| | logo  ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO | | --- | | Departamento de Engenharia de Computação e Sistemas Digitais | |  |   **PCS3635 – LABORATÓRIO DIGITAL I** | |
| --- | --- | --- | --- | --- |
| EXPERIÊNCIA 3 – Projeto de uma Unidade de Controle  Planejamento da Bancada A3 – Turma 2 – Prof. Reginaldo  Data de Emissão: 22 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.

1. **Descrição do Projeto**

A visão conceitual do projeto descreve o desenvolvimento de um sistema digital baseado em Verilog que funciona como um jogo de advinhação, 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.”*

1. **Detalhamento do Projeto Lógico**
   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 *sinal*. 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 *~sinal*, uma vez que o detector pode resetar quando a jogada deixar de ser feita.

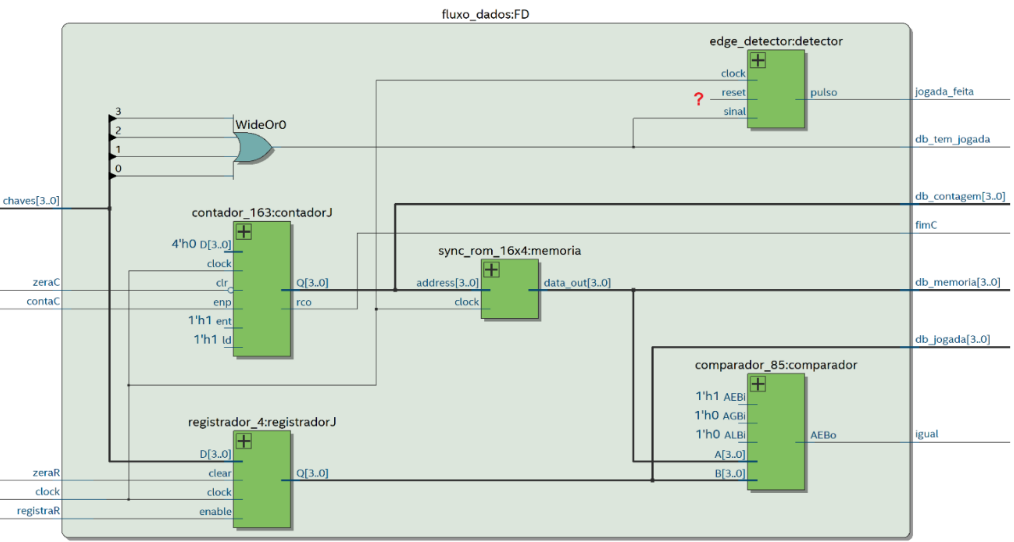


Figura 1: Estrutura interna do fluxo de dados

* 1. **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:

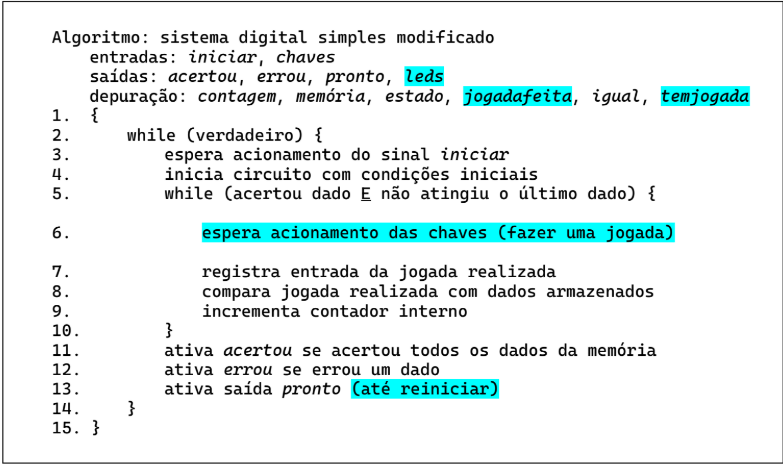


Figura 2: Pseudocódigo da lógica de execução do circuito

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.

