Prática Nº8 - SEL0606

Alunos (Bancada 5):
Vitor Alexandre Garcia Vaz - 14611432
Gabriel Dezejácomo Maruschi - 14571525

Sumário

Lista de Imagens

1 Introdução

1.1 Objetivos

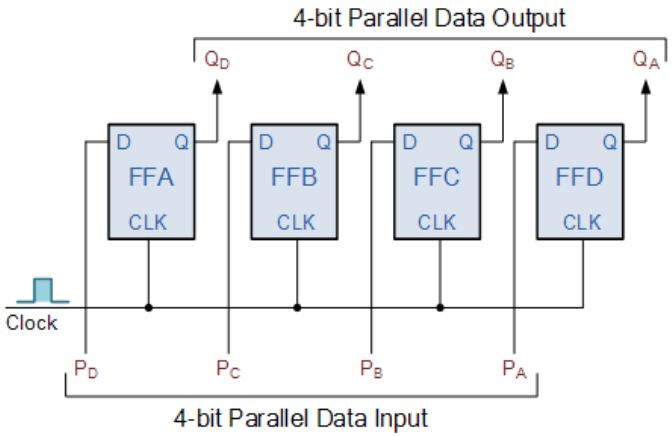
Nesta prática do Laboratório de Sistemas Digitais, implementamos um registrador de N bits, com clock e enable, utilizando a linguaguem VHDL no software Quartus, e testamos no kit DE10-LITE (MAX 10 10M50DAF484C7G) bem como no software ModelSim.

Para isso, visamos elaborar o registrador através de uma associção de flip-flops tipo D em paralelo (ligados ao mesmo sinal de clock), sendo cada flip-flop responsável pelo registro de um dos N bits pertencentes à entrada do registrador, e tendo como saída Q tal bit registrado.

1.2 Registrador

De maneira geral, registradores são circuitos sequenciais que formam a memória do computador, podendo ser, ou não, visíveis ao programador. Nesse sentido, os registradores tem como principal função guardar um dado de N bits em sua saída através, geralmente, de um conjunto de flip-flops tipo D associados em paralelo.

Figure 1: Esquemático de registrador paralelo



Fonte: Site da internet

Além disso, tais registradores funcionam conforme um sinal de clock, podendo possuir sinais de reset e enable para ditarem o seu funcionamento a partir de um certo momento.

FF D
O Flip-Flop tipo D possui apenas as entradas **D** e **Clk** e comporta-se espelhando o sinal de D na saída Q nos eventos de borda de subida do clock. A tabela verdade é demonstrada abaixo.

Figure 2: Tabela verdade do FF-D

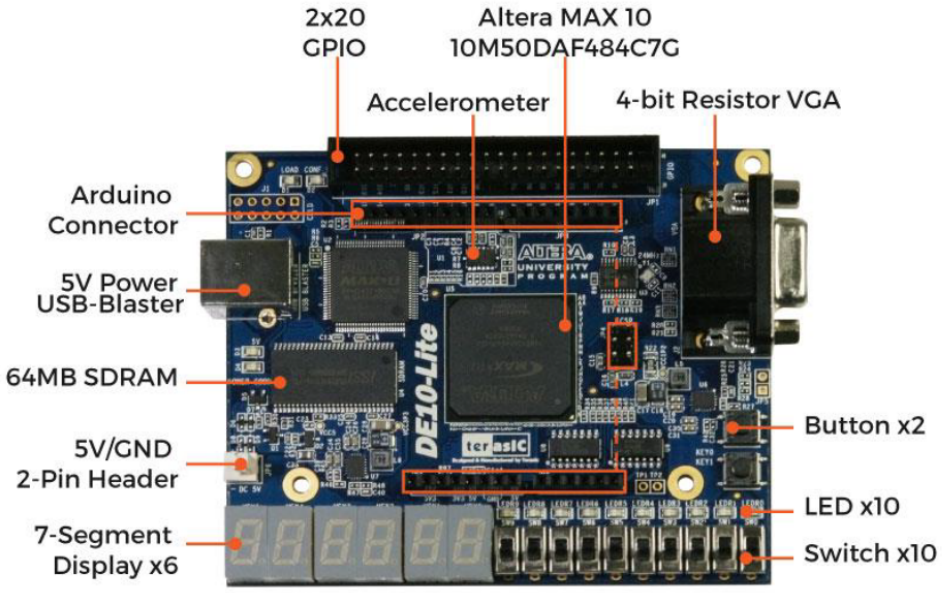
D	*Q*	*C*	*Q**	
0	0	⎍	0	Store 0
0	1	⎍	0	
1	0	⎍	1	Store 1
1	1	⎍	1	

