

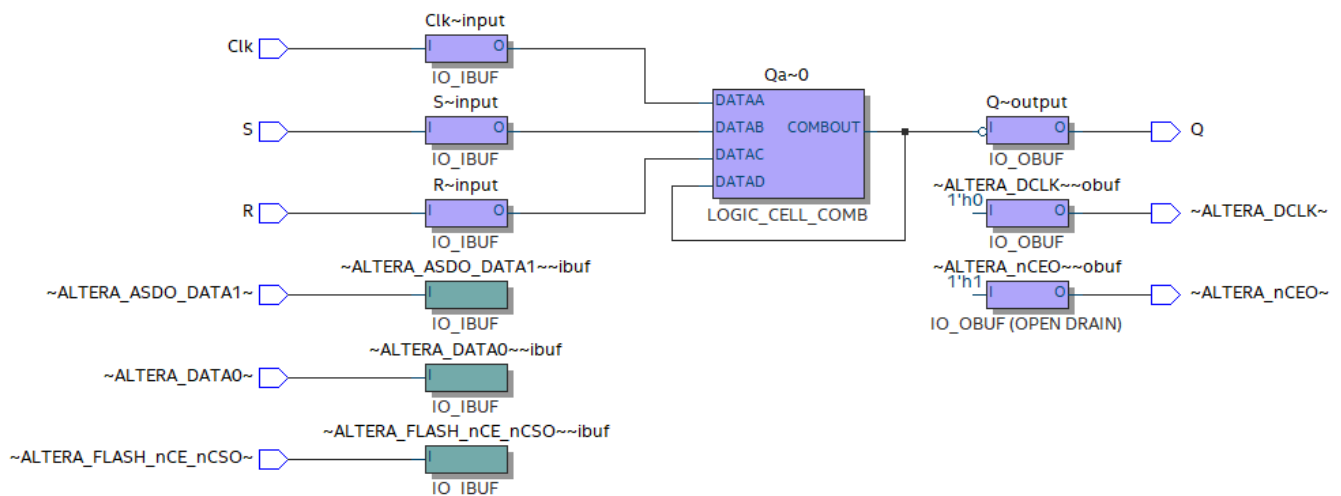
Aula 01 de Prática em Sistemas Digitais

Grupo: Bruna Izabel da Silva Pereira - 15635441

Camila Piscioneri Magalhães - 15697249

• Parte 1:

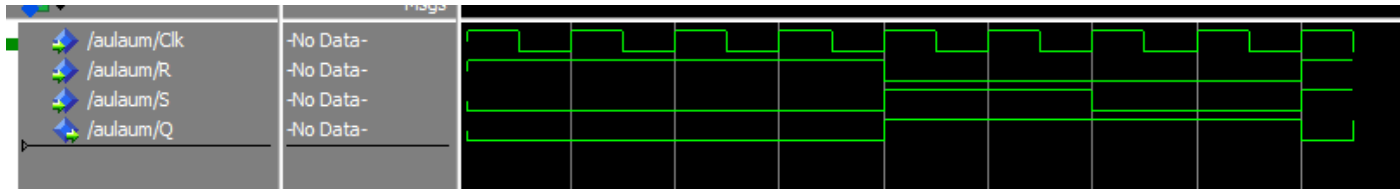
1. Crie um novo projeto Quartus para sua placa DE-series.
(Projeto disponibilizado no GitHub)
2. Gere um arquivo VHDL com o código fornecido na Figura 2 e inclua-o no projeto.
(Projeto disponibilizado no GitHub)
3. Compile o código. Utilize a ferramenta RTL Viewer do Quartus para examinar o circuito em nível de porta produzido pelo código e use a ferramenta Technology Map Viewer para verificar se o latch está implementado conforme mostrado na Figura 3b



