

**UNIVERSIDADE DE SÃO PAULO**  
**ESCOLA POLITÉCNICA**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E**  
**SISTEMAS DIGITAIS**  
**PCS3635 - LABORATÓRIO DIGITAL I**



**RELATÓRIO DA EXPERIÊNCIA 6**

Felipe Luis Korbes - NUSP: 13682893

Henrique Eduardo dos Santos de Souza - NUSP: 13679972

João Felipe de Souza Melo - NUSP: 13682913

**Turma: 5**

**Professor: Reginaldo Arakaki**

**Data da experiência: 21/02/2024**

São Paulo

2024

## Sumário

<b>1. Introdução e Objetivos.....</b>	<b>1</b>
<b>2. Atividades pré-laboratório.....</b>	<b>1</b>
<b>2.1. Atividade 1: Projeto Lógico.....</b>	<b>1</b>
<b>2.1.1 Testes.....</b>	<b>4</b>
<b>2.2. Atividade 2 – Implementação e Síntese.....</b>	<b>15</b>
<b>3. Planejamento da aula prática.....</b>	<b>16</b>
<b>4. Relatório.....</b>	<b>17</b>
<b>4.1. Atividade 2: Testes e sinais de depuração.....</b>	<b>18</b>
<b>4.2. Atividade 3 – Desafio: Modificação do Sistema Digital.....</b>	<b>18</b>
<b>4.2.1. Testes e sinais de depuração.....</b>	<b>20</b>

## 1. Introdução e Objetivos

Esta experiência propõe a criação de uma versão inicial do circuito do Jogo do Desafio da Memória, inspirado nos brinquedos Genius (Estrela) e Simon (Hasbro). Seguindo as especificações da apostila e as dicas de projeto fornecidos, o desenvolvimento segue com o modelo do projeto anterior, passando por algumas mudanças para a adaptação dessa experiência. O circuito será testado e verificado com uma tabela de testes, cuja simulação será feita no ModelSim, e após isso será preparado para síntese com o Intel Quartus Prime.

## 2. Atividades pré-laboratório

### 2.1. Atividade 1: Projeto Lógico

Iniciou-se a montagem do circuito mediante a adaptação da unidade de controle, incorporando alguns estados adicionais. Para isso, utilizou-se como guia o diagrama de estados fornecido, o qual está representado na Figura 1.

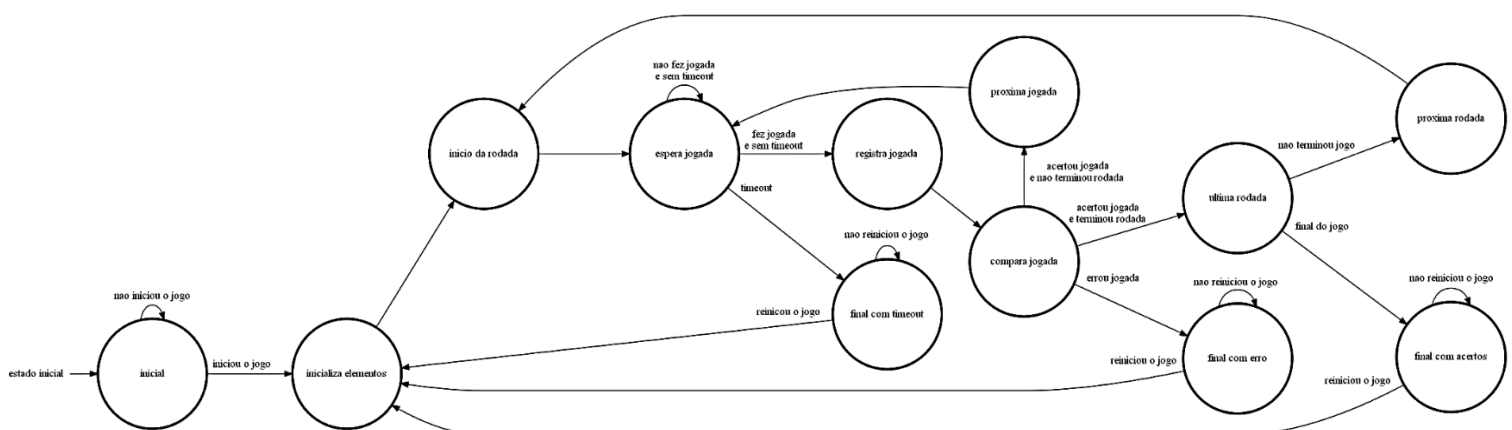


Figura 1 - Diagrama de estados de alto nível do diagrama da experiência

Após a conclusão dessa adaptação, procedeu-se à instanciação de um novo contador no fluxo de dados, e o *circuito\_jogo\_base* foi ajustado para acomodar essas alterações. Além disso, foi feita a modificação dos nomes das entradas dos módulos, bem como o nome dos arquivos e suas conexões internas, assegurando assim a consistência e alinhamento com as especificações do projeto.

A estrutura de fluxo de dados do projeto é composta por dois contadores binários com reset e contagens assíncronas, uma ROM que armazena as jogadas a serem realizadas, um registrador que registra a jogada feita pelo jogador, dois comparadores (um utilizado para comparar as jogadas e outro como controle para

as rodadas), um *edge\_detector* e uma porta OR que colaboram para o registro da jogada do jogador. Por fim, há um contador normal síncrono destinado a contabilizar a inatividade do jogador. O diagrama de blocos do fluxo de dados pode ser visualizado na Figura 2.