Fonte: V. P. Nelson, *Digital Logic Circuit analysis and design.* 1995

2 Materiais e métodos

O código utilizado para desenvolver os módulos dos Flip-Flops e do Registrador foi escrito em VHDL e compilado no Quartus. A fim de testá-los no DE10-LITE, desenvolvemos também um interface cujo objetivo foi redirecionar os sinais dos pinos para os módulos e vice-versa. Utilizamos também o software ModelSim da Intel para visualizar as ondas de entrada e saída do Registrador desenvolvido.

Figure 3: Kit DE10-LITE



Fonte: DE10-Lite User Manua

2.1 Interface

A interface, codificada como é mostrado abaixo, direciona as entradas e saídas para as portas e leds do kit DE10-LITE:

Figure 4: Código da interface

```
1  -- Version: 1.1
2  -- Date: 07/11/2024
3  -- Owners: Gabriel D. Maruschi
4  --            Vitor Garcia Vaz
5
6  ENTITY DE10_LITE_Reg IS
7  PORT(
8          SW  : IN BIT_VECTOR (9 DOWNTO 0);
9          KEY : IN BIT_VECTOR (1 DOWNTO 0);
10         LEDR    : OUT BIT_VECTOR (7 DOWNTO 4)
11     );
12 END DE10_LITE_Reg;
13
14 ARCHITECTURE interface OF DE10_LITE_Reg IS
15 BEGIN
16     REG: entity work.Reg
17         GENERIC MAP(
18         n => 4)
19         PORT MAP(
20         D => SW(7 DOWNTO 4), clk => KEY(0), cr => KEY(1), en => SW(9), Q => LEDR(7 DOWNTO 4));
21 END interface;
```

Fonte: os autores

Para a chamada do módulo do registrador, as entradas de Clock e Clear, **clk** e **cr**, respectivamente, foram direcionados para os botões 0 e 1 do kit. O barramento **D** tem os bits selecionados pelas chaves de 4 a 7, e o enable (**en**) é controlado pela chave 9. A saída pôde ser visualizada nos LEDs 4 a 7 do kit. Vale ressaltar o **GENERIC MAP** especificado na chamada do registrador, nele definimos o tamanho do registrador paralelo ao fornecer o valor 4 para a variável genérica **n**. Este conceito será demostrado na definição do registrador.

2.2 Registrador

O Registrador paralelo desenvolvido em VHDL conta com o uso do loop FOR-GENERATE disponível na linguagem para fazer **n** chamadas da entidade de um Flip-Flop Tipo-D.
Além disso, para fins de generalidade, definimos o inteiro genérico n. Este possibilita que chamadas do módulo do registrador escolham o barramento deste. O valor *default* de **n** é 4.
Há também no registrador o sinal de enable e o clear assíncrono. Nesta implementação, $\mathbf{en} = 0$ anula o clock e não transmite alterações para a saída. Entretanto $\mathbf{cr} = 1$ gera $\mathbf{Q} = 0$ independente de **en**.

