E208 – Eletrônica Digital II **4º Período / 1º Semestre de 2021**

Professores: Bruno de Oliveira Monteiro bruno@inatel.br

Monitores: Felipe Pereira Silveira felipepereira@gea.inatel.br

Carlos Daniel Borges Vilela Marques carlos.marques@gea.inatel.br

Gualter Machado Mesquita machadomgualter@gmail.com

Maíra Alves Chagas mairaalves@gec.inatel.br

Pedro Henrique Praxedes dos Reis pedro.reis@gea.inatel.br

Thalita Fortes Domingos thalita.fortes@gec.inatel.br

Maria Luiza Rosestolato Araújo maria.luiza@gec.inatel.br

Marcos Henrique Rodrigues Lopes marcos.lopes@gea.inatel.br

Thiago da Rocha Miguel thiago.miguel@gec.inatel.br

Aluno: _____ **Matrícula:** _____ **Período:** ___ **Data:** ___ / ___ / ___

Assunto da semana: Flip Flop D e T

Relatório 3

Teoria

Flip-Flop D

Apesar do flip-flop tipo JK possuir várias configurações uteis para nós, em certos casos utilizamos apenas algumas de suas funções. Na figura 1 mostramos o diagrama lógico de um flip-flop D (*Data* - Dados), implementado com uma porta Inversora:

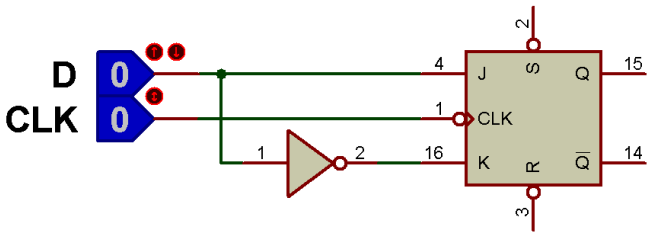


Figura 1

O flip-flop tipo D herda apenas as configurações de quando J e K são diferentes (J = 0, logo K = 1 ou J = 1, logo K = 0), desse modo ele funciona como um carregador de Dado, ou seja o valor

lógico da entrada de dados D é transferido para a saída Q quando ocorre uma transição na entrada de CLOCK.

A Figura 2 mostra o diagrama em bloco do flipo flop D e a Tabela 1 mostra a tabela da verdade.

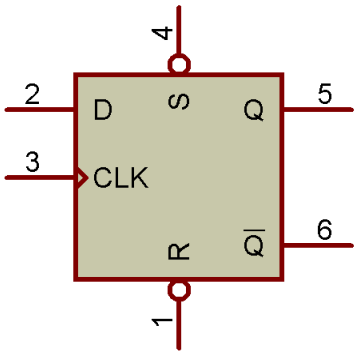


Figura 2

CL	PR	CK	D	Q
0	0	X	X	*
0	1	X	X	0
1	0	X	X	1
1	1	↓	X	Qa
1	1	↑	0	0
1	1	↑	1	1

Tabela 1

* Condição não permitida (sem sentido físico).

Flip-Flop T

O flip-flop tipo T herda apenas as configurações as quais o J e K são iguais (J = K = 0 ou J = K = 1), pois existe um curto nos terminas de J para K. Assim, ele guarda o valor de dado da saída ou inverte o valor anterior da mesma quando ocorre uma transição na entrada de CLK. Dessa forma, o Flip-Flop T (*Toggle* - Alternar) age como um "alternador", que inverte o estado da saída quando é ativado e o mantém constante quando é desativado. A figura 3 mostra a implementação:

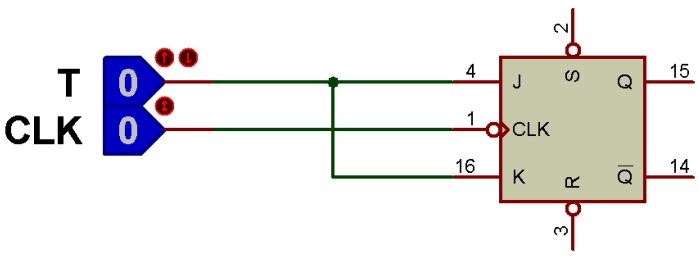


Figura 3

CLR	PR	CK	T	Q
0	0	X	X	*
0	1	X	X	0
1	0	X	X	1
1	1	↓	X	Qa

A Figura 4 mostra o diagrama em bloco do flipo flop T e a Tabela 2 mostra a tabela da verdade do flip-flop T.