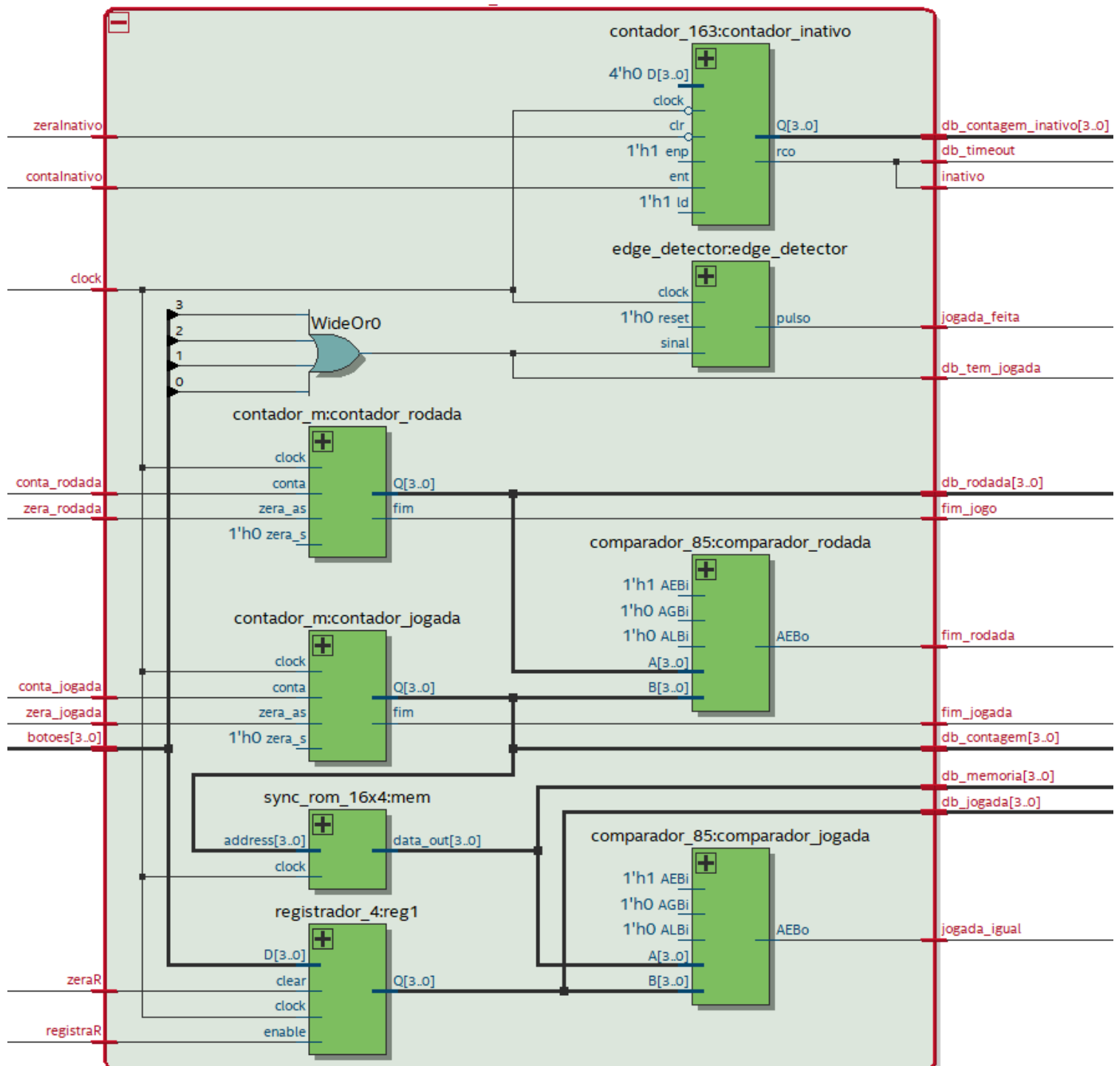
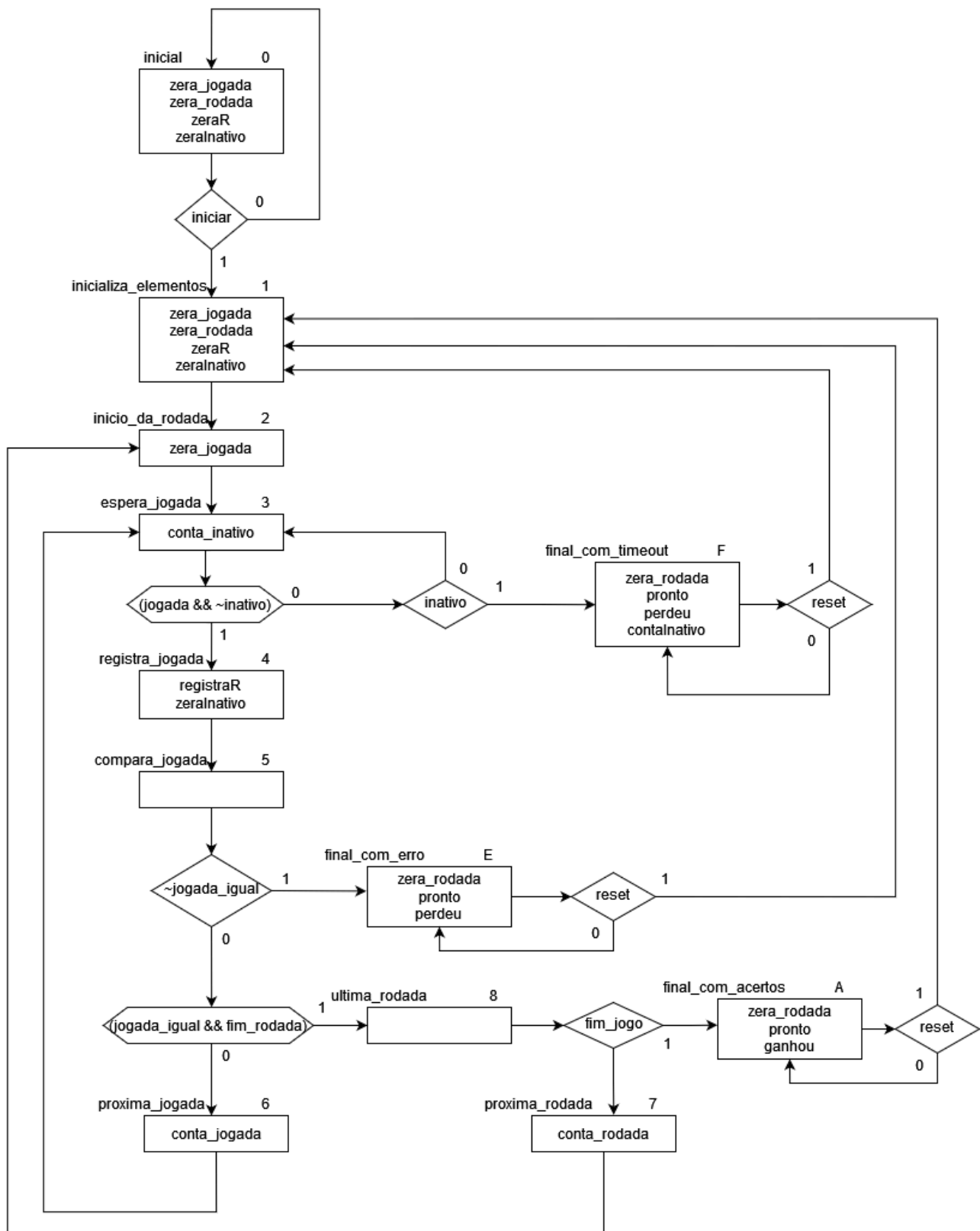


Figura 2 - Diagrama de blocos do fluxo de dados

Abaixo está o diagrama de transição de estados em um nível mais detalhado do fluxo de dados do circuito.



*Figura 3 - Diagrama de transição de estados*

### 2.1.1 Testes

Com o fluxo de dados concluído e o circuito implementado em Verilog, foram criados três casos de teste para avaliar o desempenho do circuito. Cada um desses casos de teste está detalhado nas tabelas a seguir.

Cenário 1 - Erro na rodada 4					
#	Operação	Sinais de controle	Resultado esperado	Resultado observado	Veredito
c.i.	Condições iniciais	reset=0 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	
1	Resetar o circuito	reset=1 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	
2	Iniciar	reset=0 iniciar=1	ganhou=0 perdeu=0 pronto=0 db_iniciar=0-1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	
3	Joga rodada 1	reset=0 iniciar=0 botões=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=1 db_tem_jogada=0	

			timeout=0 db_jogadafeita=1 db_rodada=1 db_contagem=0 db_memoria=1 db_estado=4-5-8-7-2-3	timeout=0 db_jogadafeita=1 db_rodada=1 db_contagem=0 db_memoria=1 db_estado=4-5-8-7-2-3	
4	Continua jogando até rodada 4	reset=0 iniciar=0 botões=0-1 0-1-2 0-1-2-3 3 (jogada errada)	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4-8 db_rodada=2-3-4 db_contagem=1-2-3 db_memoria=1-2-4-8 db_estado=4-5-6-3 e 4-5-8-7-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4-8 db_rodada=2-3-4 db_contagem=1-2-3 db_memoria=1-2-3-4 db_estado=4-5-6-3 e 4-5-8-7-2-3	
5	Erra jogada	reset=0 iniciar=0 chaves=3	ganhou=0 perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=8 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=4-5-E	ganhou=0 perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=8 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=4-5-E	

Tabela 1 - Teste do circuito para erro na rodada 4

Cenário 2 - Acertar todas as rodadas					
#	Operação	Sinais de controle	Resultado esperado	Resultado observado	Veredito
c.i.	Condições iniciais	reset=0 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	
1	Resetar o circuito	reset=1 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0	

			db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	
2	Iniciar	reset=0 iniciar=1	ganhou=0 perdeu=0 pronto=0 db_iniciar=0-1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	
3	Jogar até a rodada E	reset=0 iniciar=0 botões= 0-1 0-1-2 0-1-2-3 0-1-2-3-2 0-1-2-3-2-1 0-1-2-3-2-1-0 0-1-2-3-2-1-0-0 0-1-2-3-2-1-0-0-1 0-1-2-3-2-1-0-0-1-1 0-1-2-3-2-1-0-0-1-1-2 0-1-2-3-2-1-0-0-1-1-2-2 0-1-2-3-2-1-0-0-1-1-2-2-3 0-1-2-3-2-1-0-0-1-1-2-2-3-3 0-1-2-3-2-1-0-0-1-1-2-2-3-3-0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0-1 timeout=0 db_jogadafeita=1-2-4-8 db_rodada=1-2-3-4-5-6-7-8-9-A-B-C-D-E db_contagem=1-2-3-4-5-6-7-8-9-A-B-C-D-E db_memoria=1-2-4-8 db_estado=4-5-6-3 e 4-5-8-7-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0-1 timeout=0 db_jogadafeita=1-2-4-8 db_rodada=1-2-3-4-5-6-7-8-9-A-B-C-D-E db_contagem=1-2-3-4-5-6-7-8-9-A-B-C-D-E db_memoria=1-2-4-8 db_estado=4-5-6-3 e 4-5-8-7-2-3	
4	Jogar última rodada	reset=0 iniciar=0 botões= 0-1-2-3-2-1-0-0-1-1-2-2-3-3-1-2	ganhou=1 perdeu=0 pronto=1 db_iniciar=0 db_igual=0-1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4-8 db_rodada=E-F db_contagem=1-2-3-4-5-6-7-8-9-A-B-C-D-E-F db_memoria=1-2-4-8 db_estado=4-5-6-3 e 4-5-8-A	ganhou=1 perdeu=0 pronto=1 db_iniciar=0 db_igual=0-1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4-8 db_rodada=E-F db_contagem=1-2-3-4-5-6-7-8-9-A-B-C-D-E-F db_memoria=1-2-4-8 db_estado=4-5-6-3 e 4-5-8-A	

