



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

PCS3635 – LABORATÓRIO DIGITAL I

EXPERIÊNCIA 6 – Projeto Base do Jogo de Memória

Relato da Bancada A3 – Turma 2 – Prof. Reginaldo

Data de Emissão: 10 de Fevereiro 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 INTRODUÇÃO

Esta experiência tem como objetivo a síntese do circuito do jogo genius.

2 DESCRIÇÃO DO PROJETO

A visão conceitual do projeto descreve o desenvolvimento de um sistema digital baseado em Verilog que funciona como um jogo de adivinhação, adaptada da experiência quatro.

O fluxo de dados, adaptado da experiência anterior, é integrado com a unidade de

controle, também adaptada da experiência anterior para formar um sistema cujo funcionamento pode ser descrito pela descrição abaixo. O projeto também conta com síntese do circuito para validações experimentais na FPGA e sua descrição foi dada da seguinte maneira:

Como jogar:

- 1. Para iniciar o jogo, aperte o botão JOGAR.*
- 2. O circuito apresenta a primeira jogada, ativando um dos LEDs de saída.*
- 3. O jogador deve acionar o botão de entrada associado ao LED da jogada apresentada anteriormente.*
- 4. Em seguida, o circuito repete a primeira jogada e apresenta mais uma jogada nos LEDs.*
- 5. O jogador (ou o próximo jogador) deve repetir estas duas primeiras jogadas.*
- 6. O circuito repete as duas primeiras jogadas e acrescenta mais uma jogada nos LEDs de saída.*
- 7. O jogador (ou o próximo jogador) deve repetir estas três jogadas na mesma sequência em que foram apresentadas.*
- 8. O circuito vai aumentar a quantidade de jogadas uma a uma, e o jogador (ou um dos jogadores) deve continuar a repetir a sequência de jogadas apresentadas.*
- 9. O jogador vai ganhar o jogo se conseguir repetir a maior sequência, composta por 16 jogadas. Isso é sinalizado pelo sinal de saída GANHOU.*
- 10. Se cometer um erro em qualquer jogada, o jogador vai perder o jogo. Isso é sinalizado pelo sinal de saída PERDEU.*
- 11. Se o jogador não fizer uma jogada em até 5 segundos, o jogador também vai perder o jogo e isso é sinalizado pelos sinais de saída TIMEOUT e PERDEU.*
- 12. Em TODAS as situações de término (itens “9”, “10” e “11”), o final de jogo*

também é sinalizado pela saída PRONTO.

13. Depois do final do jogo anterior, para iniciar um novo jogo, basta apertar o botão JOGAR.

14. Se desejar parar um jogo em andamento, basta apertar o botão RESET.

3 DETALHAMENTO DO PROJETO LÓGICO

3.1 PROJETO DO FLUXO DE DADOS

O projeto do Fluxo de Dados foi adaptado do Desafio da Experiência 5. Foi retirada a função de nível e feitas as seguintes modificações:

Timers

- adicionado contadorLedsOn = timer do tempo que os leds ficam ligados;
- saída fimLedsOn = sinaliza quando o timer dos leds ligados chega ao fim;
- adicionado contadorLedsOff = timer do tempo que os leds ficam desligados;
- saída fimLedsOff = sinaliza quando o timer dos leds desligados chega ao fim;

Registrador com leds

- adicionado registradorLeds = guarda, pelo tempo que os leds estão acesos, o valor que os leds irão mostrar;
- saída leds = agora os led mostram a sequência de jogadas que precisa ser feita pelo jogador.

Registrador timeout - um registrador síncrono de 1bit para armazenar o timeout. No nosso antigo circuito, o timeout era só um pulso, agora armazenamos ele para em caso de timeout, ele fique ativo até um reset ou um jogar

O contadorE de endereço está sendo responsável por varrer a memória tanto na mostra dos leds quanto durante a jogada

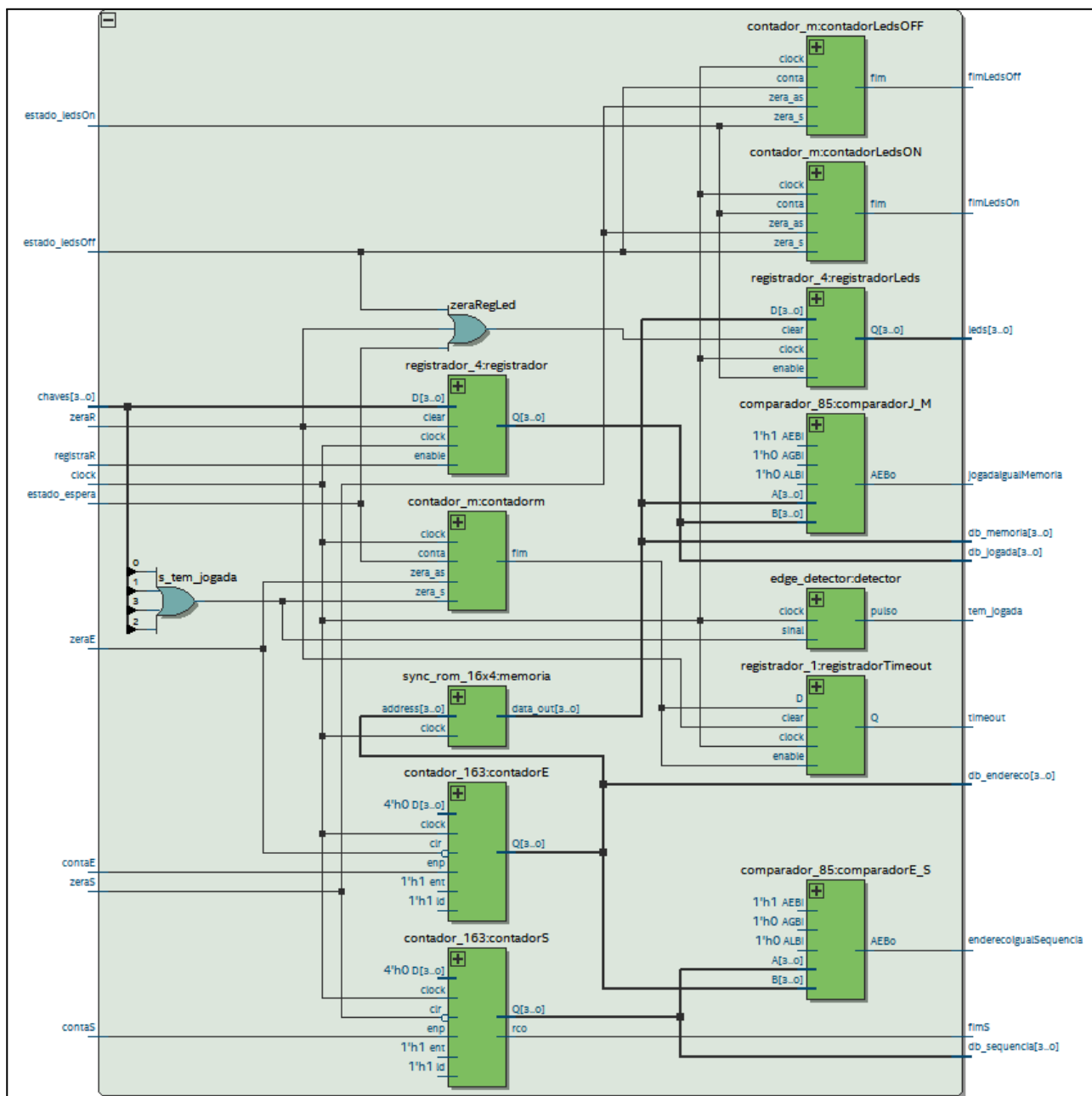


Figura 1: Estrutura interna do fluxo de dados

3.2 PROJETO DA UNIDADE DE CONTROLE

A lógica de execução do programa foi adaptada da experiência 5, e pode ser resumida pelo bloco de pseudocódigo a seguir:

```
Algoritmo: Jogo Base do Desafio da Memória
entradas:   jogar, botoes
saídas:    leds, ganhou, perdeu, timeout, pronto
depuração (sugestão): contagem, memória, sequencia, estado, jogada_feita,
                        enderecoIgualSequencia, chavesIgualMemoria

1. {
2.   while (verdadeiro) {
3.     espera acionamento do sinal jogar
4.     inicia circuito para condições iniciais do jogo
5.     inicia circuito para a sequência inicial (rodada 1)
6.     while (não atingir final do jogo e não ocorrer uma jogada errada ou timeout) {
7.       // primeiro, mostra a sequência da rodada nos leds
8.       while (não atingir jogada final da sequência da rodada atual) {
9.         mostra na saída leds a jogada armazenada na memória
10.        mantém a saída leds ativada pelo tempo de apresentação (0,5s)
11.        atualiza (incrementa) endereço para a próxima jogada da rodada
12.      }
13.      volta para início da sequência de jogadas da rodada atual
14.      // na próxima etapa, jogador interage e faz as jogadas da rodada,
15.      // que foram mostradas anteriormente
16.      while (não atingir jogada final da rodada atual e jogada foi correta e
17.             não ocorreu timeout da jogada) {
18.        espera jogada decorrente de acionamento de botoes
19.        armazena jogada efetuada
20.        compara jogada efetuada com jogada armazenada
21.        atualiza (incrementa) endereço para a próxima jogada
22.      }
23.    }
24.  }
```

Figura 2: Fragmento do Pseudocódigo da lógica de execução do circuito

A unidade de controle também foi adaptada da experiência anterior. O diagrama a seguir mostra em cores os novos estados propostos para atender às exigências do novo enunciado. Cinco novos estados foram adicionados dentro do macroestado de controle dos leds.

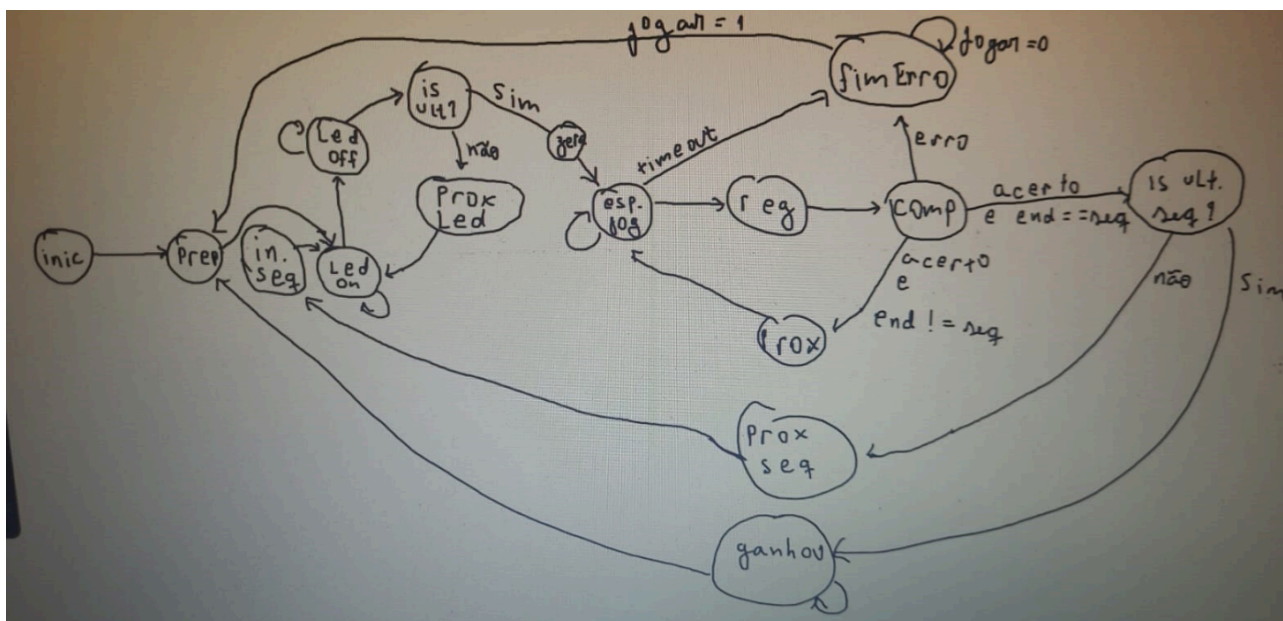


Figura 3: Diagrama de estados da unidade de controle

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, responsável por zerar o registrador e o contador	Preparação	Quando o iniciar é ativado ele muda de estado na subida de clock. Até lá continua no estado inicial.
Preparação	Zera o contador e o registrador	Espera jogada	Na próxima subida de clock.
Inicia Sequência	Zera contadorE, marca o início do macroestado de controle de leds	espera Jogada	Na próxima subida de clock.
Leds On	Leds mostram por 0,5s a jogada a ser feita	Leds Off	Quando receber o sinal fimLedsOn, na próxima subida de clock

Nome do Estado	Descrição do Estado	Próximo Estado	Condições e Justificativas para a Transição entre Estados
Leds Off	Leds ficam apagados por 0,5s	isUltimoLed	Quando receber o sinal fimLedsOff, na próxima subida de clock
isUltimoLed	Verifica se há mais jogadas para serem mostradas pelos Leds	1: Próximo Led 2: Zera Endereço	1: se endereçoIguarSequencia = 0 2: se endereçoIguarSequencia = 1
Zera Endereço	Zera o contador E para passar para o macroestado de jogadas	Espera Jogada	Na próxima subida de clock
Próximo Led	Incrementa 1 no contador de enderecos (contaE=1) para mostrar a próxima jogada a ser feita pelo jogador	Leds On	Na próxima subida de clock
Espera Jogada	Espera ativação das chaves (sinal jogada_feita) Marca o início do macroestado de jogadas	Registra	Quando jogada_feita é ativado ele muda de estado na subida de clock. Até lá continua no mesmo estado.
Registra	Registrar o valor das chaves	Comparação	Na próxima subida de clock
Comparação	Verifica se a contagem terminou	1:Próximo 2:Fim com	1: se jogadaIguarMemoria = 0 e os números comparados são

Nome do Estado	Descrição do Estado	Próximo Estado	Condições e Justificativas para a Transição entre Estados
	e compara o valor do contador e as chaves.	Acerto 3:Fim com Erro	diferentes, vai para o estado Final com Erro na próxima subida de clock 2: se jogadalqualMemoria = 1 e estamos na última sequência, vai para o estado Acerto na próxima subida de clock 3: se JogadalqualMemoria =1 e não estamos na última sequência, vai para o estado Próximo na próxima subida de clock
Próximo	Acrescenta um na contagem do contador	Espera Jogada	Na Próxima subida de clock
Última Sequência	É a última sequência	1. Fim com Acerto 2.Proxima Sequência	1; se FimS = 1, vai para o estado Fim com Acerto na próxima subida de clock 2. se FimS - 0 vai pra o estado Próxima Sequência na próxima subida de clock
Proxima Sequência	contaS = 1(aumenta o tamanho da próxima sequência)	Inicia sequência	Na proxima subida de clock
Fim com Acerto	Indica o fim acertando todos os números, além de	Preparação	Só muda de estado com iniciar

Nome do Estado	Descrição do Estado	Próximo Estado	Condições e Justificativas para a Transição entre Estados
	levantar a flag acertou		
Fim com Erro	Indica o fim acertando todos os números, além de levantar a flag errou	Preparação	Só muda de estado com iniciar

Em **VERDE**, o macroestado de controle e exibição de leds;

Em **AMARELO**, o macroestado de jogadas.

3.3 PROJETO DO SISTEMA DIGITAL

Sinais de Controle: A novidade agora são os sinais que controlam a exibição das jogadas pelos Leds: estadoLedsOn e estadoLedsOff, que são usados como enable dos contadores timers dos leds on e off.

Sinais de Condição: Os novos sinais de condição são fimLedsOn e fimLedsOff, suas respectivas funções podem ser encontradas na tabela de transição de estados.

Sinais de Depuração: As saídas com nome display_(nome) serão exibidas no display de 7 segmentos da placa FPGA.

- db_enderecolgualSequencia
- db_fimS
- db_jogadalgualMemoria
- db_tem_jogada
- display_sequencia (HEX03)
- display_jogada (HEX02)

- **display_memoria (HEX01)**
- **display_endereco (HEX00)**
- **display_estado (HEX04)**

Saídas:

- **ganhou**
- **perdeu**
- **pronto**
- **timeout**
- **leds[3:0]**

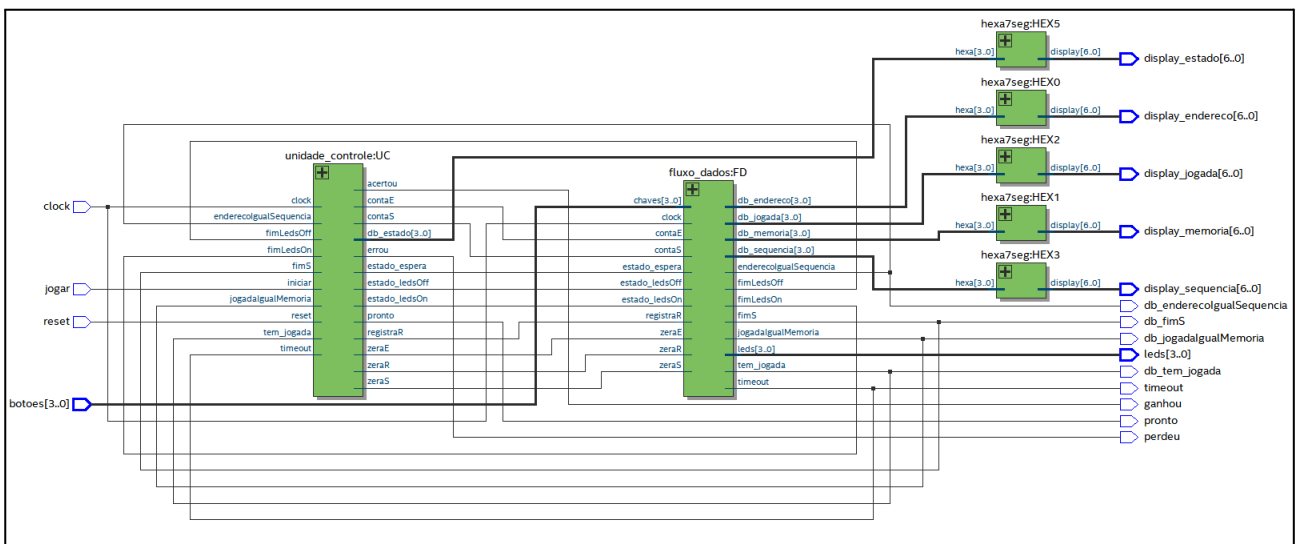


Figura 4: Esquema do circuito completo com RTLView

4 PLANO DE TESTES DO SISTEMA E SIMULAÇÕES

4.1 CENÁRIO DE TESTE 1 – VENCEDOR NO PRIMEIRO JOGO E ACERTO DAS 3 PRIMEIRAS

SEQUÊNCIAS E ERRO NA 2ª JOGADA DA 4ª SEQUÊNCIA NO SEGUNDO JOGO

Para o caso de testes 1, o ModelSim apresentou um problema de overflow do contador de 32 bits para a temporização, uma vez que estávamos contando segundos com uma escala de nanossegundos. Para evitar isso, a simulação foi feita com uma exibição de leds com 0,05s em vez de 0,5s seguindo recomendação do Professor Edson

Midorikawa. Para a implementação do teste na FPGA, foi utilizado o tempo normal de 0,5s

Tabela 2 – Descrição e Resultados Simulados do Cenário de Teste 1

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
c.i.	Condições Iniciais	reset, iniciar e chaves zerados	perdeu, ganhou, fim e LEDs zerados	sim
1	“Resetar” circuito	acionar reset	perdeu, ganhou, fim e LEDs zerados	sim
2	Aguardar alguns segundos	reset, iniciar e chaves zerados	perdeu, ganhou, fim e LEDs zerados	sim
3	Acionar sinal iniciar	acionar iniciar	perdeu, ganhou, fim e LEDs zerados	sim
4	Sequência 1	acionar chave(0)	LED(0) liga	sim
5	Sequência 2	acionar chave(0) acionar chave(1)	LED(0) liga LED(1) liga	sim
6	Sequência 3	acionar chave(0) acionar chave(1) acionar chave(2)	LED(0) liga LED(1) liga LED(2) liga	sim
7	Sequência 4	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga	sim
8	Sequencia 5	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga	sim
9	Sequencia 6	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga	sim
10	Sequencia 7	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga	sim
11	Sequencia 8	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga	sim

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
		acionar chave(0)	LED(0) liga	
12	Sequencia 9	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(1)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga	sim
13	Sequencia 10	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(1) acionar chave(1)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga LED(1) liga	sim
14	Sequencia 11	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(1) acionar chave(1) acionar chave(2)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga LED(1) liga LED(2) liga	sim
15	Sequência 12	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(1) acionar chave(1) acionar chave(2) acionar chave(2)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga LED(1) liga LED(2) liga LED(2) liga	sim
16	Sequencia 13	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga	sim

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
		acionar chave(1) acionar chave(1) acionar chave(2) acionar chave(2) acionar chave(3)	LED(1) liga LED(1) liga LED(2) liga LED(2) liga LED(3) liga	
17	Sequencia 14	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(1) acionar chave(1) acionar chave(2) acionar chave(2) acionar chave(3) acionar chave(3)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga LED(1) liga LED(2) liga LED(2) liga LED(3) liga LED(3) liga	sim
18	Sequencia 15	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(1) acionar chave(1) acionar chave(2) acionar chave(2) acionar chave(3) acionar chave(3) acionar chave(0)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga LED(1) liga LED(2) liga LED(2) liga LED(3) liga LED(3) liga LED(0) liga	sim
19	Sequencia 16	acionar chave(0) acionar chave(1) acionar chave(2) acionar chave(3) acionar chave(2) acionar chave(1) acionar chave(0) acionar chave(0) acionar chave(0) acionar chave(1) acionar chave(1) acionar chave(1) acionar chave(2) acionar chave(2) acionar chave(3) acionar chave(3) acionar chave(3) acionar chave(0) acionar chave(2)	LED(0) liga LED(1) liga LED(2) liga LED(3) liga LED(2) liga LED(1) liga LED(0) liga LED(0) liga LED(1) liga LED(1) liga LED(1) liga LED(2) liga LED(2) liga LED(3) liga LED(3) liga LED(3) liga LED(0) liga LED(2) liga saídas pronto e ganhou	sim

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
			ativadas	
20	Acionar sinal iniciar	acionar iniciar	perdeu, ganhou, fim e LEDs zerados	sim
21	Sequência 1	acionar chave(0)	LED(0) liga	sim
22	Sequência 2	acionar chave(0) acionar chave(1)	LED(0) liga LED(1) liga	sim
23	Sequência 3	acionar chave(0) acionar chave(1) acionar chave(2)	LED(0) liga LED(1) liga LED(2) liga	sim
24	Sequência 4	acionar chave(0) acionar chave(3)	LED(0) liga, LED(3) liga, saídas pronto e erro ativadas	sim

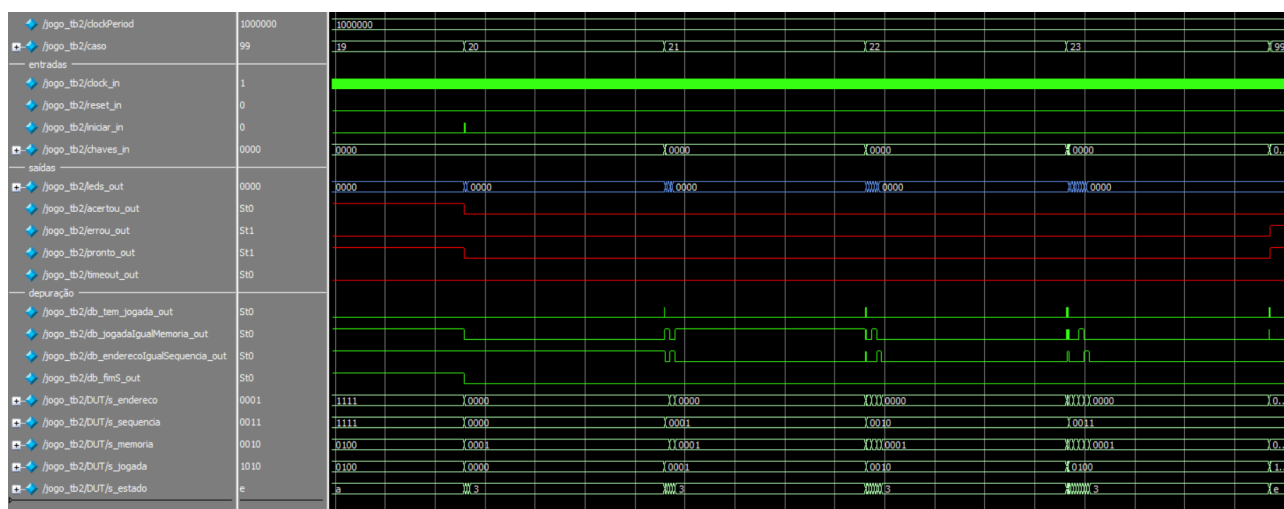


Figura 5: Fragmento final do Cenário de Testes 1

4.2 CENÁRIO DE TESTE 2 – TIMEOUT

Tabela 3 – Descrição e Resultados Simulados do Cenário de Teste 2

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
c.i.	Condições Iniciais	reset, iniciar e chaves zerados	perdeu, ganhou, fim e LEDs zerados	sim
1	“Resetar” circuito	acionar reset	perdeu, ganhou, fim e LEDs zerados	sim
2	Aguardar alguns segundos	reset, iniciar e chaves zerados	perdeu, ganhou, fim e LEDs zerados	sim
3	Acionar sinal iniciar	acionar iniciar	perdeu, ganhou, fim e LEDs zerados	sim

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
4	Sequência 1	acionar chave(0)	LED(0) liga	sim
5	Sequência 2	acionar chave(0)	LED(0) liga	sim
6	Timeout	espera 5 segundos	timeout ativado	sim

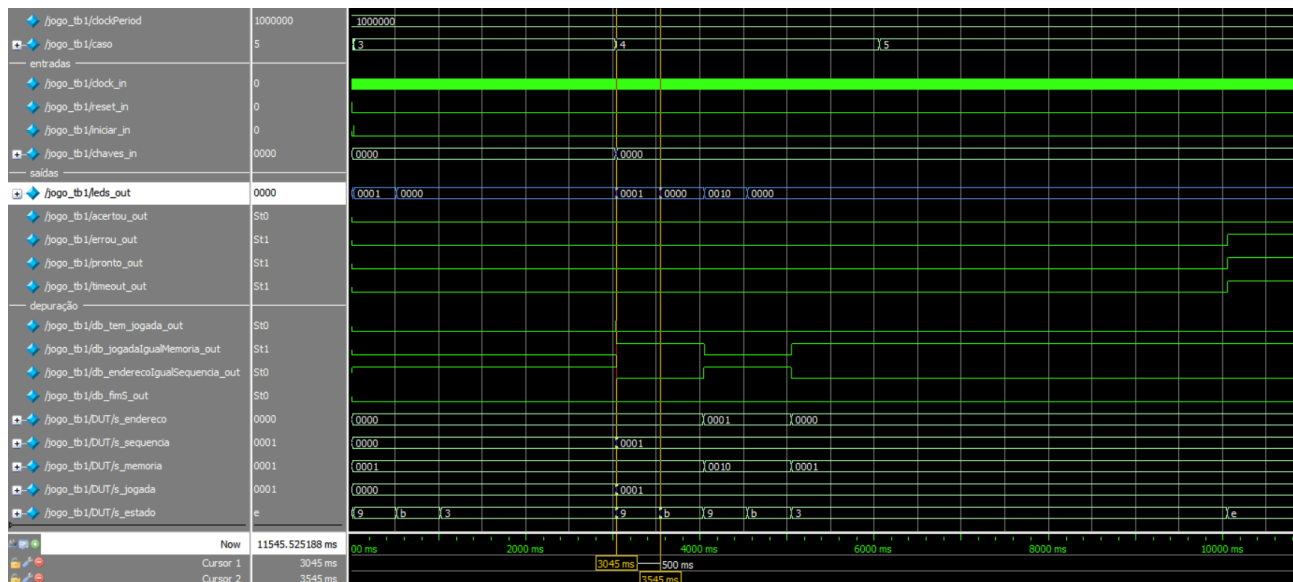


Figura 6: Simulação do Cenário de Teste 2.

5 IMPLANTAÇÃO DO PROJETO

5.1 PINAGEM DA PLACA FPGA

clock	Unknown	PIN_B16	7A	B7A_NO	2.5 V (default)
reset	Unknown	PIN_C16	7A	B7A_NO	2.5 V (default)
jogar	Unknown	PIN_K20	7A	B7A_NO	2.5 V (default)
botoes[0]	Unknown	PIN_K22	5B	B5B_NO	2.5 V (default)
botoes[1]	Unknown	PIN_M21	5B	B5B_NO	2.5 V (default)
botoes[2]	Unknown	PIN_R22	5A	B5A_NO	2.5 V (default)
botoes[3]	Unknown	PIN_T22	5A	B5A_NO	2.5 V (default)
ganhou	Unknown	PIN_A12	7A	B7A_NO	2.5 V (default)
perdeu	Unknown	PIN_B12	7A	B7A_NO	2.5 V (default)
pronto	Unknown	PIN_B13	7A	B7A_NO	2.5 V (default)
timeout	Unknown	PIN_D13	7A	B7A_NO	2.5 V (default)
leds[0]	Unknown	PIN_G17	7A	B7A_NO	2.5 V (default)
leds[1]	Unknown	PIN_J18	7A	B7A_NO	2.5 V (default)
leds[2]	Unknown	PIN_G11	7A	B7A_NO	2.5 V (default)
leds[3]	Unknown	PIN_J11	7A	B7A_NO	2.5 V (default)
db_tem_jogada	Unknown	PIN_AA1	2A	B2A_NO	2.5 V (default)
db_jogada...alMemoria	Unknown	PIN_W2	2A	B2A_NO	2.5 V (default)
db_ledsIg...Sequencia	Unknown	PIN_AA2	2A	B2A_NO	2.5 V (default)
db_fimS	Unknown	PIN_N2	2A	B2A_NO	2.5 V (default)
db_endere...Sequencia	Unknown	PIN_Y3	2A	B2A_NO	2.5 V (default)
display_memoria[0]	Unknown	PIN_AA20	4A	B4A_NO	2.5 V (default)
display_memoria[1]	Unknown	PIN_AB20	4A	B4A_NO	2.5 V (default)
display_memoria[2]	Unknown	PIN_AA19	4A	B4A_NO	2.5 V (default)
display_memoria[3]	Unknown	PIN_AA18	4A	B4A_NO	2.5 V (default)
display_memoria[4]	Unknown	PIN_AB18	4A	B4A_NO	2.5 V (default)
display_memoria[5]	Unknown	PIN_AA17	4A	B4A_NO	2.5 V (default)
display_memoria[6]	Unknown	PIN_U22	4A	B4A_NO	2.5 V (default)
display_jogada[0]	Unknown	PIN_Y19	4A	B4A_NO	2.5 V (default)
display_jogada[1]	Unknown	PIN_AB17	4A	B4A_NO	2.5 V (default)
display_jogada[2]	Unknown	PIN_AA10	3B	B3B_NO	2.5 V (default)
display_jogada[3]	Unknown	PIN_Y14	4A	B4A_NO	2.5 V (default)
display_jogada[4]	Unknown	PIN_V14	4A	B4A_NO	2.5 V (default)
display_jogada[5]	Unknown	PIN_AB22	4A	B4A_NO	2.5 V (default)
display_jogada[6]	Unknown	PIN_AB21	4A	B4A_NO	2.5 V (default)
display_sequencia[0]	Unknown	PIN_Y16	4A	B4A_NO	2.5 V (default)
display_sequencia[1]	Unknown	PIN_W16	4A	B4A_NO	2.5 V (default)
display_sequencia[2]	Unknown	PIN_Y17	4A	B4A_NO	2.5 V (default)
display_sequencia[3]	Unknown	PIN_V16	4A	B4A_NO	2.5 V (default)
display_sequencia[4]	Unknown	PIN_U17	4A	B4A_NO	2.5 V (default)
display_sequencia[5]	Unknown	PIN_V18	4A	B4A_NO	2.5 V (default)
display_sequencia[6]	Unknown	PIN_V19	4A	B4A_NO	2.5 V (default)
display_estado[0]	Unknown	PIN_U20	4A	B4A_NO	2.5 V (default)
display_estado[1]	Unknown	PIN_Y20	4A	B4A_NO	2.5 V (default)
display_estado[2]	Unknown	PIN_V20	4A	B4A_NO	2.5 V (default)
display_estado[3]	Unknown	PIN_U16	4A	B4A_NO	2.5 V (default)
display_estado[4]	Unknown	PIN_U15	4A	B4A_NO	2.5 V (default)
display_estado[5]	Unknown	PIN_Y15	4A	B4A_NO	2.5 V (default)
display_estado[6]	Unknown	PIN_P9	3B	B3B_NO	2.5 V (default)
display_endereco[0]	Unknown	PIN_U21	4A	B4A_NO	2.5 V (default)
display_endereco[1]	Unknown	PIN_V21	4A	B4A_NO	2.5 V (default)
display_endereco[2]	Unknown	PIN_W22	4A	B4A_NO	2.5 V (default)
display_endereco[3]	Unknown	PIN_W21	4A	B4A_NO	2.5 V (default)
display_endereco[4]	Unknown	PIN_Y22	4A	B4A_NO	2.5 V (default)
display_endereco[5]	Unknown	PIN_Y21	4A	B4A_NO	2.5 V (default)
display_endereco[6]	Unknown	PIN_AA22	4A	B4A_NO	2.5 V (default)

Figura 6: Tabela de pinagem do Intel Quartus

5.2 ESTRATÉGIA DE MONTAGEM

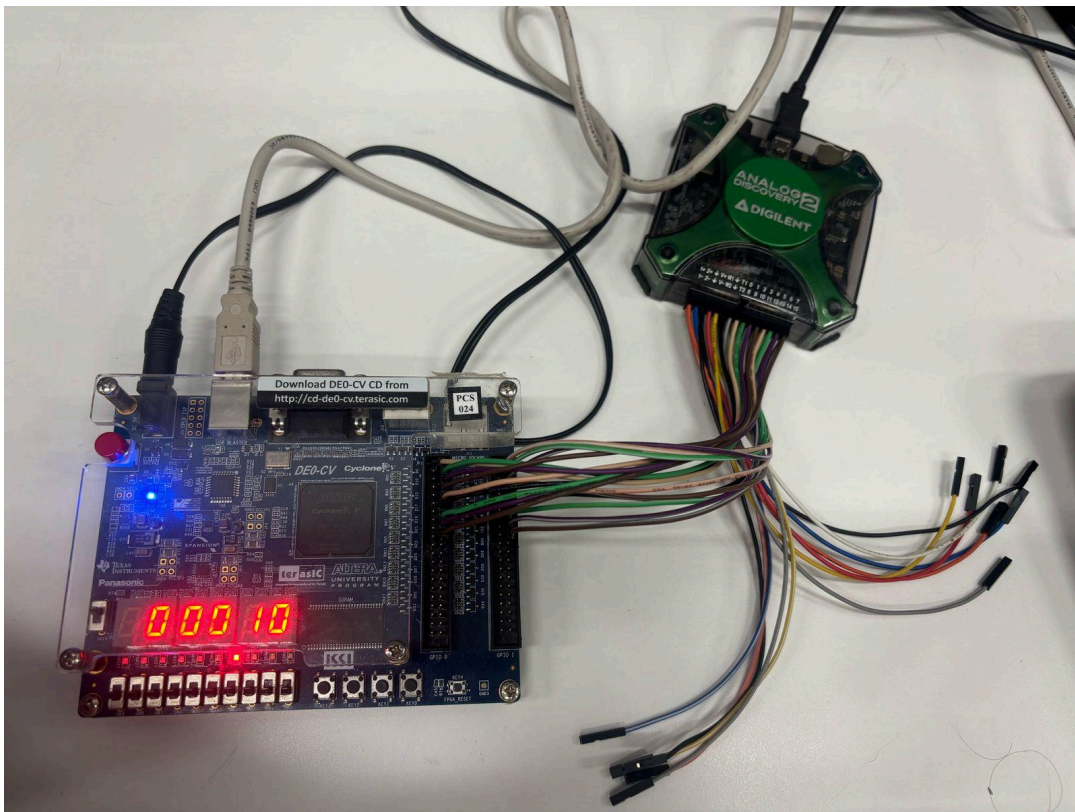


Figura 7: Montagem da placa FPGA + Condições Iniciais

5.3 ESTRATÉGIA DE DEPURAÇÃO

Os sinais de depuração estão sendo mostrados na placa conforme especificado na montagem, alguns são LEDs e outros displays de 7 segmentos. Se surgir algum problema, podemos identificá-lo durante os testes por meio dos sinais de depuração junto com as saídas. Quando um problema for detectado, devemos investigar sua origem realizando testes mais aprofundados.

Se for um problema lógico, as alterações necessárias devem ser feitas no código Verilog para corrigir o erro, e o programa deve ser testado novamente, inicialmente utilizando testbench e ferramentas como o Modelsim. Se o problema for relacionado à montagem, é preciso refazer a montagem corretamente e testar novamente o circuito. Sempre que realizarmos modificações no projeto, seja por causa de um desafio ou outro motivo, devemos repetir os testes anteriores para garantir que as novas alterações não comprometeram o funcionamento das partes que já haviam sido testadas.

5.4 EXECUÇÃO PRÁTICA DO CENÁRIO DE TESTES 1

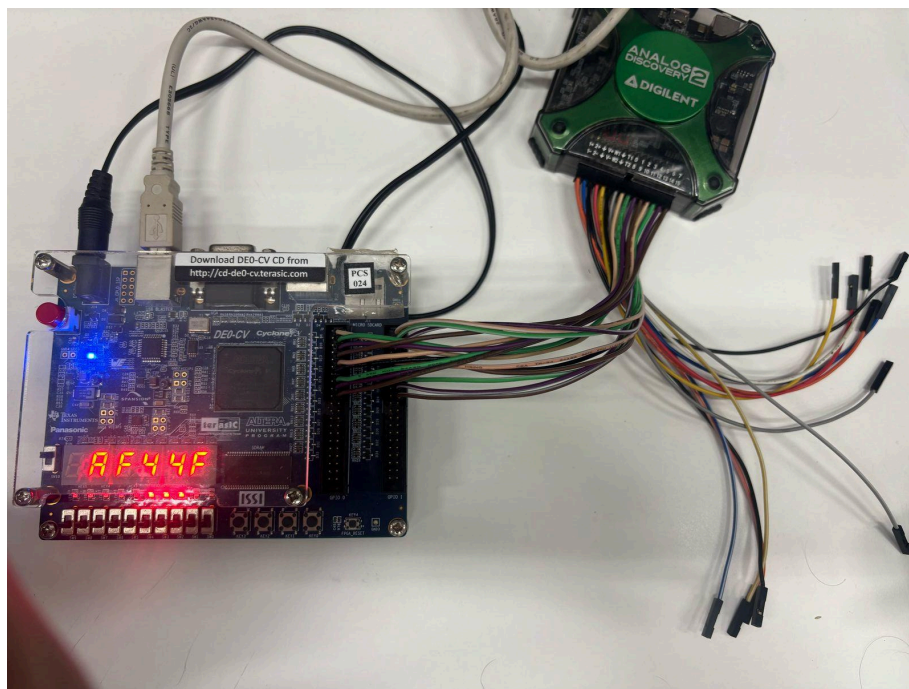


Figura 8: Acerto das 16 seqüências antes do reinício do cenário de testes 1

5.5 EXECUÇÃO PRÁTICA DO CENÁRIO DE TESTES 2

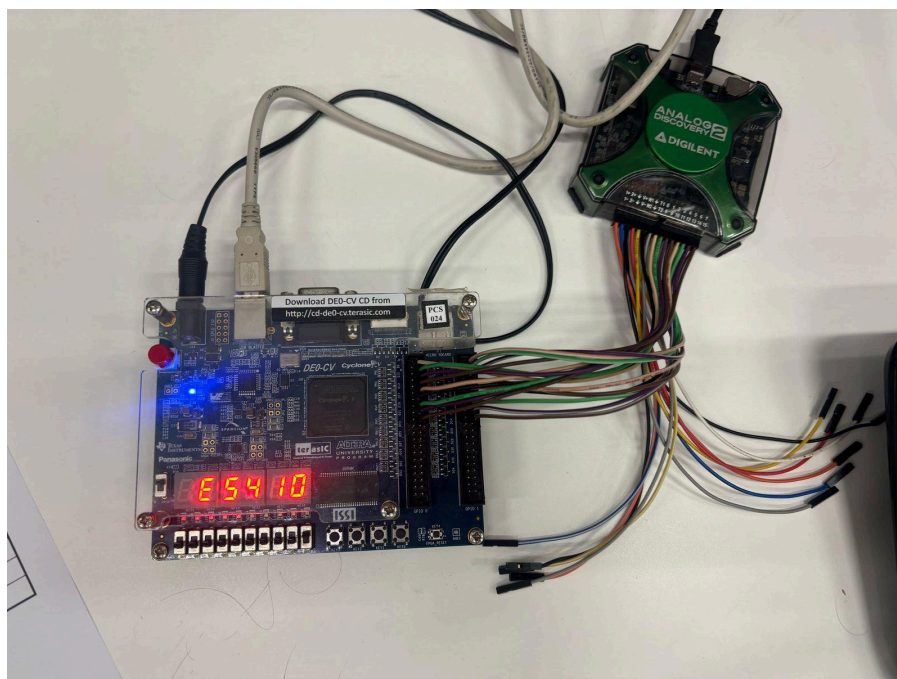


Figura 9: Resultado do timeout do cenário de testes 2

6 PROJETO DO DESAFIO DA EXPERIÊNCIA

6.1 DESCRIÇÃO DO DESAFIO

O objetivo desta atividade é alterar o projeto básico da Atividade 2 da Experiência 6 para implementar a seleção da memória de jogadas a partir de uma nova entrada memória (mapeada na chave SW8 da placa DE0-CV). Para memória=0, deve-se adotar o conteúdo original de jogadas fornecido nas experiências anteriores. O grupo deverá propor o conteúdo da memória de jogadas quando memória=1, obedecendo a restrição de que todas as entradas de jogadas devem ser acionadas nas 16 posições disponíveis.

6.2 DESCRIÇÃO DO PROJETO LÓGICO

A modificação proposta será no Fluxo de Dados, adicionando uma memória adicional e controlando qual memória será usada com um novo sinal de entrada: memória, e um multiplexador.

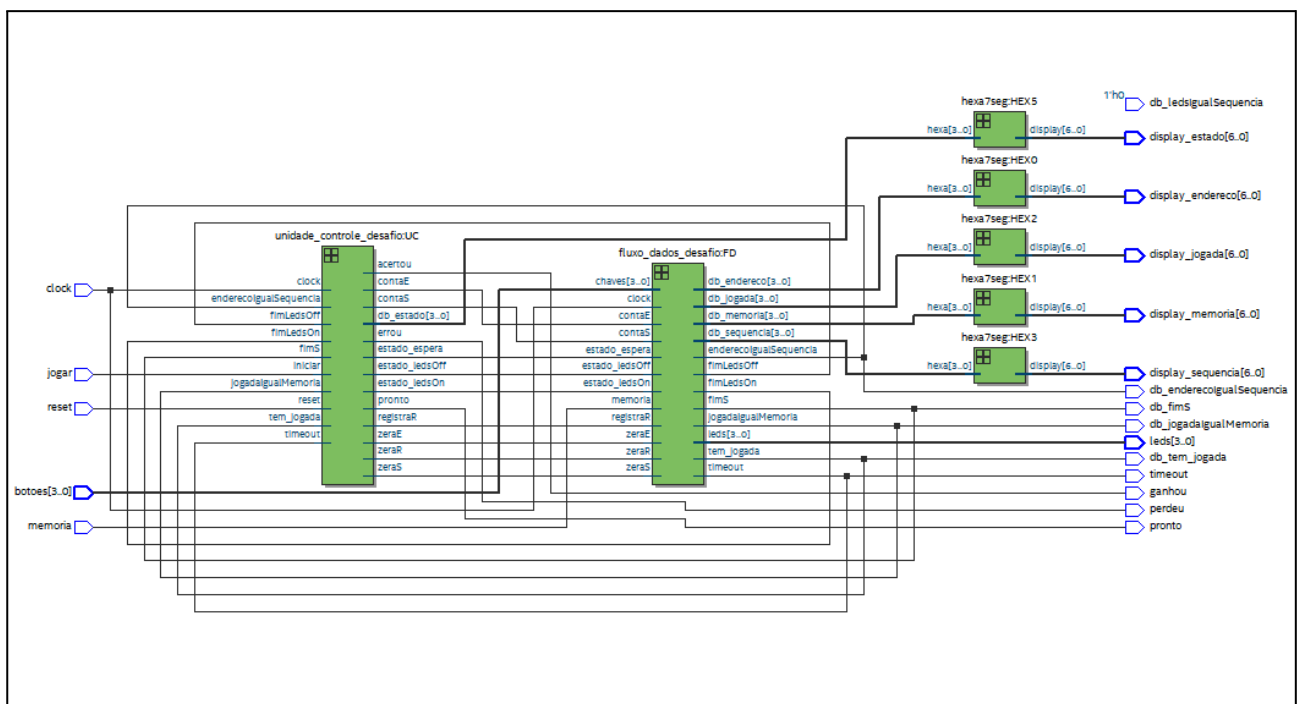


Figura 10: Projeto Lógico do Desafio

out	ganhou	Output	PIN_A12	7A	B7A_NO	PIN_A12	2.5 V
in	jogar	Input	PIN_K20	7A	B7A_NO	PIN_K20	2.5 V
out	leds[3]	Output	PIN_J11	7A	B7A_NO	PIN_J11	2.5 V
out	leds[2]	Output	PIN_G11	7A	B7A_NO	PIN_G11	2.5 V
out	leds[1]	Output	PIN_J18	7A	B7A_NO	PIN_J18	2.5 V
out	leds[0]	Output	PIN_G17	7A	B7A_NO	PIN_G17	2.5 V
in	memoria	Input	PIN_U13	4A	B4A_NO	PIN_U13	2.5 V
out	perdeu	Output	PIN_B12	7A	B7A_NO	PIN_B12	2.5 V
out	pronto	Output	PIN_B13	7A	B7A_NO	PIN_B13	2.5 V
in	reset	Input	PIN_C16	7A	B7A_NO	PIN_C16	2.5 V
out	timeout	Output	PIN_D13	7A	B7A_NO	PIN_D13	2.5 V
<<new node>>							

Figura 11: Pinagem do novo sinal de entrada Memória

A nova memória (memória da posição 1) está carregada da seguinte forma:

```
always @ (posedge clock)
begin
    case (address)
        4'b0000: data_out = 4'b0001;
        4'b0001: data_out = 4'b0001;
        4'b0010: data_out = 4'b0010;
        4'b0011: data_out = 4'b0010;
        4'b0100: data_out = 4'b0100;
        4'b0101: data_out = 4'b0100;
        4'b0110: data_out = 4'b1000;
        4'b0111: data_out = 4'b1000;
        4'b1000: data_out = 4'b0100;
        4'b1001: data_out = 4'b0010;
        4'b1010: data_out = 4'b0001;
        4'b1011: data_out = 4'b0010;
        4'b1100: data_out = 4'b0100;
        4'b1101: data_out = 4'b1000;
        4'b1110: data_out = 4'b0001;
        4'b1111: data_out = 4'b0100;
    endcase
end
```

Figura 12: Nova memória

6.2.1 ALTERAÇÕES NO FLUXO DE DADOS

Adicionamos uma nova memória e um multiplexador para selecionar a memória, de acordo com o sinal `memoria`.

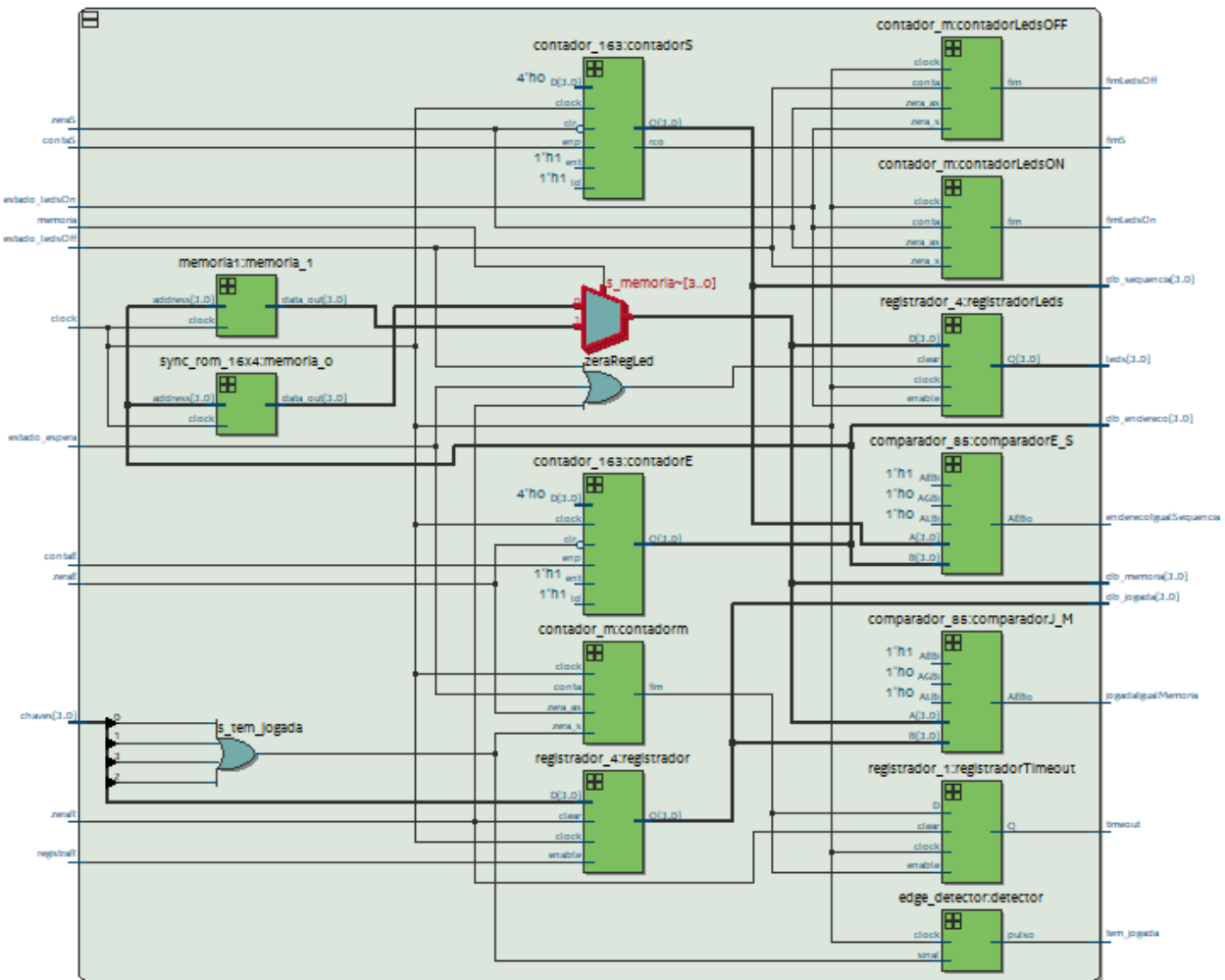


Figura 13: Fluxo de Dados do desafio.

6.3 VERIFICAÇÃO E VALIDAÇÃO DO DESAFIO

Novamente, para os testes usaremos um tempo de exibição dos leds em 0,05s para evitar problemas com o ModelSim. Na FPGA utilizamos o tempo normal de 0,5s. Para testes, repetiremos o Cenário de Testes 1 da atividade, mas dessa vez, começaremos com a chave de memória em um valor, e após o reinício do jogo, trocaremos para outra posição, alterando a memória escolhida.

6.3.1 CENÁRIO DE TESTES 1 - DESAFIO

Memória inicia na posição 0 e depois segue para a posição 1.

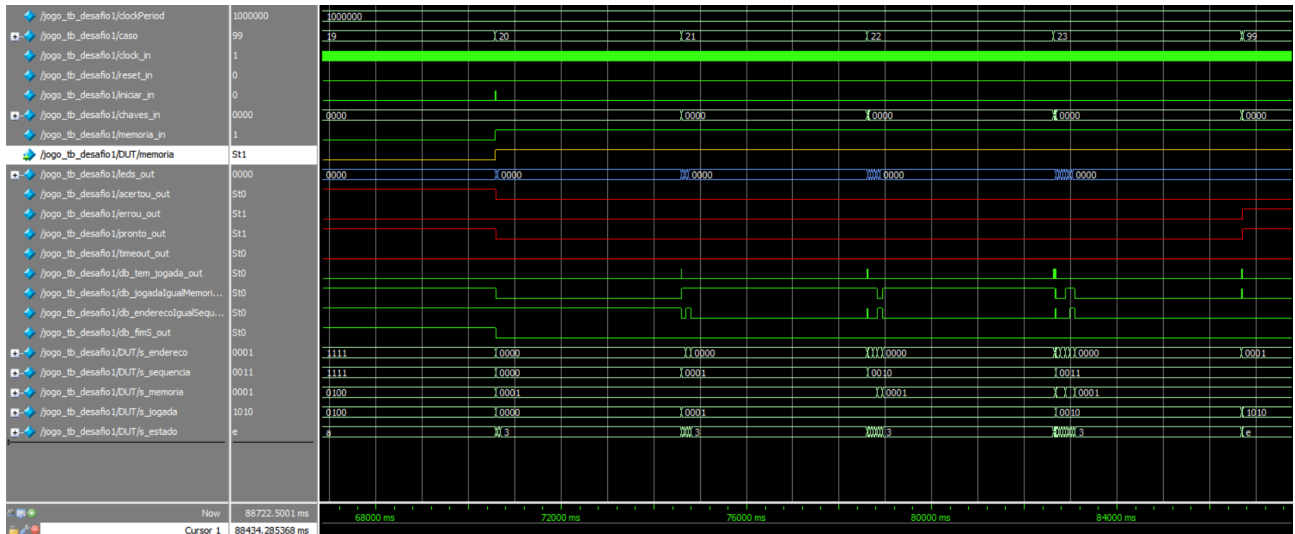


Figura 14: Final do Cenário de Testes 1 do Desafio

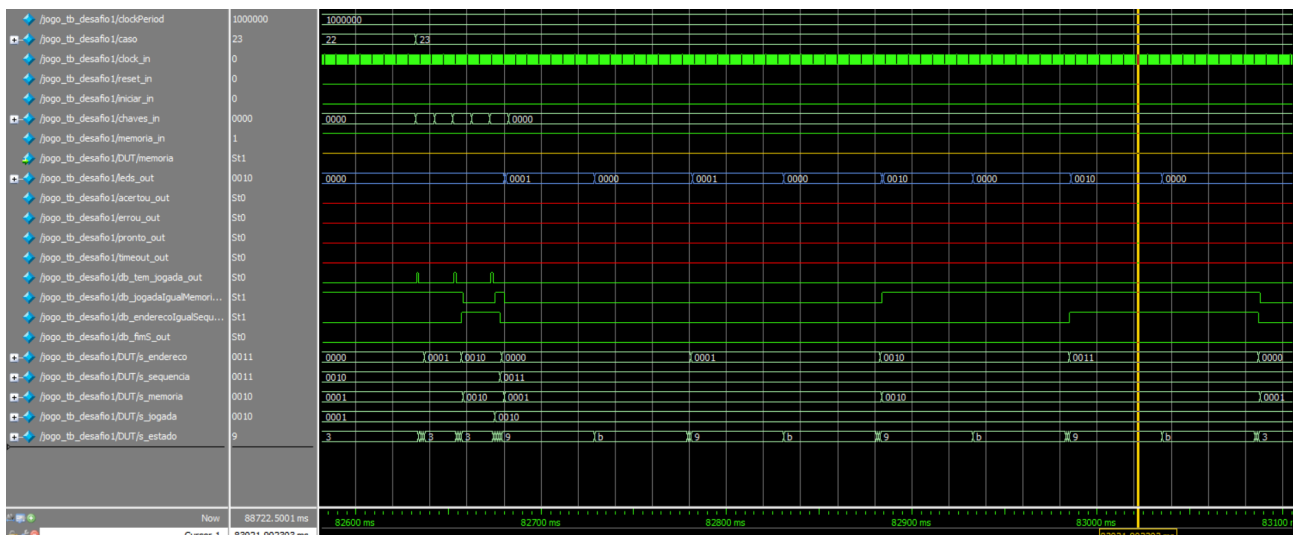


Figura 15: Fragmento do final do Cenário de Testes 1 do Desafio, observe que os leds mostram os dados da memória 1 após o restart do jogo

6.3.2 CENÁRIO DE TESTES 2 - DESAFIO

Memória inicia na posição 1 e depois segue para a 0

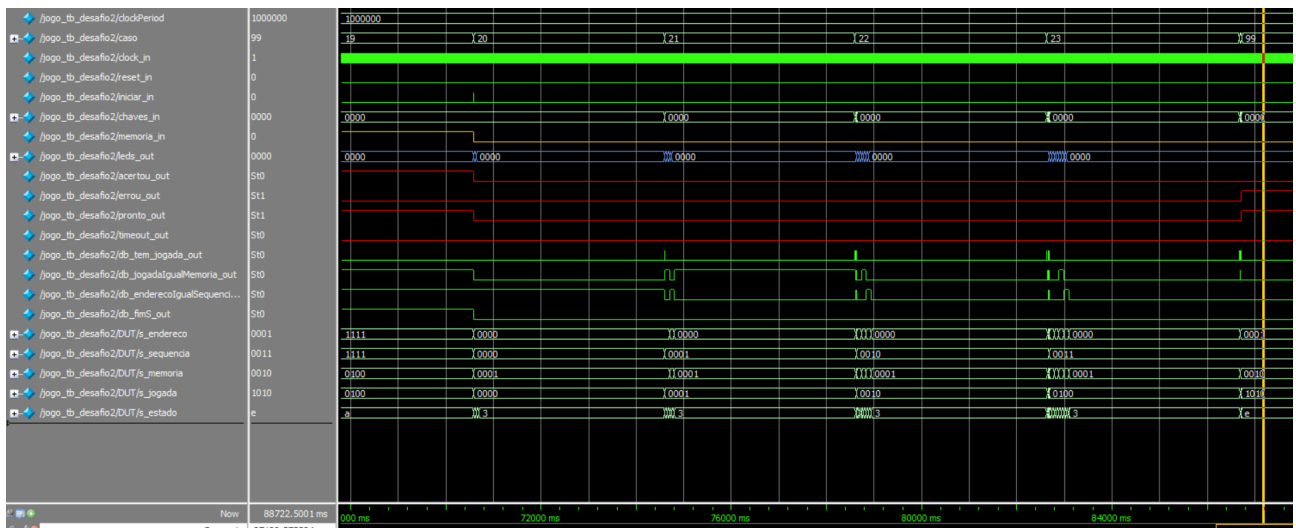


Figura 16: Final do Cenário de Testes 2 do Desafio

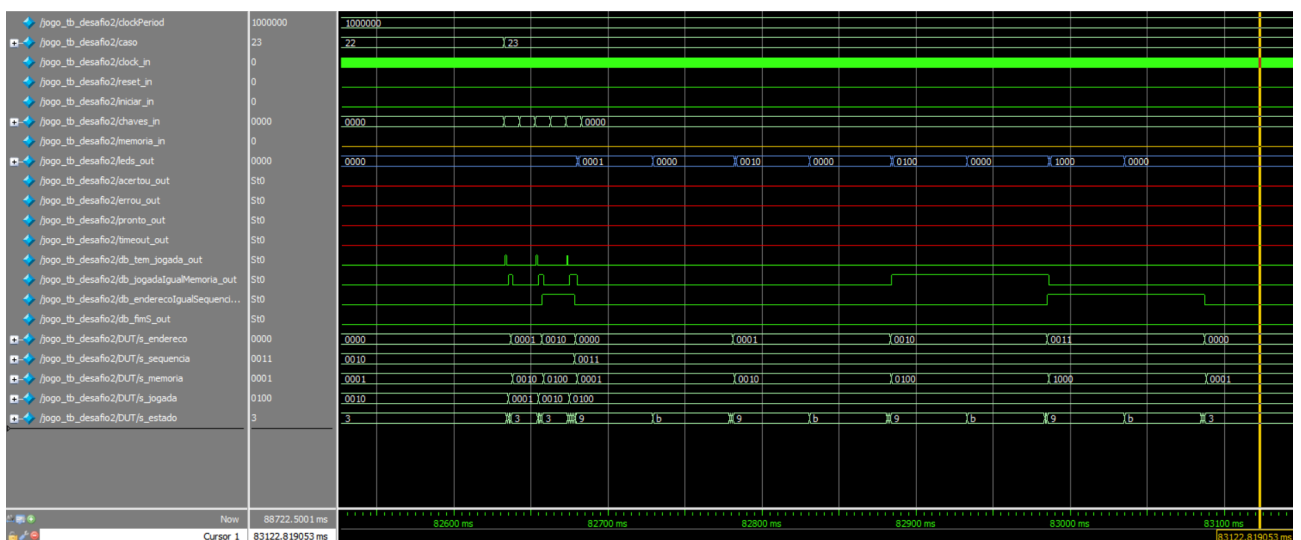


Figura 17: Fragmento do final do Cenário de Testes 2 do Desafio, observe que os leds mostram os dados da memória 0 após o restart do jogo

7 CONCLUSÕES

O desenvolvimento do projeto aconteceu como esperado e todas as saídas bateram com o esperado. Conseguimos desenvolver um sistema parecido com o Genius, e aprendemos a como sintetizá-lo em uma placa FPGA. Durante a aula de laboratório conseguimos testar o circuito. Todos os resultados bateram comprovando o funcionamento do projeto.

Não tivemos grandes problemas para realizar o desafio, conseguindo testar tudo durante a aula.

