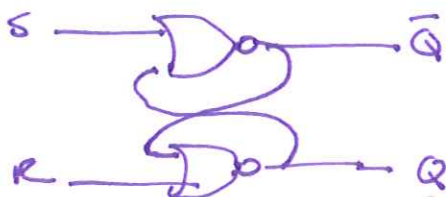


S	R	Q	$\bar{Q}$
1	1	1	0
0	1	1	0
1	0	0	1
0	0	1	1

} both stable  
 ← "set"  
 ← "reset" or clean  
 ← not wanted

Note:  $\overline{\text{L}} \text{D} \text{H} = \text{D}$

crossed NOR  $\Rightarrow$  active high SRFF



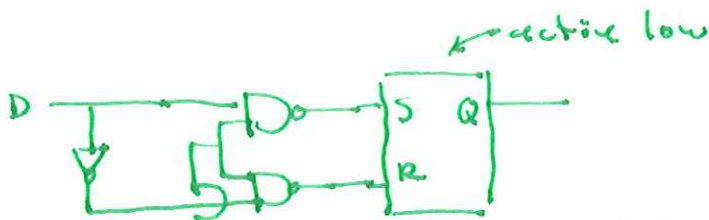
S	R	Q	$\bar{Q}$
0	0	1	0
1	0	1	0
0	1	0	1
1	1	0	0

} both stable  
 ← "set"  
 ← "reset" or clean  
 ← not wanted

Note:  $\text{L} \text{D} \text{H} = \text{D}$

Note: words "SRFF" do not define a unique TT need to know if is active low (usual) or high

Level clocked ("Transparent") D Latch: Q Followed D if Clock H; if clock L holds value.



active low  
 L CLK  $\rightarrow$  if CLK is L S, R inputs high - hold mode  
 if CLK is H  $D = \text{D}$  ?  
 $D = H \Rightarrow (S=L, R=H)$  set  
 $D = L \Rightarrow (S=H, R=L)$  reset

edge trigger vs level trigger: limit the time the FF responds to D input to an "instant" of transition



(positive edge trigger)



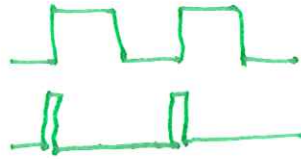
or



(negative edge trigger)

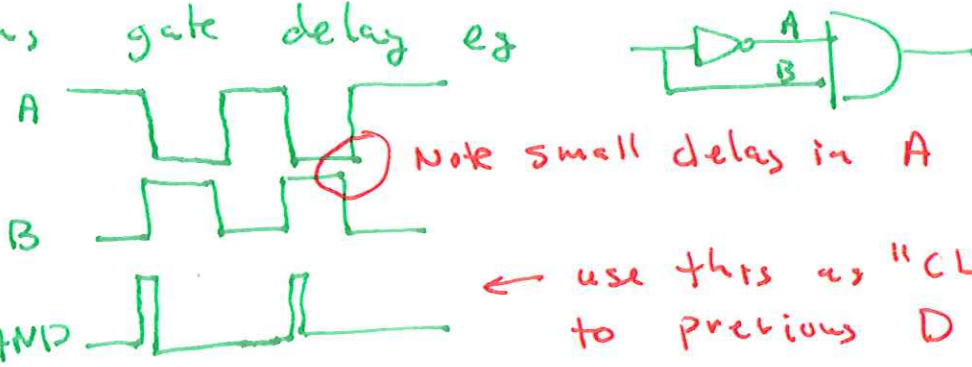


How to do? convert:



(circuit symbol)

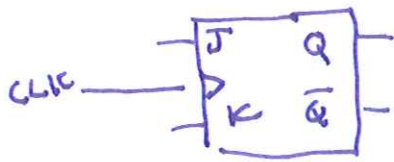
using gate delay eg



Note small delay in A

← use this as "CLK" input to previous D latch

Important (if odd) chip: JKFF



J & K are inputs (controls)

chip may well have other inputs like immediate (aka jam, asynchronous) CLEAR or SET

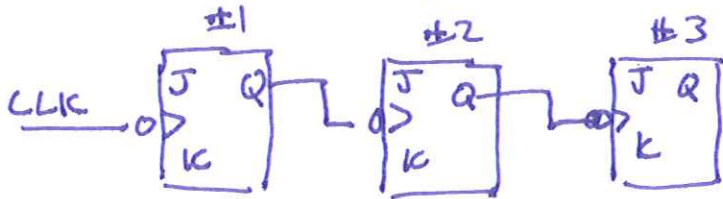
↓ & ↑ versions available.

[eventually I'll show how to make this from a DFF but for time being trust that these things exist]

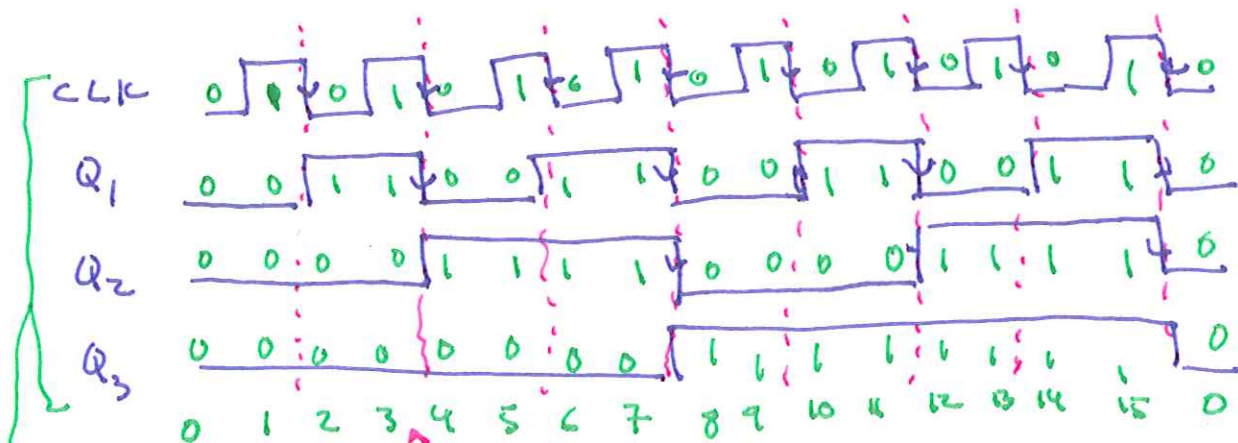
Truth Table:

J	K	
0	0	hold value
0	1	Q → 0 on clock edge
1	0	Q → 1 on clock edge
1	1	toggle mode: 0 → 1 on clock edge, 1 → 0

A simple (synchronous) JKFF circuit:



$J=K=1$  for all  
negative edge triggers



parallel binary number counting up

Notes: because of gate delays this is asynchronous

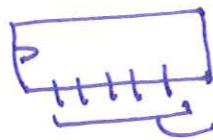
eg 

1	0	0	0
1	1	0	0
0	0	0	1
0	0	0	0

 For a few nanosec count will be wrong

We'll learn later how to make synchronous counters, but this would be fine for human read read-outs.

"Counters"



parallel binary number output

options: up, down, immediate or synchronous clear

load a value. Count itself might be synchronous or asynchronous. Top value (before returns to zero) might be 9 ("decade counter") or a power of 2 ("binary counter")

Positive or negative edge trigger

methods to deal with a "carry" to/from another counter