

Sequential Logic

Ref: Appendix B

1

Combinational vs. Sequential

- Combinational: output depends completely on the value of the inputs.
 - time doesn't matter.
- Sequential: output also depends on the *state a little while ago*.
 - can depend on the value of the output some time in the past.

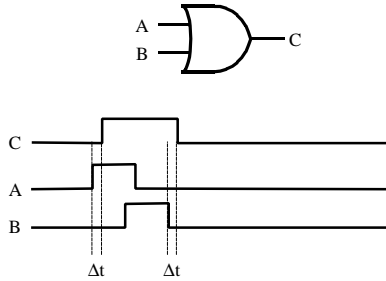
2

Memory

- Think about how you might design a combinational circuit that could be used as a single bit *memory*.
- Use your *memory* to recall that the output of a gate can change whenever the inputs change.

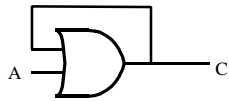
3

Gate Timing



4

Feedback

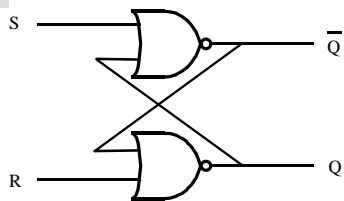


- What happens when A changes from 1 to 0?

5

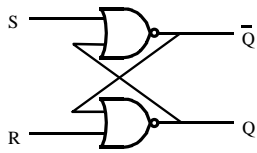
A	B	A nor B
0	0	1
0	1	0
1	0	0
1	1	0

S-R latch



6

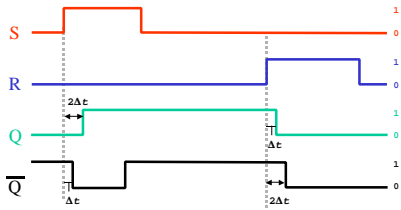
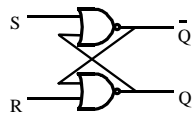
S-R latch Truth Table



Q_t	S_t	R_t	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	?
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	?

7

S-R latch Timing



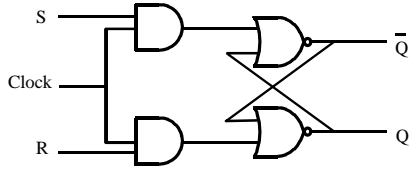
8

Clocked S-R Latch

- Inside a computer we want the output of gates to change only at specific times.
- We can add some circuitry to make sure that changes occur only when a *clock* changes (when the clock changes from 0 to 1).

9

Clocked S-R Latch



- Q only changes when the Clock is a 1.
- If Clock is 0, neither S or R reach the NOR gates.

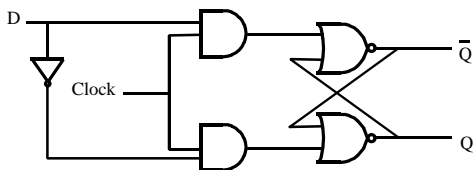
10

What if S=R=1?

- The truth table shows ? when S=R=1.
- The value of Q is undetermined.
 - The circuit is not *stable*.
- We can make sure that S=R=1 now that we have a clock.

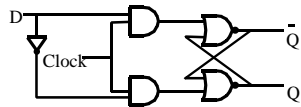
11

Avoiding S=R=1: D Flip-Flop



12

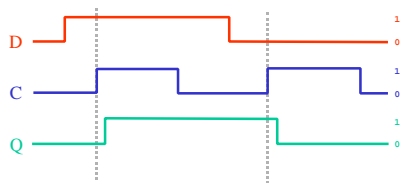
D Flip-Flop



- Now have only one input: D.
- If D is a 1 when the clock becomes 1, the circuit will *remember* the value 1 ($Q=1$).
- If D is a 0 when the clock becomes 1, the circuit will *remember* the value 0 ($Q=0$).

13

D Flip-Flop Timing



14

8 Bit Memory

- We can use 8 D Flip-Flops to create an 8 bit memory.
- We have 8 inputs that we want to *store*, all are *written* at the same time.
 - all 8 flip-flops use the same clock.

15

