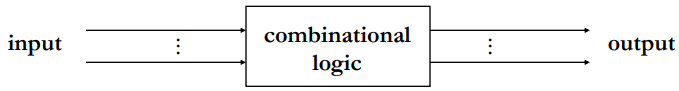
Computer System Lecture 11

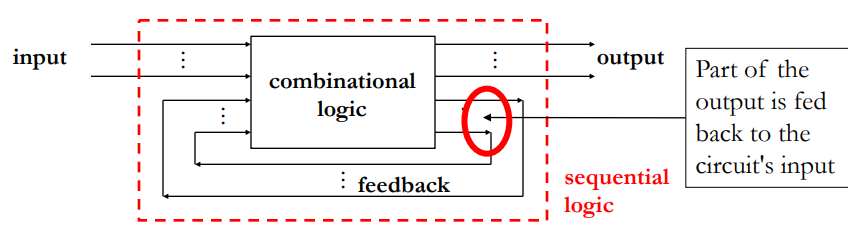
Combinational Logic Review

The output of combinational logic depends only on the current inputs, it doesn’t remember previous outputs.



Memory is needed for more complex operations

Sequential Logic Circuits

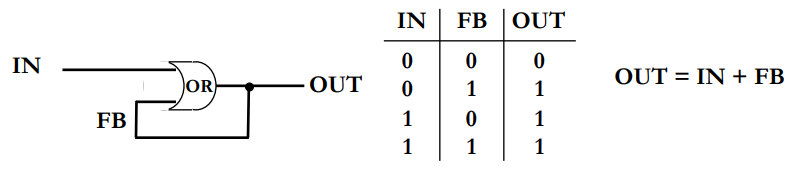


The output depends on current and (some) past inputs, the circuit has memory

Sequences of inputs generate sequences of outputs, with n feedback signals there are up to 2n stable states.

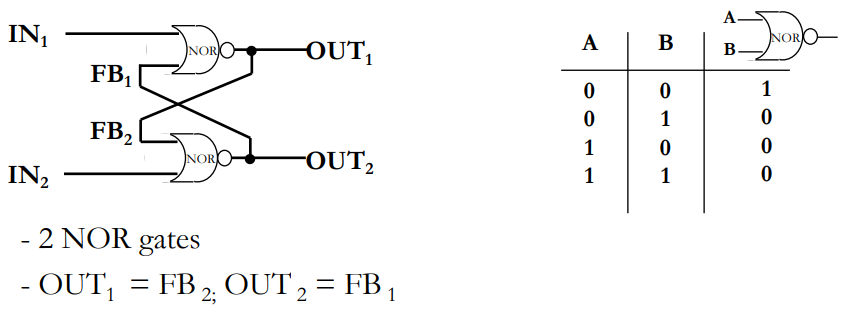
Sequential Circuits

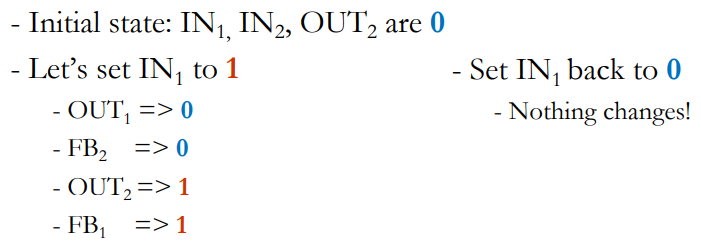
OR with feedback



Once the or has output a 1, it will remember that 1 forever.

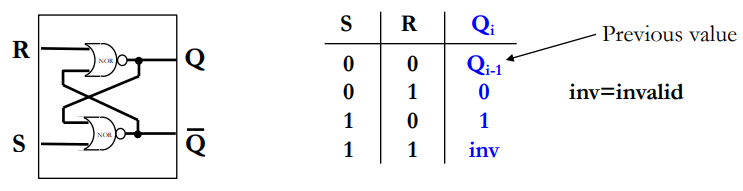
More Practical Sequential Circuits



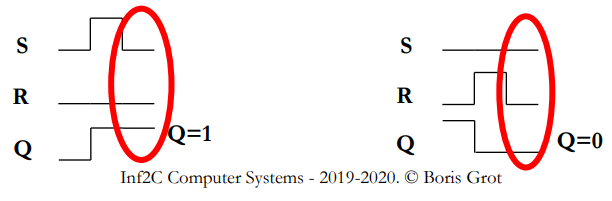


The outputs for this are the opposite of each other, calling one of the outputs Q setting In1 to 1 resets the value of Q back to 0 and In2 sets the value to 1.

