



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

PCS3635 – LABORATÓRIO DIGITAL I

EXPERIÊNCIA 4 – Desenvolvimento de Projeto de Circuitos Digitais em FPGA

Planejamento da Bancada 07 – Turma 1 – Prof. Edson Midorikawa

Data de Emissão: 26 de Janeiro de 2024.

Nome: Gabriel Chaves Lopes Silva	Número USP: 12555839
Nome: Luiz Mariano dos Santos Silva	Número USP: 13684467
Nome: Luca Bompiani	Número USP: 12552760

1 INTRODUÇÃO

Esta experiência tem como objetivo desenvolver e implementar um sistema digital sequencial em FPGA, integrando um fluxo de dados e uma unidade de controle, para processar entradas externas acionadas por chaves e compará-las com dados armazenados em memória. Além disso, o projeto visa consolidar práticas de síntese em FPGA usando o Intel Quartus Prime e validação funcional por meio de simulações no ModelSim e testes físicos na placa DE0-CV.

O trabalho dá continuidade ao projeto desenvolvido na experiência anterior, introduzindo adaptações críticas, como a detecção de bordas para tratamento de sinais de entrada de longa duração e a inclusão de estados de controle para gerenciar ciclos de comparação e finalização do jogo.

2 DESCRIÇÃO DO PROJETO

O sistema digital desenvolvido consiste em um circuito sequencial que opera com um sinal de *clock* periódico, integrando um fluxo de dados que executa operações físicas como contagem, armazenamento e comparação, e uma unidade de controle para gerenciamento de estados.

O funcionamento do sistema pode ser representado pelo pseudocódigo a seguir:

```

1. {
    2. while (verdadeiro) {
        3. espera acionamento do sinal iniciar
        4. inicia circuito com condições iniciais
        5. while (acertou dado E não atingiu o último dado) {
            6. espera acionamento das chaves (fazer uma jogada)
            7. registra entrada da jogada realizada
            8. compara jogada realizada com dados armazenados
            9. incrementa contador interno
        10. }
        11. ativa acertou se acertou todos os dados da memória
        12. ativa errou se errou um dado
        13. ativa saída pronto (até reiniciar)
    14. }
15. }

```

O núcleo do projeto é uma memória interna de 16 dados de 4 bits, acessada por um contador que percorre seus endereços. Após o acionamento do sinal *iniciar*, o sistema aguarda a entrada de jogadas via chaves, compara esses valores com os dados armazenados na memória e atualiza saídas de depuração (como *db_igual*, *db_contagem* e *db_estado*) para monitoramento em displays de sete segmentos e LEDs

3 DETALHAMENTO DO PROJETO LÓGICO

3.1 PROJETO DO FLUXO DE DADOS

O fluxo de dados é composto por quatro componentes principais, sendo eles um Contador de 4 bits, que gera endereços para a memória interna, um Registrador de 4 bits, que armazena temporariamente as jogadas acionadas pelas chaves, uma Memória ROM síncrona 16x4, que armazena os 16 dados de referência para comparação, e um Comparador 4 bits, que verifica a igualdade entre o valor das chaves e o dado da memória, gerando o sinal `db_igual`.

A figura abaixo apresenta a estrutura do fluxo de dados:

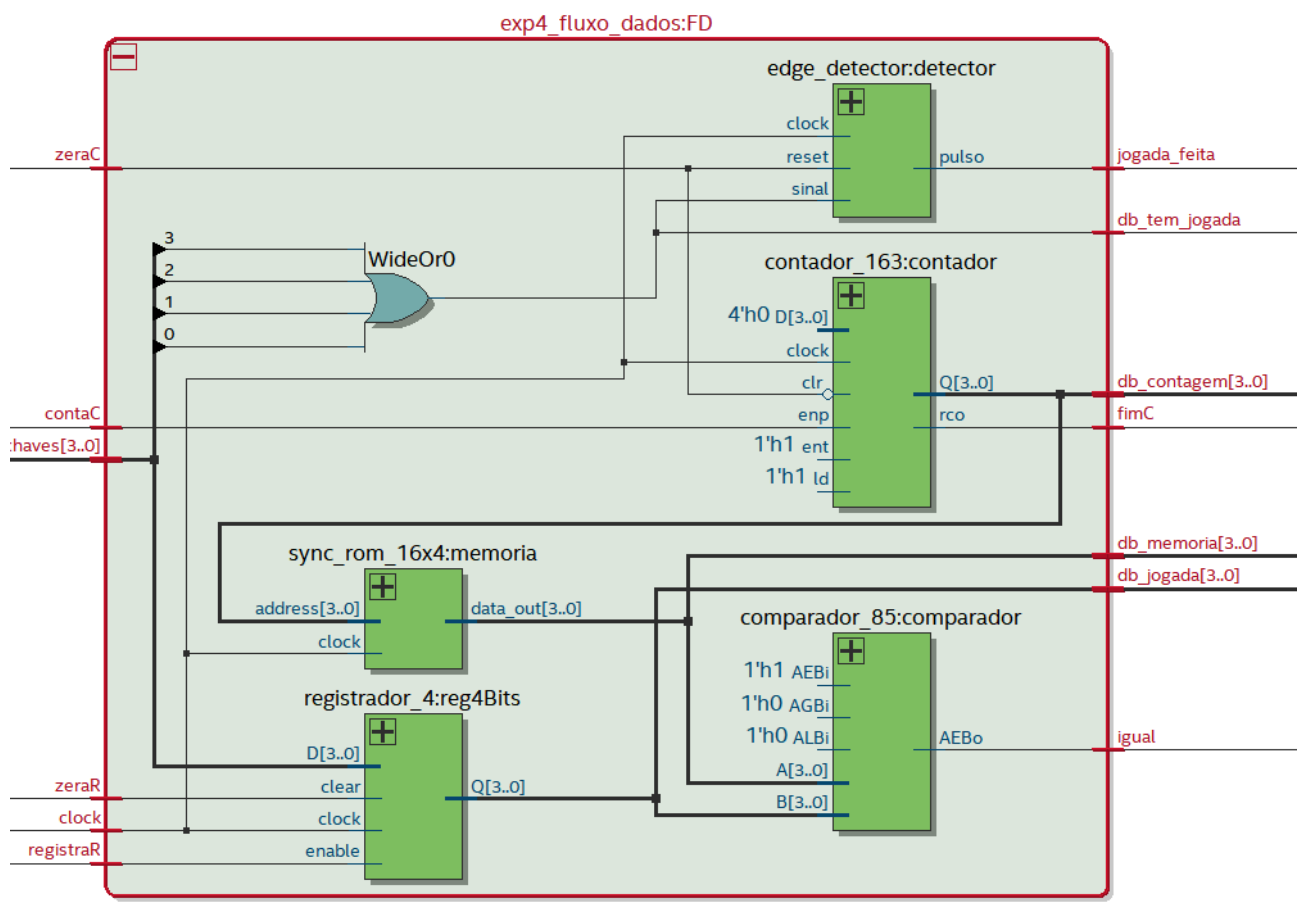


fig 1. Fluxo de Dados

Dentre as saídas de depuração, destacam-se os sinais `db_contagem` (endereço atual da memória), `db_memoria` (dado lido da memória), `db_jogadafeita` (jogada registrada) e `db_estado` (código do estado atual da unidade de controle).

A principal adaptação em relação à experiência anterior foi a adição de um detector de borda (*edge detector*) para tratar pulsos longos das chaves, convertendo-os em sinais de 1 período de *clock*. Isso evita múltiplas leituras acidentais devido à natureza mecânica das chaves. Além disso, para o sinal de reset do *edge detector*, foi utilizado o princípio que sempre que o circuito começa (`inicializa_elementos`) ou quando transicional para próxima jogada (`proxima_jogada`) o sinal é ativado permitindo que uma nova detecção seja feita.

