



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

Relato da Bancada A3 – Turma 2 – Prof. Reginaldo

Data de Emissão: 27 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 INTRODUÇÃO

Esta experiência tem como objetivo o estudo de aspectos de projeto de circuitos digitais usando um sinal periódico como entrada de clock e a interface de elementos externos como entrada de dados. O circuito da experiência é baseado no projeto desenvolvido na experiência anterior.

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

três.

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:

“O circuito do sistema digital sequencial inclui um conjunto de 16 dados de 4 bits que é armazenado em uma memória interna, cujos endereços são percorridos por meio de um contador interno. Depois do acionamento do sinal reset, o circuito deve aguardar o início de sua operação até o acionamento do sinal de entrada iniciar.

Depois de iniciar seu funcionamento, o circuito deve aguardar o acionamento de uma das chaves de entrada. A ocorrência desse acionamento deve ser indicada pela saída de depuração db_temjogada. O dado das chaves (jogada realizada) deve ser armazenado pelo circuito e apresentado nos leds de saída e também na saída de depuração db_jogadafeita. Em seguida, deve-se comparar o conteúdo armazenado da entrada de jogada com o respectivo dado da memória e deve-se indicar, na saída de depuração db_igual, o resultado da comparação. Em seguida, o contador interno deve ser incrementado para posicionar o acesso à memória interna para o próximo dado. As saídas de depuração db_contagem e db_memoria indicam, respectivamente, o endereço e o dado armazenado pela memória, ao passo que a saída de depuração db_estado, por sua vez, deve indicar o código do estado vigente da Unidade de Controle em determinado instante do funcionamento do sistema digital. Essas quatro saídas (db_jogadafeita, db_contagem, db_memoria e db_estado) devem ser projetadas para serem exibidas em displays de sete segmentos.

O ciclo de espera pela jogada, armazenamento das chaves, comparação e reposicionamento da memória deve prosseguir enquanto o jogador acertar o dado armazenado na memória e até que todos os 16 dados da memória sejam verificados. Se o jogador acertar todos os dados, o sinal de saída acertou deve ser ativado. Se o jogador errar um dado, o ciclo deve ser imediatamente interrompido, e o sinal de saída errou deve ser ativado. Ao final da operação, o sinal de saída pronto também deve ser ativado. Depois disso, o circuito deve aguardar o próximo acionamento do sinal iniciar. Essas três

saídas (acertou, errou e pronto), quando ativadas, devem permanecer ativadas até o reinício da operação do circuito.”

3 DETALHAMENTO DO PROJETO LÓGICO

3.1 PROJETO DO FLUXO DE DADOS

O projeto do fluxo de dados apresentado é extremamente similar ao utilizado na experiência anterior. Dessa forma, apresenta-se nesse relatório apenas as alterações e adaptações feitas.

No fluxo de dados dessa experiência, a principal novidade é a inclusão de um detector de borda, que tem a função de sinalizar para a unidade de controle quando uma jogada é feita através de chaves. O sinal de condição jogado para a unidade de controle é denominado **jogada_feita**. Para isso, o sinal do detector é um OR lógico de todos os sinais das chaves, chamado no diagrama de *senal*. O reset do detector de borda foi deixado em aberto e o circuito funcionou perfeitamente. Apesar disso, o grupo chegou na conclusão de que o sinal de reset do detector poderia ter sido atribuído como $\sim\text{senal}$, uma vez que o detector pode resetar quando a jogada deixar de ser feita.

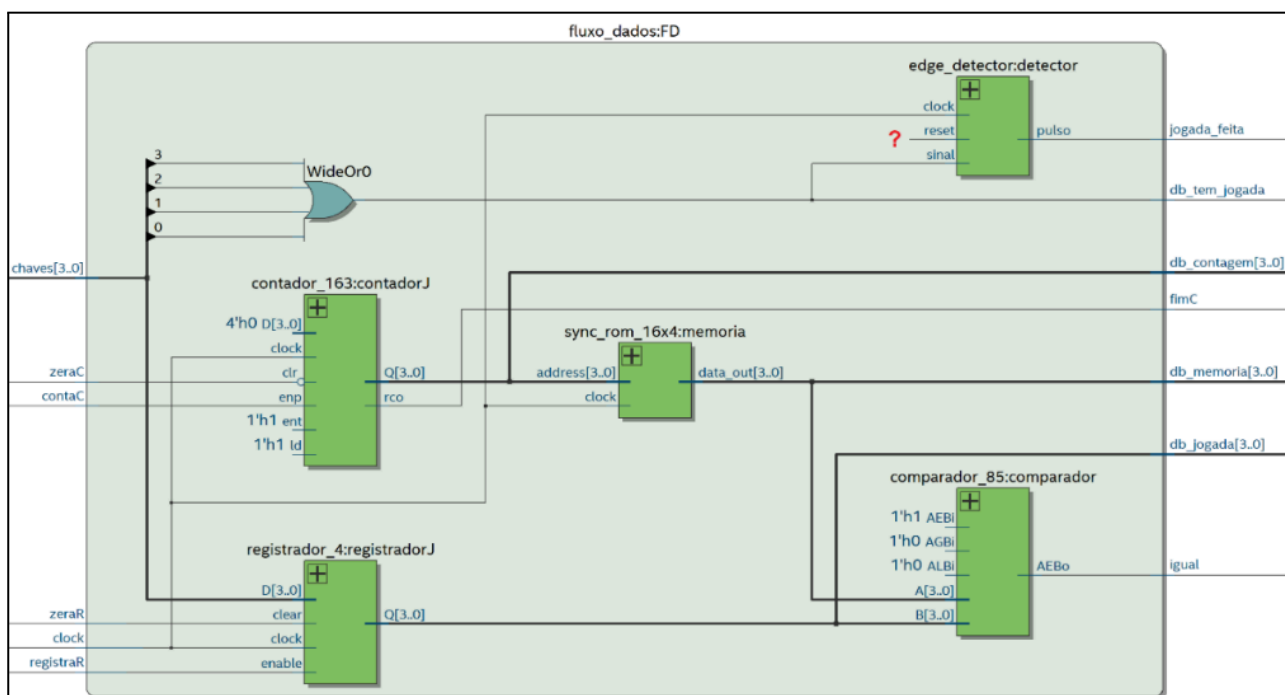


Figura 1: Estrutura interna do fluxo de dados

3.2 PROJETO DA UNIDADE DE CONTROLE

A lógica de execução do programa foi adaptada da experiência 3, e pode ser resumida pelo bloco de pseudocódigo a seguir:

```

Algoritmo: sistema digital simples modificado
entradas: iniciar, chaves
saídas: acertou, errou, pronto, leds
depuração: contagem, memória, estado, jogadafeita, igual, temjogada
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. }

```

Figura 2: 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. Três estados foram adicionados e novas condições para acionamento de sinais de controle e transição de estados foram estabelecidas. Uma alteração em relação ao diagrama é que no estado de preparação, o registrador também é zerado.

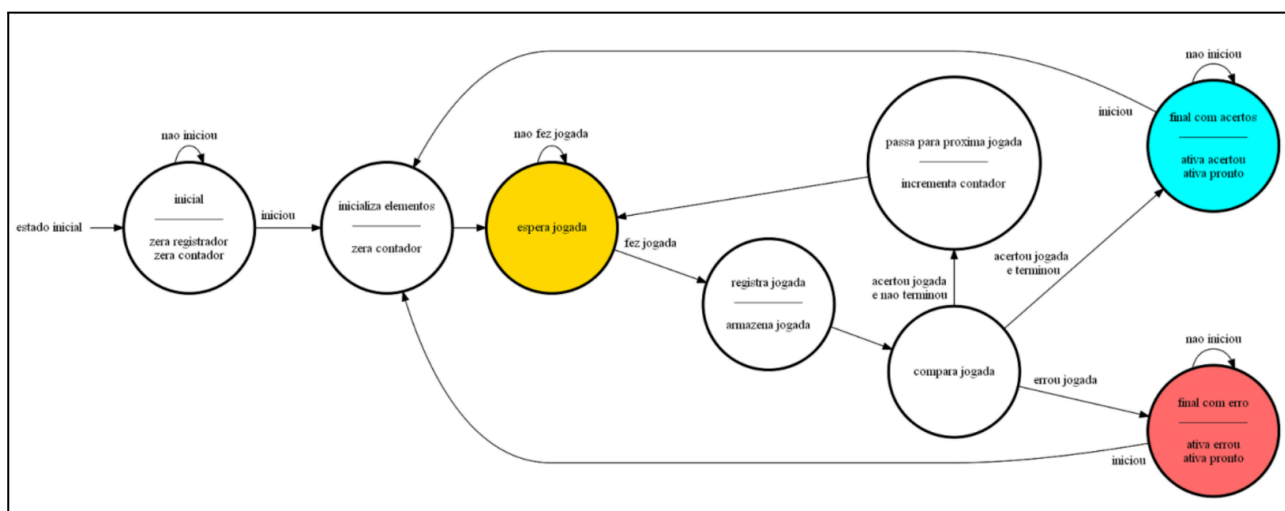


Figura 3: Diagrama de estados da unidade de controle

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.
Espera Jogada	Espera ativação das chaves (sinal jogada_feita)	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 e compara o valor do contador e as chaves.	1:Próximo 2:Fim com Acerto 3:Fim com Erro	<p>1: se fimC = 0 e os números comparados são iguais, vai para o estado Próximo na próxima subida de clock</p> <p>2: se fimC = 1 e os números comparados são iguais vai para o estado Fim com Acerto na próxima subida de clock</p> <p>3: se os números comparados são diferentes, independente de fimC vai para o estado Fim com Erro na próxima subida de clock</p>
Próximo	Acrescenta um na contagem do contador	Espera Jogada	Na Próxima subida de clock
Fim com	Indica o fim	Preparação	Só muda de estado com iniciar

Nome do Estado	Descrição do Estado	Próximo Estado	Condições e Justificativas para a Transição entre Estados
Acerto	acertando todos os números, além de 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

3.3 PROJETO DO SISTEMA DIGITAL

Sinais de Controle: Idem experiência 3.

Sinais de Condição: A novidade agora é o sinal de condição `tem_jogada` enviado do fluxo de dados à unidade de controle, responsável por ativar a transição de estados de “Espera Jogada” para o estado “Registra”, isto é, um indicativo para a unidade de controle de quando a jogada foi feita.

Sinais de Depuração:

- **db_jogadafeita_out:** enviado a um display de 7 segmentos mostrando qual número está sendo jogado pelas chaves.
- **db_tem_jogada_out:** indica se há jogada ou não. Representa o sinal de depuração do sinal de condição `tem_jogada`.

Saídas:

- `acertou_out`, `errou_out`, `pronto` e `leds`. Os leds representam quais chaves estão ativadas, devem mostrar, em binário, o mesmo número mostrado no display de 7 segmentos do sinal de depuração `db_jogadafeita_out`.

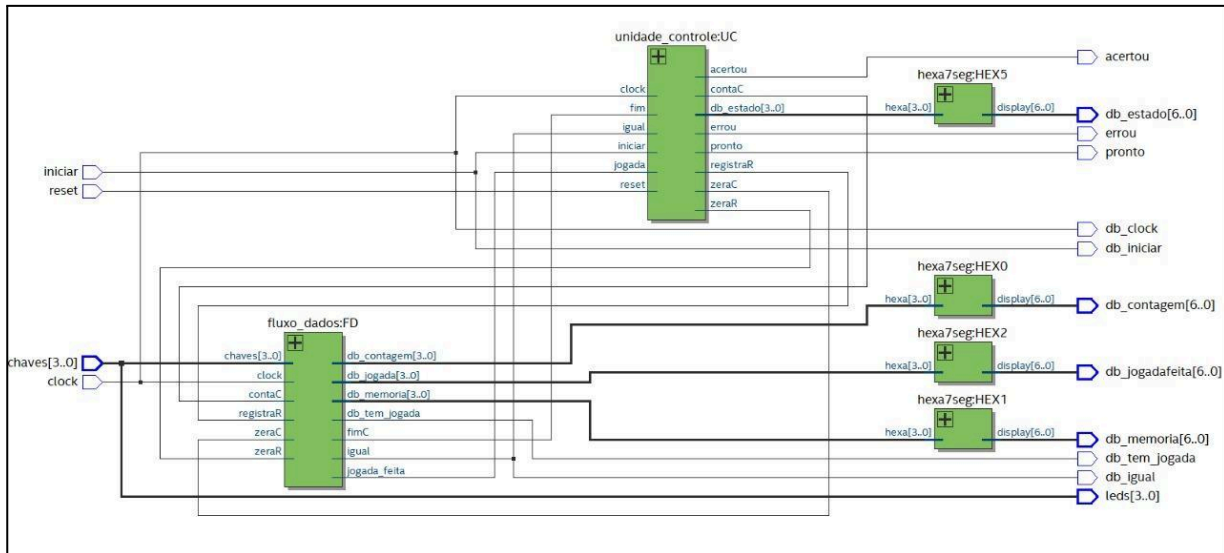


Figura 4: Esquema do circuito completo com RTLView

4 PLANO DE TESTES DO SISTEMA E SIMULAÇÕES

4.1 CENÁRIO DE TESTE 1 – ACERTO DAS 16 JOGADAS

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	errou, acertou, fim e LEDs zerados	sim
1	“Resetar” circuito	acionar reset	errou, acertou, fim e LEDs zerados	sim
2	Aguardar alguns segundos	reset, iniciar e chaves zerados	errou, acertou, fim e LEDs zerados	sim
3	Acionar sinal iniciar	acionar iniciar	errou, acertou, fim e LEDs zerados	sim
4	Acionar primeira entrada (jogada 1)	acionar chave(0)	LED(0) liga	sim
5	Acionar segunda entrada (jogada 2)	acionar chave(1)	LED(1) liga	sim
6	Acionar terceira entrada (jogada 3)	acionar chave(2)	LED(2) liga	sim
7	Acionar quarta entrada (jogada 4)	acionar chave(3)	LED(3) liga	sim
8	Acionar quinta entrada (jogada 5)	acionar chave(2)	LED(2) liga	sim
9	Acionar sexta entrada (jogada 6)	acionar chave(1)	LED(1) liga	sim
10	Acionar sétima entrada (jogada 7)	acionar chave(0)	LED(0) liga	sim
11	Acionar oitava entrada (jogada 8)	acionar chave(0)	LED(0) liga	sim
12	Acionar nona entrada (jogada 9)	acionar chave(1)	LED(1) liga	sim
13	Acionar 10ª entrada (jogada 10)	acionar chave(1)	LED(1) liga	sim
14	Acionar 11ª entrada (jogada 11)	acionar chave(2)	LED(2) liga	sim
15	Acionar 12ª entrada (jogada 12)	acionar chave(2)	LED(2) liga	sim
16	Acionar 13ª entrada (jogada 13)	acionar chave(3)	LED(3) liga	sim
17	Acionar 14ª entrada (jogada 14)	acionar chave(3)	LED(3) liga	sim
18	Acionar 15ª entrada (jogada 15)	acionar chave(0)	LED(0) liga	sim
19	Acionar 16ª entrada (jogada 16)	acionar chave(2)	LED(2) liga, saídas pronto e acertou ativadas	sim

Observações a Respeito da Tabela:

O cenário de testes apresentado anteriormente aborda o caso em que o jogador acerta todas as 16 jogadas. Esse resultado é indicado pelo sinal de saída *acertou*. Como o clock do sistema opera a uma frequência constante e significativamente mais rápida que a entrada do jogador, cada operação é processada em mais de 10 ciclos de clock. Isso garante que a velocidade das jogadas do jogador não afete o funcionamento do sistema.

Contudo, no desafio isso vai mudar.

Imagens do Primeiro Cenário de Testes (Modelsim):

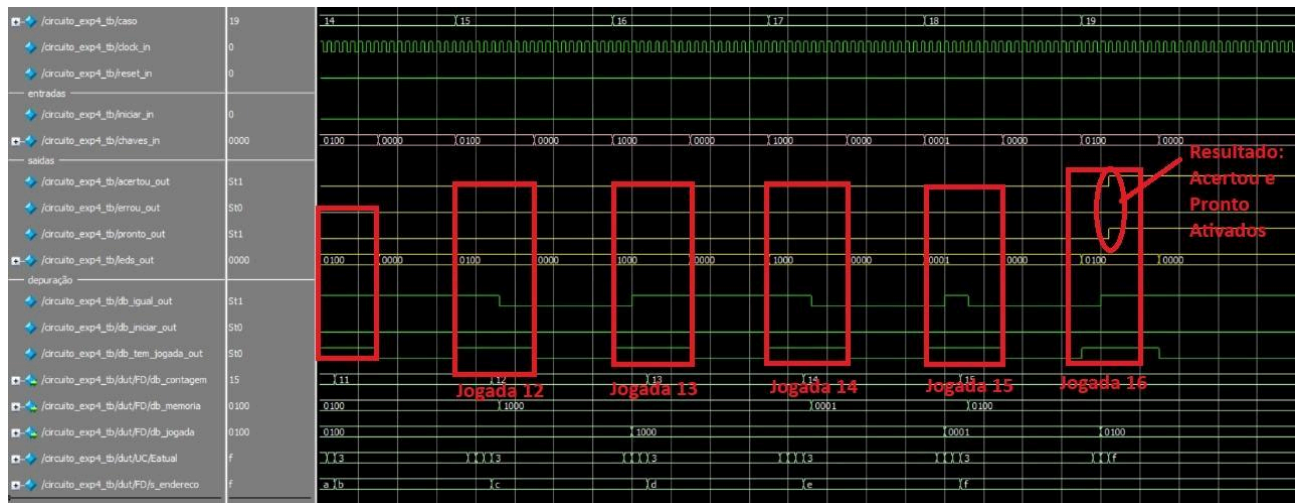


Figura 5: Imagem do Modelsim Testando o Final do Cenário de Teste 1

4.2 CENÁRIO DE TESTE 2 – ACERTO DAS 3 PRIMEIRAS JOGADAS E ERRO NA 4ª JOGADA

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	errou, acertou, fim e LEDs zerados	sim
1	“Resetar” circuito	acionar reset	errou, acertou, fim e LEDs zerados	sim
2	Aguardar alguns segundos	reset, iniciar e chaves zerados	errou, acertou, fim e LEDs zerados	sim
3	Acionar sinal iniciar	acionar iniciar	errou, acertou, fim e LEDs zerados	sim
4	Acionar primeira entrada (jogada 1)	acionar chave(0)	LED(0) liga	sim
5	Acionar segunda entrada (jogada 2)	acionar chave(1)	LED(1) liga	sim
6	Acionar terceira entrada (jogada 3)	acionar chave(2)	LED(2) liga	sim
7	Acionar quarta entrada (jogada 4)	acionar chave(0)	LED(0) liga, saídas pronto e errou ativadas	sim

Observações a Respeito da Tabela:

Nessa tabela o jogador erra na quarta rodada colocando a chave na posição 0 ao invés da posição 3. Essa jogada incorreta levanta o sinal *errou*, caracterizando assim a “derrota” do jogador.

Imagens do Segundo Cenário de Testes (Modelsim):

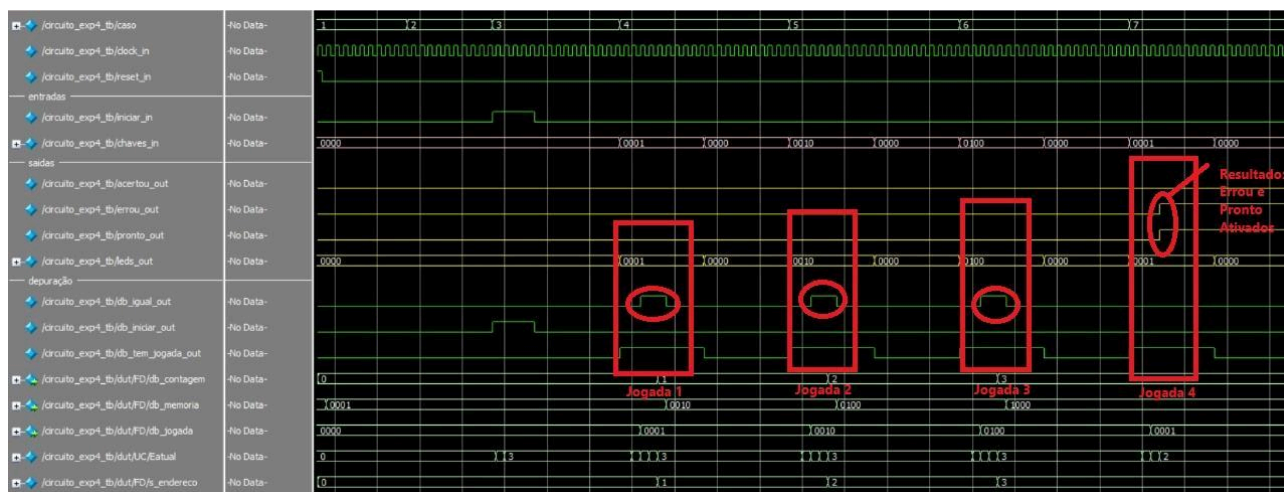


Figura 6: Imagem do Modelsim Testando o Cenário de Teste 2

5 IMPLANTACÃO DO PROJETO

5.1 PINAGEM DA PLACA FPGA

Tabela 4 – Pinagem da Placa FPGA no Pin-Planner do Quartus

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	er Analog Setting:
acertuo	Output	PIN_N1	2A	B2A_NO	PIN_N1	2.5 V		12mA (default)	1 (default)		
chaves[3]	Input	PIN_P19	5A	B5A_NO	PIN_P19	2.5 V		12mA (default)			
chaves[2]	Input	PIN_N19	5B	B5B_NO	PIN_N19	2.5 V		12mA (default)			
chaves[1]	Input	PIN_T22	5A	B5A_NO	PIN_T22	2.5 V		12mA (default)			
chaves[0]	Input	PIN_R22	5A	B5A_NO	PIN_R22	2.5 V		12mA (default)			
clock	Input	PIN_N16	5B	B5B_NO	PIN_N16	2.5 V		12mA (default)			
db_clock	Output	PIN_L2	2A	B2A_NO	PIN_L2	2.5 V		12mA (default)	1 (default)		
db_contagem[6]	Output	PIN_AA22	4A	B4A_NO	PIN_AA22	2.5 V		12mA (default)	1 (default)		
db_contagem[5]	Output	PIN_Y21	4A	B4A_NO	PIN_Y21	2.5 V		12mA (default)	1 (default)		
db_contagem[4]	Output	PIN_Y22	4A	B4A_NO	PIN_Y22	2.5 V		12mA (default)	1 (default)		
db_contagem[3]	Output	PIN_W21	4A	B4A_NO	PIN_W21	2.5 V		12mA (default)	1 (default)		
db_contagem[2]	Output	PIN_W22	4A	B4A_NO	PIN_W22	2.5 V		12mA (default)	1 (default)		
db_contagem[1]	Output	PIN_V21	4A	B4A_NO	PIN_V21	2.5 V		12mA (default)	1 (default)		
db_contagem[0]	Output	PIN_U21	4A	B4A_NO	PIN_U21	2.5 V		12mA (default)	1 (default)		
db_estado[6]	Output	PIN_W19	4A	B4A_NO	PIN_W19	2.5 V		12mA (default)	1 (default)		
db_estado[5]	Output	PIN_C2	2A	B2A_NO	PIN_C2	2.5 V		12mA (default)	1 (default)		
db_estado[4]	Output	PIN_C1	2A	B2A_NO	PIN_C1	2.5 V		12mA (default)	1 (default)		
db_estado[3]	Output	PIN_P14	4A	B4A_NO	PIN_P14	2.5 V		12mA (default)	1 (default)		
db_estado[2]	Output	PIN_T14	4A	B4A_NO	PIN_T14	2.5 V		12mA (default)	1 (default)		
db_estado[1]	Output	PIN_M8	3B	B3B_NO	PIN_M8	2.5 V		12mA (default)	1 (default)		
db_estado[0]	Output	PIN_N9	3B	B3B_NO	PIN_N9	2.5 V		12mA (default)	1 (default)		
db_igual	Output	PIN_U1	2A	B2A_NO	PIN_U1	2.5 V		12mA (default)	1 (default)		
db_iniciar	Output	PIN_y16	4A	B4A_NO	PIN_V13	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[6]	Output	PIN_AB21	4A	B4A_NO	PIN_AB21	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[5]	Output	PIN_AB22	4A	B4A_NO	PIN_AB22	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[4]	Output	PIN_V14	4A	B4A_NO	PIN_V14	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[3]	Output	PIN_Y14	4A	B4A_NO	PIN_Y14	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[2]	Output	PIN_AA10	3B	B3B_NO	PIN_AA10	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[1]	Output	PIN_AB17	4A	B4A_NO	PIN_AB17	2.5 V		12mA (default)	1 (default)		
db_jogadafeita[0]	Output	PIN_Y19	4A	B4A_NO	PIN_Y19	2.5 V		12mA (default)	1 (default)		
db_memoria[6]	Output	PIN_U22	4A	B4A_NO	PIN_U22	2.5 V		12mA (default)	1 (default)		
db_memoria[5]	Output	PIN_AA17	4A	B4A_NO	PIN_AA17	2.5 V		12mA (default)	1 (default)		
db_memoria[4]	Output	PIN_AB18	4A	B4A_NO	PIN_AB18	2.5 V		12mA (default)	1 (default)		
db_memoria[3]	Output	PIN_AA18	4A	B4A_NO	PIN_AA18	2.5 V		12mA (default)	1 (default)		
db_memoria[2]	Output	PIN_AA19	4A	B4A_NO	PIN_AA19	2.5 V		12mA (default)	1 (default)		
db_memoria[1]	Output	PIN_AB20	4A	B4A_NO	PIN_AB20	2.5 V		12mA (default)	1 (default)		
db_memoria[0]	Output	PIN_AA20	4A	B4A_NO	PIN_AA20	2.5 V		12mA (default)	1 (default)		
db_tem_jogada	Output	PIN_L1	2A	B2A_NO	PIN_L1	2.5 V		12mA (default)	1 (default)		
erro	Output	PIN_U2	2A	B2A_NO	PIN_U2	2.5 V		12mA (default)	1 (default)		
iniciar	Input	PIN_U13	4A	B4A_NO	PIN_U13	2.5 V		12mA (default)			
leds[3]	Output	PIN_Y3	2A	B2A_NO	PIN_Y3	2.5 V		12mA (default)	1 (default)		
leds[2]	Output	PIN_W2	2A	B2A_NO	PIN_W2	2.5 V		12mA (default)	1 (default)		
leds[1]	Output	PIN_AA1	2A	B2A_NO	PIN_AA1	2.5 V		12mA (default)	1 (default)		
leds[0]	Output	PIN_AA2	2A	B2A_NO	PIN_AA2	2.5 V		12mA (default)	1 (default)		
pronto	Output	PIN_N2	2A	B2A_NO	PIN_N2	2.5 V		12mA (default)	1 (default)		
reset	Input	PIN_B16	7A	B7A_NO	PIN_B16	2.5 V		12mA (default)			
<<new node>>											

5.2 ESTRATÉGIA DE MONTAGEM

A síntese do circuito foi feito para a placa usando o Quartus, utilizando os guias disponibilizados pelo professor foi feito a conexão do analog discovery, a Figura 7 mostra o resultado da montagem. Nela é possível observar os sinais de monitoramento, a seguir descreveremos eles.

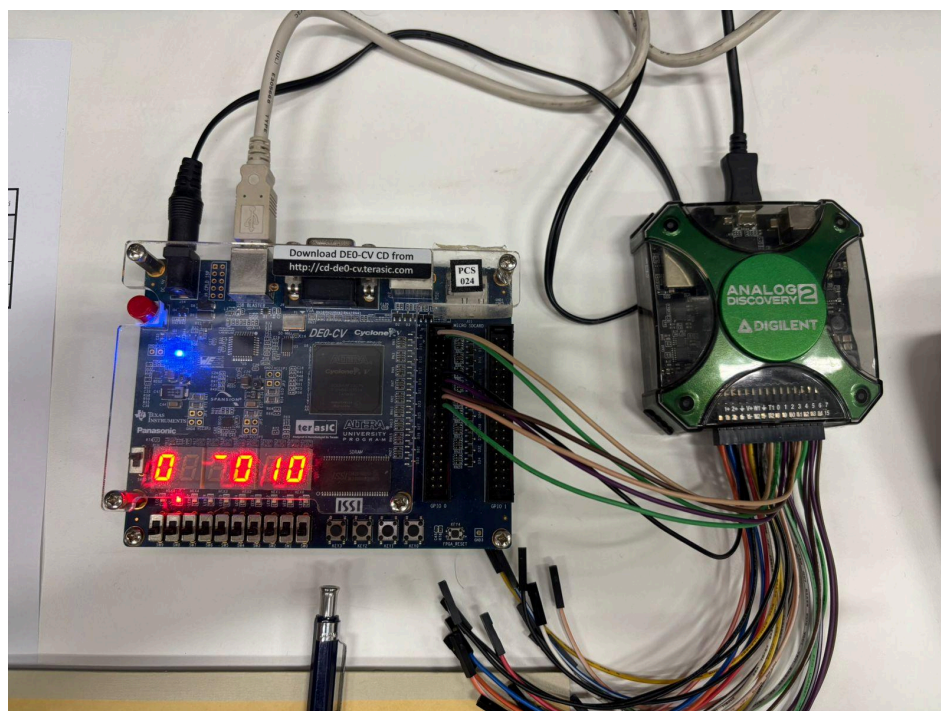


Figura 7: Montagem da Experiência

O primeiro display de 7 segmentos da esquerda pra direita (HEX5) mostra o estado atual do programa, ele não possui muita utilidade na nossa aplicação pois os estados mudam muito rápido tornando inviável observá-lo bem, mas mesmo assim resolvemos implementá-lo para observarmos se está no estado esperaJogada enquanto jogamos. O segundo display utilizado está conectado apenas na entrada mais de cima e ele equivale a entrada iniciar negada, portanto na figura o iniciar é igual a zero. O terceiro display (HEX2) equivale ao sinal de depuração jogada feita, sinal da última jogada feita que fica armazenada. O quarto display (HEX1) é o sinal de depuração da memória, que mostra a saída atual da memória e portanto qual a jogada “correta” a ser feita. O quinto display (HEX0) é a contagem de acertos. O primeiro LED também da esquerda para a direita é temJogada (LED9). O segundo é o *clock* (LED8) que está sempre ligado pois é muito rápido. O terceiro (LED7) é o que indica que a saída da memória é o mesmo das chaves. O quarto (LED6) é o *pronto*, que indica o fim do circuito. O quinto (LED5) e o

sexto(LED4) são *acertou* e *errou* respectivamente. Os próximos 4 LEDs são os conectados às saídas das chaves e ficam acesos enquanto a jogada é feita.

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.

Caso aconteça algum problema, temos como identificá-lo durante os cenários de teste usando os sinais de depuração em conjunto com as saídas. A partir do momento que um problema é identificado temos que buscar a origem do problema testando mais a fundo. Caso o problema for lógico, devemos fazer as alterações necessárias no verilog afim de corrigir o problema, e novamente testar o programa primeiramente através de testbench e softwares como o Modelsim. Caso o problema for de montagem temos que montar da forma correta e testar novamente o circuito. Quando implementarmos mudanças no projeto, por causa do desafio ou qualquer outro motivo, devemos refazer as baterias de teste anteriores a fim de garantir que as novas implementações não prejudicaram o funcionamento do que já tinha sido previamente testado.

5.4 EXECUÇÃO PRÁTICA DO CENÁRIO DE TESTE 1 – ACERTO DAS 16 JOGADAS

Replicamos os testes feitos digitalmente e os resultados foram os esperados.

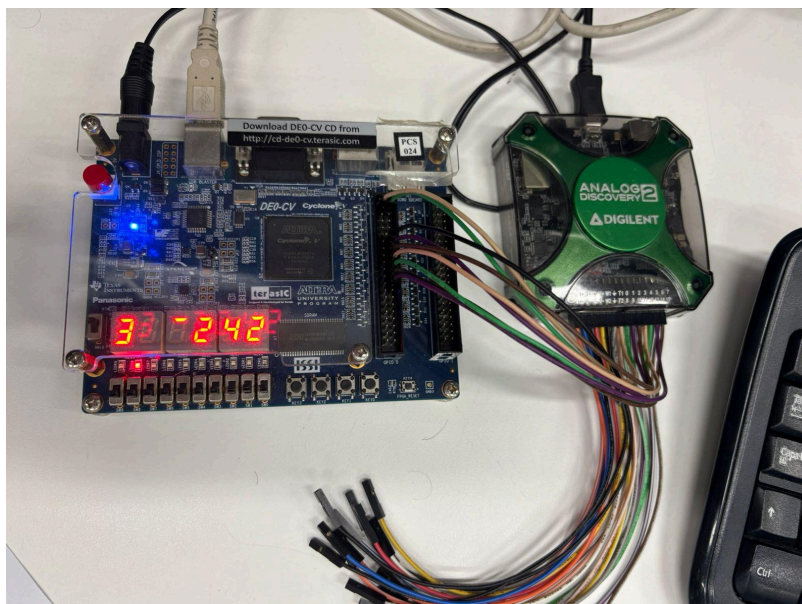


Figura 8: Operação 6, terceiro acerto

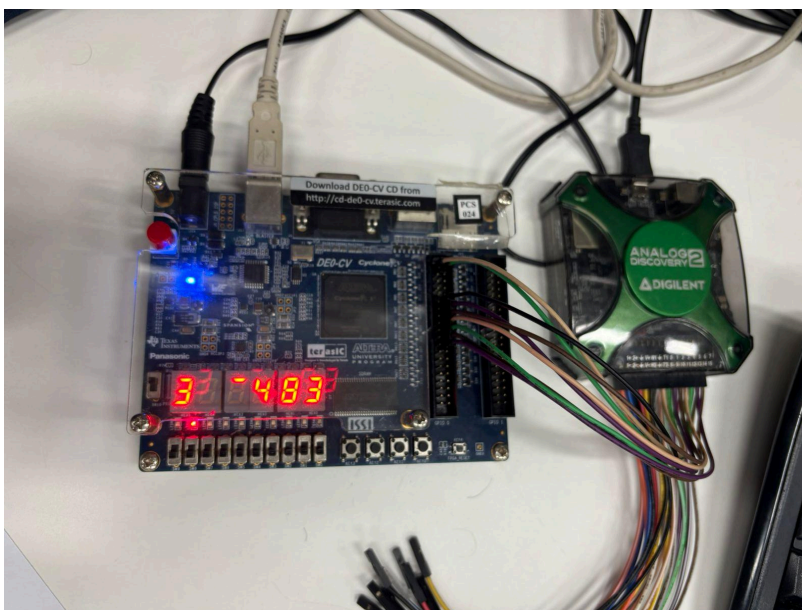


Figura 9: Operação 7, quarto acerto

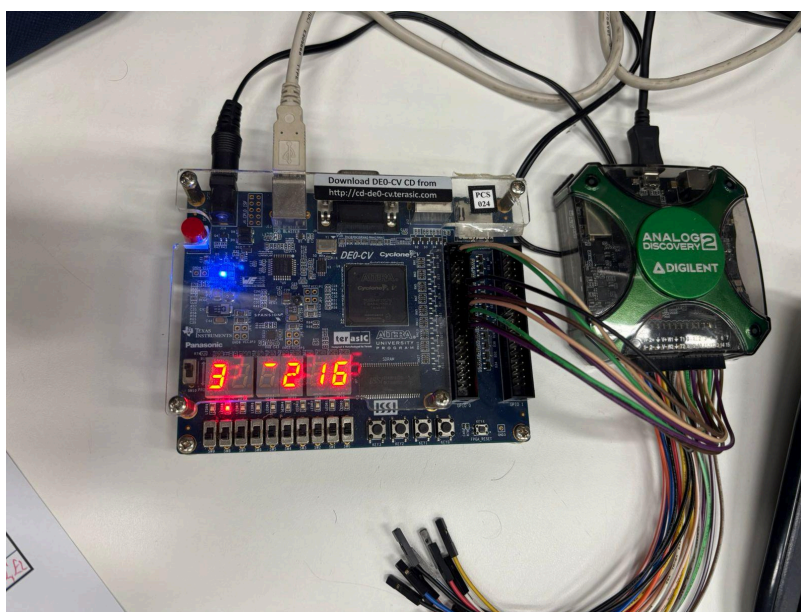


Figura 10: Operação 10, sétimo acerto

5.5 EXECUÇÃO PRÁTICA DO CENÁRIO DE TESTE 2 – ACERTO DAS 3 PRIMEIRAS JOGADAS E ERRO NA 4ª JOGADA

Replicamos os testes feitos digitalmente e os resultados foram os esperados

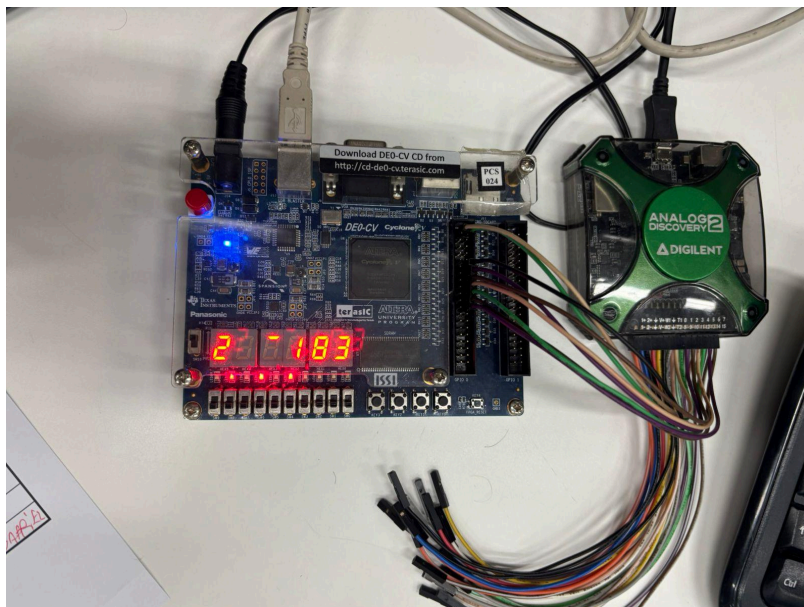


Figura 11: Operação 7, erro e fim

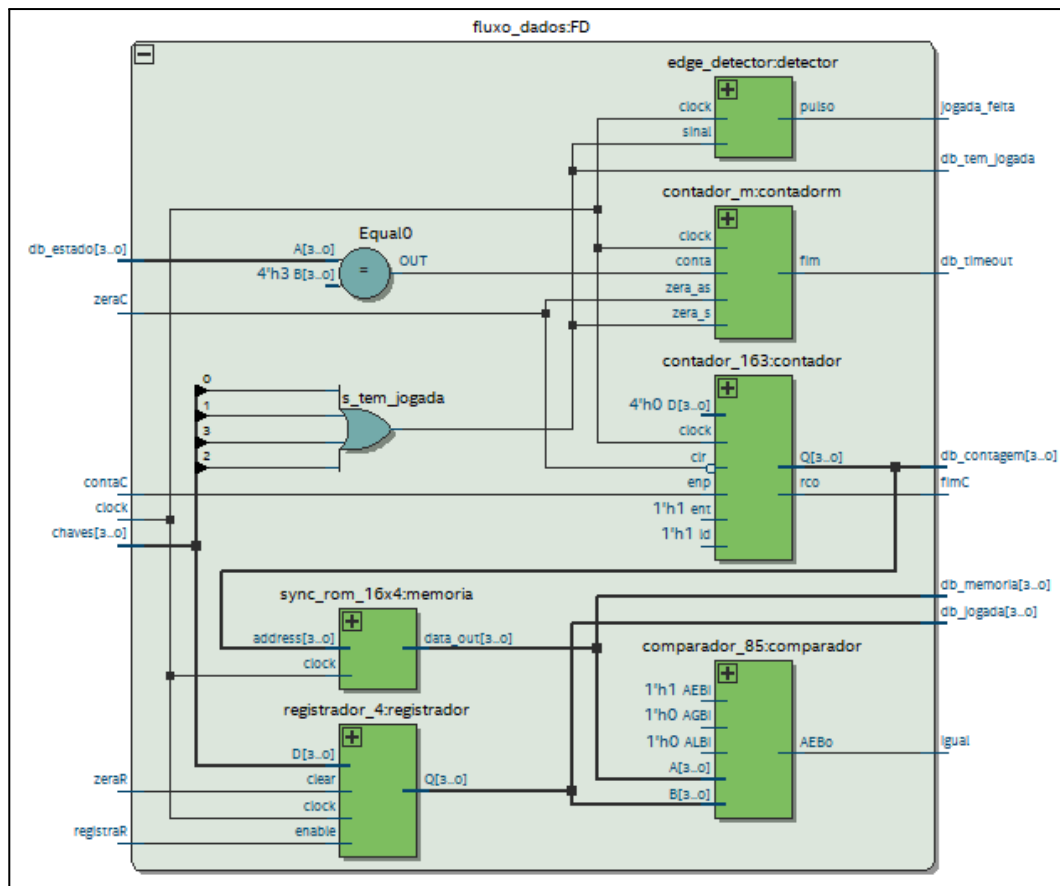
6 PROJETO DO DESAFIO DA EXPERIÊNCIA

6.1 DESCRIÇÃO DO DESAFIO

O desafio da experiência de hoje é colocar um tempo máximo em que o jogador deve realizar a jogada, esse tempo é de 3 segundos. Caso o jogador não consiga executar a jogada, o jogador toma “timeout”.

Para implantarmos isso adicionamos um estado adicional de “timeout”, o programa só vai para esse estado partindo do estado de espera. No estado de espera, com auxílio de um contador de 12 bits, contamos quantos ciclos de clock são necessários para se passar 3 segundos. Então, caso o jogador não realize a jogada, o estado muda para o de timeout e ativa os sinais *fim* e *timeout*.

6.2 DESCRIÇÃO DO PROJETO LÓGICO



6.3 VERIFICAÇÃO E VALIDAÇÃO DO DESAFIO

A estratégia de testes implementada foi a de testar o novo circuito com os testes antigos e criar um novo cenário de teste de timeout.

6.3.1 CENÁRIO DE TESTE 1 – ACERTO DAS 16 JOGADAS

Os testes bateram com o que foi previamente testado

6.3.2 CENÁRIO DE TESTE 2 – ACERTO DAS 3 PRIMEIRAS JOGADAS E ERRO NA 4ª JOGADA

Os testes bateram com o que foi previamente testado

6.3.2 CENÁRIO DE TESTE 3 – TIMEOUT NA SEGUNDA JOGADA

Tabela 5 – Descrição e Resultados Simulados do Cenário de Teste 3

#	Operação	Entradas	Saídas Esperadas	Resultado Simulado OK?
c.i.	Condições Iniciais	reset, iniciar e chaves zerados	errou, acertou, timeout, fim e LEDs zerados	sim
1	“Resetar” circuito	acionar reset	errou, acertou, timeout, fim e LEDs zerados	sim
2	Aguardar alguns segundos	reset, iniciar e chaves zerados	errou, acertou, timeout, fim e LEDs zerados	sim
3	Acionar sinal iniciar	acionar iniciar	errou, acertou, timeout, fim e LEDs zerados	sim
4	Acionar primeira entrada (jogada 1)	acionar chave(0)	LED(0) liga	sim
5	Aguardar mais de 3 segundos para timeout	reset, iniciar e chaves zerados	timeout e fim ativados	sim

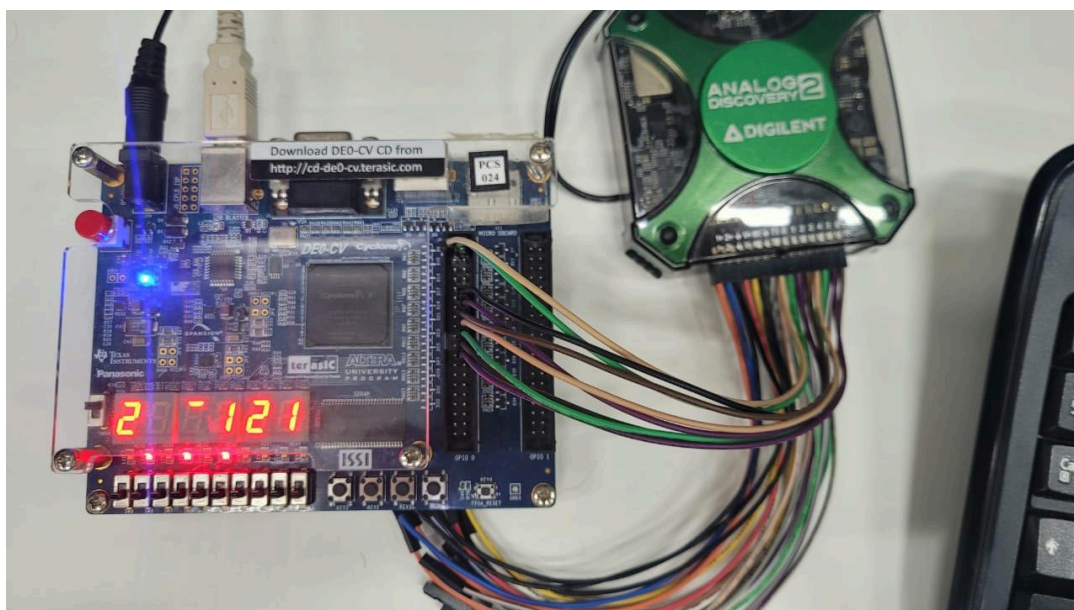


Figura 12: Operação 5, timeout

7 CONCLUSÕES

O desenvolvimento do projeto aconteceu como esperado e todas as saídas bateram com o esperado. Nosso projeto cumpre com o objetivo de criar uma interface de circuitos digitais com elementos externos de entrada de dados, conseguimos implementar as mudanças propostas pela descrição do projeto, sendo a mais importante a fase de espera do input do jogador.

Durante a aula conseguimos realizar todos os testes e todos bateram com o esperado. As figuras coladas nos capítulos de teste são justamente para documentar esse processo. Os objetivos da experiência foram cumpridos durante a aula, até mesmo realizando o desafio.

O desafio foi de longe a parte mais problemática do projeto, houve um problema de lógica no verilog do desafio que impedia o enable do contador de ser ativado, impedindo o timeout. Através de testes foi feita a identificação do problema e com auxílio dos monitores conseguimos mudar a lógica em verilog. Após isso conseguimos realizar todos os testes e assim validar o circuito do desafio.