

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

PCS3635 – LABORATÓRIO DIGITAL I

EXPERIÊNCIA 2 – Um Fluxo de Dados Simples

Relato da Bancada A3 – Turma 2 – Prof. Reginaldo Arakaki

Data de Emissão: 13 de JANEIRO DE 2025.

Nome: Pedro Henrique Zanato da Costa	Número USP: 13874761
Nome: Enzo Koichi Jojima	Número USP: 14568285
Nome: Eduardo Ribeiro do Amparo Rodrigues de Souza	Número USP: 14567346

1. ATIVIDADE 1 - ESTUDO E FAMILIARIZAÇÃO COM DESCRIÇÕES VERILOG.

Nesta etapa iremos conhecer os componentes a serem utilizados nas atividades seguintes. São eles um contador de 4bits e um comparador de magnitude. Os componentes e seus respectivos códigos em Verilog estão descritos a seguir, assim como planos de testes e seus respectivos resultados.

a. Componente contador_163

```
module contador_163 ( CLK, CLR, LD, ENP, ENT, D, Q, RCO );
    input CLK, CLR, LD, ENP, ENT;
    input [3:0] D;
    output reg [3:0] Q;
    output reg RCO;
```

```
always @ (posedge CLK) // Create the counter f-f behavior
    if (~CLR)           Q <= 4'd0;
    else if (~LD)        Q <= D;
    else if (ENT && ENP) Q <= Q + 1;
    else                 Q <= Q;
```

```
always @ (Q or ENT) // Create RCO combinational output
```

```

if (ENT && (Q == 4'd15)) RCO = 1;
else RCO = 0;
endmodule

```

O componente foi analisado pelo código em Verilog e também com sua esquematização no Digital. Trata-se de um contador integral síncrono com reset (CLR) e load (LD) em ativo baixo.

Tabela 1: Plano de Teste para o Componente "contador_163"

Teste	Descrição	Sinais de Entradas	Saídas esperadas
condições iniciais	Todas as entradas inativadas	CLR=1, LD=1, D=0000, ENP=0, ENT=0, CLOCK=0	Q=0000, RCO=0
1	Aciona clock 2x com entradas inativadas	CLR=1, LD=1, D=0000, ENP=0, ENT=0, acionar CLOCK 2 vezes	Q=0000, RCO=0
2	Aciona clock (somente) com clear ativado	CLR=0, LD=1, D=0000, ENP=0, ENT=0, acionar CLOCK	Q=0000, RCO=0
3	Aciona clock 5x com sinais de enable ativados	CLR=1, LD=1, D=0000, ENP=1, ENT=1, acionar CLOCK 5 vezes	Q=0101, RCO=0
4	Aciona clock com load ativado e dado=1011	CLR=1, LD=0, D=1011, ENP=0, ENT=0, acionar CLOCK	Q=1011, RCO=0
5	Aciona clock 4x com sinais de enable ativados	CLR=1, LD=1, D=0000, ENP=1, ENT=1, acionar CLOCK 4 vezes	Q=1111, RCO=1
6	Aciona clock 2x com enp=0 e ent=1	CLR=1, LD=1, D=0000, ENP=0, ENT=1, acionar CLOCK 2 vezes	Q=1111, RCO=1
7	Aciona clock 2x com enp=1 e ent=0	CLR=1, LD=1, D=0000, ENP=1, ENT=0, acionar CLOCK 2 vezes	Q=1111, RCO=0
8	Aciona clock 2x com sinais de enable ativados	CLR=1, LD=1, D=0000, ENP=1, ENT=1, acionar CLOCK 2 vezes	Q=0001, RCO=0
9	Aciona clock com clear e load ativados e dado=1001	CLR=0, LD=0, D=1001, ENP=1, ENT=1, acionar CLOCK	Q=0000, RCO=0
10	Aciona clock com load ativado e dado=1011	CLR=1, LD=0, D=1001, ENP=1, ENT=1, acionar CLOCK	Q=1001, RCO=0
11	Aciona clock 6x com sinais de enable ativados	CLR=1, LD=1, D=1001, ENP=1, ENT=1, acionar CLOCK 6 vezes	Q=1111, RCO=1

