**Sequential Logic Design**

In the combinational circuits, output definitely depends on inputs.

* Tuşa basınca lamba açılır, tuşa basınca lamba kapanır.

Sequential circuits keep the current state of the circuit. So output does not only depend on the input, but the output both depends on input and output (our current state).

* Tuşa başınca garaj kapısı açıksa kapanır, kapalıysa açılır.

**Storing one bit**

There are some different circuits to store 1 bit.

0 🡪 Basic storage with feedback

1 🡪 SR Latch

2 🡪 Level-sensitive SR Latch

3 🡪 Level-sensitive D Latch

4 🡪 Edge Triggered D-Flip-Flop (Master Slave Inverter)

Basic Storage Element

A picture containing line chart

Description automatically generatedA picture containing text

Description automatically generated   
 {Dashed lines are for 0 level.}

Assume initially s and q are both 0.

After s turns 1, there is going to be delay to q to be 1 because OR gates include some transistors. You apply a voltage to transistors, it will collect some electrons to gate, and then it becomes conducter. This process takes some time.

Once you set the output to 1, there is no way to return back.

A picture containing polygon

Description automatically generatedSR Latch

We will use NOR gates.

x can be 0 or 1, it doesn’t change the output.

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | OR | NOR |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

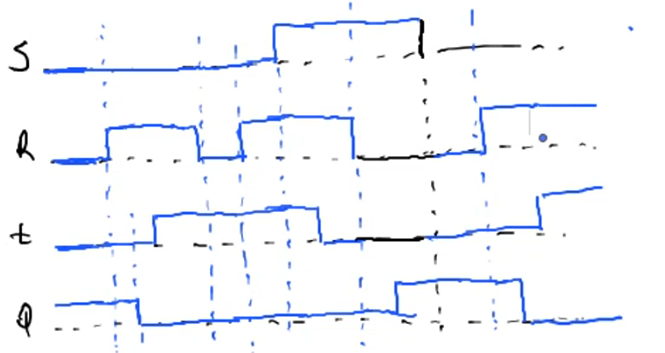
A picture containing diagram

Description automatically generated

Outputu başta 1 varsaydık. Q (output), o andaki değeri neyse onunla başlayacak.

Reset 1’se, S outputta tutulmaz.

**Don’t forget the delays!!!**



What happens when S and R become 1 at the same time and return 0 at the same time?

Chart

Description automatically generated with medium confidence

This is a problem with SR Latch.

Setting S and R to 1 at the same time causes oscilation.

Prevent S and R to become 1 at the same time.

Text

Description automatically generatedLevel-sensitive SR Latch

Diagram

Description automatically generated

2 ns

Our aim by placing NOT gate is to prevent S and R to become 1 at the same time. Each gate create delay. Lets say the latency of a gate is 1 ns. We will have 2 ns latency. After X becomes 1, it takes 2 ns to R to be 0.  
In the red part, both X (or S, they are same) and R become 1 at the same time, we didn’t solve the problem. Original Level-sensitive SR Latch is below:

Icon

Description automatically generatedA picture containing diagram

Description automatically generated

Q’ is opposite of Q.



X ve Y değişince C’yi de değiştiririz. Buradaki sorun sürekli C ile uğraşmak.

A picture containing text, computer

Description automatically generated

C is used to pass dangerous (2 ns) red region.

C is called clock signal.

- Clock Signal

Clock signal generator (oscilator):

A close-up of a magnifying glass

Description automatically generated with medium confidence

There were some transistors inside NOT gate. There is some latency.

Clock signal:



How can we change frequency of clock signal?

A picture containing diagram

Description automatically generatedIf you increase the size of transistors gate, that means you are increasing the latency, and if you decrease the size of transistors, that means you are decreasing the latency (collectiong electrons to drain will take less time).

Level-sensitive D Latch

Diagram

Description automatically generated with low confidence

When C is 1, input D is going to transferred to the output Q.

When C is 0, output is going to stay same. Don’t change anything in the output.

D Latch stores the value at D when C=1 and stores it even C=0.

A picture containing text

Description automatically generated Diagram

Description automatically generated

Lets say you have multiple Level-sensitive D Latch:

Diagram

Description automatically generated

They all have same clock signal so I am activating them at the same time.

Lets say I have 1001. We start with 1.

At 1 clock signal I move 1 to one latch right. Then I put next one at the 1001 to the moved latch.

Yani datamızı adım adım sağa iterate ederek her latchin outputunda 1 bit saklıyoruz. 4 biti 4 latch ile saklıyabiliyoruz.

What happens if frequency of the clock signal is very low?

In this case, at each clock signal you can have multiple movements bc duration is quite large.

What happens if frequency of the clock signal is very high?

It may not be enough to activate D Latches.

So frequency of the clock signal is quite critical.

So D Latch is not useful.

Edge Triggered D-Flip-Flop (Master Slave Inverter)

Diagram

Description automatically generatedIncludes 2 D Latch + 1 inverter.

A picture containing text, person

Description automatically generated



Diagram

Description automatically generated



We have made real changing when clk (C - clock signal)  
changes from 0 to 1.



Before clock signal is 0, master was active, slave is waiting.  
Slave is waiting for clock signal to become 1.  
So transition between 0 to 1 for clock signal, makes that data to be at the output.

When clock signal becomes 1 again above, Qs will become 0 after some delay.

So transition is quite important in edge triggered D flip flop bc transition is so fast. 0 to 1 transition is quite fast.

Look at rising clock signal.

*0 to 1 transition makes the input at D to be at output (Q).*

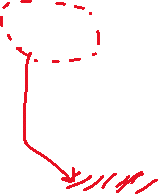
Diagram, engineering drawing

Description automatically generated

Example:

Diagram

Description automatically generated



Main principle of flip flops are storing 1 bit data.

What about if you want to store more bits at the same time? 🡪 You need registers.