Tabela 2 - Teste do circuito de acertar todas as rodadas



Cenário 3 - Inatividade na rodada 3					
#	Operação	Sinais de controle	Resultado esperado	Resultado observado	Veredito
c.i.	Condições iniciais	reset=0 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	
1	Resetar o circuito	reset=1 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	
2	Iniciar	reset=0 iniciar=1	ganhou=0 perdeu=0 pronto=0 db_iniciar=0-1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	
3	Joga até a rodada 3	reset=0 iniciar=0 botões=0 0-1 0-1-2	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4 db_rodada=1-2-3 db_contagem=1-2 db_memoria=1-2-4 db_estado=4-5-6-3 e 4-5-8-7-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=1 db_tem_jogada=0 timeout=0 db_jogadafeita=1 db_rodada=1 db_contagem=0 db_memoria=1 db_estado=4-5-8-7-2-3	
4	Inatividade	reset=0	ganhou=0	ganhou=0	

		iniciar=0	perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=1-0 db_jogadafeita=4 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=F	perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=1-0 db_jogadafeita=4 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=F	
--	--	-----------	---	---	--

Tabela 3 - Teste do circuito para inatividade na rodada 3

Abaixo, na figura 4, o resultado da simulação do cenário 1, para o qual há erro na rodada 4.

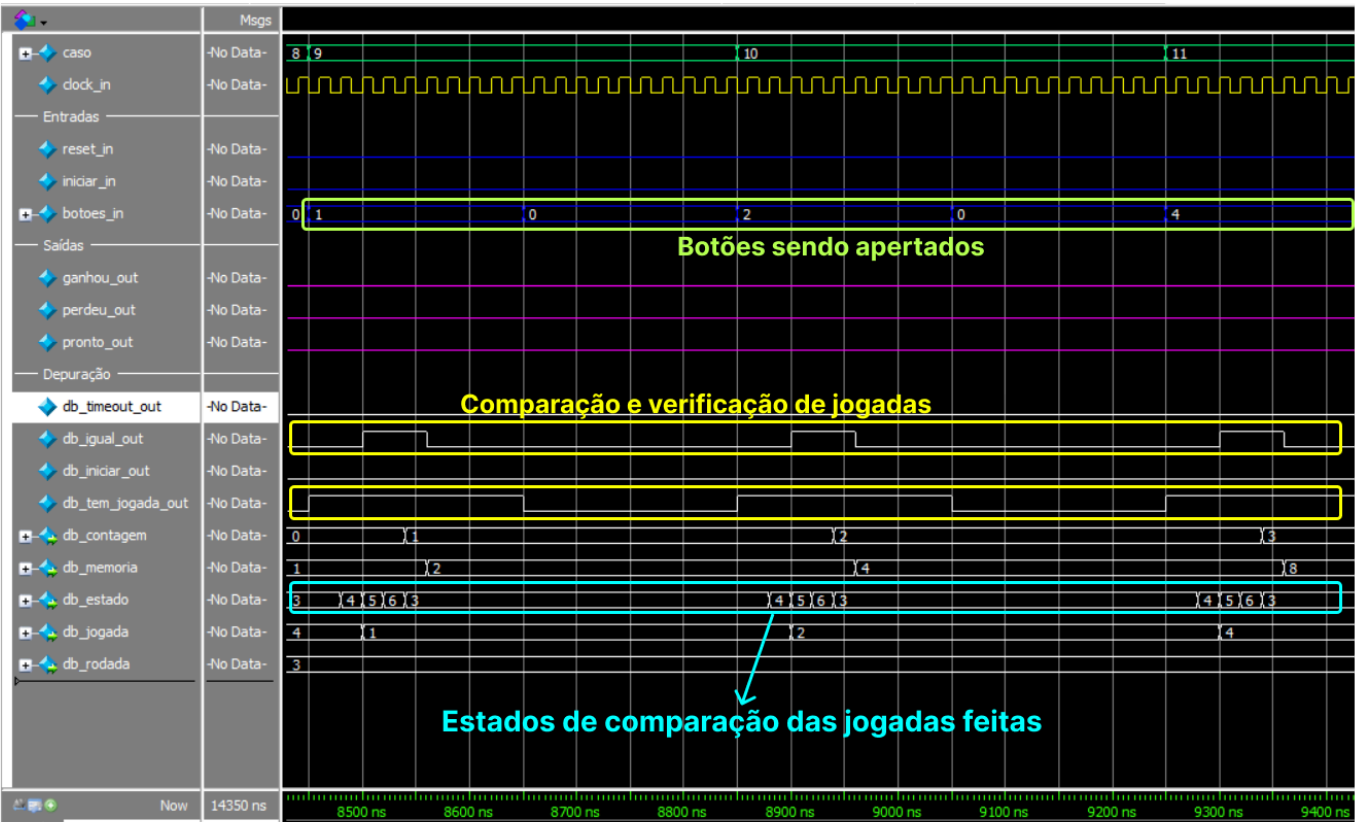


Figura 4 - Formato de onda das jogadas do cenário 1

Nessa figura, percebe-se as jogadas do jogador nos instantes de ativação de “botoes\_in”. Para as três primeiras jogadas da rodada, percebe-se que o estado varia na sequência 4, 5, 6 e 3, que corresponde às transições para o caso de sucesso, ou seja, jogadas certas.

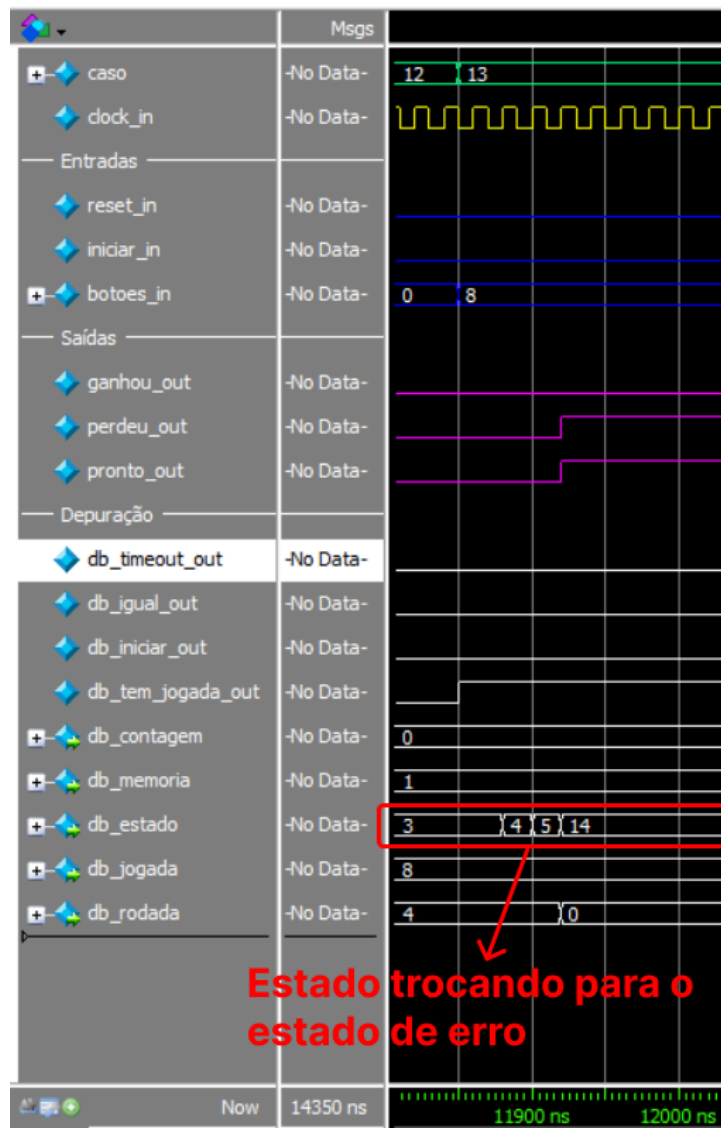


Figura 5 - Formato de onda das jogadas do caso 1 com erro

Para a quarta e última jogada, representada na figura 5, a transição observada foi 4, 5, 14, que corresponde ao final com erro, em conformidade com o esperado pelo caso de teste.