Figure 5: Código do Registrador Paralelo

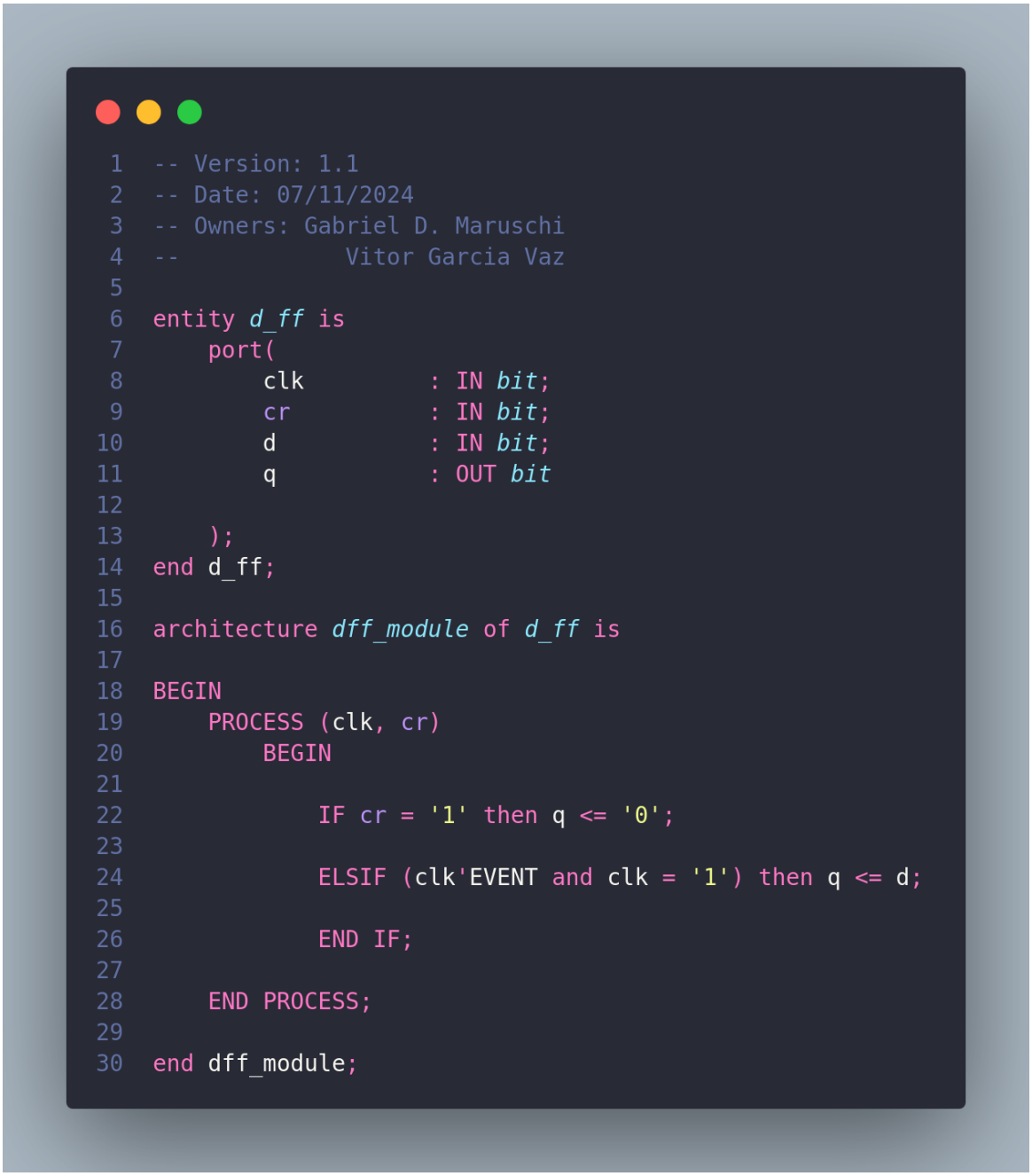
```
1  -- Version: 1.1
2  -- Date: 07/11/2024
3  -- Owners: Gabriel D. Maruschi
4  --            Vitor Garcia Vaz
5
6  ENTITY Reg IS
7      GENERIC (n      :   INTEGER :=  4);
8      PORT (clk   :   IN  BIT;
9              en      :   IN BIT;
10             cr      :   IN BIT;
11             D       :   IN BIT_VECTOR (n-1 DOWNTO 0);
12             Q       :   OUT BIT_VECTOR (n-1 DOWNTO 0));
13 END Reg;
14
15 ARCHITECTURE estrutural OF Reg IS
16
17     SIGNAL aux  : BIT;
18
19 BEGIN
20
21     aux <= clk AND en;
22
23     regis_gen: FOR i IN 0 TO n-1 GENERATE
24
25             FF: entity work.d_ff
26                 PORT MAP(
27                     clk => aux, cr => cr, d => D(i), q => Q(i)
28                     );
29
30         END GENERATE regis_gen;
31
32 END estrutural;
33
```

