

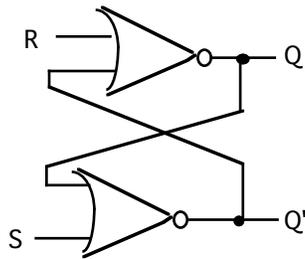
CS240 Laboratory 7

Memory Circuits

Basic Memory Circuits

Latch Single-bit memory, level-triggered
Flip-Flop Also single-bit, but edge-triggered

SR (Set Reset) Latch



S	R	Q	
0	0	Q ₀	remember
0	1	0	reset (clear)
1	0	1	set
1	1		unstable/non-deterministic

SR latch uses 2 cross-coupled NOR gates:

A	B	(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

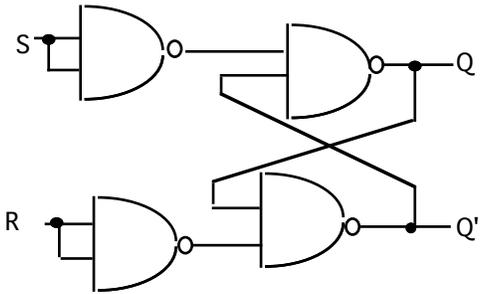
What does **unstable/non-deterministic** mean? Notice in a NOR gate, if either input = 1 to a gate, its output = 0 (deterministic input)

You wouldn't usually try to set and reset at the same time (it doesn't make sense), but if you did, $Q = Q' = 0$. This is predictable.

However, when you go back to the remember state ($S=R=0$) from $S=R=1$, Q and Q' do not stay at 0 (the circuit does not 'remember' the previous output). Instead, Q and Q' change to the complement of one another.

But, you cannot predict whether Q will be 1 or 0! The final state depends on which of the inputs (S or R) being set to 0 is sensed first in the circuit. It is for **this** reason that the $S=R=1$ state is called non-deterministic, or unstable.

Here is an equivalent circuit for the SR latch, using NAND gates only:

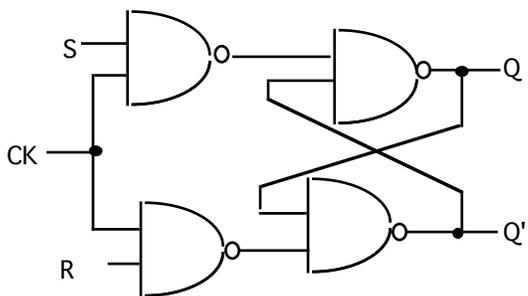


Works the same, except when $S = R = 1$, $Q = Q' = 1$ (instead of 0 as in the NOR gate configuration). Why?

For the NAND gate, if either input is 0, the output is 1 (so 0 is the deterministic input)

A	B	$(AB)'$
0	0	1
0	1	1
1	0	1
1	1	0

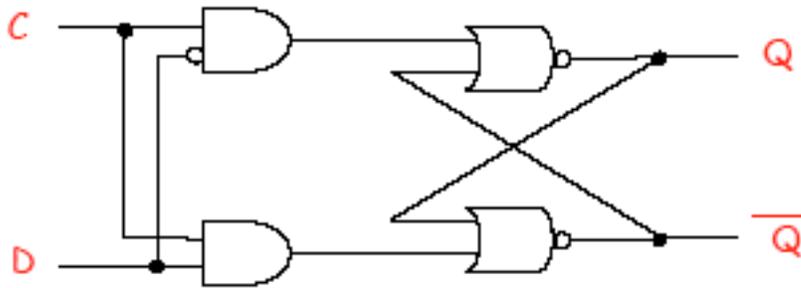
Clocked SR Latch incorporates a clock input



Change occurs only on a high clock level.

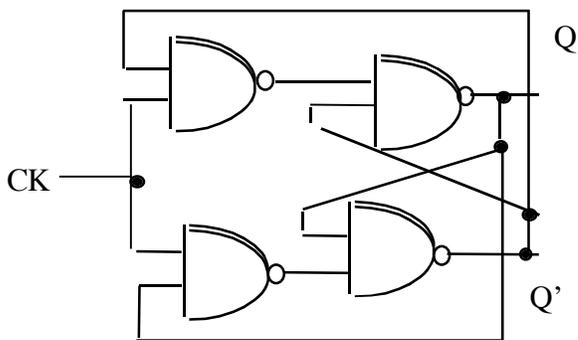
3 circuits which avoid non-deterministic state