Latch implementado conforme a figura

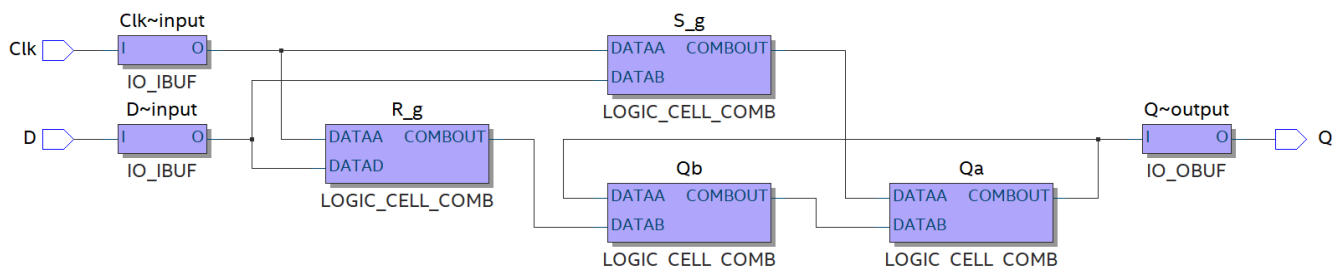
4. Simule o comportamento do código VHDL utilizando o recurso de simulação do software Quartus. Primeiro, crie um arquivo de forma vetorial (*.vwf), usando o software Quartus para especificar as entradas e saídas do circuito. Em seguida, use os comandos disponíveis na ferramenta Simulation Waveform Editor para executar a simulação do circuito. O procedimento para utilizar o Simulation Waveform Editor está descrito no tutorial "Quartus Introduction", disponível no site do Intel FPGA University Program (instruções mais

detalhadas estão no tutorial "Introduction to Quartus Simulation"). Um exemplo de arquivo de forma vetorial é mostrado na Figura 4. As formas de onda começam configurando Clk = 1 e R = 1, o que permite que a ferramenta de simulação inicialize todos os sinais dentro do latch com valores conhecidos. Se as formas de onda fornecidas à ferramenta de simulação não permitirem a inicialização de todos os sinais no circuito, a ferramenta pode sair com uma condição de erro.



• Parte 2:

1. Crie um novo projeto no Quartus. Gere um arquivo VHDL usando o estilo de código da Figura 2 para o latch D com porta. Use a diretiva KEEP para garantir que elementos lógicos separados sejam usados para implementar os sinais R, S_g, R_g, Qa e Qb. (Projeto disponibilizado no GitHub)
2. Compile seu projeto e use a ferramenta Technology Map Viewer para examinar o circuito implementado.

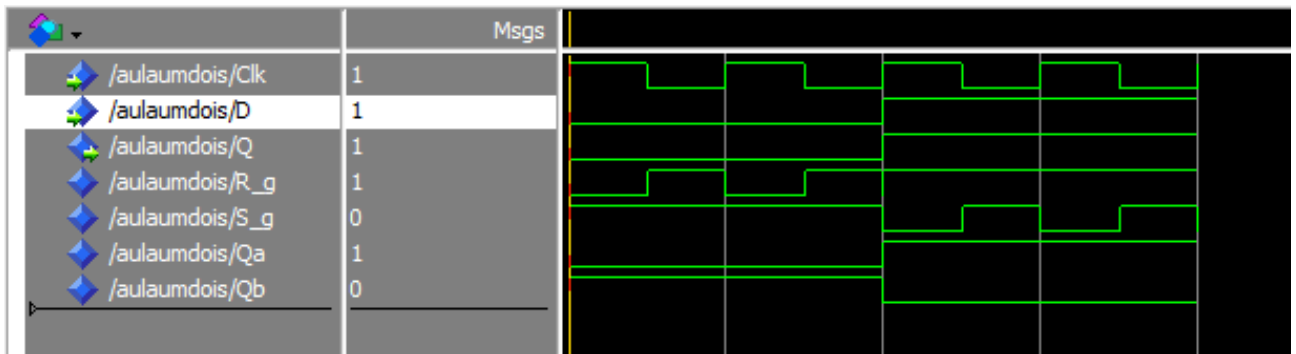


Latch implementado corretamente

3. Verifique se a trava funciona corretamente para todas as condições de entrada usando simulação funcional. Examine as características de temporização do circuito usando simulação de temporização.
4. Crie um novo projeto Quartus que será usado para implementação do gated D latch em sua série DE. Este projeto deve consistir em um módulo de nível superior que contém as portas de entrada e saída apropriadas (pinos) para sua placa. Instancie sua trava neste módulo de nível superior. Use a chave SW0 para acionar a entrada D da trava

e use SW1 como entrada Clk. Conecte a saída Q ao LEDR0.
(Projeto testado em laboratório; Projeto disponibilizado no GitHub)

- Inclua as atribuições de pinos necessárias e então compile seu projeto e baixe o circuito compilado em sua placa da série DE.
(Projeto disponibilizado no GitHub)
- Teste a funcionalidade do seu circuito alternando as chaves D e Clk e observando a saída Q.

