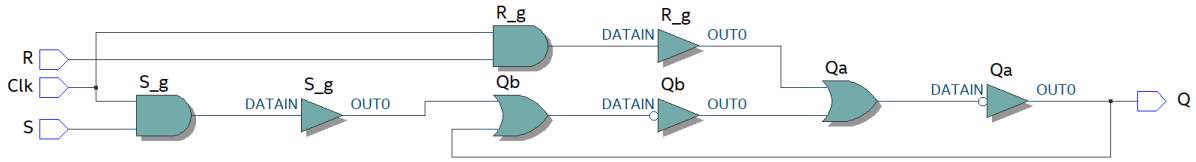
Relatório

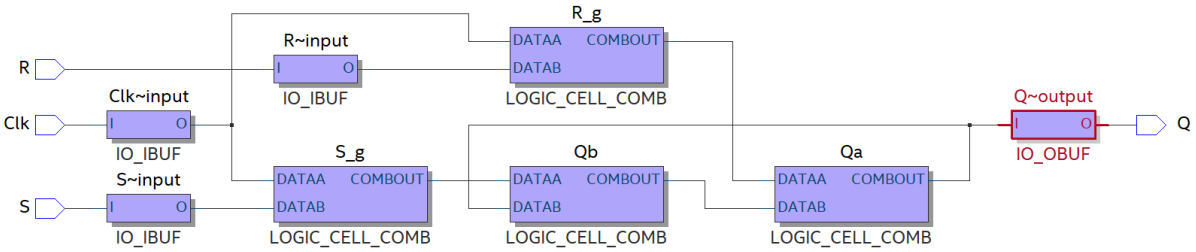
Tópico: Aula 2 & 3 - Atividade com Latch e FF
Grupo: Frederico Scheffel Oliveira 15452718
Pedro Henrique Perez Dias 15484075

Parte 1

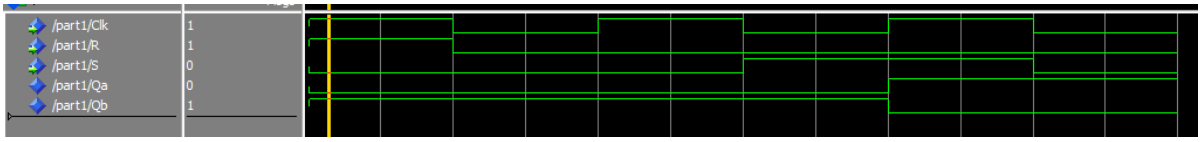
Attribute keep possibilita verificar valores intermediários de saída das portas lógicas
RTL Viewer:



Technology Map Viewer:



Simulação no ModelSim



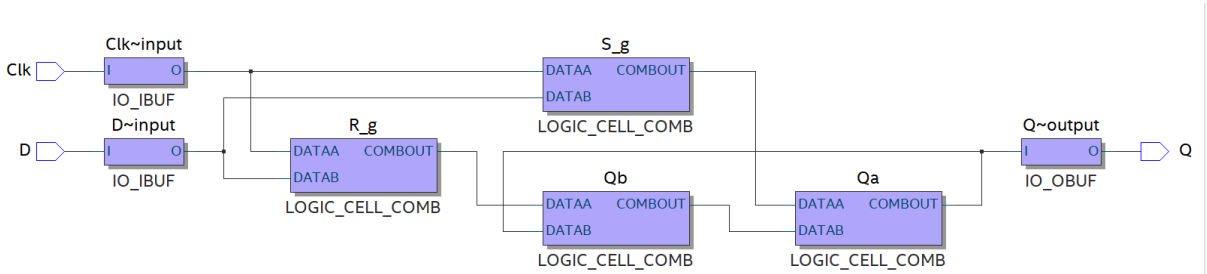
Código VHDL:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY part1 IS
PORT ( Clk, R, S : IN STD_LOGIC;
Q : OUT STD_LOGIC);
END part1;
ARCHITECTURE Structural OF part1 IS
SIGNAL R_g, S_g, Qa, Qb : STD_LOGIC ;
ATTRIBUTE KEEP : BOOLEAN;
ATTRIBUTE KEEP OF R_g, S_g, Qa, Qb : SIGNAL IS TRUE;
```

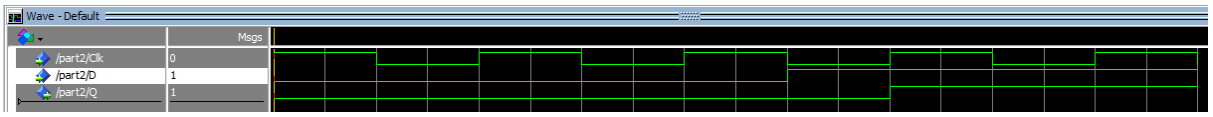
```
BEGIN
R_g <= R AND Clk;
S_g <= S AND Clk;
Qa <= NOT (R_g OR Qb);
Qb <= NOT (S_g OR Qa);
Q <= Qa;
END Structural;
```

Parte 2

Technology Map Viewer:



Simulação no ModelSim

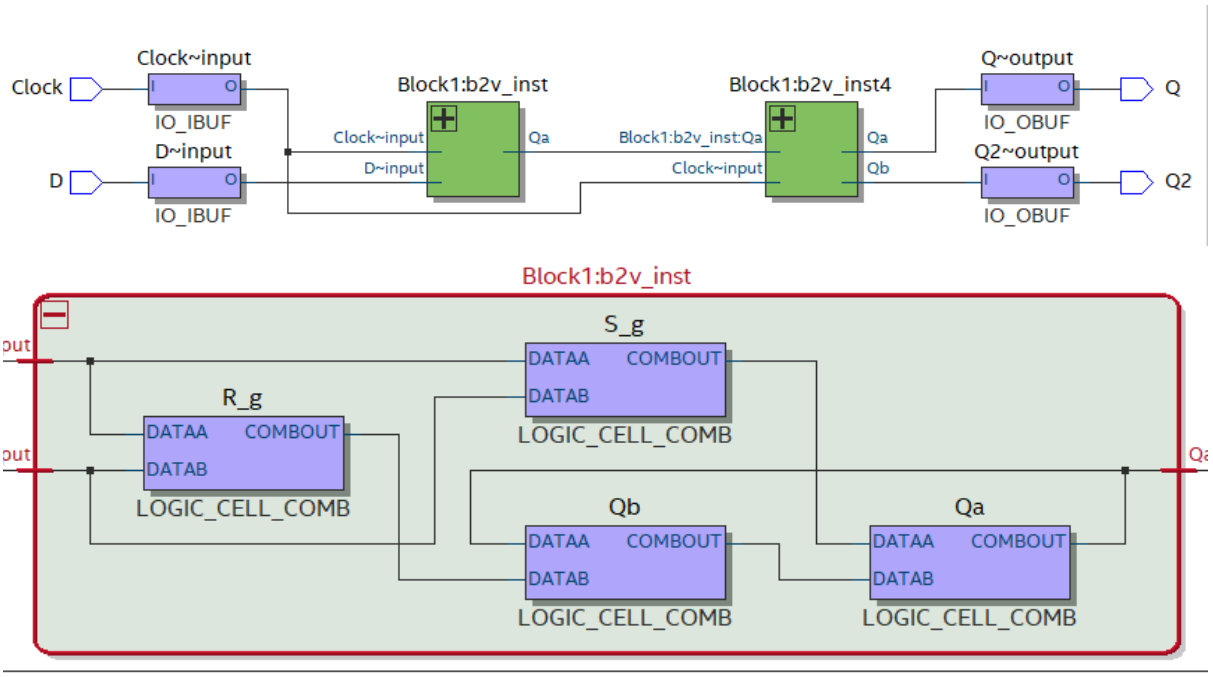


Código VHDL:

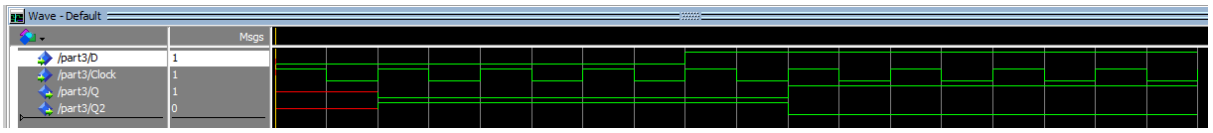
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY part2 IS
PORT ( Clk, D : IN STD_LOGIC;
Q : OUT STD_LOGIC);
END part2;
ARCHITECTURE Structural OF part2 IS
SIGNAL R_g, S_g, Qa, Qb, S, R : STD_LOGIC ;
ATTRIBUTE KEEP : BOOLEAN;
ATTRIBUTE KEEP OF R_g, S_g, Qa, Qb : SIGNAL IS TRUE;
BEGIN
S <= D;
R <= NOT (D);
R_g <= NOT (R AND Clk);
S_g <= NOT (S AND Clk);
Qa <= NOT (S_g AND Qb);
Qb <= NOT (R_g AND Qa);
Q <= Qa;
END Structural;
```

