main 1 Branch 0 Tags Go to file Go to file Code

This branch is 1 commit ahead of naavaneetha/D-FLIPDLOP-NEGEDGE:main .

23008859 Update README.md 2 minutes ago

| | | |
|---|---|---|
| db | EXP8 | 3 months ago |
| incremental_db | EXP8 | 3 months ago |
| output_files | EXP8 | 3 months ago |
| simulation | EXP8 | 3 months ago |
| DFLIPFLOPNEGEDGE.qpf | EXP8 | 3 months ago |
| DFLIPFLOPNEGEDGE.qsf | EXP8 | 3 months ago |
| DFLIPFLOPNEGEDGE.qws | EXP8 | 3 months ago |
| DFLIPFLOPNEGEDGE.v | EXP8 | 3 months ago |
| DFLIPFLOPNEGEDGE.v.bak | EXP8 | 3 months ago |
| LICENSE | Initial commit | 3 months ago |
| README.md | Update README.md | 2 minutes ago |
| Waveform1.vwf | EXP8 | 3 months ago |

README License

# D-FLIPDLOP-NEGEDGE

**AIM:**

To implement D flipflop using verilog and validating their functionality using their functional tables

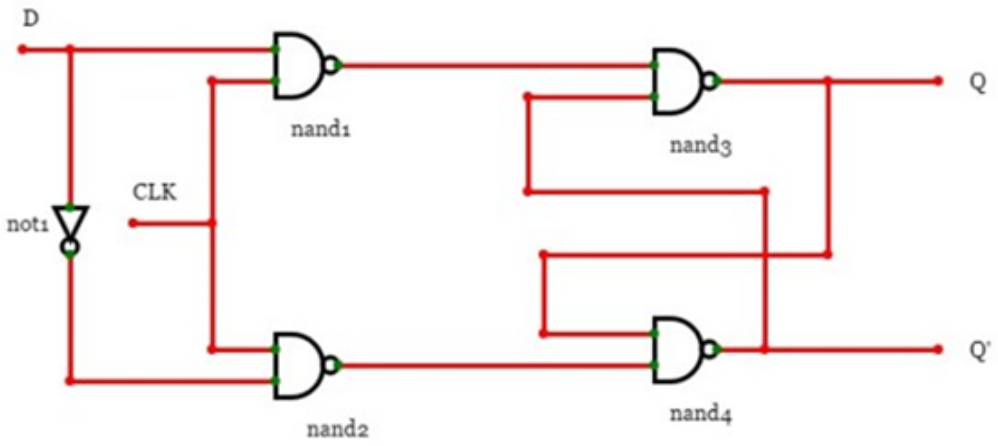
**SOFTWARE REQUIRED:**

Quartus prime

**THEORY**

**D Flip-Flop**

D flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal. That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal. The circuit diagram of D flip-flop is shown in the following figure.




This circuit has single input D and two outputs Qtt & Qtt'. The operation of D flip-flop is similar to D Latch. But, this flip-flop affects the outputs only when positive transition of the clock signal is applied instead of active enable. The following table shows the state table of D flip-flop.

| D | Qt + 1t + 1 |
|---|---|
| 0 | 0 |
| 1 | 1 |

Therefore, D flip-flop always Hold the information, which is available on data input, D of earlier positive transition of clock signal. From the above state table, we can directly write the next state equation as Qt+1t+1 = D

| D | Q $t+1$ |
|---|---|
| 0 | Q $t$ |
| 1 | Q $t$ ' |

Next state of D flip-flop is always equal to data input, D for every positive transition of the clock

signal. Hence, D flip-flops can be used in registers, shift registers and some of the counters.

**Procedure**

1.Type the program in Quartus software.

2.Compile and run the program.

3.Generate the RTL schematic and save the logic diagram.

4.Create nodes for inputs and outputs to generate the timing diagram.

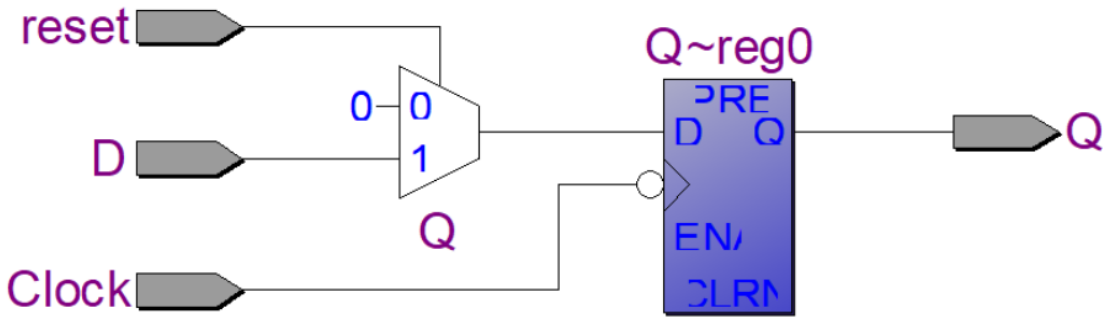
5.For different input combinations generate the timing diagram.

**PROGRAM**

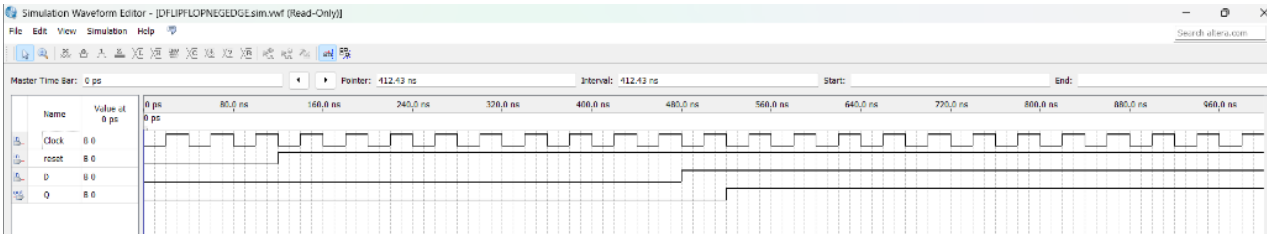
/* Program for flipflops and verify its truth table in quartus using Verilog programming. Developed by: ROSHINI S RegisterNumber: 212223230174

```
module DFLIPFLOPNEGEDGE(D,Clock,reset,Q);
input D,reset,Clock;
output reg Q;
always @ (negedge Clock)
if(!reset)
Q <= 0;
else
Q <= D;
endmodule
```

**RTL LOGIC FOR FLIPFLOPS**



### TIMING DIGRAMS FOR FLIP FLOPS




### RESULTS

Thus the program to implement a D flipflop using verilog and validating their functionality using their functional tables.

## Releases

No releases published

## Packages

No packages published

## Languages

**VHDL** 49.0% **Stata** 21.0% **HTML** 18.7% **Verilog** 9.6% **Standard ML** 1.7%