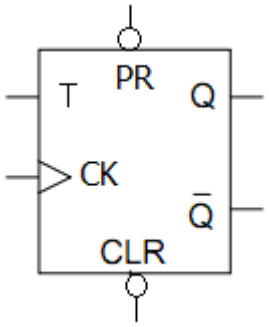


Figura 3

1	1	↑	0	Qa
1	1	↑	1	$\overline{Qa}$

Tabela 2

* Condição não permitida (sem sentido físico).

Exercício Teórico

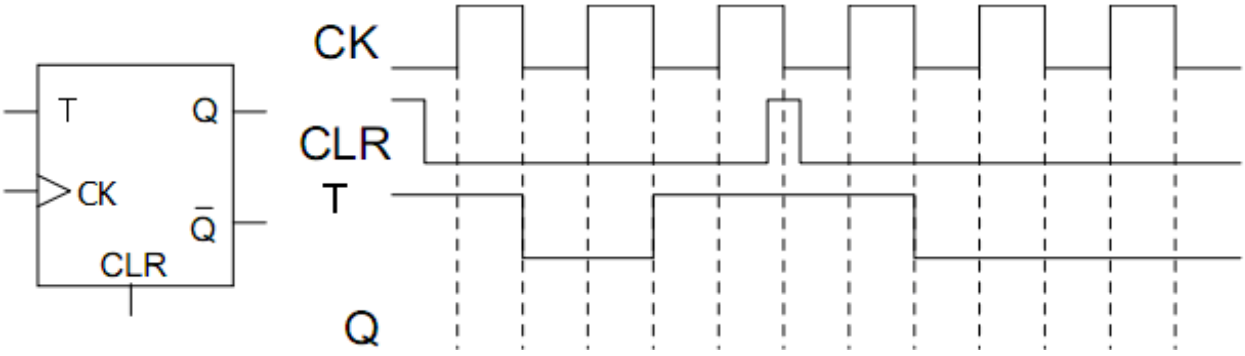
Questão 1. Configure cada flip flop dos itens abaixo. Desenhe cada um dos flip flops com entradas PRESET e CLEAR e desenhe o ajuste necessário.

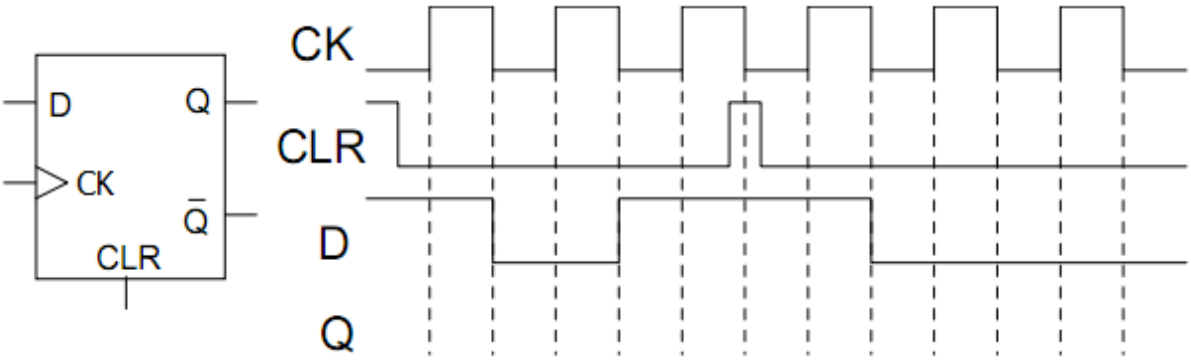
a. F.F. D para operar somente na função Toggle.

b. F.F. JK para operar como um F.F. T.

c. F.F. JK para operar como um F.F. D.

Questão 2. Desenhe a forma de onda na saída de cada flip flop.