1. Resultado observado corresponde à saída esperada;
2. Resultado observado corresponde à saída esperada;
3. Resultado observado corresponde à saída esperada;
4. Resultado observado corresponde à saída esperada;
5. Resultado observado corresponde à saída esperada;
6. Resultado observado corresponde à saída esperada;
7. Resultado observado corresponde à saída esperada;
8. Resultado observado corresponde à saída esperada;
9. Resultado observado corresponde à saída esperada;
10. Resultado observado corresponde à saída esperada;
11. Resultado observado corresponde à saída esperada;

b. Componente comparador_85

```
module comparador_85 (ALBi, AGBi, AEBi, A, B, ALBo, AGBo, AEBo);
```

```

input[3:0] A, B;
input ALBi, AGBi, AEBi;
output ALBo, AGBo, AEBo;
wire[4:0] CSL, CSG;

assign CSL = ~A + B + ALBi;
assign ALBo = ~CSL[4];
assign CSG = A + ~B + AGBi;
assign AGBo = ~CSG[4];
assign AEBo = ((A == B) && AEBi);

```

```
endmodule
```

A análise do componente foi feita de maneira semelhante à realizada no ítem a. Trata-se de um comparador de magnitude combinatório.

Para o componente comparador_85, completamos os espaços em branco do Plano de Testes com os resultados observados para cada execução.

Tabela 2: Plano de Teste para o Componente "comparador_85"

Teste	Sinais de Entradas	Saídas esperadas
condições iniciais	A>Bin=0, A=Bin=1, A<Bin=0, A=0000, B=0000	A>Bout=0, A=Bin=1, A<Bout=0
1	A>Bin=0, A=Bin=1, A<Bin=0, A=0110, B=0110	A>Bout=0, A=Bin=1, A<Bout=0
2	A>Bin=1, A=Bin=0, A<Bin=0, A=0110, B=0110	A>Bout=1, A=Bin=0, A<Bout=0
3	A>Bin=0, A=Bin=0, A<Bin=1, A=0110, B=0110	A>Bout=0, A=Bin=0, A<Bout=1
4	A>Bin=0, A=Bin=1, A<Bin=0, A=0001, B=0000	A>Bout=? , A=Bin=? , A<Bout=?
5	A>Bin=1, A=Bin=0, A<Bin=0, A=0001, B=0000	A>Bout=? , A=Bin=? , A<Bout=?
6	A>Bin=0, A=Bin=0, A<Bin=1, A=0001, B=0000	A>Bout=? , A=Bin=? , A<Bout=?
7	A>Bin=0, A=Bin=1, A<Bin=0, A=0011, B=1100	A>Bout=? , A=Bin=? , A<Bout=?
8	A>Bin=1, A=Bin=0, A<Bin=0, A=0011, B=1100	A>Bout=? , A=Bin=? , A<Bout=?
9	A>Bin=0, A=Bin=1, A<Bin=0, A=0011, B=1100	A>Bout=? , A=Bin=? , A<Bout=?

1. Resultado observado corresponde à saída esperada;
2. Resultado observado corresponde à saída esperada;
3. Resultado observado corresponde à saída esperada;
4. A>Bout=1, A=Bout=0, A<Bout=0;
5. A>Bout=1, A=Bout=0, A<Bout=0;
6. A>Bout=1, A=Bout=0, A<Bout=0;
7. A>Bout=0, A=Bout=0, A<Bout=1;
8. A>Bout=0, A=Bout=0, A<Bout=1;
9. A>Bout=0, A=Bout=0, A<Bout=1;

2. ATIVIDADE 2 - PROJETO E SIMULAÇÃO DE UM FLUXO DE DADOS

Essa etapa conciste no projeto de um fluxo de dados envolvendo os dois componentes estudados na etapa anterior. O fluxo de dados seguirá o modelo da figura 1.