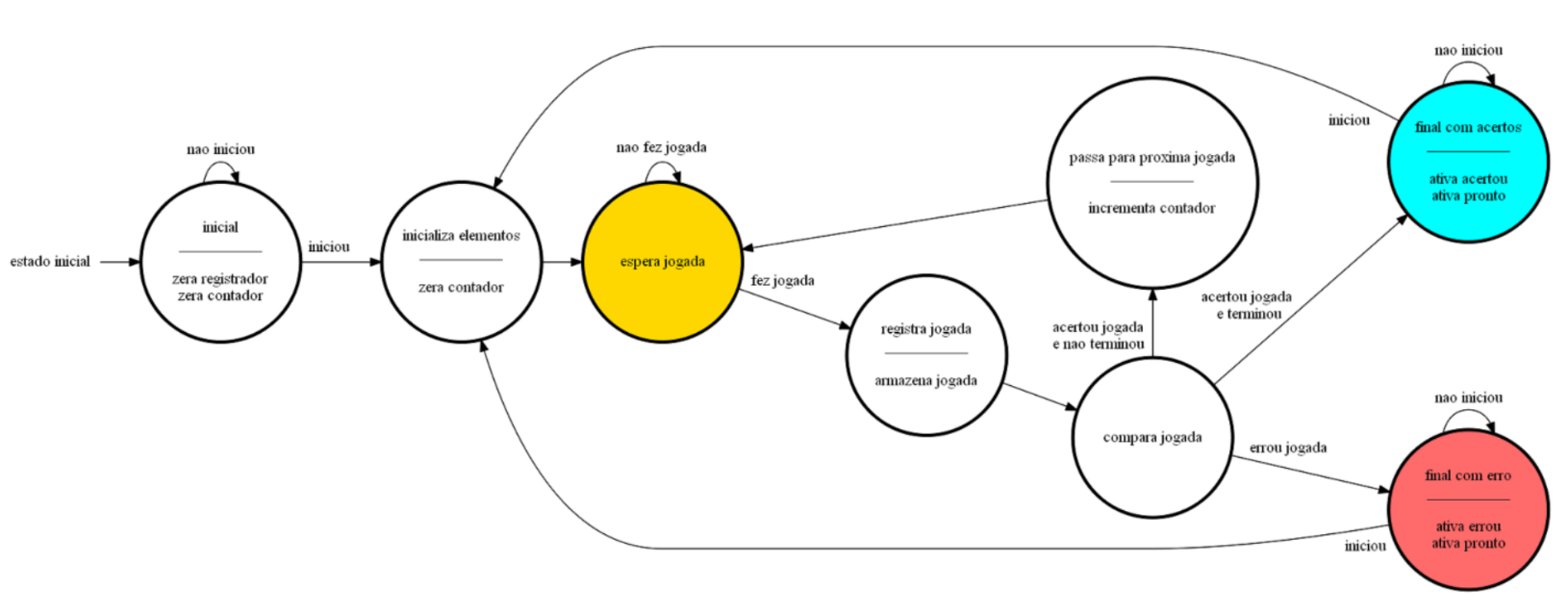


Figura 3: Diagrama de estados da unidade de controle

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 | 1: se fimC = 0 e os números comparados são iguais, vai para o estado Próximo na próxima subida de clock  2: se fimC = 1 e os números comparados são iguais vai para o estado Fim com Acerto na próxima subida de clock  3: se os números comparados são diferentes, independente de fimC vai para o estado Fim com Erro na próxima subida de clock |
| Próximo | Acrescenta um na contagem do contador | Espera Jogada | Na Próxima subida de clock |
| Fim com Acerto | Indica o fim acertando todos os números, além de levantar a flag acertou | Preparação | Só muda de estado com iniciar |
| Fim com Erro | Indica o fim acertando todos os números, além de levantar a flag errou | Preparação | Só muda de estado com iniciar |

* 1. **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**: envidado 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.

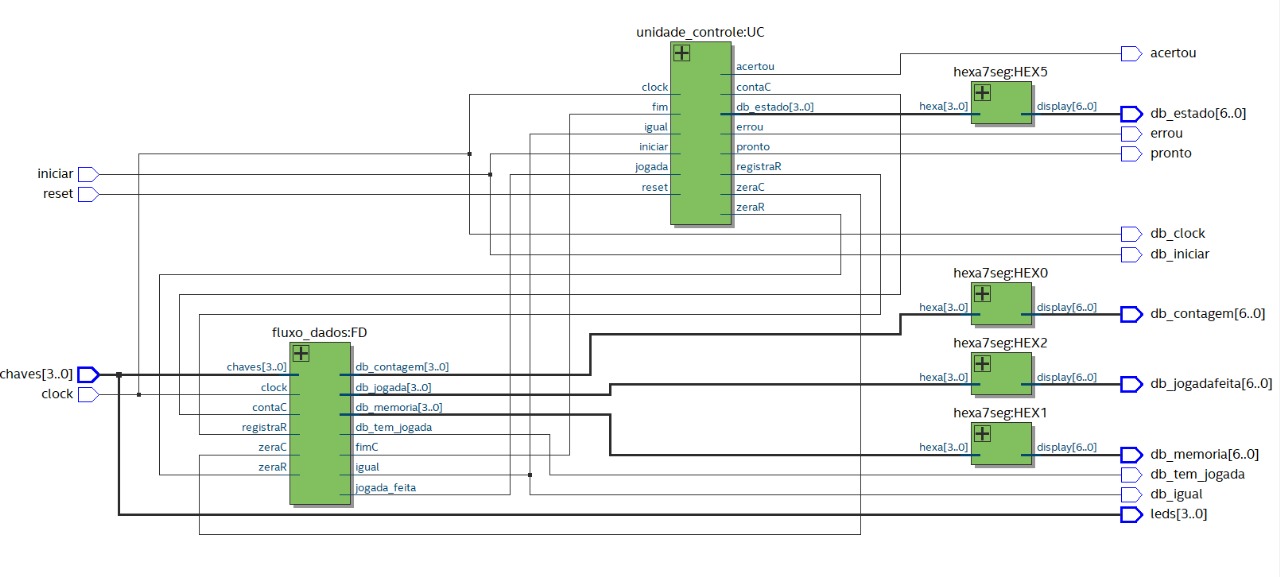


Figura 4: Esquema do circuito completo com RTLView

1. **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, segundo o professor, isso mudará nas próximas atividades.

**Imagens do Primeiro Cenário de Testes (Modelsim):**

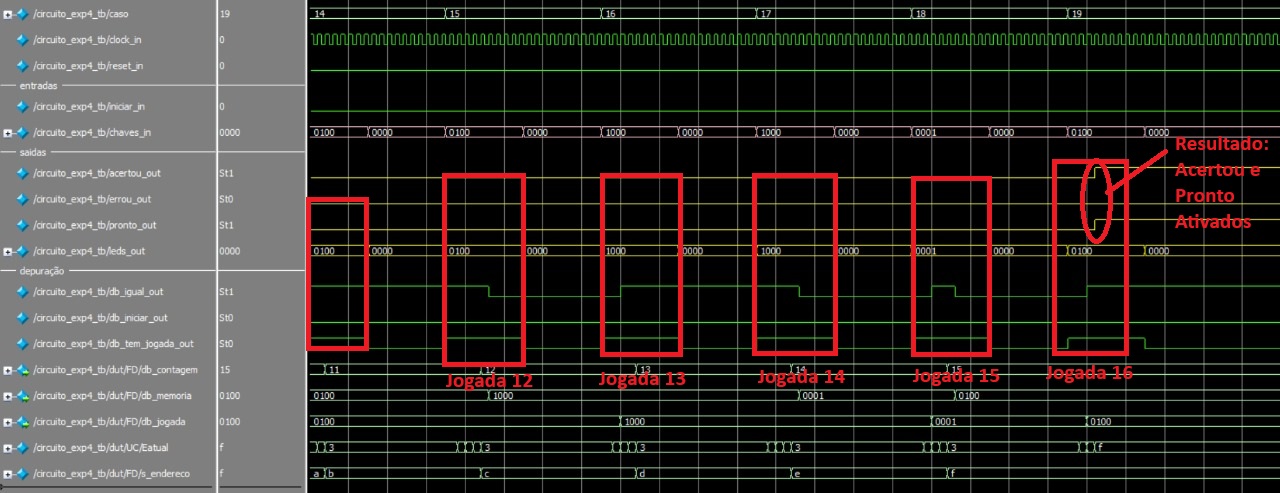


Figura 5: Imagem do Modelsim Testando o Final do Cenário de Teste 1

## 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):**

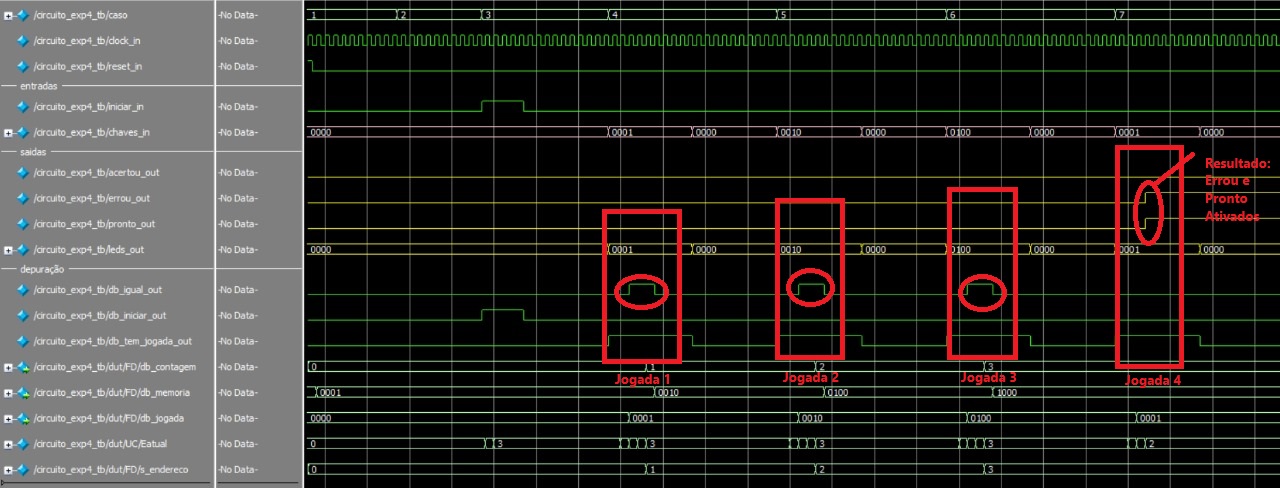


Figura 6: Imagem do Modelsim Testando o Cenário de Teste 2

# Implantação do Projeto

Será documentado durante a próxima aula.

# Projeto do Desafio da Experiência

Será documentado durante a próxima aula.

# 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.