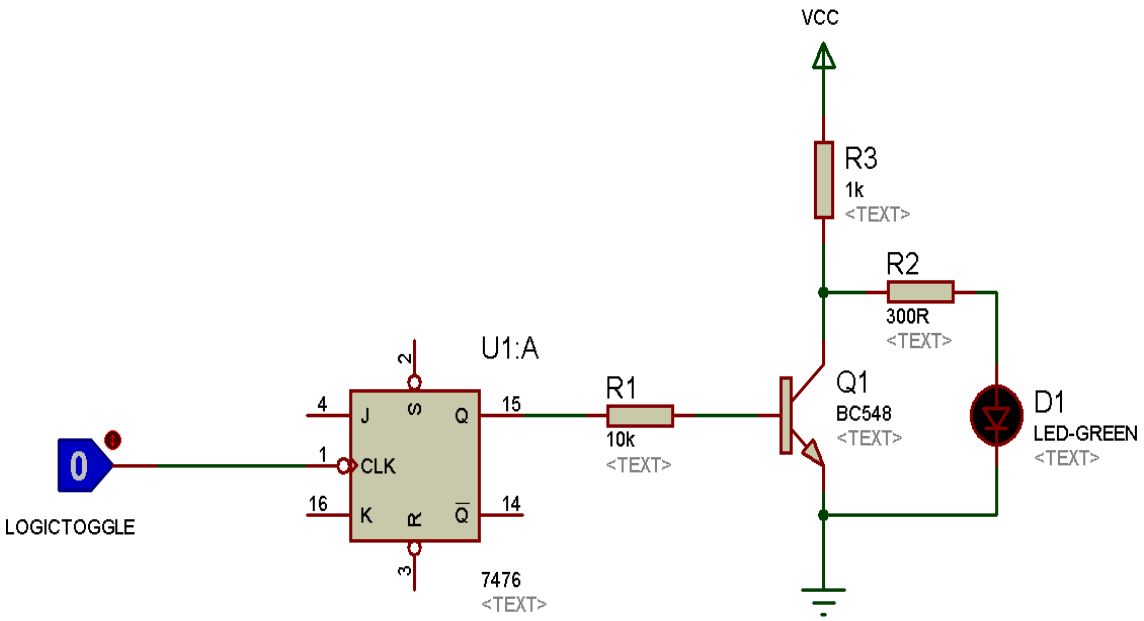


Exercício Prático

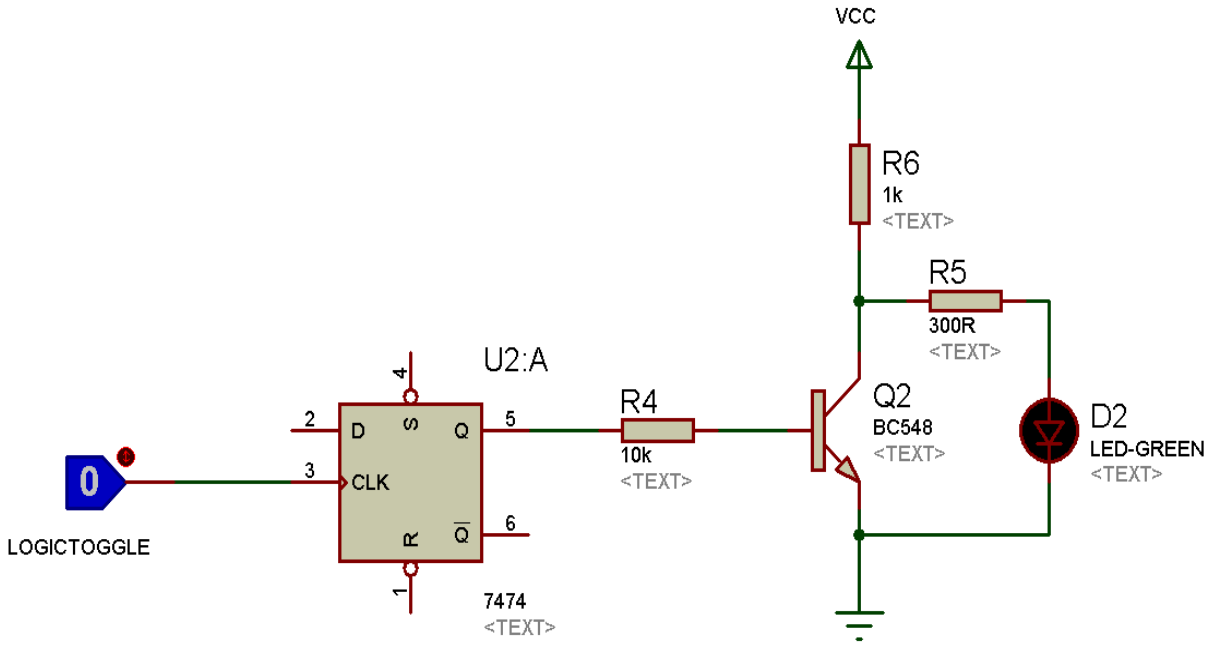
Questão 1. Configure cada circuito abaixo e faça a simulação no Proteus ISIS e no módulo digital:

Obs: Todas as atividades que solicitarem montagem no módulo digital devem ser realizadas no TinkerCad enquanto as aulas não retornarem de forma presencial.

a. Configurar o F. F. J K para operar como um F. F. D (utilizar o CI para aplicar a lógica INVERSORA).



b. Configurar o F. F. D para operar somente na função Toggle.



Questão 2. Realize a simulação de um flip-flop D no software ISE.