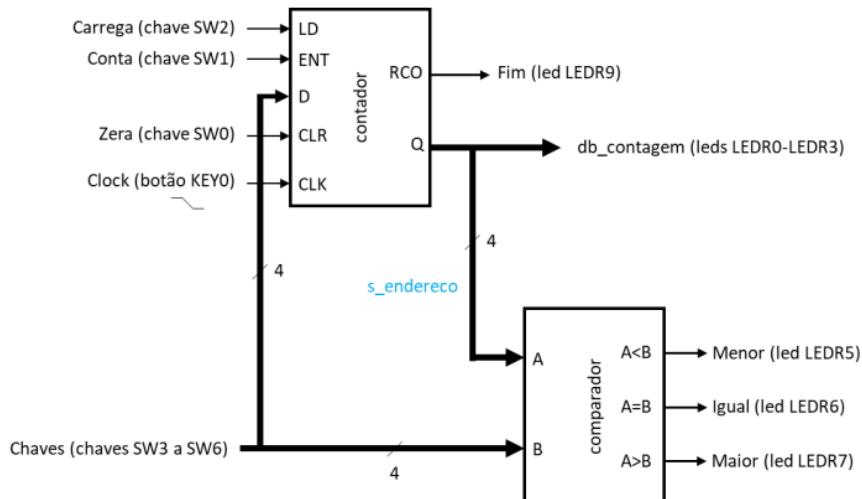


Figura 1: Diagrama de blocos do circuito a ser implementado na Atividade 2

O resultado do código em verilog para o fluxo de dados está representado a seguir:

```
module circuito_exp2_ativ2 (clock, zera, carrega, conta, chaves,
                           menor, maior, igual, fim, db_contagem);
    input      clock;
    input      zera;
    input      carrega;
```

```

input      conta;
input [3:0] chaves;
output     menor;
output     maior;
output     igual;
output     fim;
output [3:0] db_contagem;

wire [3:0] s_contagem; // sinal interno para interligacao dos componentes

// contador_163
contador_163 contador (
    .clock( clock ),
    .clr ( ~zera ),
    .ld  ( ~carrega ),
    .ent ( 1'b1 ),
    .enp ( conta ),
    .D   ( chaves ),
    .Q   ( s_contagem ),
    .rco ( fim )
);

// comparador_85
comparador_85 comparador (
    .A   ( s_contagem ),
    .B   ( chaves ),
    .ALBi( 1'b0 ),
    .AGBi( 1'b0 ),
    .AEBi( 1'b1 ),
    .ALBo( menor ),
    .AGBo( maior ),
    .AEBo( igual )
);

// saida de depuracao
assign db_contagem = s_contagem;

endmodule

```

A equipe montou os circuitos abaixo no Digital para seguir com os testes propostos.

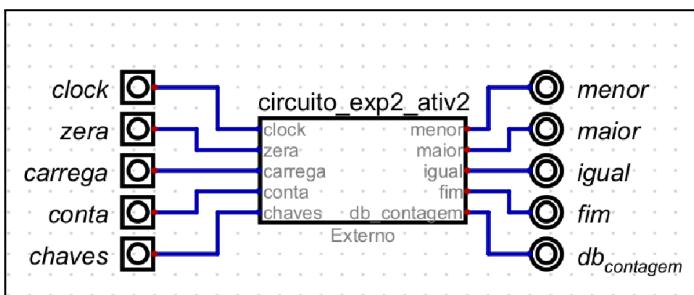


Figura 2: Circuito 1

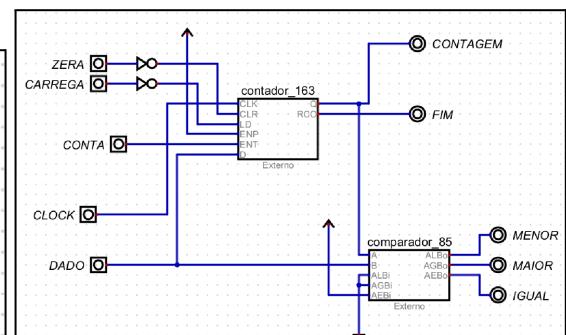


Figura 3: Circuito 2

Os testes e os respectivos resultados estão descritos a seguir:

Tabela 3: Plano de Testes para o Circuito Digital da Atividade 2

#	Caso de teste	Sinais de controle	Resultado esperado
c.i.	<i>Condições iniciais</i>	clock=1 zera=0 carrega=0 conta=0 chaves=0000	contagem=0, fim=0, maior=0, menor=0, igual=1
1	<i>Zerar contador e observar a saída da contagem</i>	zera=1 clock ↑	contagem=0, fim=0, maior=0, menor=0, igual=1
2	<i>Ajustar chaves para 0001</i>	chaves=0001	contagem=0, fim=0, maior=0, menor=1, igual=0
3	<i>Incrementar contador e chaves=0001</i>	conta=1 clock ↑	contagem=1, fim=0, maior=0, menor=0, igual=1
4	<i>Incrementar contador para 3 e ajustar chaves para 0010</i>	conta=1 clock ↑ (2x) chaves=0010	contagem=3, fim=0, maior=1, menor=0, igual=0
5	<i>Ajustar chaves para 0110</i>	chaves=0110	contagem=3, fim=0, maior=0, menor=1, igual=0
6	<i>Incrementar contador até 1110</i>	conta=1 clock ↑ (11x)	contagem=14, fim=0, maior=1, menor=0, igual=0
7	<i>Incrementar contador</i>	conta=1 clock ↑	contagem=15, fim=1, maior=1, menor=0, igual=0

Tabela 4: Resultados dos Testes da Atividade 2

Teste	Circuito 1	Circuito 2
1	contagem=0,fim=0,maior=0,menor=0,igual=1	contagem=0,fim=0,maior=0,menor=0,igual=1
2	contagem=0,fim=0,maior=0,menor=1,igual=0	contagem=0,fim=0,maior=0,menor=1,igual=0
3	contagem=1,fim=0,maior=0,menor=0,igual=1	contagem=1,fim=0,maior=0,menor=0,igual=1
4	contagem=3,fim=0,maior=1,menor=0,igual=0	contagem=3,fim=0,maior=1,menor=0,igual=0
5	contagem=3,fim=0,maior=0,menor=1,igual=0	contagem=3,fim=0,maior=0,menor=1,igual=0
6	contagem=14(E),fim=0,maior=1,menor=0,igual=0	contagem=14(E),fim=0,maior=1,menor=0,igual=0
7	contagem=15(F),fim=1,maior=1,menor=0,igual=0	contagem=15(F),fim=1,maior=1,menor=0,igual=0

3. ATIVIDADE 3 - SÍNTESE E TESTE DO CIRCUITO NA PLACA FPGA

Nessa etapa, seguimos o tutorial em vídeo disponível no Moodle ([link](#)) do professor Edson Midorikawa para a criação e configuração do Intel Quartus. A configuração seguiu como o esperado e, com ajuda da monitoria, o projeto foi sintetizado na placa FPGA com facilidade.

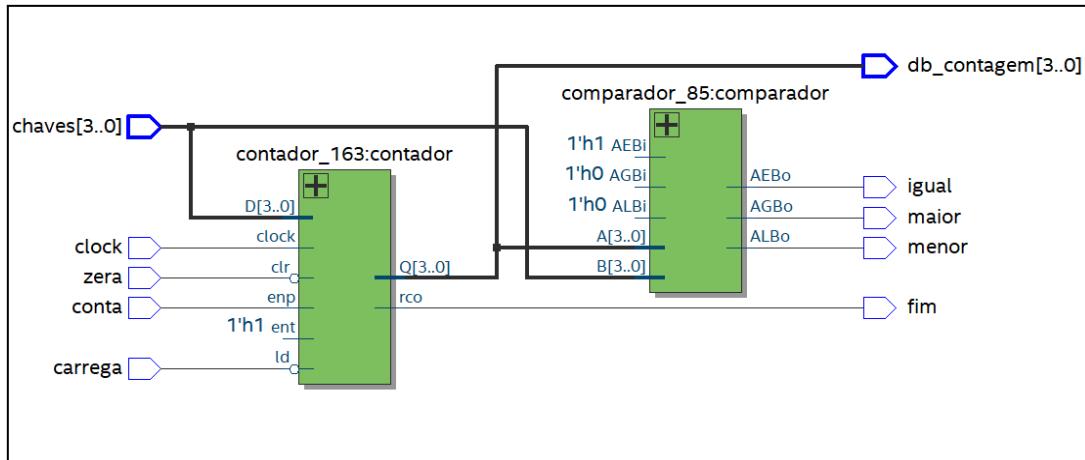


Figura 4: RTL View do circuito da atividade 3 sintetizado no Intel Quartus

Tabela 5: Pinagem do circuito com a placa FPGA

Sinal	Pino na Placa DE0-CV	Pino na FPGA
CLOCK	botão KEY0	PIN_U7
ZERA	chave SW0	PIN_U13
CONTA	chave SW1	PIN_V13
CARREGA	chave SW2	PIN_T13
CHAVES(0)	chave SW3	PIN_T12
CHAVES(1)	chave SW4	PIN_AA15
CHAVES(2)	chave SW5	PIN_AB15
CHAVES(3)	chave SW6	PIN_AA14
DB_CONTAGEM(0)	Led LEDR0	PIN_AA2
DB_CONTAGEM(1)	Led LEDR1	PIN_AA1
DB_CONTAGEM(2)	Led LEDR2	PIN_W2
DB_CONTAGEM(3)	Led LEDR3	PIN_Y3
MENOR	Led LEDR5	PIN_N1
IGUAL	Led LEDR6	PIN_U2
MAIOR	Led LEDR7	PIN_U1
FIM	Led LEDR9	PIN_L1

O experimento seguiu com o plano de testes da Tabela 3, dessa vez na placa FPGA. Esse plano de testes irá verificar o funcionamento do circuito sintetizado e a pinagem feita pela equipe. O resultado dos testes pode ser verificado nas figuras e está descrito na tabela a seguir.

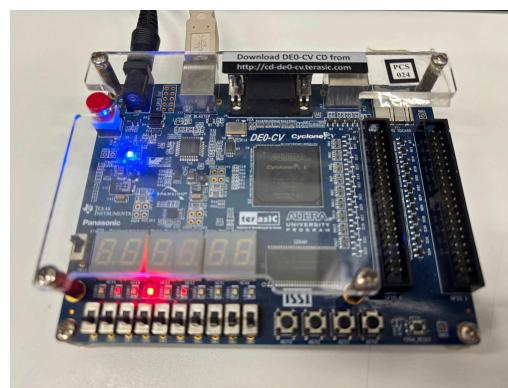


Figura 5: Condições iniciais do teste

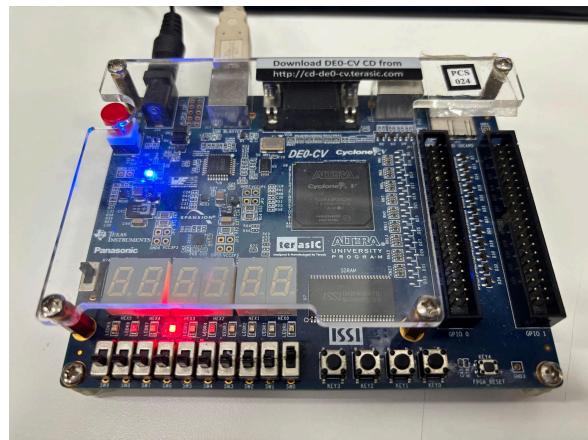


Figura 6: Teste 1

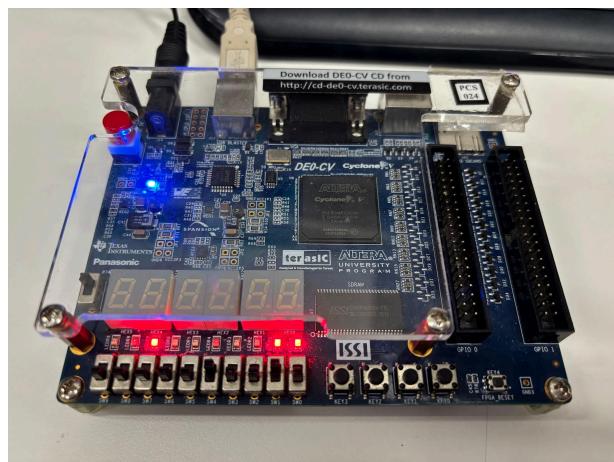


Figura 7: Teste 4

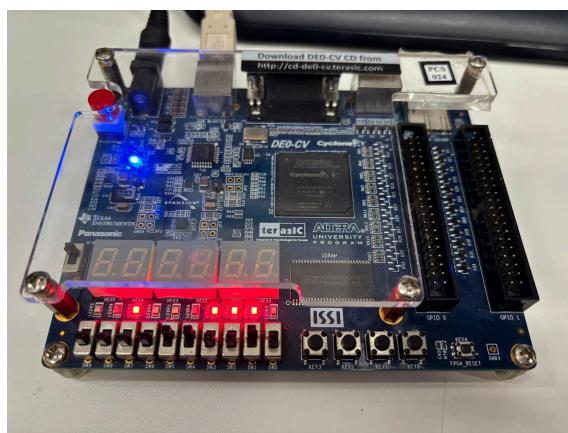


Figura 8: Teste 6

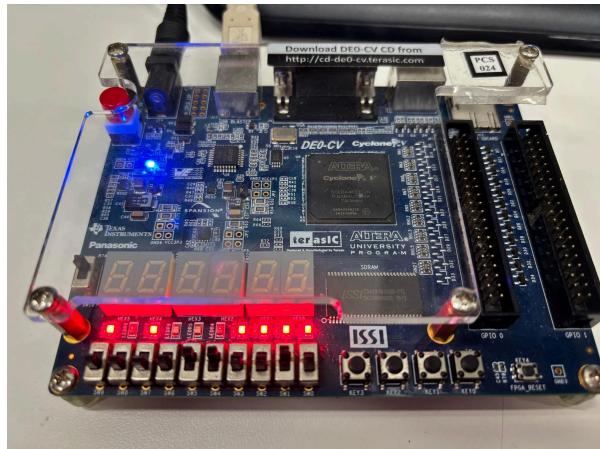


Figura 9: Teste 7

Tabela 6: Resultado dos Testes com a placa FPGA

Teste	Circuito sintetizado na FPGA
1	contagem=0,fim=0,maior=0,menor=0,igual=1
2	contagem=0,fim=0,maior=0,menor=1,igual=0
3	contagem=1,fim=0,maior=0,menor=0,igual=1
4	contagem=3,fim=0,maior=1,menor=0,igual=0
5	contagem=3,fim=0,maior=0,menor=1,igual=0
6	contagem=14(E),fim=0,maior=1,menor=0,igual=0
7	contagem=15(F),fim=1,maior=1,menor=0,igual=0

4. ATIVIDADE 4 - DESAFIO

O desafio da experiência 2 enuncia a conversão da representação do contador para um display de 7 segmentos, utilizando um módulo disponibilizado hexa7seg que realiza essa conversão. Cabe, portanto, à equipe realizar a pinagem adequada e adaptar o fluxo de dados do circuito para incluir o módulo de conversão para o display de 7 segmentos.

Ao analisar o código em Verilog do conversor hexa7seg, temos um circuito combinatório que realiza a conversão da saída, no caso do contador_163, para o display de 7 segmentos segundo o seguinte mapeamento:

```
/*
```

```

*   ---
*   | 0 |
* 5 | 1 |
*   |
*   ---
*   | 6 |
* 4 | 2 |
*   |
*   ---
*   | 3 |
*/

```

A pinagem foi feita segundo o material disponibilizado no Moodle ([link](#)), e o fluxo de dados do circuito sintetizado no Quartus pode ser visto no esquema abaixo:

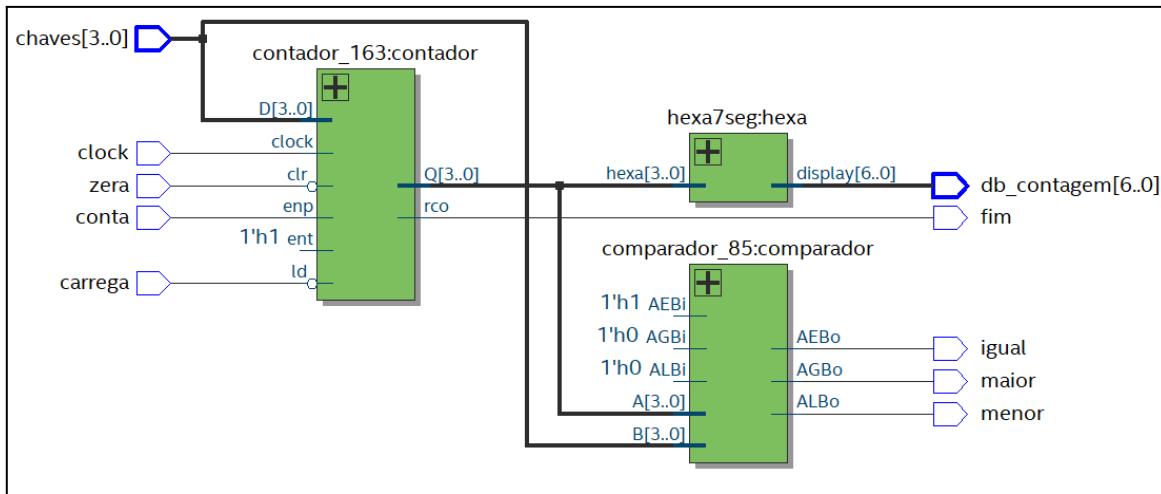


Figura 10: RTL View do circuito do desafio sintetizado no Intel Quartus

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength
chaves[3]	Unknown	PIN_AA14	4A	B4A_NO	2.5 V (default)		12mA (default)
carrega	Unknown	PIN_T13	4A	B4A_NO	2.5 V (default)		12mA (default)
chaves[2]	Unknown	PIN_AB15	4A	B4A_NO	2.5 V (default)		12mA (default)
chaves[1]	Unknown	PIN_AA15	4A	B4A_NO	2.5 V (default)		12mA (default)
chaves[0]	Unknown	PIN_T12	4A	B4A_NO	2.5 V (default)		12mA (default)
clock	Unknown	PIN_U7	3A	B3A_NO	2.5 V (default)		12mA (default)
conta	Unknown	PIN_V13	4A	B4A_NO	2.5 V (default)		12mA (default)
fim	Unknown	PIN_L1	2A	B2A_NO	2.5 V (default)		12mA (default)
igual	Unknown	PIN_U2	2A	B2A_NO	2.5 V (default)		12mA (default)
maior	Unknown	PIN_U1	2A	B2A_NO	2.5 V (default)		12mA (default)
menor	Unknown	PIN_N1	2A	B2A_NO	2.5 V (default)		12mA (default)
zera	Unknown	PIN_U13	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[0]	Unknown	PIN_U21	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[1]	Unknown	PIN_V21	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[2]	Unknown	PIN_W22	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[3]	Unknown	PIN_W21	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[4]	Unknown	PIN_Y22	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[5]	Unknown	PIN_Y21	4A	B4A_NO	2.5 V (default)		12mA (default)
db_contagem[6]	Unknown	PIN_AA22	4A	B4A_NO	2.5 V (default)		12mA (default)

Figura 11: Pinagem do circuito sintetizado no desafio

Os testes realizados seguem o padrão do plano de testes da Tabela 3. Segue abaixo as figuras com detalhes dos resultados e a tabela de resultados.

