Lab Work 3 (Part 1)

Implementing Sequential Logic

Sequential logic circuits mainly provides storage for a digital system. Thus, for example, it allows implementation of processes that need the previous state of a digital logic.

Only two basic types of sequential logic circuit:

- Latch : level-triggered temporary storage device with 2 stable states
- Flip-flop : edge-triggered storage device

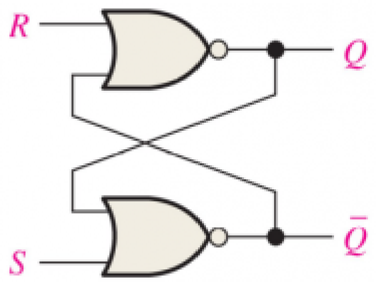
***Note**: We will be using the button (@reset) switch (instead of the normal toggle DIP-switch). Try to find out why this is a better option.*

Implementing Latches

For a basic latch, it needs to have the means to achieve any stable state (HI or LO) at its output and to maintain that output state.

S-R Latch

A basic S-R Latch:



Truth table:

S	R	Q_n	$\overline{Q_n}$
0	0	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
0	1	0	1
1	0	1	0
1	1	0*	0*

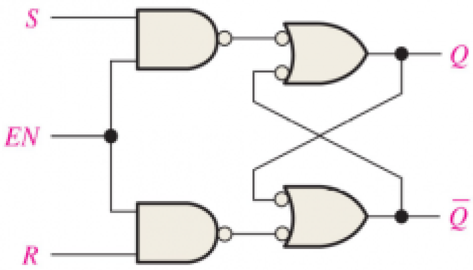
Note: '' indicates invalid condition*

***Disclaimer**: The image above is extracted from resources available for Digital Fundamentals 11th Edition (Global Edition)*

Notice that the invalid condition will happen when the S & R inputs are both at logic 1. In a real digital system, we need to prevent this from happening.

Gated S-R Latch

A gated S-R Latch:



Disclaimer: *The image above is extracted from resources available for Digital Fundamentals 11th Edition (Global Edition)*

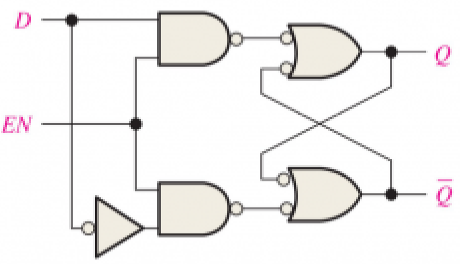
Truth table for a S-R Latch:

S	R	EN	Q_n	$\overline{Q_n}$
0	0	0	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
0	1	0	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
1	0	0	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
1	1	0	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
0	0	1	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
0	1	1	0	1
1	0	1	1	0
1	1	1	**1**	**1**

*Note:Output in **BOLD** is an invalid condition.*

Gated D Latch

A gated D Latch:



Disclaimer: *The image above is extracted from resources available for Digital Fundamentals 11th Edition (Global Edition)*

Truth table for a gated D Latch:

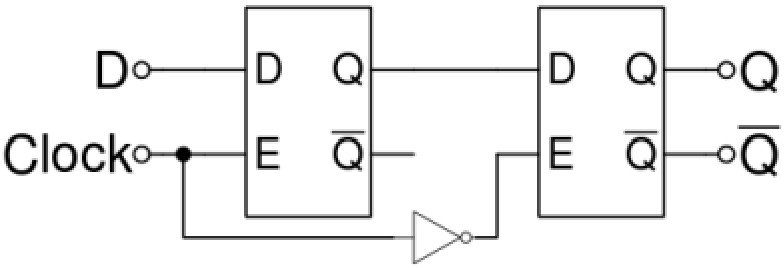
EN	D	Q_n	$\overline{Q_n}$
0	X	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
1	0	0	1
1	1	1	0

Implementing Flip-flops

Unlike latches, flip-flops requires *reference* clock signal that is used to transfer whatever signal at its input to its internal storage (latch).

D Flip-flop

A D flip-flop (DFF) can be built using two opposite level-triggered gated D latches. This is known as a master slave DFF, as shown in figure below. Is this a positive or negative edge-triggered DFF?



Disclaimer: *The image above is obtained from Wikipedia*

Truth table for a positive edge-triggered DFF:

CLK	D	Q_n	$\overline{Q_n}$
0→1	0	0	1
0→1	1	1	0

The table for negative edge-triggered DFF onlly differs at CLK column with 1→0 instead of 0→1.

J-K Flip-flop

Truth table for a J-K Flip-flop:

J	K	CLK	Q_n	$\overline{Q_n}$
0	0	0→1	$Q_{n\text{-}1}$	$\overline{Q_{n\text{-}1}}$
0	1	0→1	0	1
1	0	0→1	1	0
1	1	0→1	$\overline{Q_{n\text{-}1}}$	$Q_{n\text{-}1}$

Notice that, unlike DFF, J-K FF has a toggle mode.

T Flip-flop

The DFF can easily be modified to act as a Flip-flop that toggles on clock edges - by simply connecting the $\overline{Q}$ output to the D input.

Things To Do

THING1 Build an S-R Latch (use NOR gates) and verify.

THING2 Build a gated S-R Latch (use NAND gates) and verify.

THING3 Build a gated D Latch and verify.

THING4 Build a D Flip-flop (DFF) and verify.

THING5 Build a J-K Flip-flop and verify.

THING6 (optional) Try to verify the use of DFF as a frequency divider.