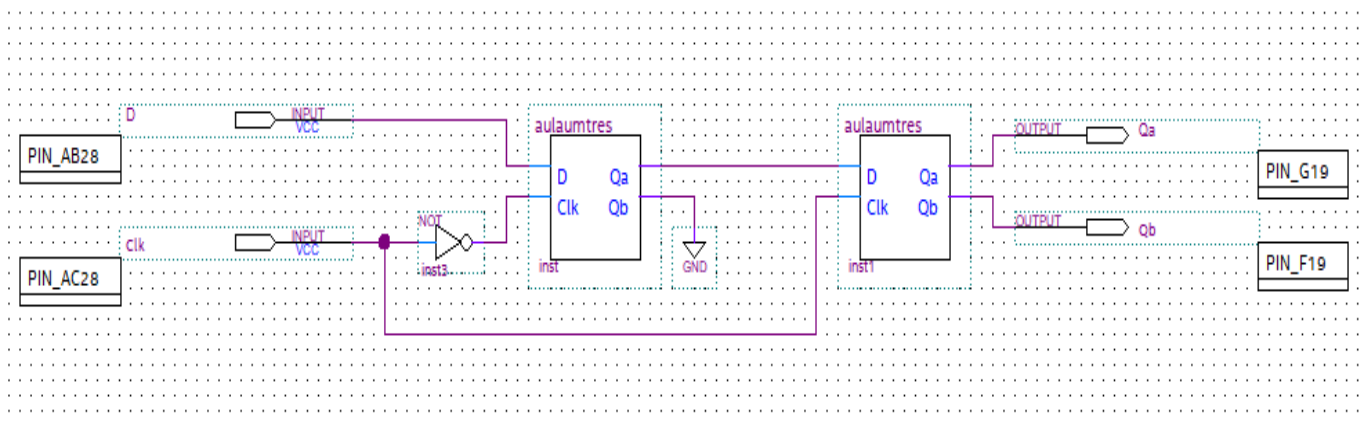


• **Parte 3:**

- Crie um novo projeto no Quartus. Gere um arquivo VHDL que instancie duas cópias do seu módulo de latch D com porta da Parte II para implementar o flip-flop mestre-escravo.

(Projeto disponibilizado no GitHub)

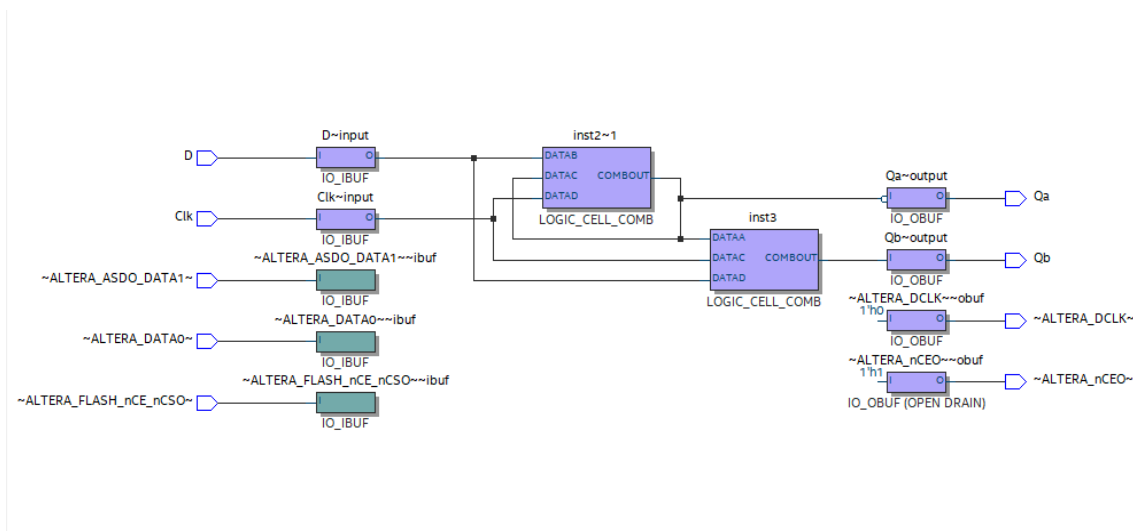
- Inclua no seu projeto as entradas e saídas apropriadas para sua placa da série DE. Use a chave SW0 para acionar a entrada D do flip-flop e a chave SW1 como a entrada de Clock. Conecte a saída Q ao LEDR0.