Além dos testes acima, foram também testadas e verificadas as formas de ondas do caso 2 e caso 3, como podem ser vistos nas figuras abaixo.

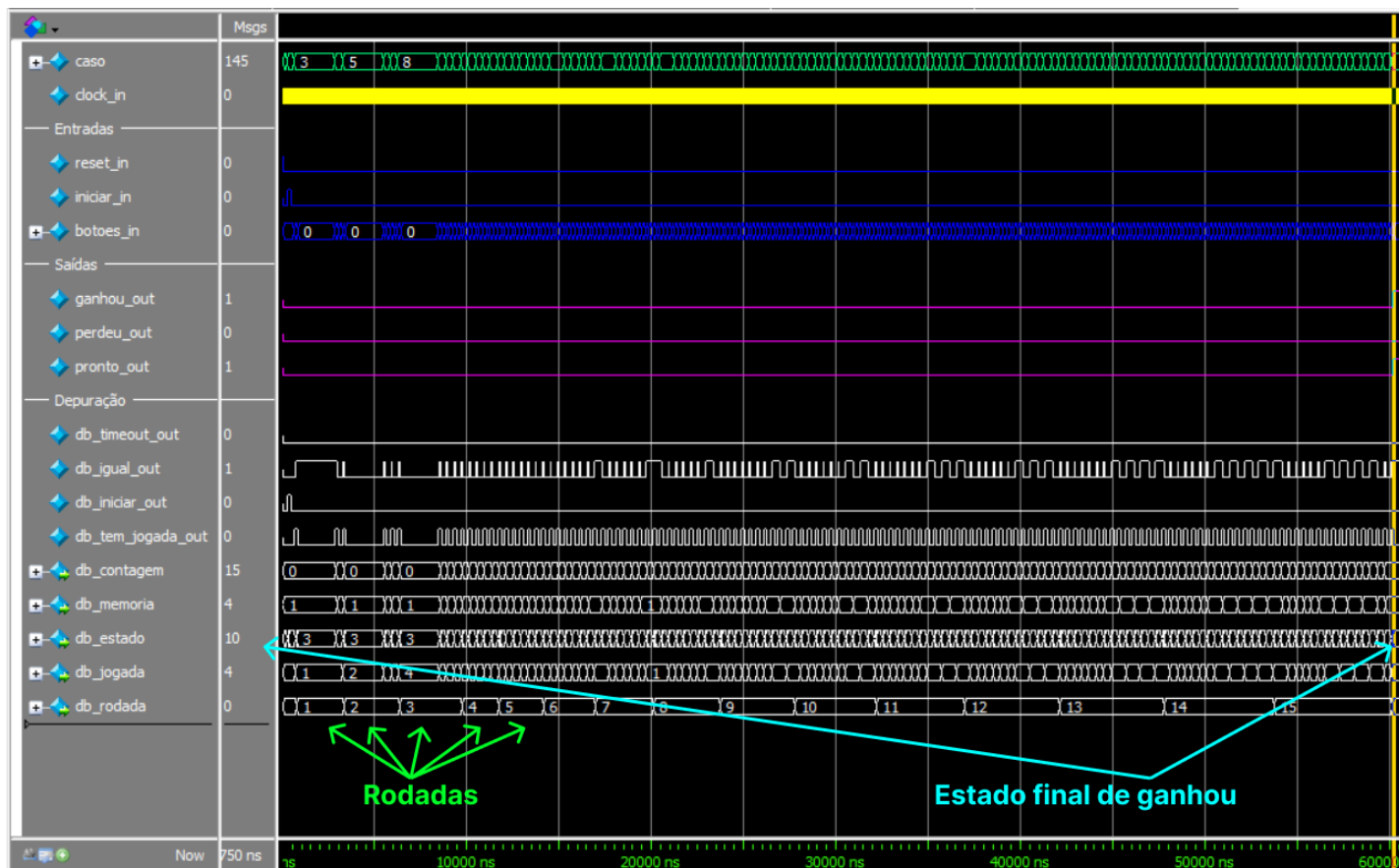


Figura 6 - Formato de onda das jogadas do caso 2 com “ganhou”

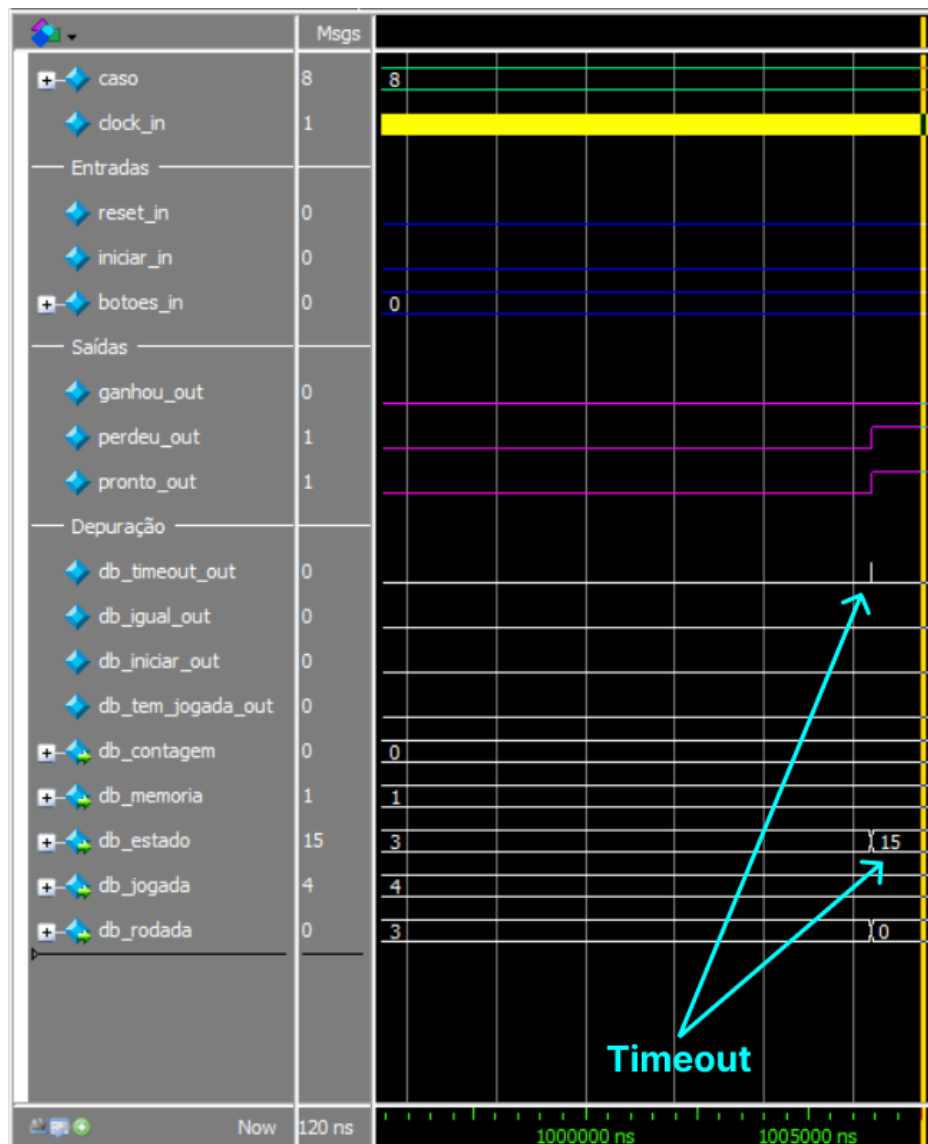


Figura 7 - Formato de onda das jogadas do caso 3 com "timeout"

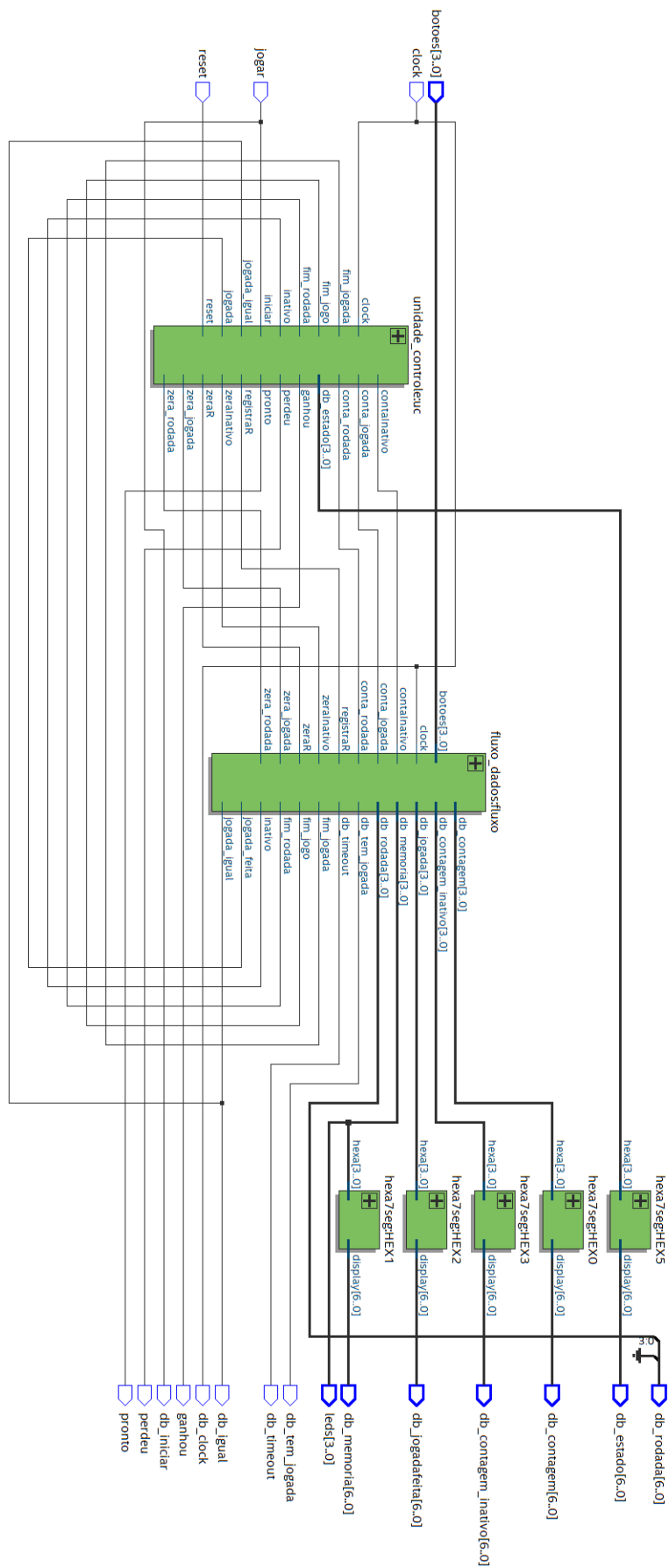


Figura 8 - Diagrama de blocos do circuito completo

As figuras 8 e 9 contém, respectivamente, o diagrama de blocos do circuito completo e a máquina de estados finita descrita pelo código em verilog, gerados pela ferramenta TRL Viewer do Quartus. Em relação ao diagrama de blocos, pode-se verificar que ele está consistente com a arquitetura proposta. A máquina de estados, por sua vez, também está de acordo com a que foi projetada. Portanto, ao nível de arquitetura, o circuito programado em verilog corresponde ao circuito que se deseja implementar.

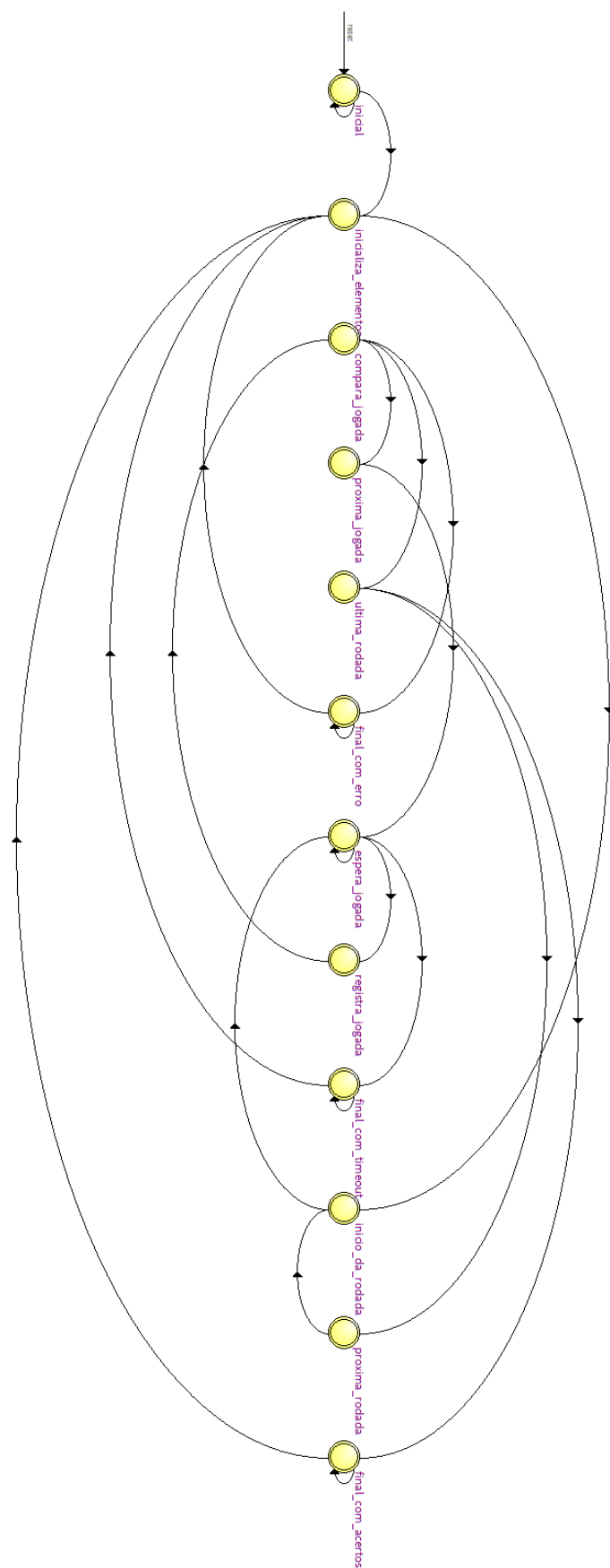


Figura 9 - Diagrama de transição de estados gerado pelo Quartus



## 2.2. Atividade 2 – Implementação e Síntese

Para essa experiência, a designação de pinos proposta encontra-se na tabela 4. Utilizou-se os leds e os displays para os sinais de depuração e os sinais da Analog Discovery para entradas do circuito e visualização dos principais sinais de controle.

Sinal	Pino na Placa DE0-CV	Pino na FPGA	Analog Discovery
CLOCK	GPIO_0_D0	PIN_N16	StaticIO – LED – DIO0 e Patterns – Clock – 1 kHz
RESET	GPIO_0_D1	PIN_B16	StaticIO – Button 0/1 – DIO1
JOGAR	GPIO_0_D3	PIN_C16	StaticIO – Button 0/1 – DIO2
BOTOES(0)	GPIO_0_D11	PIN_R22	StaticIO – Button 0/1 – DIO4
BOTOES(1)	GPIO_0_D13	PIN_T22	StaticIO – Button 0/1 – DIO5
BOTOES(2)	GPIO_0_D15	PIN_N19	StaticIO – Button 0/1 – DIO6
BOTOES(3)	GPIO_0_D17	PIN_P19	StaticIO – Button 0/1 – DIO7
LEDS(0)	GPIO_1_D17	PIN_A15	StaticIO - LED - DIO12
LEDS(1)	GPIO_1_D19	PIN_L8	StaticIO - LED - DIO13
LEDS(2)	GPIO_1_D21	PIN_B15	StaticIO - LED - DIO14
LEDS(3)	GPIO_1_D23	PIN_E14	StaticIO - LED - DIO15
PRONTO	GPIO_1_D15	PIN_J11	StaticIO - LED - DIO10
GANHOU	GPIO_1_D13	PIN_G11	StaticIO - LED - DIO9
PERDEU	GPIO_1_D11	PIN_J18	StaticIO - LED - DIO8
DB_IGUAL	Led LEDR7	PIN_U1	-
DB_CLOCK	Led LEDR0	PIN_AA2	-
DB_INICIAR	Led LEDR1	PIN_AA1	-
DB_TEM_JOGADA	Led LEDR2	PIN_W2	-

TIMEOUT	Led LEDR3	PIN_Y3	-
DB_CONTAGEM	Display HEX0	DB_CONTAGEM[0]=PIN_U21 DB_CONTAGEM[1]=PIN_V21 DB_CONTAGEM[2]=PIN_W22 DB_CONTAGEM[3]=PIN_W21 DB_CONTAGEM[4]=PIN_Y22 DB_CONTAGEM[5]=PIN_Y21 DB_CONTAGEM[6]=PIN_AA22	-
DB_MEMORIA	Display HEX1	DB_MEMORIA[0]=PIN_AA20 DB_MEMORIA[1]=PIN_AB20 DB_MEMORIA[2]=PIN_AA19 DB_MEMORIA[3]=PIN_AA18 DB_MEMORIA[4]=PIN_AB18 DB_MEMORIA[5]=PIN_AA17 DB_MEMORIA[6]=PIN_U22	-
DB_JOGADAFEITE	Display HEX2	DB_JOGADAFEITA[0]=PIN_Y19 DB_JOGADAFEITA[1]=PIN_AB17 DB_JOGADAFEITA[2]=PIN_AA10 DB_JOGADAFEITA[3]=PIN_Y14 DB_JOGADAFEITA[4]=PIN_V14 DB_JOGADAFEITA[5]=PIN_AB22 DB_JOGADAFEITA[6]=PIN_AB21	-
DB_ESTADO	Display HEX5	DB_ESTADO[0]=PIN_N9 DB_ESTADO[1]=PIN_M8 DB_ESTADO[2]=PIN_T14 DB_ESTADO[3]=PIN_P14 DB_ESTADO[4]=PIN_C1 DB_ESTADO[5]=PIN_C2 DB_ESTADO[6]=PIN_W19	-
DB_RODADA	Display HEX3	DB_RODADA[0]=PIN_Y16 DB_RODADA[1]=PIN_W16 DB_RODADA[2]=PIN_Y17 DB_RODADA[3]=PIN_V16 DB_RODADA[4]=PIN_U17 DB_RODADA[5]=PIN_V18 DB_RODADA[6]=PIN_V19	-

*Tabela 4 - Designação de Pinos para a Atividade 2*

### 3. Planejamento da aula prática

Iniciará-se a aula prática baixando os arquivos do projeto, armazenados em repositório remoto do github, para o diretório local. Em seguida, abrirá-se o projeto do circuito no Intel Quartus Prime e compilará-se ele, a fim de verificar se não há problemas da versão mais atual do código, seja em função de sintaxe seja em função de qualquer outra incompatibilidade. Ao mesmo tempo, um dos integrantes irá conectar os pinos da Analog Discovery à placa FPGA, de acordo com o Pin Planner. Para facilitar a leitura e a interpretação dos dados dos displays e dos leds,

o grupo trará uma folha de papel com uma tabela que mapeie cada sinal importante e seu significado, como “db\_contagem” e “display HEX0”.

Com essas etapas concluídas, realizará-se a síntese do circuito no Quartus e, com o sucesso dessa operação, programará-se a placa. Finalmente, iniciará-se então a bateria de testes. Eles serão realizados em sequência, e suas entradas serão dadas de acordo com as descritas pelas tabelas de teste 1, 2 e 3. Em cada teste, deve-se seguir o procedimento padrão de ativar os sinais de entrada em ordem e observar os sinais de depuração, nos displays e no led. Havendo compatibilidade de todos os sinais, marcará-se um verificado na etapa do teste correspondente.

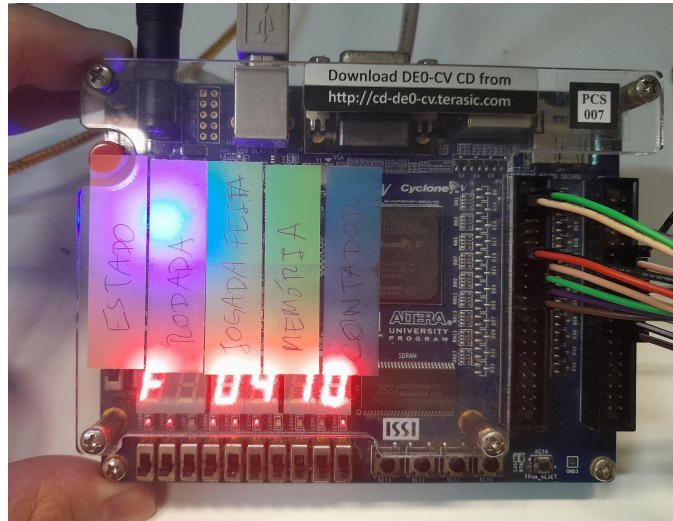
Caso haja divergência, repetirá-se o teste, desde a primeira etapa, para eliminar possíveis erros de operação, como demora excessiva para jogar, que poderia provocar timeout. Com a persistência do erro, o grupo deve discutir as hipóteses do que pode estar errado e revisar a lógica do circuito. A revisão deve iniciar-se pelo módulo defeituoso, depois para os módulos que entram diretamente em contato com ele e, em seguida, deve-se revisar o diagrama de blocos e a lógica da máquina de estados. nessa abordagem do mais específico para o mais geral, procura-se o erro em regiões cada vez mais abrangentes, que começam das proximidades do ponto de falha do circuito, estratégia que o grupo julga adequada.

Finalizados os testes, o grupo procederá então para o desafio. Ele começará pela discussão da nova lógica, em termos de que condições, estados e transições devem ser adicionados à máquina de estados. Em seguida, programará-se esses novos elementos nos arquivos verilog do circuito. Finalmente, compila-se e programa-se a placa FPGA com o novo circuito e inicia-se a onda de testes. Testa-se primeiro a retrocompatibilidade com a lógica do circuito anterior onde for cabível, para verificar que a adição traz novas funcionalidades mas não perturba as demais, já feitas. Assim, procede-se então para o teste da nova funcionalidade, proposta pelo desafio. Sendo aprovado, dá-se por encerrada a experiência e completa-se o relatório.

#### **4. Relatório**

Iniciou-se a experiência fazendo download do repositório remotos, do github, para o diretório local. Abriu-se então o projeto do Intel Quartus Prime e verificou-se a integridade e inclusão de todos os arquivos. Em seguida, o grupo passou a

conectar os fios da Analog Discovery na placa FPGA DE0-CV. Seguiu-se a ordem e a associação de fios designados pela tabela 4. Colou-se também post-its na placa de acrílico da placa FPGA, a fim de facilitar a visualização e interpretação dos sinais. O circuito montado pode ser visto na figura 10.



*figura 10 - Circuito montado*

Então, sintetizou-se e programou-se o circuito na placa FPGA. Com o software Waveforms aberto, prosseguiu-se então para os testes de cada um dos cenários, tais quais descritos pelas tabelas 1, 2 e 3.

#### **4.1. Atividade 2: Testes e sinais de depuração**

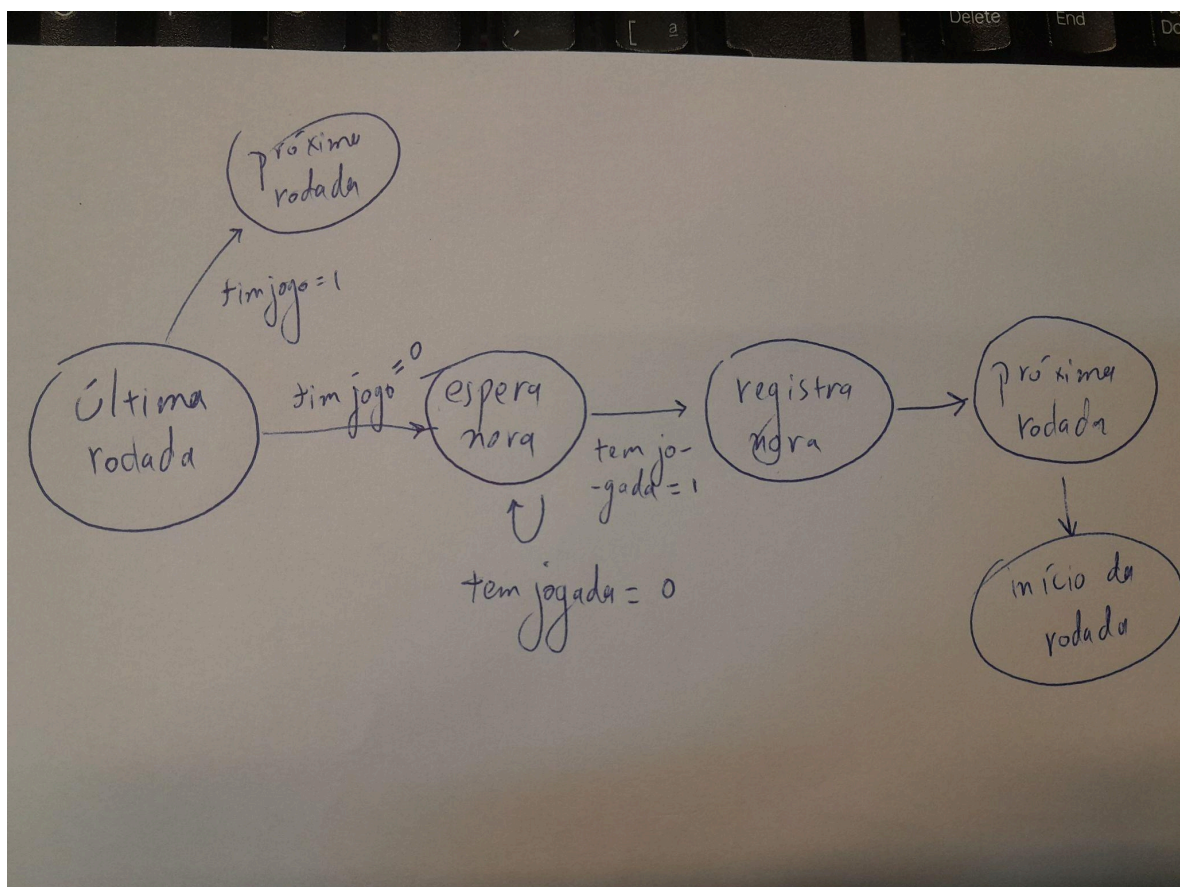
Detectou-se um problema, relacionado ao display HEX3. Após revisar a pinagem pelo Pin Planner do Quartus e a lógica do circuito, não foi encontrado nenhum defeito. O grupo passou a revisar os fios conectados e encontrou-se um mau contato com os fios. Com essa correção, o circuito comportou-se como esperado.

Com isso, deu-se continuidade à etapa de testes, antes com um clock lento, de 2Hz, para observar a transição de estados, depois com um clock rápido de 1kHz, para o teste. Para o clock lento, verificou-se corretude dos sinais de depuração, sobretudo em relação às transições de estado. Para o clock mais rápido, também verificou-se a corretude dos sinais para todos os cenários. Dessa forma, o circuito foi aprovado em todos os testes da atividade 2.

#### **4.2. Atividade 3 – Desafio: Modificação do Sistema Digital**

O desafio proposto pelo professor consistiu em implementar as rodadas com jogadas incrementais, definidas em tempo de execução pelo usuário, de forma que

a memória deixa de ser uma do tipo ROM para ser do tipo RAM, com valores armazenados variando. Para comportar a nova funcionalidade, o grupo discutiu que modificações deveriam ser realizadas no circuito, a começar pela lógica de estados. Chegou-se à conclusão de que seria necessária a adição de mais dois novos estados, denominados *espera\_nova* e *registra\_nova*, que permitem a espera por uma nova entrada do jogador, a ser salva na memória, e o registro do novo valor inserido pelo usuário, tal qual indicado na figura 10.



*Figura 11 - secção modificada da máquina de estados do circuito*

Para adequar as modificações propostas, o grupo acrescentou à unidade de controle os novos estados, a lógica de saída e criou entradas e saídas adicionais nos módulos vizinhos, para permitir a conexão desses novos sinais. Em seguida, sintetizou-se o circuito e programou-se ele na placa FPGA. De início, ele não funcionou como esperado, porque os estados travaram na condição de

registra\_nova e o contador da memória continuava contando sem critério. O grupo discutiu possíveis fontes de erro para o problema e passou a revisar sistematicamente os módulos correspondentes, seus vizinhos e a instanciação de todos os módulos no arquivo principal.

De início, percebeu-se a falta de uma lógica de saída para o estado registra\_nova, e a necessidade de adaptação para a de outros dois estados, relacionados à escrita na memória. Feita essa correção, o circuito continuava a dar sinais de depuração incoerentes. Nesse momento, após mais uma revisão adicional dos estados e dos sinais defeituosos, o grupo passou a suspeitar de mal contato nos fios. Então, após reforçar as conexões, pressionando melhor os fios na placa, o circuito não apresentou mais o problema de travar no estado de registra\_nova.

O outro problema detectado foi que a memória não era adequadamente escrita. Os valores fornecidos por ela, tal qual pode ser visto no sinal de depuração db\_memoria, do display HEX1, estava inconsistente com as novas jogadas fornecidas pelo usuário. Então, a RAM não estava registrando o botão que o usuário apertava. Ao revisar-se a lógica do circuito, percebeu-se uma falha lógica no sinal de load da RAM. Então, atribuiu-se um novo fio write\_enable para a RAM e uma lógica correta. Com a nova modificação, o circuito parou de funcionar e exibia sinais de depuração inconsistentes. Após mais revisões, voltou-se a suspeitar de mal contato dos fios. Com outro reforço das conexões, o circuito voltou a funcionar.

Finalmente, o circuito funcionou conforme esperado. A partir daí, o grupo testou-o sistematicamente contra os cenários de teste de falha na quarta rodada, sucesso e timeout, para clock lento de 2Hz, depois rápido de 10kHz.

#### **4.2.1. Testes e sinais de depuração**

O grupo propôs dois cenários de teste, um primeiro cenário erro na primeira jogada da quarta rodada, e um segundo cenário timeout na primeira jogada da terceira rodada. Houve também um teste, não descrito detalhadamente aqui, para acerto em todas as rodadas e jogadas, que deveria acabar em final com acerto.

A rotina detalhada de testes, para os cenários 1 e 2, estão descritos nas tabelas 5 e 6. As respectivas formas de onda estão nas figuras 12 e 13.

Cenário Desafio 1 - Erro na quarta rodada					
#	Operação	Sinais de controle	Resultado esperado	Resultado observado	Veredito
c.i.	Condições iniciais	reset=0 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	✓
1	Resetar o circuito	reset=1 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	✓
2	Iniciar	reset=0 iniciar=1	ganhou=0 perdeu=0 pronto=0 db_iniciar=0-1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=1 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	✓
3	Joga até a rodada 3	reset=0 iniciar=0 botões=0-1 0-1-2 0-1-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4 db_rodada=1-2-3 db_contagem=1-2 db_memoria=1-2-4 db_estado=4-5-6-3 e 4-5-8-7-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=1 db_tem_jogada=0 timeout=0 db_jogadafeita=1 db_rodada=1 db_contagem=0 db_memoria=1 db_estado=4-5-8-7-2-3	✓
4	rodada 4	reset=0	ganhou=0	ganhou=0	✓

		iniciar=0 botões=2	perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4 db_rodada=4 db_contagem=1-2 db_memoria=8 db_estado=4-5-6-3 e 4-5-E	perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4 db_rodada=4 db_contagem=1-2 db_memoria=8 db_estado=4-5-6-3 e 4-5-E	
--	--	-----------------------	---	---	--

Tabela 5 - rotina do cenário de teste 1 do desafio

Abaixo está a forma de onda para o teste acima:

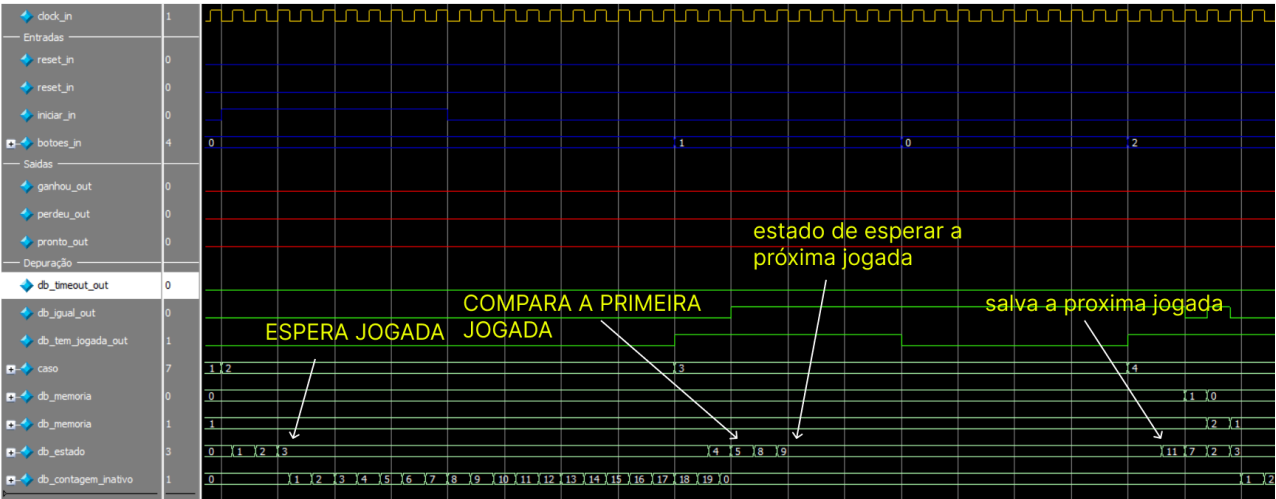


Figura 13 - Formas de onda para cenário de teste 1 do desafio

Continuação:



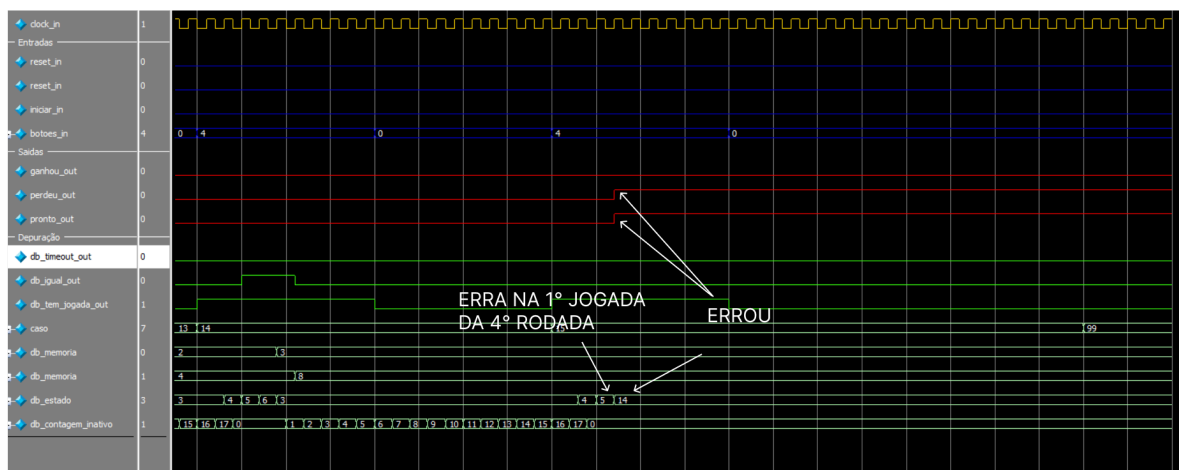


Figura 14 - Continuação das formas de onda para cenário de teste 1 do desafio

Cenário 2 - timeout na quarta rodada					
#	Operação	Sinais de controle	Resultado esperado	Resultado observado	Veredito
c.i.	Condições iniciais	reset=0 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	✓
1	Resetar circuito	reset=1 iniciar=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=0	✓
2	Iniciar	reset=0 iniciar=1	ganhou=0 perdeu=0 pronto=0 db_iniciar=0-1 db_igual=0	ganhou=0 perdeu=0 pronto=0 db_iniciar=1 db_igual=0	✓

			db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	db_tem_jogada=0 timeout=0 db_jogadafeita=0 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=1-2-3	
3	Joga até a rodada 3	reset=0 iniciar=0 botões=0-1 0-1-2 0-1-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=0-1 db_tem_jogada=0 timeout=0 db_jogadafeita=1-2-4 db_rodada=1-2-3 db_contagem=1-2 db_memoria=1-2-4 db_estado=4-5-6-3 e 4-5-8-7-2-3	ganhou=0 perdeu=0 pronto=0 db_iniciar=0 db_igual=1 db_tem_jogada=0 timeout=0 db_jogadafeita=1 db_rodada=1 db_contagem=0 db_memoria=1 db_estado=4-5-8-7-2-3	✓
4	Inatividade	reset=0 iniciar=0	ganhou=0 perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=1-0 db_jogadafeita=4 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=F	ganhou=0 perdeu=1 pronto=1 db_iniciar=0 db_igual=0 db_tem_jogada=0 timeout=1-0 db_jogadafeita=4 db_rodada=0 db_contagem=0 db_memoria=1 db_estado=F	✓

*Tabela 6 - rotina do cenário de teste 2 do desafio*

*Abaixo estão as formas de ondas:*

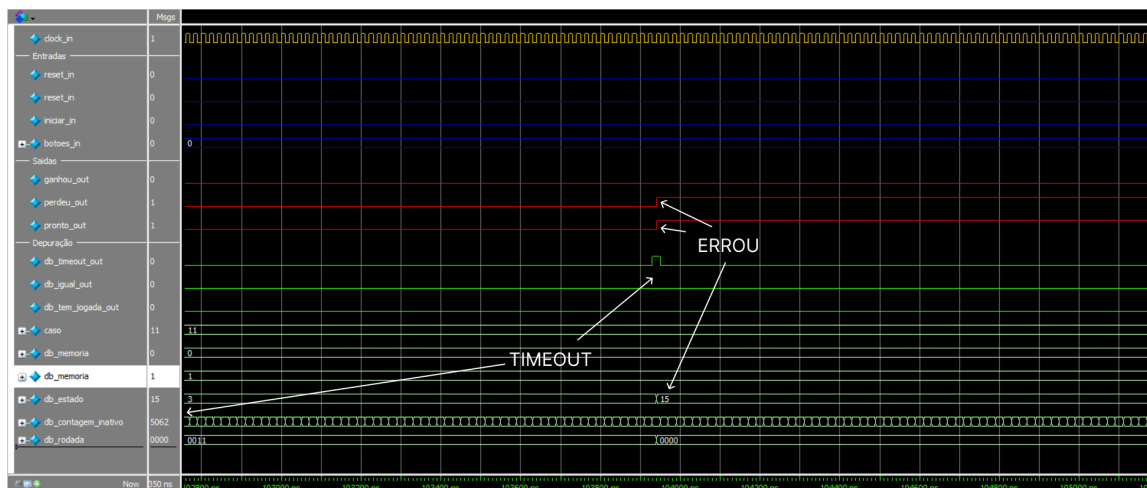


Figura 16 - formas de onda para cenário de teste 2 do desafio

Para os testes em clock lento, o circuito foi aprovado em todos os testes, para os dois primeiros cenários, e constatou-se a corretude de todas as transições de estado e sinais de depuração. Para o clock rápido, também. Ao final, o circuito passou em todos os casos de teste de todos os dois cenários propostos e foi aprovado.