A instanciação dos componentes em Verilog é apresentada no código a seguir:

1- Criação dos sinais intermediários para as novas funções e novas condições

```
wire sig_jogada_feita;
wire sig_db_tem_jogada;
assign sig_db_tem_jogada = |chaves;
wire reset_detector;
assign reset_detector = zeraC;
```

2- Instanciamento do novo componente

```
//Detector de jogadas
edge_detector detector (
    .clock    ( clock ),
    .reset    ( reset_detector ),
    .sinal    ( sig_db_tem_jogada ),
    .pulso    ( sig_jogada_feita )
);
```

3.2 PROJETO DA UNIDADE DE CONTROLE

A unidade de controle é projetada como uma máquina de estados finita que gerencia a transição entre os diferentes estados do sistema e gera os sinais de controle necessários para o fluxo de dados. O diagrama de transição de estados da unidade de controle é apresentado a seguir:

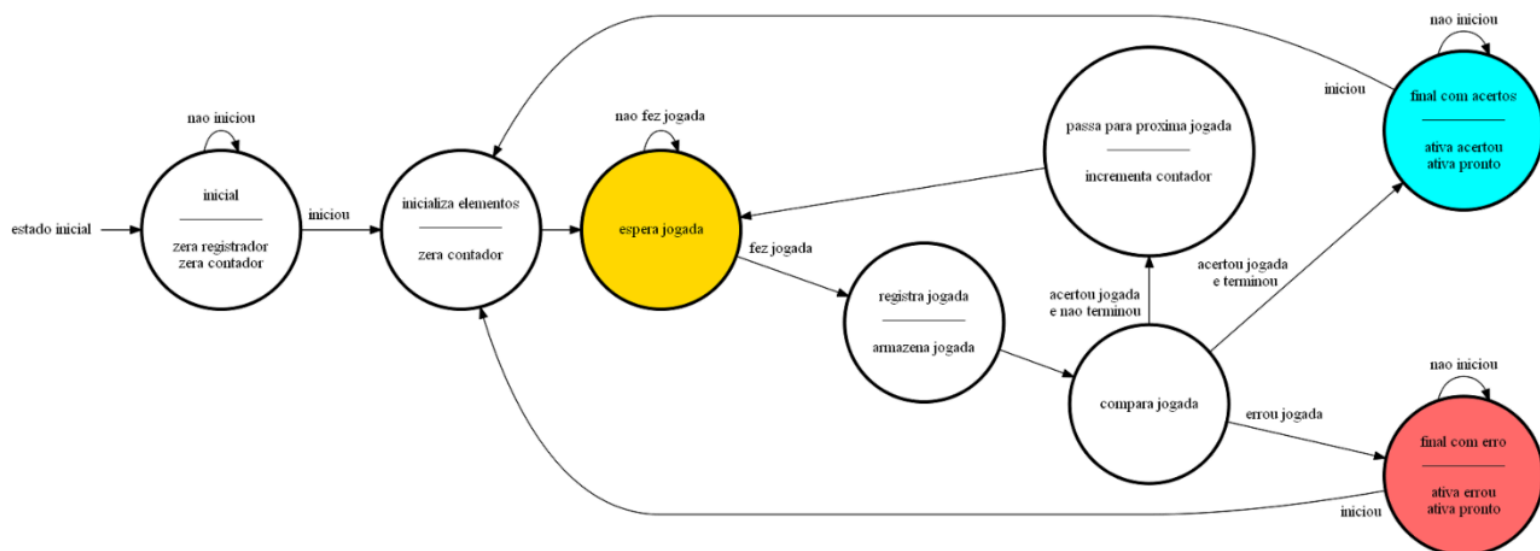


fig 2. Diagrama de estados da Unidade de Controle

Algumas mudanças pontuais foram feitas no arquivo VERILOG da UC, visto que compartilha de muitas semelhanças com o projeto anterior.

1 - Adicionado o sinal que captura a jogada para saber se houve alguma interação do jogador com o sistema.

```
input    jogada,
```

2- Novos estados respectivos a nova FSM foram adicionados e que são melhores descritos na tabela que se segue logo abaixo. Além de mudar para uma abordagem que segue os princípios de Moore.

```
// Define estados

parameter inicial          = 4'b0000; // 0

parameter inicializa_elementos = 4'b0001; // 1
```

```

parameter espera_jogada      = 4'b0011;  // 3

parameter registra_jogada    = 4'b0100;  // 4

parameter proxima_jogada     = 4'b0101;  // 5

parameter compara_jogada     = 4'b0110;  // 6

parameter final_acertou      = 4'b0111;  // 7

parameter final_errou        = 4'b1000;  // 8

```

Tabela 1 – Descrição da Unidade de Controle do Sistema

Nome do Estado	Descrição do Estado	Próximo Estado	Condições e Justificativas para a Transição entre Estados
Inicial	estado inicial do jogo	inicializa_elementos	iniciar = 1
inicializa_elementos	faz a inicialização dos elementos, preparando para jogada	espera_jogada	
espera_jogada	enquanto o jogador não fizer uma jogada nao inicia	registra_jogada	jogada=1
registra_jogada	armazena no registrador a jogada	compara_jogada	
compara_jogada	faz a comparação entre a jogada e o valor da memória ROM e continua enquanto não errar e não chegar ao fim	proxima_jogada	igual=0 e fim=0
compara_jogada	ao chegar no final se todas as jogadas foram iguais transiciona para o estado de acerto	final_acertou	igual=1 e fim=1
compara_jogada	ao errar uma jogada transiciona para o estado de erro	final_errou	igual=0
proxima_jogada	faz o loop para a próxima jogada	espera_jogada	
final_acertou	final do jogo com acerto	inicializa_elementos	iniciar=0
final_errou	final do jogo com erro	inicializa_elementos	iniciar=0

Para fins de validação, é realizado a partir da vista do FSM no Quartus, uma comparação entre o proposto acima e o gerado. Demonstra-se então correto funcionamento do circuito, conformando-se com o previsto:

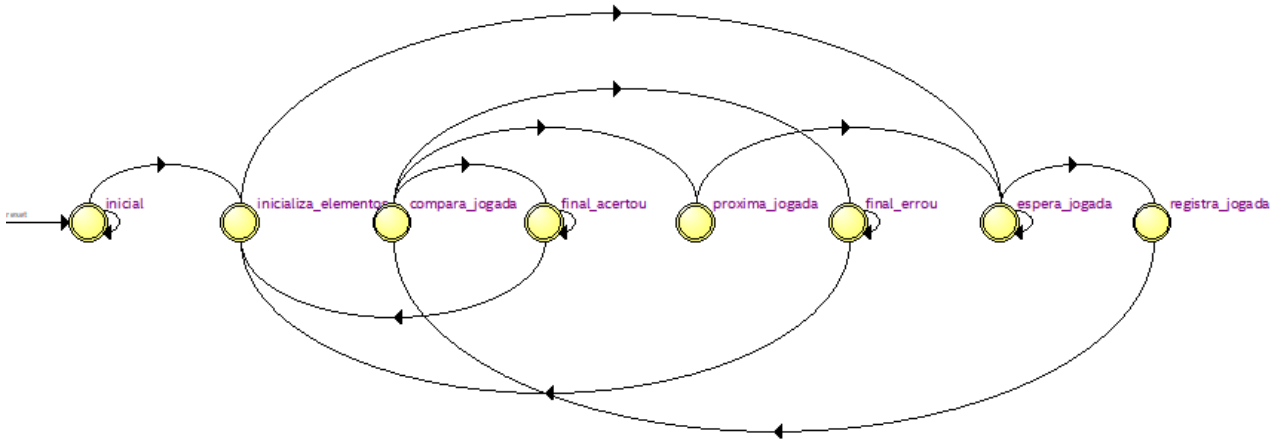


fig 3. FSM Unidade de Controle

3.3 PROJETO DO SISTEMA DIGITAL

A integração entre fluxo de dados e unidade de controle é realizada por sinais de controle e status:

- **Sinais de controle da UC para o FD:** `registraR`, `contaC`, `zeraC`, `zeraR`.
- **Sinais de status do FD para a UC:** `igual`, `jogada`, `fim` (fim da contagem).

Além disso, os sinais `db_temjogada` (que indica detecção de uma jogada) e `db_clock` (cópia do sinal de `clock` para verificação externa) podem ser utilizados para depuração adicional caso necessário

A figura abaixo ilustra a integração do sistema digital:

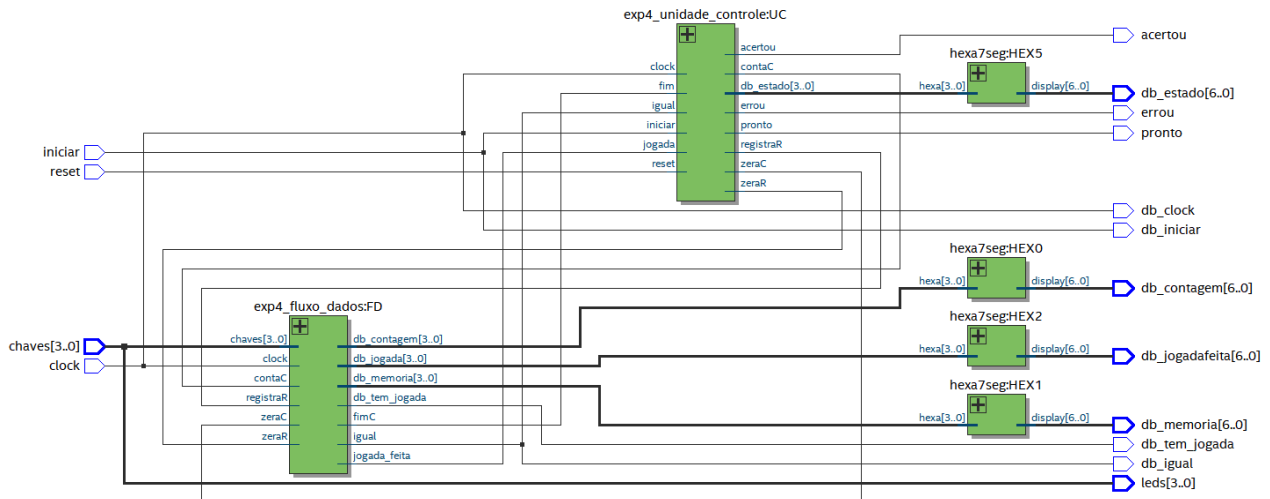


fig 4. RTL Viewer

Como na experiência anterior, a interação entre os dois componentes permite que o sistema funcione de maneira sincronizada, pois a unidade de controle gerencia a execução do fluxo de dados em cada ciclo de clock.

4 PLANO DE TESTES DO SISTEMA E SIMULAÇÕES

Este capítulo tem o objetivo de documentar a estratégia dos testes do sistema desenvolvido e documentado no capítulo 3 e apresentar os resultados das respectivas simulações realizadas.

4.1 CENÁRIO DE TESTE 1 – TESTE DO CIRCUITO COMPLETO COM TODAS AS JOGADAS ACERTADAS

Nessa seção, tomamos como objetivo validar que ao acertar todas as sequências propostas pela memória ROM o jogador termina no estado de acertou com todos os devidos sinais ativados corretamente, onde tais detalhes podem ser consultados a partir da tabela abaixo, que lista todos os sinais e o comportamento observado.

Tabela 2 – Plano de Teste para Acertar

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Condições Iniciais	iniciar=0 reset=0 chaves=0000	acertou=0 erro=0 pronto=0 db_igual=0 db_temjogada=0 db_estado=0	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
		db_jogadafeita=0 db_memoria=0 db_contagem=0 leds=0	
Acionar o Reset para ir ao estado de início	iniciar=0 reset=1 chaves=0000	zeraC=1 zeraR=1	ok
Aguardar alguns segundos			
Acionar o Iniciar	iniciar=1 reset=0 chaves=0000	zeraC=1 db_estado=0001	ok
Acionar primeira entrada (jogada 1) Aguardar ate o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 2) Aguardar ate o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	erro
Acionar primeira entrada (jogada 3) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 4) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 5) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 6) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 7)	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001	

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Aguardar ate o fim da operacao quando db_estado = 0110		db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 8) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 9) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 10) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 11) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 12) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 13) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 14) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 15) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Acionar primeira entrada (jogada 16) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	acertou=1 pronto=1 db_igual=1 db_estado=0111 db_jogada=0100 db_memoria=0100 leds=0100	

A validação foi realizada a partir da carta de tempo gerada no *ModelSim*, e pode ser observada logo a baixo com suas devidas observações.

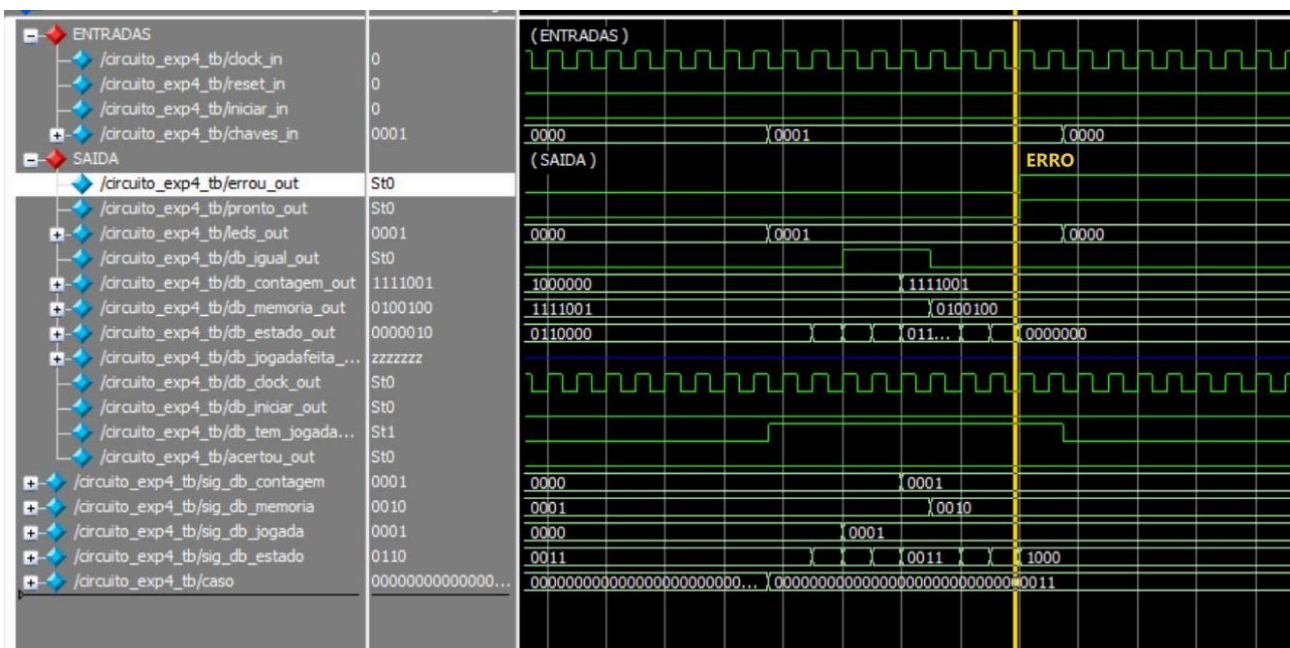


fig 6. Carta de tempo do primeiro teste

Como é possível notar, a partir da segunda jogada o circuito começou a apresentar um comportamento anômalo, então para uma verificação mais visual foi montado o mesmo utilizando o *Digital* e replicando os testes para observar onde estaria o problema.

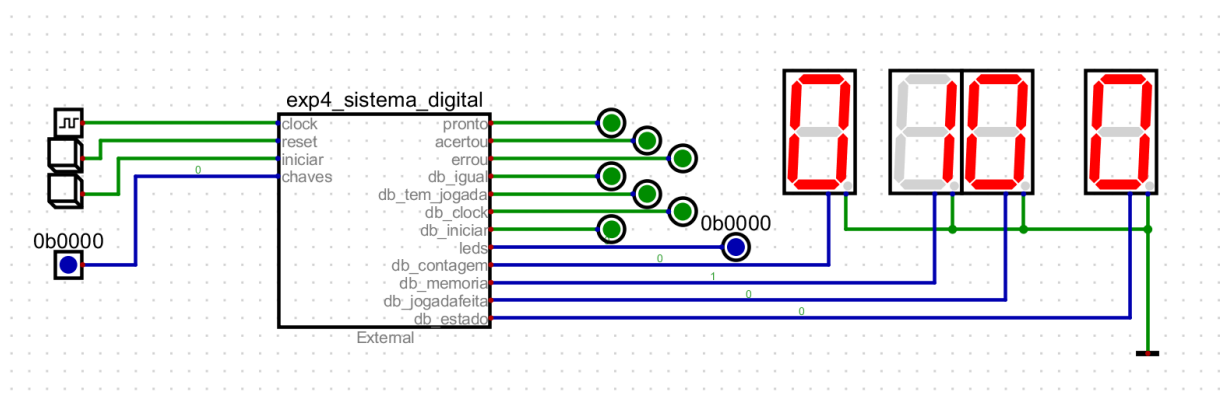


fig 6. Condições Iniciais

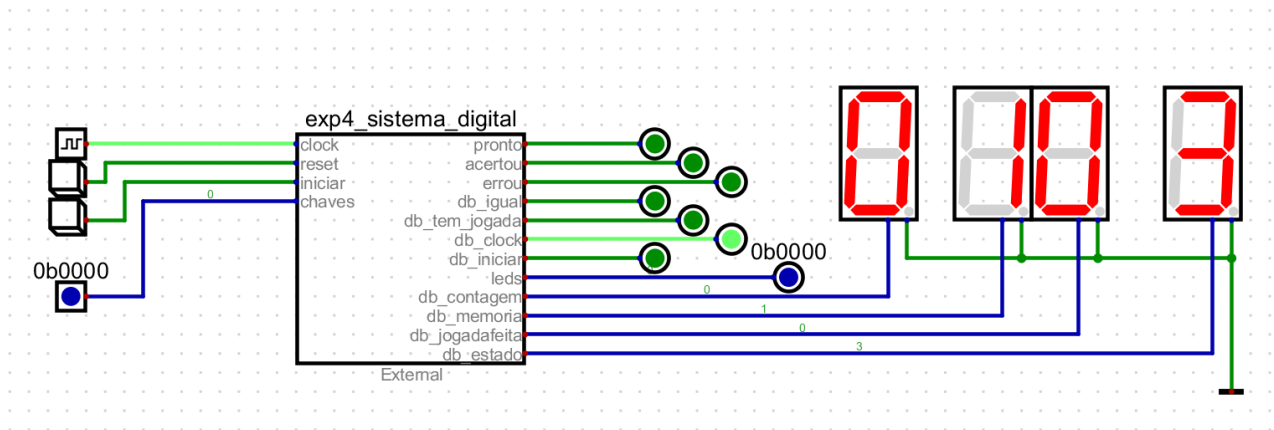


fig 7. Sinal de iniciar Ativado

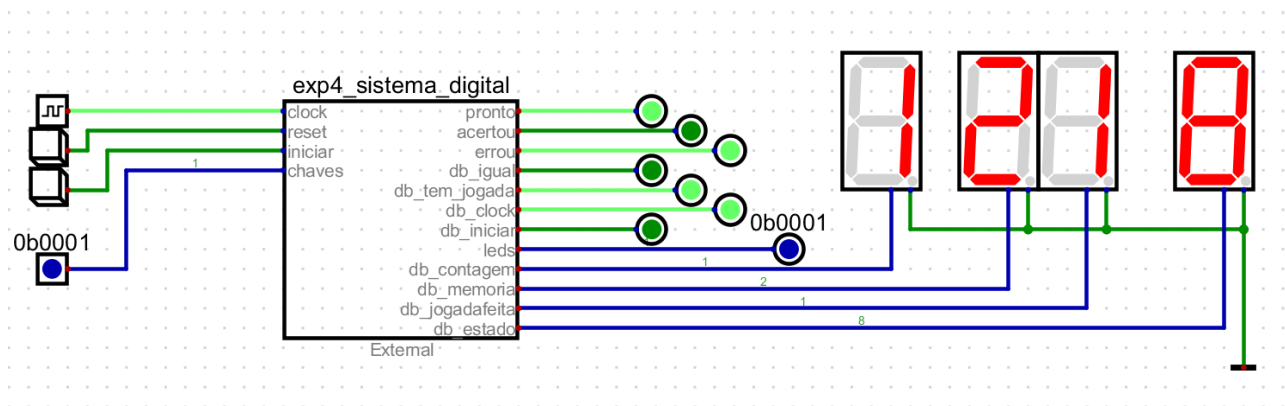


fig 8. Erro ao se iniciar a segunda jogada

Foi possível observar, que ao se realizar a segunda interação, o circuito detecta que uma jogada já foi realizada, utilizando a jogada anterior, o que faz parecer que o usuário errou, quando na verdade ele ainda não adicionou sua jogada.

Depois de feita a depuração, foi identificado um erro no sinal que reseta a jogada, onde a cada nova contagem ele era zerado e o que não deveria acontecer, apenas quando os elementos forem inicializados. Com as devidas alterações feitas no fluxo de dados, novamente o circuito foi posto a teste.

- Antes

```
assign reset_detector = zeraC | contaC;
```

- Depois

```
assign reset_detector = zeraC;
```


Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
da operacao quando db_estado = 0110		db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 4) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 5) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 6) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 7) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 8) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 9) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 10) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 11) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira	iniciar=0 reset=0	db_igual=1 db_temjogada=1	ok

4.2 CENÁRIO DE TESTE 2 – TESTE DO CIRCUITO COMPLETO COM ERRO NA QUARTA JOGADA

Após testar o Primeiro Cenário, um teste similar foi elaborado para testar quando o jogador erra uma jogada.

Tabela 4 – Plano de Teste para Errar na quarta jogada

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Condições Iniciais	iniciar=0 reset=0 chaves=0000	acertou=0 errou=0 pronto=0 db_igual=0 db_temjogada=0 db_estado=0 db_jogadafeita=0 db_memoria=0 db_contagem=0 leds=0	ok
Acionar o Reset para ir ao estado de inicio	iniciar=0 reset=1 chaves=0000	zeraC=1 zeraR=1	ok
Aguardar alguns segundos			ok
Acionar o Iniciar	iniciar=1 reset=0 chaves=0000	zeraC=1 db_estado=0001	ok
Acionar primeira entrada (jogada 1) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 2) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 3) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 4) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	errou=1 pronto=1 db_igual=0 db_estado=1000 db_jogada=0001 db_memoria=1000 leds=0001	ok

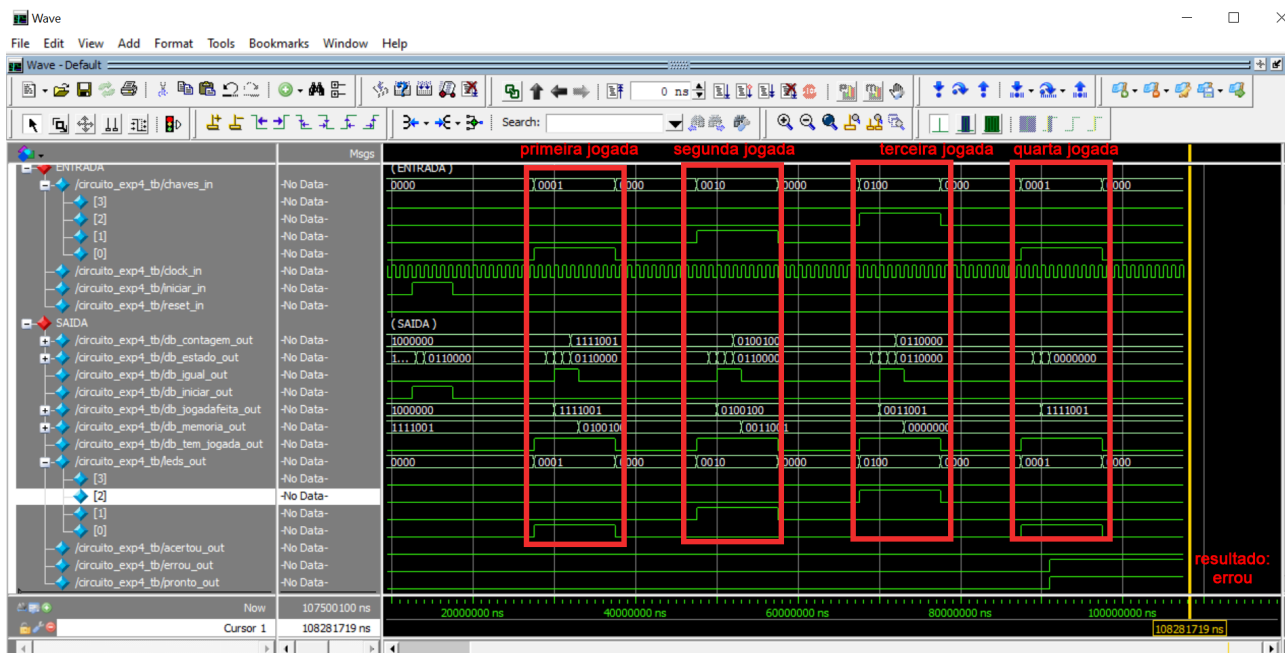


fig 10. Carta de Tempo Errou quarta jogada

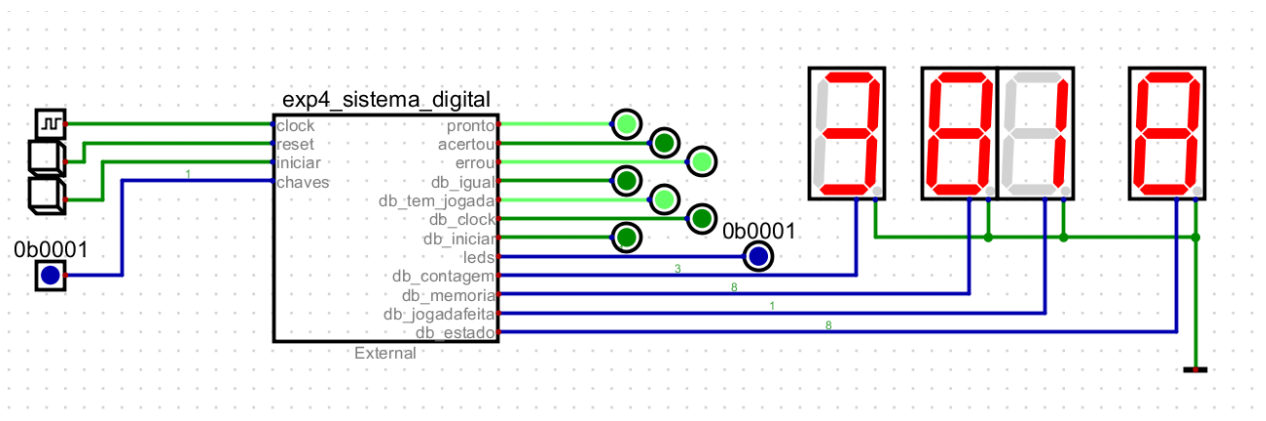


fig 11. Simulação do Sistema Digital

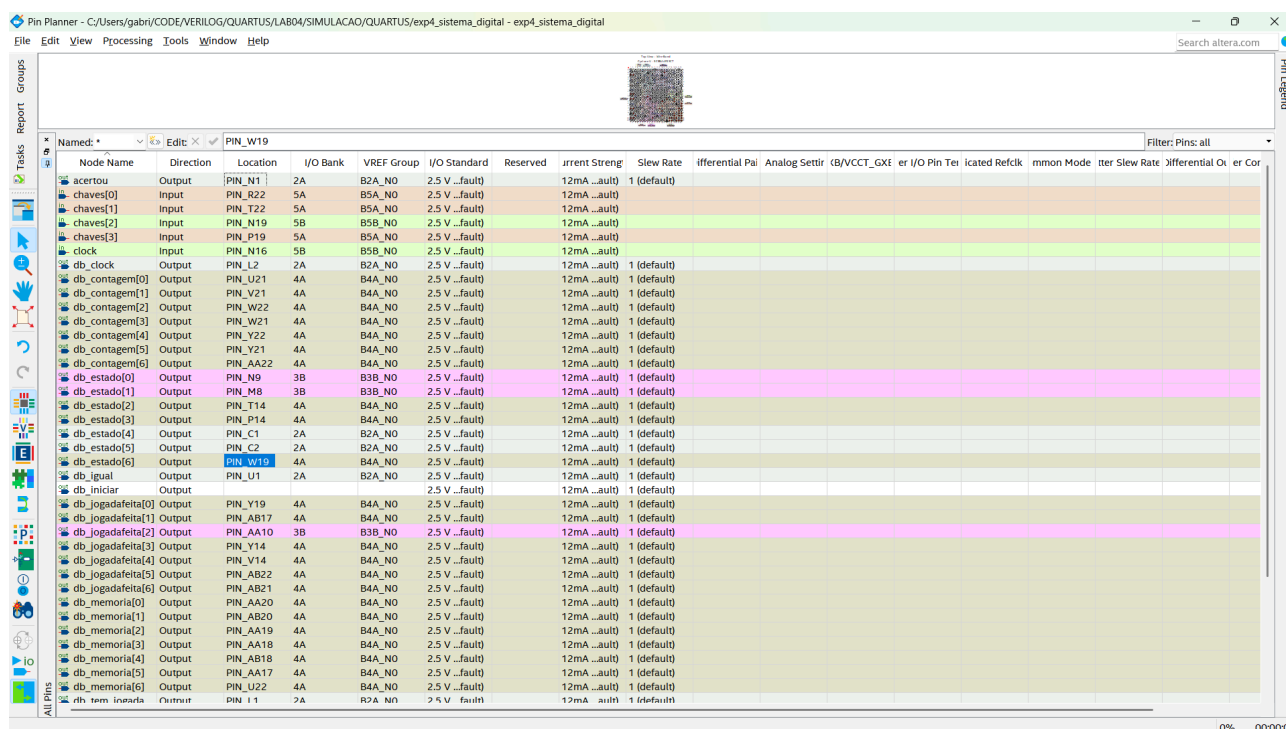
5 IMPLANTAÇÃO DO PROJETO

Este capítulo tem o objetivo de documentar as atividades práticas de execução do projeto desenvolvido e documentado no capítulo 3 no ambiente do Laboratório Digital.

(imagem da placa FPGA no lab)

5.1 PINAGEM DA PLACA FPGA

A pinagem foi definida no Intel Quartus Prime utilizando o *Pin Planner*, associando os sinais do projeto aos pinos físicos da FPGA Cyclone V (placa DE0-CV). A figura abaixo apresenta o plano de pinagem detalhado:



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Analog Setting	CB/VCCCT_GX1	er I/O Pin Ter	icated Refclk	mmion Mode	Iter Slew Rate	Differential O	er Cor
acertou	Output	PIN_N1	2A	B2A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
chaves[0]	Input	PIN_R22	5A	B5A_NO	2.5 V ..fault		12mA ..aut										
chaves[1]	Input	PIN_T22	5A	B5A_NO	2.5 V ..fault		12mA ..aut										
chaves[2]	Input	PIN_N19	5B	B5B_NO	2.5 V ..fault		12mA ..aut										
chaves[3]	Input	PIN_P19	5A	B5A_NO	2.5 V ..fault		12mA ..aut										
clock	Input	PIN_N16	5B	B5B_NO	2.5 V ..fault		12mA ..aut										
db_clock	Output	PIN_L2	2A	B2A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[0]	Output	PIN_U21	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[1]	Output	PIN_V21	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[2]	Output	PIN_W22	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[3]	Output	PIN_W21	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[4]	Output	PIN_V22	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[5]	Output	PIN_Y21	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_contagem[6]	Output	PIN_AA22	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[0]	Output	PIN_N9	3B	B3B_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[1]	Output	PIN_M8	3B	B3B_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[2]	Output	PIN_T14	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[3]	Output	PIN_P14	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[4]	Output	PIN_C1	2A	B2A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[5]	Output	PIN_C2	2A	B2A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_estado[6]	Output	PIN_W19	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_igual	Output	PIN_U1	2A	B2A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_iniciar	Output	PIN_Y19	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_jogadafeita[0]	Output	PIN_AB17	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_jogadafeita[1]	Output	PIN_AA10	3B	B3B_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_jogadafeita[2]	Output	PIN_Y14	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_jogadafeita[3]	Output	PIN_V14	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_jogadafeita[4]	Output	PIN_AB22	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_jogadafeita[5]	Output	PIN_AB21	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[0]	Output	PIN_AA20	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[1]	Output	PIN_AB20	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[2]	Output	PIN_AA19	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[3]	Output	PIN_AA18	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[4]	Output	PIN_AB18	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[5]	Output	PIN_AA17	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_memoria[6]	Output	PIN_U22	4A	B4A_NO	2.5 V ..fault		12mA ..aut	1 (default)									
db_tem_inicial	Output	PIN_L1	2A	B2A_NO	2.5 V ..fault		12mA ..aut	1 (default)									

fig 12. Pin planner

5.2 ESTRATÉGIA DE MONTAGEM

A implementação física do projeto segue três etapas principais. Em primeiro lugar, a programação da FPGA é realizada no Intel Quartus Prime, onde o projeto é compilado para gerar o arquivo **.sof**, transferido posteriormente para a placa DE0-CV por meio do *Programmer* via cabo USB.

Em seguida, as conexões físicas são estabelecidas: o dispositivo Analog Discovery é conectado aos pinos GPIO da FPGA para gerar os sinais de *clock* (canal **DI00**) e *reset* (canal **DI01**), enquanto as chaves da placa (**SW0** a **SW3**) são mapeadas para as entradas **CHAVES[0]** a **CHAVES[3]**. Os LEDs (**LEDR0** a **LEDR9**) e os displays de sete segmentos (**HEX0** a **HEX5**) são configurados para exibir as saídas de depuração, como **db_contagem** e **db_estado**. Por fim, o sinal de *clock* é configurado na ferramenta *Patterns* do Waveforms (Analog Discovery) para operar em 1 kHz, garantindo a

sincronização adequada das operações.

5.3 ESTRATÉGIA DE DEPURAÇÃO

A depuração do projeto deve seguir uma abordagem sistemática para identificar e resolver problemas, assegurando o correto funcionamento do sistema. Os LEDs são utilizados para verificar sinais críticos como **pronto**, **acertou** e **errou**, enquanto os displays de sete segmentos permitem a leitura em tempo real de **db_contagem** (endereço da memória), **db_estado** (código do estado da UC) e **db_jogadafeita** (valor das chaves registrado)

No laboratório, não é necessário o uso de equipamentos especiais além dos já citados, pois a análise dos sinais diretamente nos displays e LEDs é suficiente para a depuração física do projeto. Se forem detectados problemas no circuito lógico, são feitos ajustes no código Verilog, e a FPGA é recompilada. Para questões de montagem, as conexões físicas são verificadas e corrigidas, se necessário. Após cada modificação, testes específicos são realizados para assegurar que as funcionalidades já validadas não foram afetadas, garantindo a integridade e o funcionamento completo do sistema.

Durante a execução dos testes foi observado inconsistências dos resultados, sendo assim necessária uma etapa de depuração, tanto no código como na implementação do Analog Discovery e pinagem da placa. Como os leds estavam todos sendo utilizados foi selecionado um segmento de um dos displays de 7 segmentos para fazer a debugagem do sinal iniciar.

5.4 EXECUÇÃO PRÁTICA DO CENÁRIO DE TESTE 1 – INTEGRAÇÃO COM A FPGA PARA APENAS ACERTOS

Este plano de testes busca avaliar o funcionamento do circuito construído anteriormente na placa FPGA. A ordem de execução e as entradas de cada etapa de teste são as mesmas realizadas na simulação digital deste. Desse modo, no caso da aparição de algum erro, é possível identificar com maior facilidade seus motivos.

Tabela 3 – Plano de Teste para Acertar

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Condições Iniciais	iniciar=0 reset=0 chaves=0000	acertou=0 errou=0 pronto=0 db_igual=0 db_temjogada=0 db_estado=0 db_jogadafeita=0 db_memoria=0 db_contagem=0 leds=0	ok
Acionar o Reset para ir ao estado de início	iniciar=0 reset=1 chaves=0000	zeraC=1 zeraR=1	ok
Aguardar alguns segundos			ok
Acionar o Iniciar	iniciar=1 reset=0 chaves=0000	zeraC=1 db_estado=0001	ok
Acionar primeira entrada (jogada 1) Aguardar ate o fim da operação quando db_estado = 0011	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 2) Aguardar ate o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 3) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 4) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 5) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 6) Aguardar ate o fim	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
da operacao quando db_estado = 0110		db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 7) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 8) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 9) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 10) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 11) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 12) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 13) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 14) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira	iniciar=0 reset=0	db_igual=1 db_temjogada=1	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
entrada (jogada 15) Aguardar ate o fim da operacao quando db_estado = 0110	chaves=0001	db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 16) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	acertou=1 pronto=1 db_igual=1 db_estado=0111 db_jogada=0100 db_memoria=0100 leds=0100	ok

5.5 EXECUÇÃO PRÁTICA DO CENÁRIO DE TESTE 2 – INTEGRAÇÃO COM A FPGA PARA ERROS

Este plano de testes busca avaliar o funcionamento do circuito construído anteriormente na placa FPGA. A ordem de execução e as entradas de cada etapa de teste são as mesmas realizadas na simulação digital deste. Desse modo, no caso da aparição de algum erro, é possível identificar com maior facilidade seus motivos.

Tabela 4 – Plano de Teste para Error na quarta jogada

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Condições Iniciais	iniciar=0 reset=0 chaves=0000	acertou=0 errou=0 pronto=0 db_igual=0 db_temjogada=0 db_estado=0 db_jogadafeita=0 db_memoria=0 db_contagem=0 leds=0	ok
Acionar o Reset para ir ao estado de inicio	iniciar=0 reset=1 chaves=0000	zeraC=1 zeraR=1	ok
Aguardar alguns segundos			ok
Acionar o Iniciar	iniciar=1 reset=0 chaves=0000	zeraC=1 db_estado=0001	ok
Acionar primeira entrada (jogada 1) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 2) Aguardar ate o fim da operacao quando db_estado =	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
0110		db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 3) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 4) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	errou=1 pronto=1 db_igual=0 db_estado=1000 db_jogada=0001 db_memoria=1000 leds=0001	ok

6 PROJETO DO DESAFIO DA EXPERIÊNCIA

6.1 DESCRIÇÃO DO DESAFIO

O desafio da atividade visa implementar no circuito já existente, um contador que detecta o intervalo entre 2 jogadas. O jogo deve terminar e retornar erro caso haja uma diferença de tempo entre as mesmas maior que 3 segundos.

6.2 DESCRIÇÃO DO PROJETO LÓGICO

6.2.1 Alterações do Fluxo de Dados

Em relação ao fluxo de dados, é adicionado o contador que recebe o sinal **clock** como entrada e retorna o sinal **timer** quando este atinge 3 segundos. Ressalta-se aqui que o **clock** opera a uma frequência de 1 kHz, e o sinal será retornado após 3000 periodos de clock .

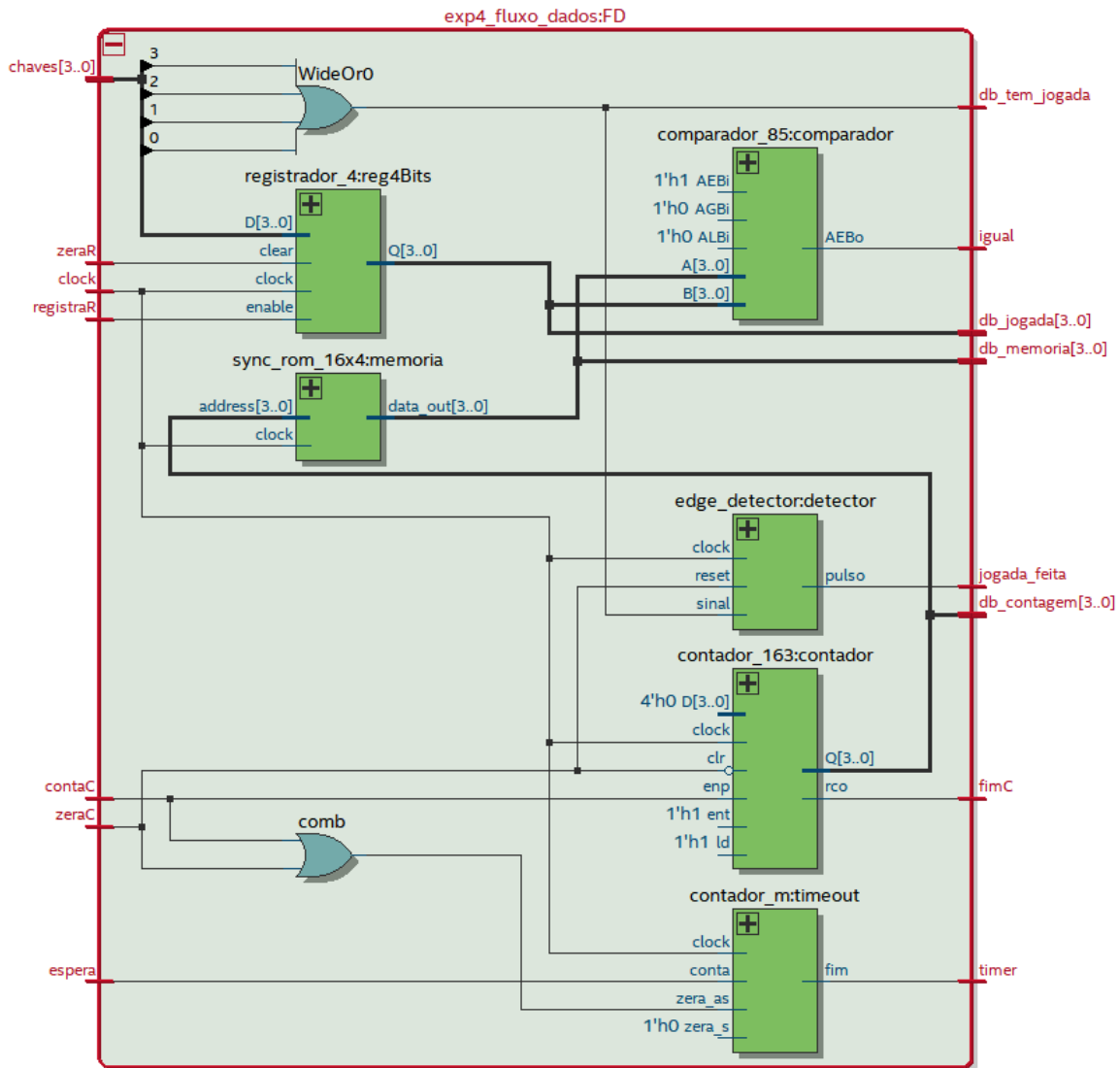


fig 13. Diagrama RTL Fluxo de Dado

6.2.2 Alterações da Unidade de Controle

É adicionado à Unidade de Controle um novo estado de timeout, este será acessado quando houver timeout no jogo, este estado será ativado pelo sinal `timer=1`.

Vale notar a possibilidade de simplesmente conectar o estado de espera ao estado final com erro (estado 8), sem a necessidade de criar um novo para este caso. Entretanto, percebe-se que essa solução pode ser prejudicial ao grupo em algum desenvolvimento futuro do projeto, no qual o novo estado pode ganhar maior importância. Desse modo, é preferível a criação do do estado de timeout (estado 2).

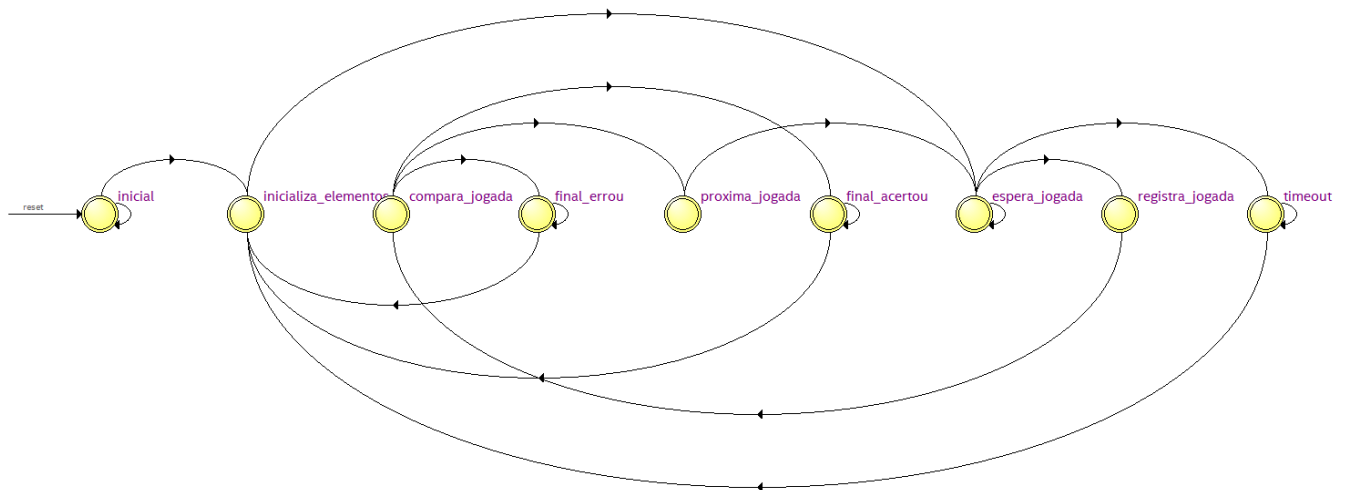


fig 14. Diagrama de estados gerado no Quartus

6.2.3 Alterações do Sistema Digital

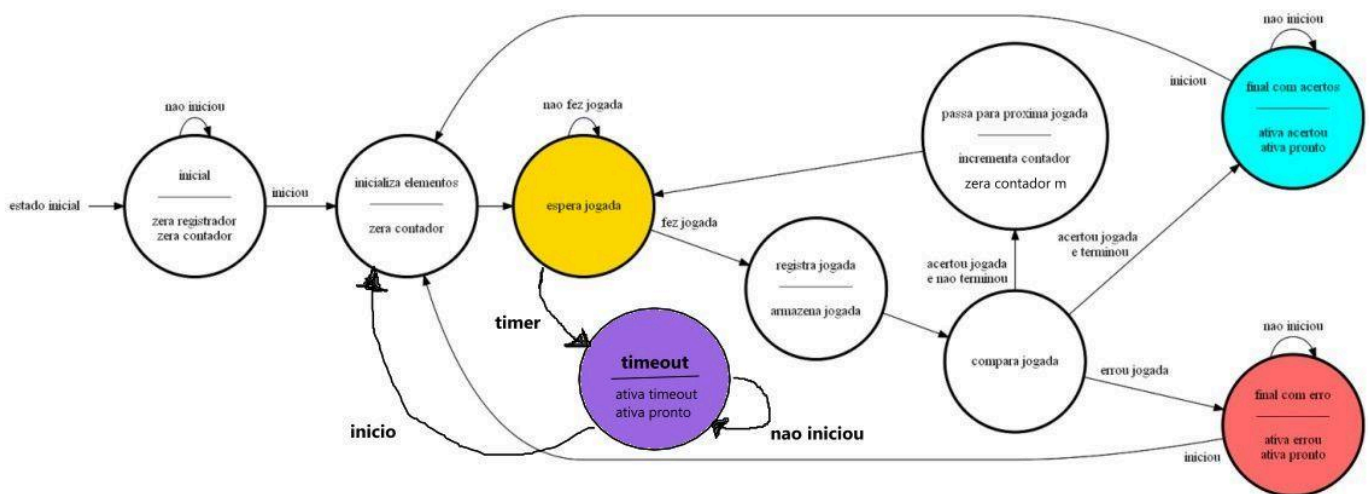


fig 15. Diagrama de estados

A integração entre o fluxo de dados e a Unidade de Controle deve ser feita como nas atividades anteriores. Adiciona-se, entretanto, o sinal **timer**, retornado pelo fluxo de dados que indica à Unidade de Controle a ocorrência de timeout e que esta deve prosseguir ao estado 2. Esta, por sua vez, retorna o sinal **time_out**, utilizado para a depuração do sistema.

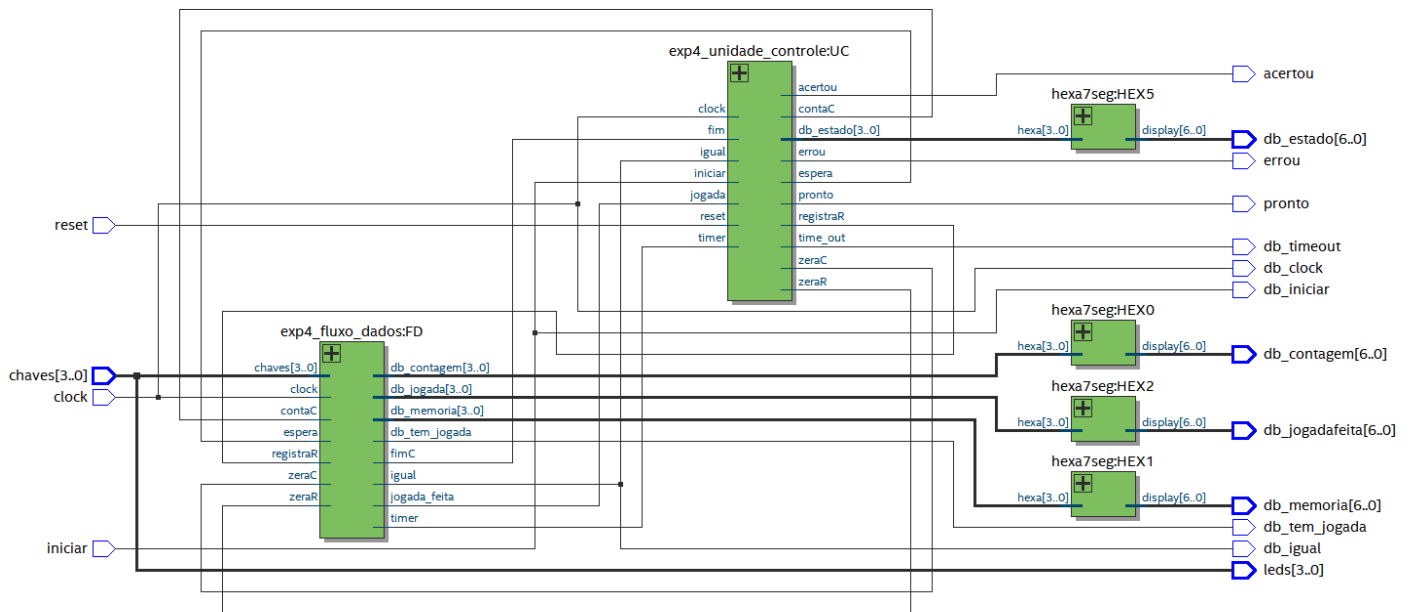


fig 16. Diagrama RTL do circuito

6.3 VERIFICAÇÃO E VALIDAÇÃO DO DESAFIO

A etapa de verificação e validação do desafio deve seguir a mesma lógica das etapas anteriores. Nota-se, entretanto, que o intervalo entre a alteração das entradas deve ser menor que o tempo máximo de execução de 3 segundos.

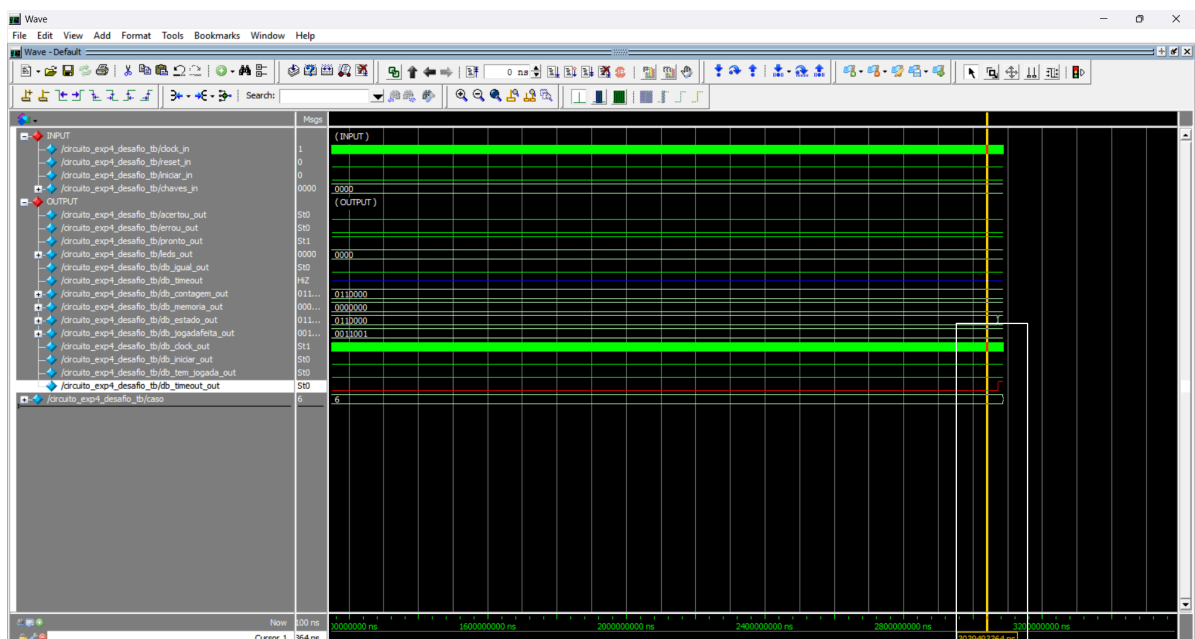


fig 15. Simulação do desafio no Modelsim com timeout no 6 caso

6.3.1 Cenário de Teste 1 – Todos os resultados corretos

Este plano de testes busca avaliar o funcionamento do circuito do desafio, construído na placa FPGA para o caso em que todos os resultados estão corretos. A ordem de execução e as entradas de cada etapa de teste são as mesmas realizadas nas atividades anteriores, deve-se apenas garantir que cada jogada seja realizada com menos de 3 segundos de intervalo. Desse modo, no caso da aparição de algum erro, é possível identificar com maior facilidade seus motivos.

Tabela 6 – Descrição e Resultados do Cenário de Teste 1 para o Desafio

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Condições Iniciais	iniciar=0 reset=0 chaves=0000	acertou=0 errou=0 pronto=0 db_igual=0 db_temjogada=0 db_estado=0 db_jogadafeita=0 db_memoria=0 db_contagem=0 leds=0	ok
Acionar o Reset para ir ao estado de início	iniciar=0 reset=1 chaves=0000	zeraC=1 zeraR=1	ok
Aguardar alguns segundos			ok
Acionar o Iniciar	iniciar=1 reset=0 chaves=0000	zeraC=1 db_estado=0001	ok
Acionar primeira entrada (jogada 1) Aguardar ate o fim da operação quando db_estado = 0011	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 2) Aguardar ate o fim da operação	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
quando db_estado = 0110			
Acionar primeira entrada (jogada 3) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 4) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 5) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 6) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
entrada (jogada 7) Aguardar ate o fim da operacao quando db_estado = 0110		db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	
Acionar primeira entrada (jogada 8) Aguardar ate o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 9) Aguardar ate o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 10) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 11) Aguardar até o fim da operação quando db_estado =	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
0110			
Acionar primeira entrada (jogada 12) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 13) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 14) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=1000	db_igual=1 db_temjogada=1 db_jogadafeita=1000 db_memoria=1000 db_contagem=1 leds=1000 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 15) Aguardar até o fim da operação quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 16) Aguardar	iniciar=0 reset=0 chaves=0100	acertou=1 pronto=1 db_igual=1 db_estado=0111 db_jogada=0100 db_memoria=0100 leds=0100	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
até o fim da operação quando db_estado = 0110			

6.3.2 Cenário de Teste 2 – Timeout após 3 acertos

.Este plano de testes busca avaliar o funcionamento do circuito do desafio, construído na placa FPGA para o caso de timeout do circuito (tempo entre execução maior que 3 segundos. Novamente, a ordem de execução e as entradas de cada etapa de teste são as mesmas realizadas nas atividades anteriores. Desse modo, no caso da aparição de algum erro, é possível identificar com maior facilidade seus motivos.

Tabela 7 – Descrição e Resultados do Cenário de Teste 2 para o Desafio

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Condições Iniciais	iniciar=0 reset=0 chaves=0000	acertou=0 errou=0 pronto=0 db_igual=0 db_temjogada=0 db_estado=0 db_jogadafeita=0 db_memoria=0 db_contagem=0 leds=0	ok
Acionar o Reset para ir ao estado de inicio	iniciar=0 reset=1 chaves=0000	zeraC=1 zeraR=1	ok
Aguardar alguns segundos			ok
Acionar o Iniciar	iniciar=1 reset=0 chaves=0000	zeraC=1 db_estado=0001	ok
Acionar primeira entrada (jogada 1) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0001	db_igual=1 db_temjogada=1 db_jogadafeita=0001 db_memoria=0001 db_contagem=1 leds=0001 db_estado=(0011 ate 0110)	ok
Acionar primeira entrada (jogada 2) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0010	db_igual=1 db_temjogada=1 db_jogadafeita=0010 db_memoria=0010 db_contagem=1 leds=0010 db_estado=(0011 ate 0110)	ok

Descrição do Estado	Sinais de Entrada	Saída Esperada	Observações
Acionar primeira entrada (jogada 3) Aguardar ate o fim da operacao quando db_estado = 0110	iniciar=0 reset=0 chaves=0100	db_igual=1 db_temjogada=1 db_jogadafeita=0100 db_memoria=0100 db_contagem=1 leds=0100 db_estado=(0011 ate 0110)	ok
Aguardar mais de 3 segundos	iniciar=0 reset=0 chaves=0000	errou=1 pronto=1 db_igual=0 db_estado=1000 db_jogada=0001 db_memoria=1000 leds=0000 db_timeout=1	ok

6.4 APROFUNDAMENTO

1. Jogadas consecutivas em menos de 3 ms

Caso fosse possível fisicamente realizar duas jogadas consecutivas em um intervalo menor que 3 ms, apenas a segunda jogada seria considerada válida. Isso ocorre devido ao funcionamento do registrador no fluxo de dados. Após ser ativado pela primeira jogada, o registrador retorna o sinal de entrada com um pequeno atraso. Se o sinal de entrada for alterado durante esse intervalo, o componente armazenará apenas o novo valor, ignorando a primeira jogada.

2. Função de seleção de nível de dificuldade

Para implementar a seleção de nível de dificuldade, as seguintes alterações seriam necessárias:

Fluxo de Dados: Adicionar um comparador que utiliza o sinal `nivel` como entrada. Se `nivel=0`, o contador interrompe o jogo ao atingir 8 jogadas corretas, ativando o sinal de fim. Caso `nivel=1`, o circuito segue o funcionamento atual, exigindo 16 acertos para finalizar o jogo.

Unidade de Controle: Introduzir um estado novo chamado `seleciona_nivel`, que será responsável por configurar o comportamento inicial do jogo conforme o valor do sinal `nivel`. Esse estado precederia o início do fluxo de jogadas, garantindo que o número correto de comparações fosse considerado ao longo do jogo.

7 CONCLUSÕES

As atividades realizadas neste projeto permitiram o desenvolvimento de um sistema digital em FPGA, consolidando o aprendizado sobre fluxos de dados e unidades de controle. Por meio de testes e simulações, foi possível validar o funcionamento correto do circuito e sua integração com a placa DE0-CV, e foi possível observar o desempenho esperado em diferentes cenários. As estratégias de depuração e montagem foram essenciais para solucionar inconsistências e assegurar a funcionalidade do projeto.

O desafio proposto trouxe um elemento adicional ao sistema, com a implementação de um contador de intervalo entre jogadas. Este recurso não apenas expandiu a complexidade do circuito, mas também exigiu uma abordagem metódica para sua validação. A introdução do estado de timeout demonstrou ser uma decisão estratégica para possíveis evoluções futuras no projeto, fortalecendo o aprendizado e a aplicação de conceitos avançados em projetos digitais.