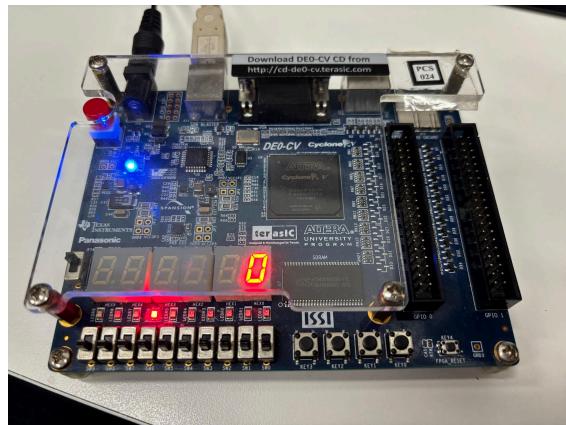


Figura 12: Condições Iniciais

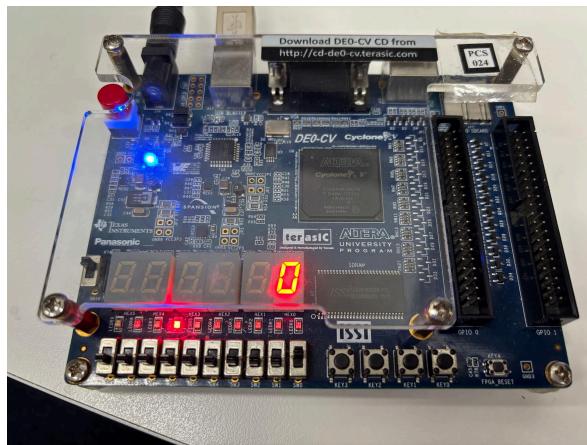


Figura 13: Teste 1

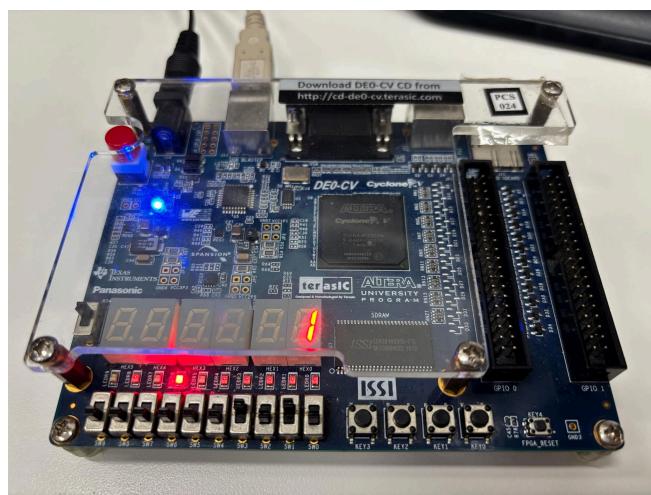


Figura 14: Teste 3

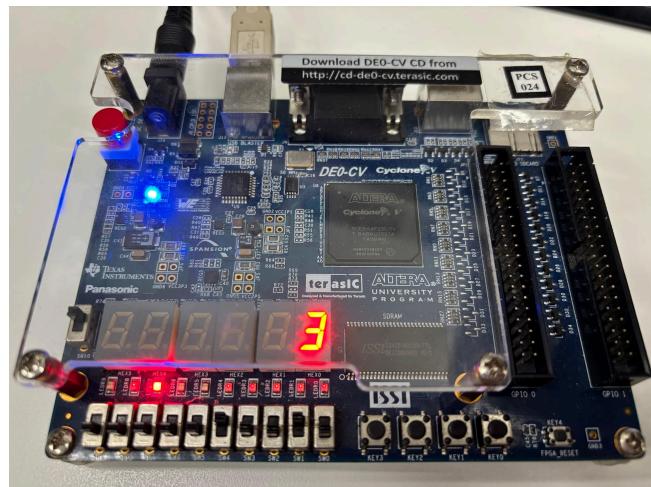


Figura 15: Teste 4

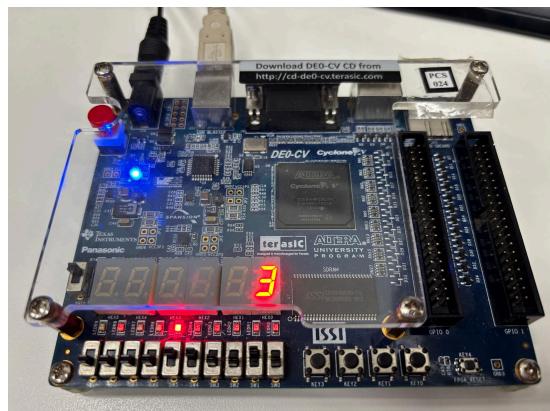


Figura 16: Teste 5

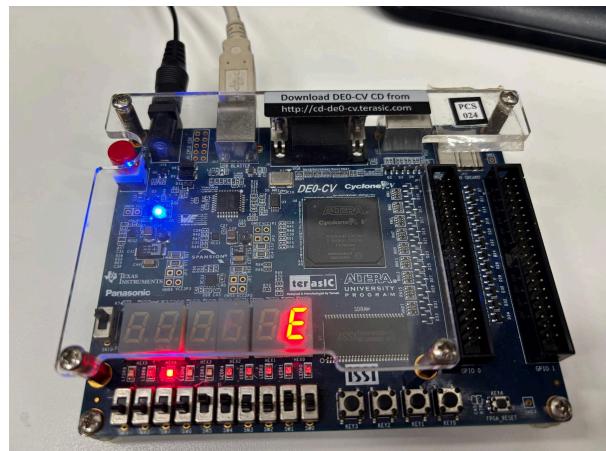


Figura 17: Teste 6

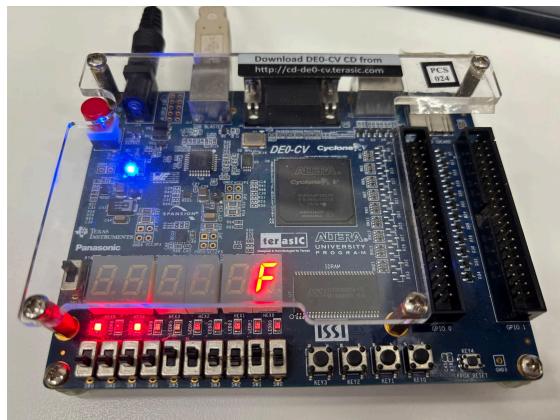


Figura 18: Teste 7

Tabela 7: Resultado dos Testes com a placa FPGA e display 7 segmentos

Teste	Circuito sintetizado na FPGA
1	contagem=0,fim=0,maior=0,menor=0,igual=1
2	contagem=0,fim=0,maior=0,menor=1,igual=0
3	contagem=1,fim=0,maior=0,menor=0,igual=1
4	contagem=3,fim=0,maior=1,menor=0,igual=0
5	contagem=3,fim=0,maior=0,menor=1,igual=0
6	contagem=14(E),fim=0,maior=1,menor=0,igual=0
7	contagem=15(F),fim=1,maior=1,menor=0,igual=0

5. CONCLUSÕES

No experimento, tivemos a oportunidade de compreender o mecanismo de compilação e a configuração de pinagem no software Quartus. Além disso, implementamos e testamos essas funcionalidades diretamente na FPGA, utilizando tanto o display de 7 segmentos quanto os LEDs. Com isso, foi possível observar o funcionamento prático do mapeamento dos sinais digitais aos pinos físicos da FPGA, bem como validar a lógica implementada no design digital. Este processo reforçou a importância de uma configuração precisa dos pinos para garantir o correto funcionamento do circuito e uma interface confiável com os periféricos conectados. Esses passos são fundamentais para o desenvolvimento de projetos em FPGAs, fornecendo uma base sólida para aplicações mais complexas no futuro.