Fonte: os autores

2.3 Flip-Flop Tipo D

O Flip-Flop tipo D também possui as entradas citadas na introdução e a entrada de Clear assíncrono. Na arquitetura, utilizamos também a contrução com PROCESS e IF, ELSIF. Quando as condições de clock são atingidas, **q** recebe o sinal de **d**, assim como deve ser.

Figure 6: Código do FF D



```vhdl
-- Version: 1.1
-- Date: 07/11/2024
-- Owners: Gabriel D. Maruschi
--            Vitor Garcia Vaz

entity d_ff is
    port(
        clk         : IN bit;
        cr          : IN bit;
        d           : IN bit;
        q           : OUT bit

    );
end d_ff;

architecture dff_module of d_ff is

BEGIN
    PROCESS (clk, cr)
        BEGIN

            IF cr = '1' then q <= '0';

            ELSIF (clk'EVENT and clk = '1') then q <= d;

            END IF;

    END PROCESS;

end dff_module;
```

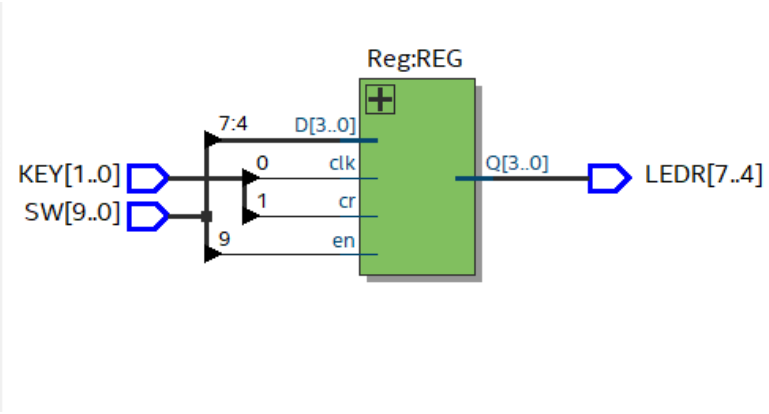
Fonte: os autores

3 Conclusão

3.1 Cicuito RTL

A partir do código em VHDL feito em laborratório, e usando a ferramenta de síntese disponibilizada pelo quartus, o seguinte circuito, referente à interface entre registrador e os pinos da FPGA, foi obtido:

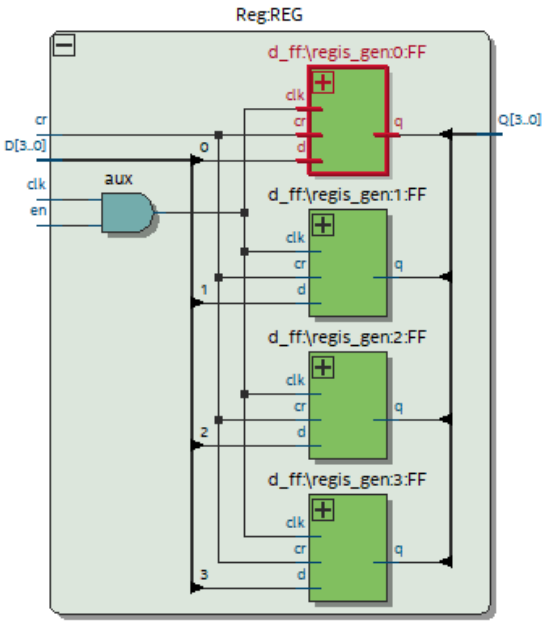
Figure 7: Visuzalização do circuito referente à interface