SR Latch

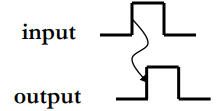


Using 1-bit memory, we can keep the value in memory by maintaining s=0 and r=0 and we can set the value to 0 (or 1) by setting R=1 (or S=1) for a short time:

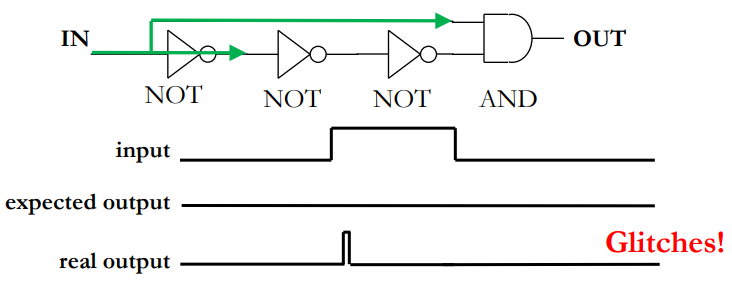


Timing Of Events

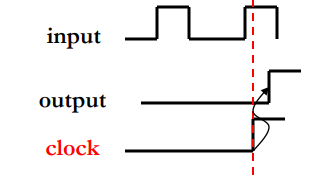
Asynchronous sequential logic: th estate of a circuit changes whenever inputs change:

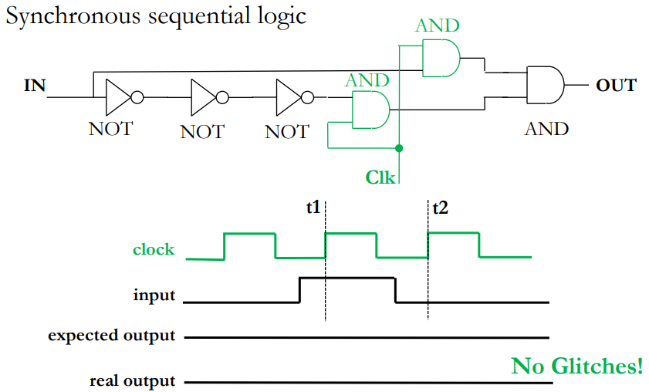


The problem is different circuit paths can have different delays:

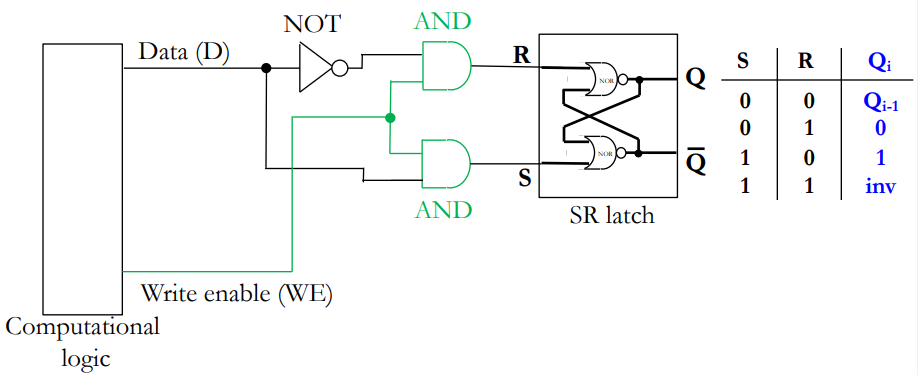


The solution to this is to use only allow the state of memory to synchronized times using an external signal, known as the clock:



This solution is inefficient, you can simply and the output with the clock.

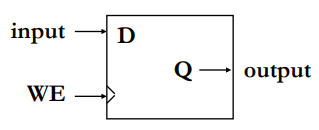
Writing SR Latch  
We now what to apply this idea to our SR latch:

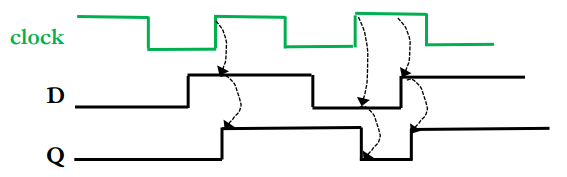
.

This will only allow the data to be saved in the latch when write enable is a 1. (we feed D to S and not D to R to get the correct values we want.

This is necessary, as without our write enable circuit, the SR latch will always just ‘save’ whatever its immediately getting.

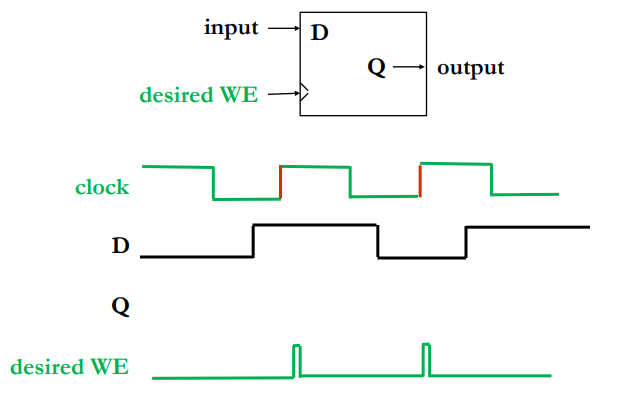
This new circuit is called a D latch

Level-Triggered D Latch



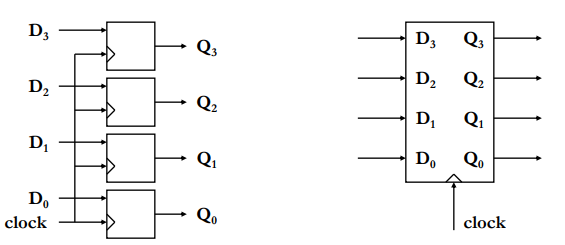
Here the output can change whenever the clock is 1.

Edge-Triggered D Latch  
We only want to be able to write during a very short time:

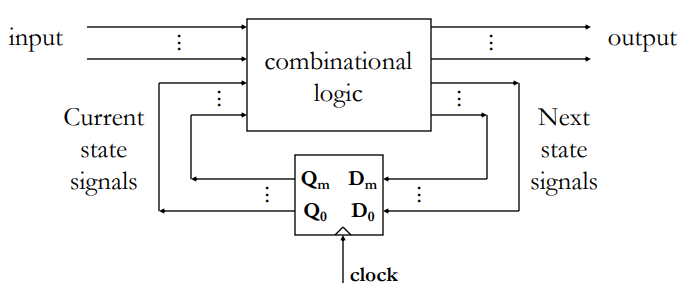


Registers

A register is simply a whole load of d flip flops that use a common clock:



General Sequential Logic Circuit



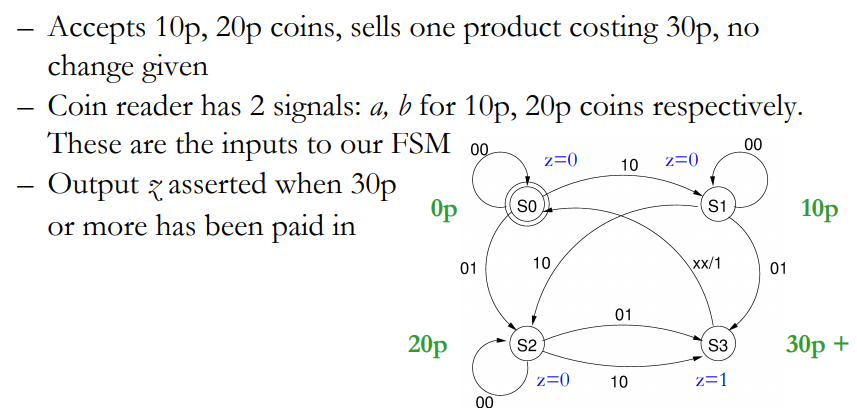
Next state signals are propagated to current state signals at every rising lcok edge.

Current state signals plus inputs work through combinational logic and generate output and next state signals.

Hardware FSM

A sequential circuit is a deterministic finite state machine .

An example of this would be a vending machine:



FSM Implementation

Methodology :

* Choose encoding for states ( S0=00, …., S3=11)
* Build the truth table for next states S1’, S0’ and output z (S1 and S0 are what we use to encode the states).
* Generate the logic equations for S1’, S0’, z (S1’ and S0’ are the next state).
* Design combinator logic from logic equations and add state-holding registers