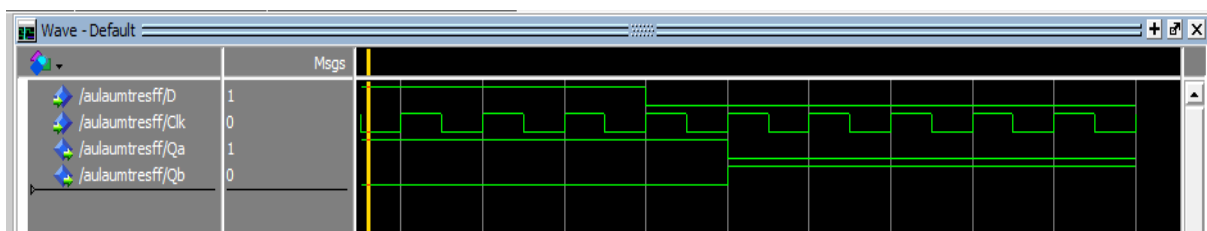
3. Inclua as atribuições de pinos necessárias e, em seguida, compile seu projeto.

(O projeto foi compilado e apresentado no laboratório e está disponível no GitHub)

4. Use o Technology Viewer para examinar o circuito do flip-flop D e utilize a simulação para verificar seu correto funcionamento.



(Technology Map Viewer)



(Simulação das ondas: infelizmente não conseguimos solucionar o problema do delay)

5. Faça o download do circuito na sua placa da série DE e teste sua funcionalidade alternando as chaves D e Clock, observando a saída Q.

(O circuito foi estado em laboratório e avaliado pelos monitores, disponível no GitHub)

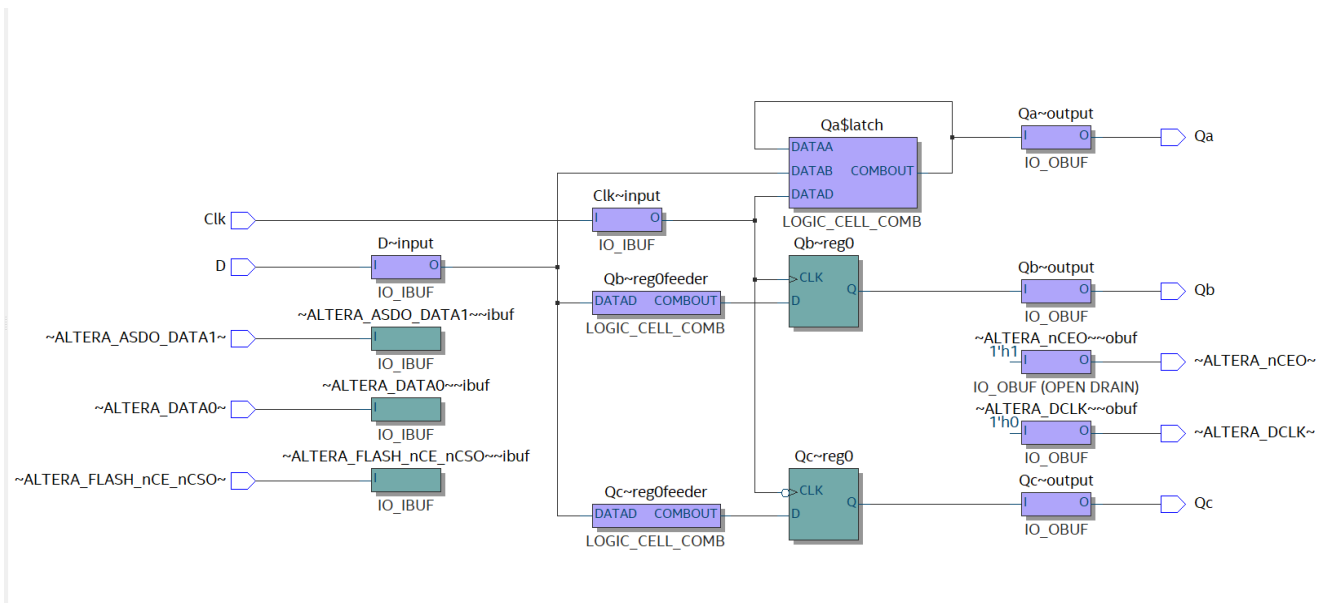
● **Parte 4:**

1. Crie um novo projeto no Quartus.

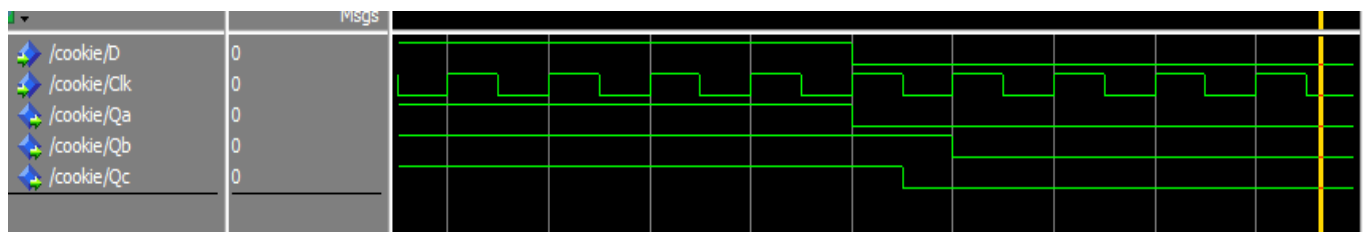
2. Escreva um arquivo VHDL que instancie os três elementos de armazenamento. Para esta parte, você não deve mais usar a diretiva KEEP (isto é, a declaração ATTRIBUTE em VHDL) das Partes I a III. A Figura 8 fornece um estilo de código VHDL comportamental que especifica o latch D com porta descrito na Figura 5. Esse latch pode ser implementado em uma tabela de busca de 4 entradas. Use um estilo de código similar para especificar os flip-flops da Figura 7.

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.all ;
3
4  ENTITY cookie IS
5  PORT ( D, clk : IN STD_LOGIC ;
6        Qa, Qb, Qc : OUT STD_LOGIC) ;
7  END cookie ;
8
9  ARCHITECTURE aulaumquatro OF cookie IS
10 BEGIN
11     PROCESS ( D, clk )
12     BEGIN
13         IF clk = '1' THEN
14             Qa <= D ;
15         END IF ;
16     END PROCESS ;
17
18     PROCESS ( D, clk )
19     BEGIN
20         IF RISING_EDGE(clk) THEN
21             Qb <= D ;
22         END IF ;
23     END PROCESS ;
24
25     PROCESS ( D, clk )
26     BEGIN
27         IF FALLING_EDGE(clk) THEN
28             Qc <= D ;
29         END IF ;
30     END PROCESS ;
31
32 END aulaumquatro;
```

3. Compile seu código e use o Visualizador de Mapa de Tecnologia (Technology Map Viewer) para examinar o circuito implementado. Verifique se o latch usa uma tabela de busca e se os flip-flops foram implementados usando os flip-flops fornecidos no FPGA alvo.



4. Crie um arquivo de forma de onda vetorial (.vwf) que especifique as entradas e saídas do circuito. Desenhe as entradas D e Clock conforme indicado na Figura 7. Use simulação funcional para obter os três sinais de saída. Observe o comportamento diferente dos três elementos de armazenamento.



(infelizmente não conseguimos arrumar o delay)