Fonte: os autores

Por outro lado, referentemente à estrutura interna do registrador paralelo, o seguinte circuito foi obtido:

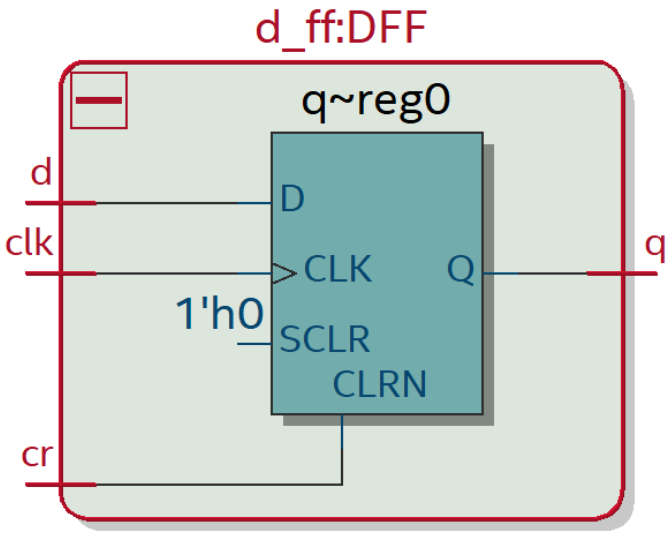
Figure 8: Visuzalização do circuito referente ao registrador



Fonte: os autores

Ademais, para cada flip-flop tipo D associado em paralelo na formação do registrador, obtemos o seguinte circuito:

Figure 9: Visualização do circuito referente ao Flip-Flop tipo D

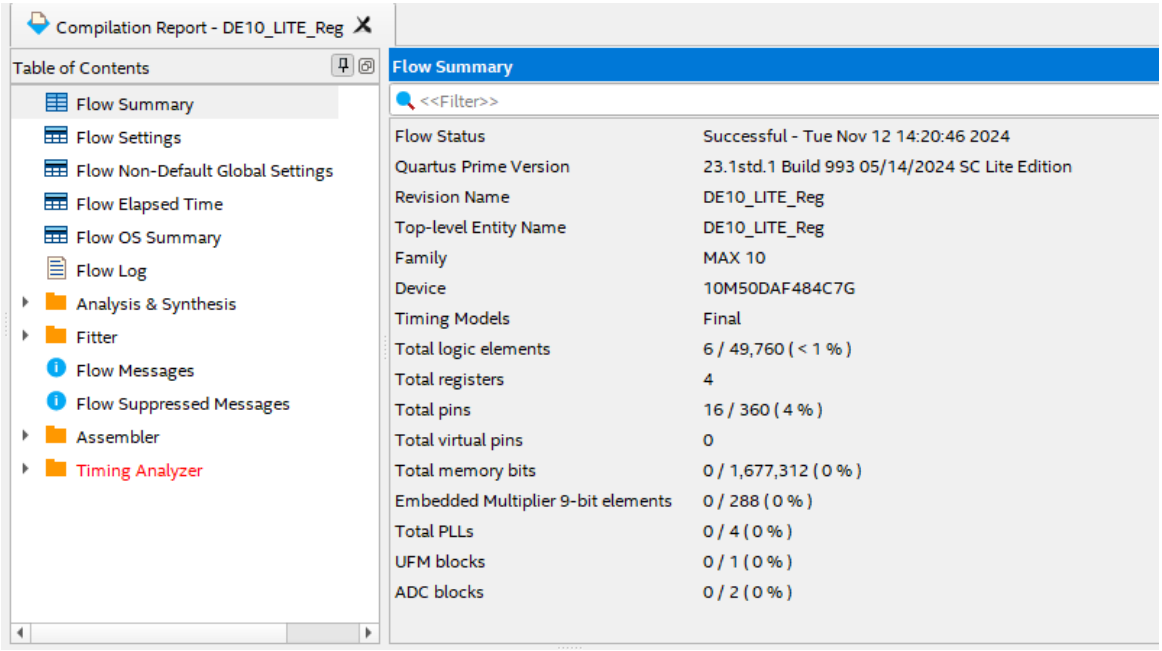


Fonte: os autores

3.2 Número de células lógicas

Ademais, por meio do resumo do funcionamento e constituição do circuito descrito em VHDL no software (fig10), chegamos à conclusão de que o circuito sintetizado apresentou um uso de 6 elentos lógicas e 16 pinos do dispositivo reconfigurável (FPGA).

Figure 10: Resumo de funcionamento



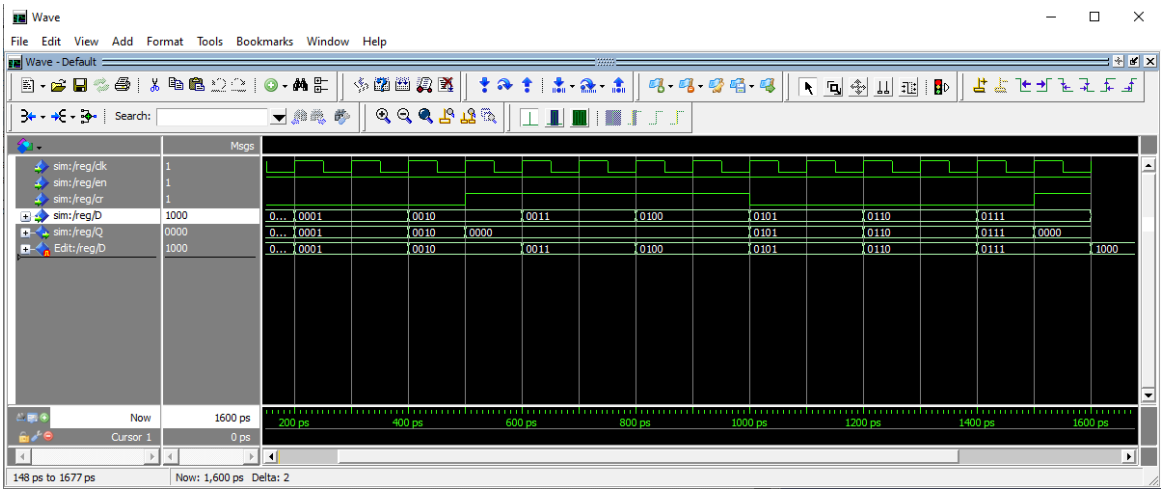
Fonte: os autores

3.3 Funcionamento do Registrador Paralelo

A respeito do funcionamento do registrador elaborado pela turma, utilizamos o software de ModelSim para simulá-lo e comprovar e validar a sua funcionalidade.

Nesse sentido, tal funcionamento pode ser observado através da fig.11, á medida em que a saída do registrador, a cada ciclo de clock, correspondeu a entrada D de 4 bits neste mesmo ciclo enquanto o reset estava desativado ($cr = 0$). Porém, após o ativamento do reset (em $500ps$ de simulação) a saída permaneceu nula ($Q = 0000_2$) até o sucetivo ativamento do reset, o qual ocorreu aos $1000ps$ de simulação, comprovando o funcionamento do reset associado ao registrador elaborado.

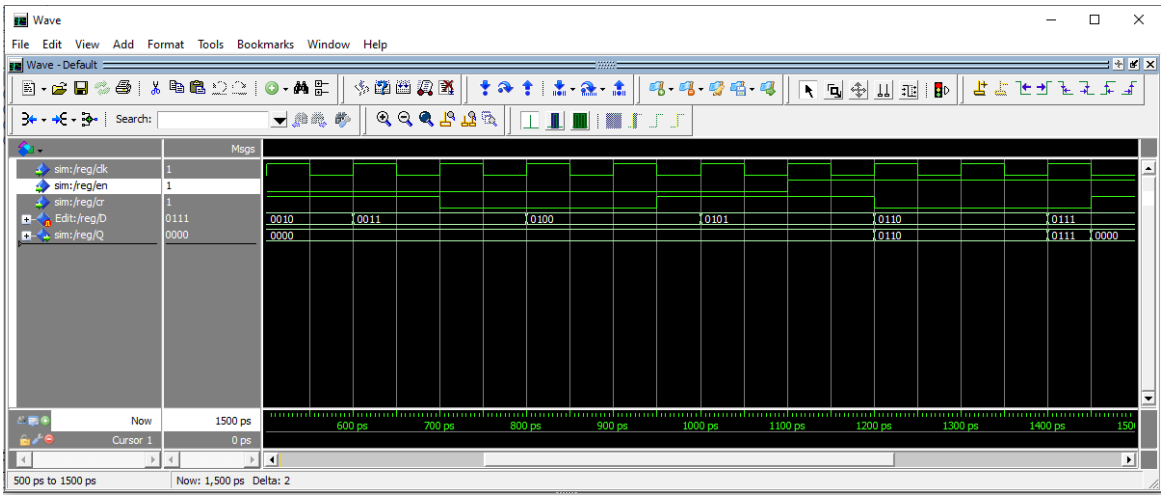
Figure 11: Primeira simulação do registrador elaborado



Fonte: os autores

Além disso, tal funcionamento também foi evidenciado através da simulação associada à fig.12, já que, através de sua análise, é possível observar que o registrador só passou a salvar os 4 bits da entrada D em sua saída a patir da primeira borda de clock positiva posterior ao ativamento do enable ($en = 1$) em $1100ps$ de simulação, e também ao desativamento do reset em $1200ps$ de simulação. Fato o qual evidencia a funcionalidade do sinal de enable.

Figure 12: Segunda simulação do registrador elaborado

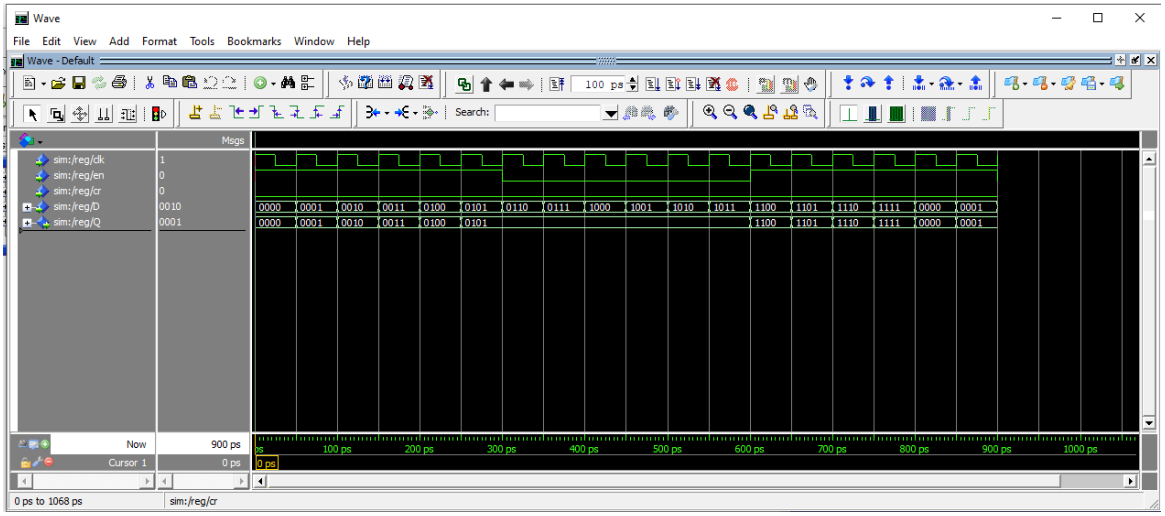


Fonte: os autores

Por fim, vale ressaltar o correto funcionamento sequencial do circuito, pois, em ambas as simulações e quando $cr = 0$ e $en = 1$, a saída do registrador Q corresponde à entrada D do atual ciclo de clock. Porém, quando o

enable é desativado, o clock do registrador também permanece no nível lógico 0 e, consequentemente, a saída Q é apenas mantida (salva), sem assumir o mesmo valor de D, conforme o visto na fig.13 associada à terceira simulação realizada.

Figure 13: Terceira simulação do registrador elaborado



Fonte: os autores