Código de VHDL

```
ENTITY FF_D is
    PORT(   d : in  STD_LOGIC;
            clk : in  STD_LOGIC;
            q : out  STD_LOGIC;
            q_bar : out  STD_LOGIC);
END FF_D;

ARCHITECTURE Behavioral OF FF_D IS

signal sinal_q: STD_LOGIC;

BEGIN
    process(clk)
BEGIN

    sinal_q <= '0';

    IF rising_edge(clk) THEN
        IF (d = '0') THEN
            sinal_q <= '0';
        ELSIF (d = '1') THEN
            sinal_q <= '1';
        END IF;
    ELSIF falling_edge(clk) THEN
        sinal_q <= sinal_q;
    END IF;

    END PROCESS;

    q <= sinal_q;
    q_bar <= not sinal_q;

END Behavioral;
```

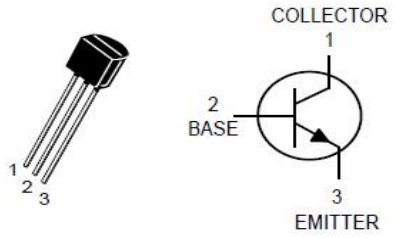
Código de teste

```
d <= '0';
wait for 50 ns;
clk <= '0';
wait for 50 ns;
clk <= '1';
wait for 50 ns;
clk <= '0';
wait for 50 ns;

d <= '1';
wait for 50 ns;
clk <= '0';
wait for 50 ns;
clk <= '1';
wait for 50 ns;
clk <= '0';
wait for 50 ns;
```

Questão 3 (Proposto). Realize a simulação de um flip-flop T no software ISE.

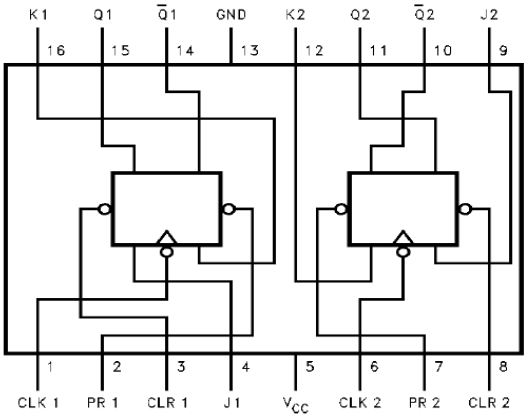
Transistor BC 548



BC 548 Transistor

CI 7476

Connection Diagram



Function Table

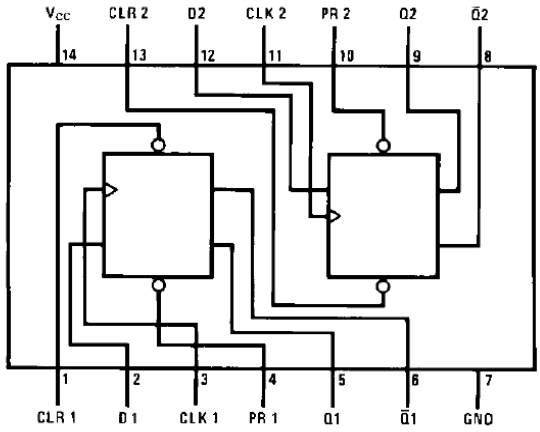
Inputs					Outputs	
PR	CLR	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H (Note 1)	H (Note 1)
H	H	⎍	L	L	Q_0	$\overline{Q}_0$
H	H	⎍	H	L	H	L
H	H	⎍	L	H	L	H
H	H	⎍	H	H	Toggle	

H = HIGH Logic Level
L = LOW Logic Level
X = Either LOW or HIGH Logic Level
⎍ = Positive pulse data. The J and K inputs must be held constant while the clock is HIGH. Data is transferred to the outputs on the falling edge of the clock pulse.
Q_0 = The output logic level before the indicated input conditions were established.
Toggle = Each output changes to the complement of its previous level on each complete active HIGH level clock pulse.

Note 1: This configuration is nonstable; that is, it will not persist when the preset and/or clear inputs return to their inactive (HIGH) level.

CI 7474

Connection Diagram



Function Table

Inputs				Outputs	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H (Note 1)	H (Note 1)
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

H = HIGH Logic Level
X = Either LOW or HIGH Logic Level
L = LOW Logic Level
↑ = Positive-going transition of the clock.
Q_0 = The output logic level of Q before the indicated input conditions were established.

Note 1: This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (HIGH) level.

Itens que devem conter no quite:

- Um protoboard;
- Um CI 7476;
- Um CI 7474;
- Um transistor BC 548;
- Um resistor de cada: 1kR, 10kR, 330R
- Um LED 5m.