Parte 3

Technology Map Viewer:



Simulação do ModelSim



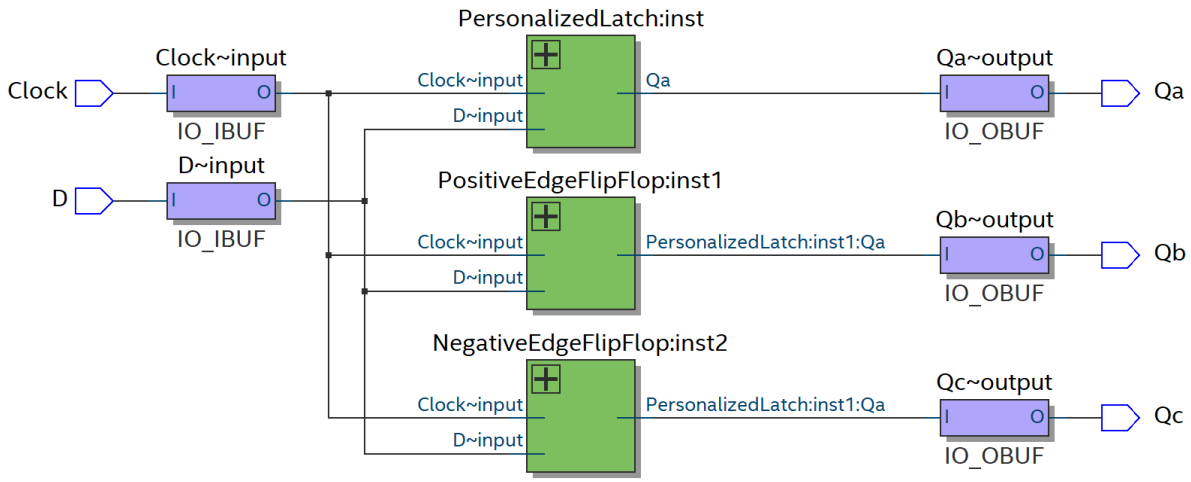
Código VHDL:

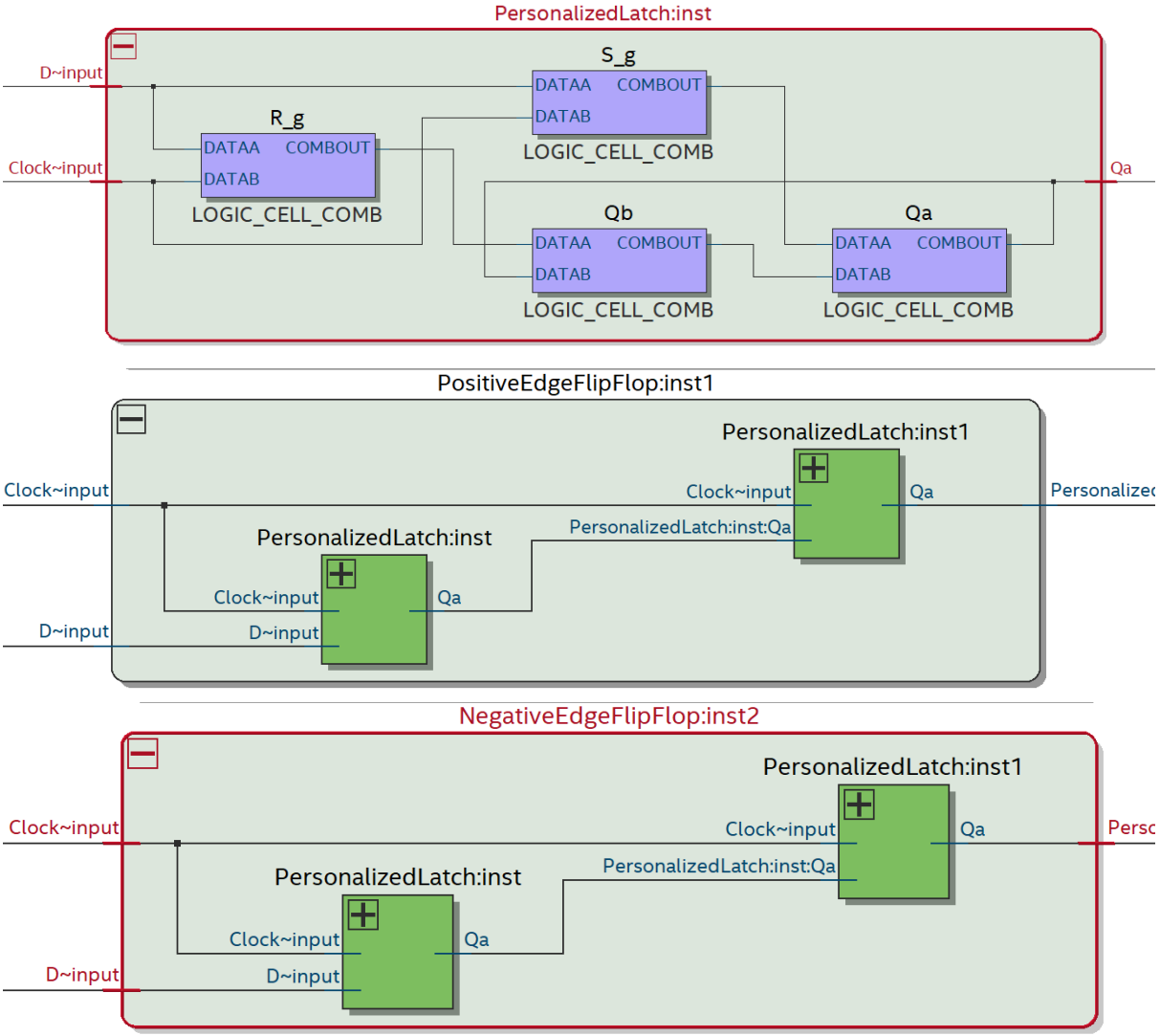
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
ENTITY part3 IS
        PORT
        (
                D :  IN  STD_LOGIC;
                Clock :  IN  STD_LOGIC;
                Q :  OUT  STD_LOGIC;
                Q2 :  OUT  STD_LOGIC
        );
END part3;
ARCHITECTURE bdf_type OF part3 IS
COMPONENT Block1
        PORT(D : IN STD_LOGIC;
                Clk : IN STD_LOGIC;
                Q : OUT STD_LOGIC;
                Q2 : OUT STD_LOGIC
```

```
        );
END COMPONENT;
SIGNAL          Qm :  STD_LOGIC;
SIGNAL          Qs :  STD_LOGIC;
SIGNAL          SYNTHESIZED_WIRE_0 :  STD_LOGIC;
BEGIN
b2v_inst : Block1
PORT MAP(D => D,
                Clk => SYNTHESIZED_WIRE_0,
                Q => Qm);
SYNTHESIZED_WIRE_0 <= NOT(Clock);
b2v_inst4 : Block1
PORT MAP(D => Qm,
                Clk => Clock,
                Q => Qs,
                Q2 => Q2);
Q <= Qs;
END bdf_type;
```

Parte 4

Technology Map Viewer:





Código VHDL:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
ENTITY part4 IS
        PORT
        (
                D :  IN  STD_LOGIC;
                Clock :  IN  STD_LOGIC;
                Qa :  OUT  STD_LOGIC;
                Qb :  OUT  STD_LOGIC;
                Qc :  OUT  STD_LOGIC
        );
END part4;
ARCHITECTURE bdf_type OF part4 IS
```

```
COMPONENT personalizedlatch
	PORT(D : IN STD_LOGIC;
		Clk : IN STD_LOGIC;
		Q : OUT STD_LOGIC
	);
END COMPONENT;
COMPONENT positiveedgeflipflop
	PORT(D : IN STD_LOGIC;
		Clock : IN STD_LOGIC;
		Q : OUT STD_LOGIC
	);
END COMPONENT;
COMPONENT negativeedgeflipflop
	PORT(D : IN STD_LOGIC;
		Clock : IN STD_LOGIC;
		Q : OUT STD_LOGIC
);
END COMPONENT;
BEGIN
b2v_inst : personalizedlatch
PORT MAP(D => D,
		Clk => Clock,
		Q => Qa);
b2v_inst1 : positiveedgeflipflop
PORT MAP(D => D,
		Clock => Clock,
		Q => Qb);
b2v_inst2 : negativeedgeflipflop
PORT MAP(D => D,
		Clock => Clock,
		Q => Qc);
END bdf_type;
```