D Latch



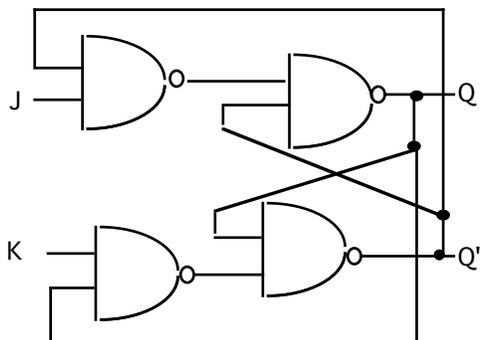
D	Qnext
0	0
1	1

T Latch



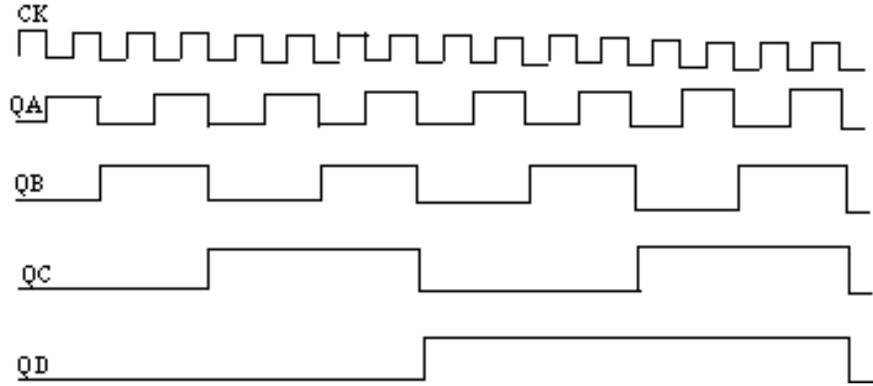
Qprev	Qnext
0	1
1	0

JK Latch

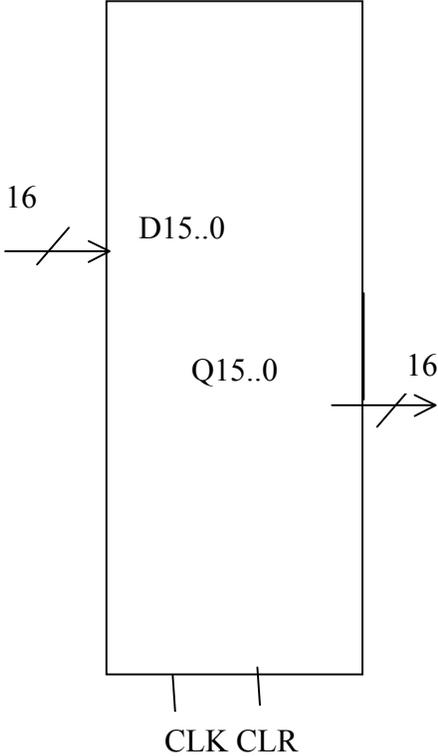


J	K	Q
0	0	Q ₀ (remembers)
0	1	0
1	0	1
1	1	toggles

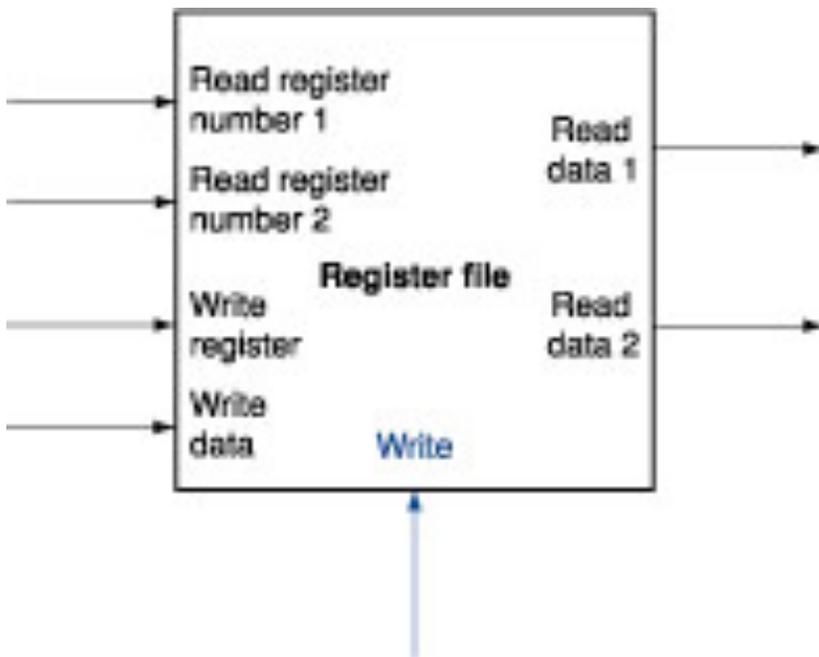
CK	QD	QC	QB	QA
1				
0	0	0	0	0
1				
0	0	0	0	1
1				
0	0	0	1	0
1				
0	0	0	1	1
1				
0	0	1	0	0
1				
0	0	1	0	1
1				
0	0	1	1	0
1				
0	0	1	1	1
1				
0	1	0	0	0
1				
0	1	0	0	1
1				
0	1	0	1	0
1				
0	1	0	1	1
1				
0	1	1	0	0
1				
0	1	1	0	1
1				
0	1	1	1	0
1				
0	1	1	1	1



Register: n-bit memory, using n flip-flops, shared clock and clear inputs

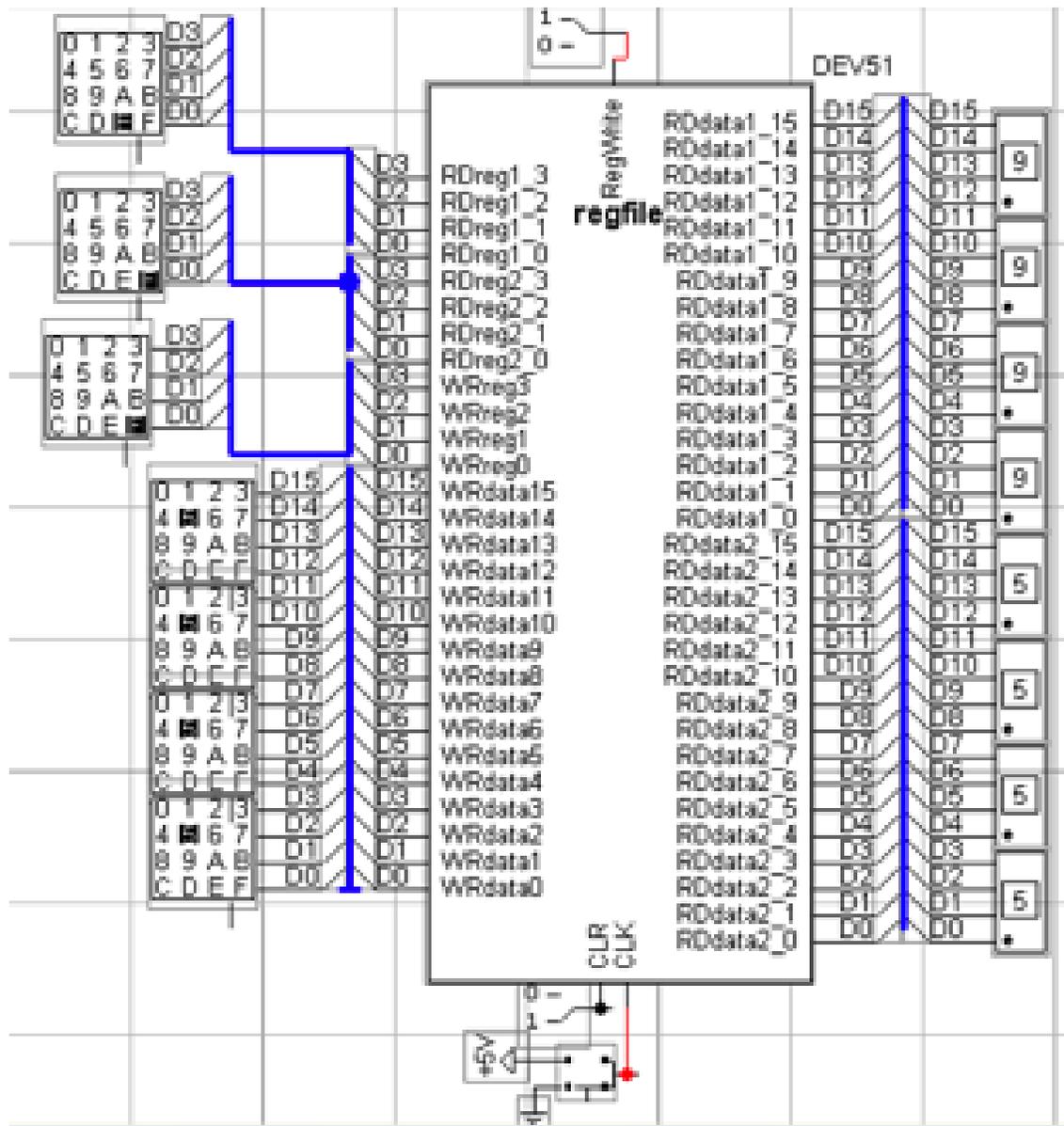


Register File: set of registers

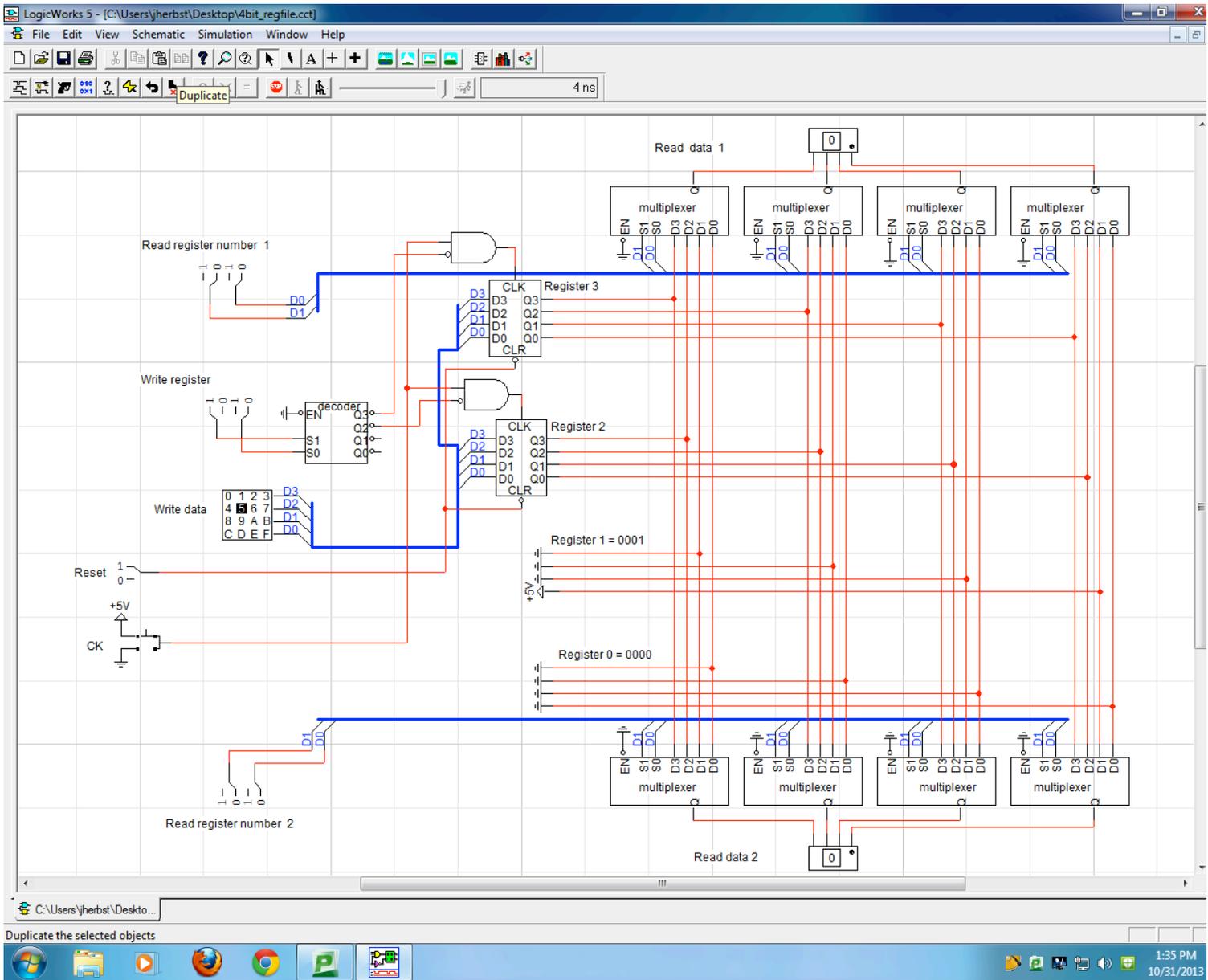


- **Write** is the write control signal.
- **Write register** is the single register to be written to at a time
- **Read register number 1** and **2** indicate which 2 registers can be read at data ports **Read data 1** and **Read data 2**
- clear and clock (**CLR** and **CLK**) are shared by all the 16 registers.
- **CLR** is active low

LogicWorks register file



Internally:



- 2 sets of 16 x 1 multiplexers select which 2 registers are currently being output at the two read ports.

- A 4x16 decoder uses the write register number to select which of the 16 registers will receive a new value on a write.