Código VHDL personalizedLatch:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY PersonalizedLatch IS
PORT ( D, Clk : IN STD_LOGIC;
Q : OUT STD_LOGIC);
END PersonalizedLatch;
ARCHITECTURE Structural OF PersonalizedLatch IS
SIGNAL R_g, S_g, Qa, Qb, S, R : STD_LOGIC ;
ATTRIBUTE KEEP : BOOLEAN;
ATTRIBUTE KEEP OF R_g, S_g, Qa, Qb : SIGNAL IS TRUE;
BEGIN
S <= D;
R <= NOT (D);
```

```
R_g <= NOT (R AND Clk);
S_g <= NOT (S AND Clk);
Qa <= NOT (S_g AND Qb);
Qb <= NOT (R_g AND Qa);
Q <= Qa;
END Structural;
```

Código VHDL positiveEdgeFlipFlop:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
ENTITY PositiveEdgeFlipFlop IS
	PORT
	(
		D :  IN  STD_LOGIC;
		Clock :  IN  STD_LOGIC;
		Q :  OUT  STD_LOGIC
	);
END PositiveEdgeFlipFlop;
ARCHITECTURE bdf_type OF PositiveEdgeFlipFlop IS
COMPONENT personalizedlatch
	PORT(D : IN STD_LOGIC;
		 Clk : IN STD_LOGIC;
		 Q : OUT STD_LOGIC
	);
END COMPONENT;
SIGNAL	SYNTHESIZED_WIRE_0 :  STD_LOGIC;
SIGNAL	SYNTHESIZED_WIRE_1 :  STD_LOGIC;
BEGIN
b2v_inst : personalizedlatch
PORT MAP(D => D,
		 Clk => SYNTHESIZED_WIRE_0,
		 Q => SYNTHESIZED_WIRE_1);
b2v_inst1 : personalizedlatch
PORT MAP(D => SYNTHESIZED_WIRE_1,
		 Clk => Clock,
		 Q => Q);
SYNTHESIZED_WIRE_0 <= NOT(Clock);
END bdf_type;
```

Código VHDL negativeEdgeFlipFlop:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
ENTITY NegativeEdgeFlipFlop IS
	PORT
```

```
        (
                D :  IN  STD_LOGIC;
                Clock :  IN  STD_LOGIC;
                Q :  OUT  STD_LOGIC
        );
END NegativeEdgeFlipFlop;
ARCHITECTURE bdf_type OF NegativeEdgeFlipFlop IS
COMPONENT personalizedlatch
        PORT(D : IN STD_LOGIC;
                Clk : IN STD_LOGIC;
                Q : OUT STD_LOGIC
        );
END COMPONENT;
SIGNAL          SYNTHESIZED_WIRE_0 :  STD_LOGIC;
SIGNAL          SYNTHESIZED_WIRE_1 :  STD_LOGIC;
BEGIN
b2v_inst : personalizedlatch
PORT MAP(D => D,
                Clk => Clock,
                Q => SYNTHESIZED_WIRE_0);
b2v_inst1 : personalizedlatch
PORT MAP(D => SYNTHESIZED_WIRE_0,
                Clk => SYNTHESIZED_WIRE_1,
                Q => Q);
SYNTHESIZED_WIRE_1 <= NOT(Clock);
END bdf_type;
```

Simulação no ModelSim: