

Desenvolvimento de um Modelo VHDL para Jogo da Memória

Beatriz Gouveia Gadelha^a, Isaac Marlon da Silva Lourenço^a

^aUniversidade Federal do Rio Grande do Norte, Instituto Metrópole Digital, Lagoa Nova, Natal - RN, 59076-560

Abstract. Este artigo apresenta o desenvolvimento de um modelo (VHSIC Hardware Description Language) para um sistema simples que simula o Jogo da Memória. Para isso, foi utilizado o software de design de hardware chamado *ModelSim*. Ademais, são apresentados os componentes de hardware necessários para uma integração e funcionamento do sistema como um todo. Por fim, foi possível a realização de testes na plataforma do *ModelSim* a fim de confirmar corretude e pleno funcionamento do sistema.

Keywords: circuitos lógicos, VHDL, simulação.

1 Introdução

A utilização de softwares para o design de hardware desempenha um papel fundamental na atual indústria de tecnologia.¹ Essa abordagem oferece uma série de benefícios que vão desde a economia de tempo e recursos até a melhoria da qualidade e da eficiência de um projeto. Nesse contexto, como forma de exercício educacional, este artigo apresenta a implementação de um modelo VHDL (VHSIC Hardware Description Language) na plataforma *ModelSim* para simular o Jogo da Memória.

O design desenvolvido inclui: um teclado numérico para seleção da carta que se deseja virar; e 16 displays de 7 segmentos para indicar as cartas viradas. Por fim, esta pesquisa destaca a aplicação da tecnologia VHDL na criação de um design de hardware modularizado e de simples compreensão.

Por fim, disponibilizamos o código do projeto VHDL desenvolvido neste artigo para discussão de possíveis aprimoramentos do sistema.

2 Desenvolvimento

Primeiramente, a fim de destacar a modularização obtida no processo de design do hardware, separamos o sistema do elevador em seis componentes, sendo eles: codificador de cartas; comparador de cartas selecionadas; máquina de estado; virador de cartas; contador de cartas viradas; display de 7 segmentos; manipulador de displays; escolhedor de cartas.

O codificador de cartas (COD) é responsável por receber o número correspondente a carta que o jogador deseja virar. Esse componente mantém um sinal de *buffer* para reconhecer quando o usuário escolheu duas cartas, i.e. deseja verificar se as cartas coincidem. A partir disso, a máquina de estados (ME) percebe que o jogador escolheu duas cartas e então aciona o estado do jogo COMPARACAO. Nesse estado o comparador de cartas selecionadas (COMP) é ativado e então compara as duas cartas selecionadas. Ademais, o resultado da comparação é repassada para o componente virador de cartas (VIRA) que é responsável por manter as duas cartas viradas ou desvirá-las caso que sejam diferentes.

O contador de cartas viradas (CONTA) é ativado sempre que o resultado da comparação é verdadeiro, permitindo que o CONTA incremente a quantidade de cartas viradas. Essa quantidade é verificada em tempo de *clock* pelo ME. Quando o jogador consegue manter desviradas todas as cartas, o ME altera o estado do sistema para FIM e então os componentes são desativados.

O manipulador de displays (MAN) é constituído como um bloco de 16 displays de 7 segmentos (D7). Cada um desses displays é responsável por exibir o símbolo da carta específica para a sua posição no teclado. Quando a carta está virada, todos os segmentos do display são ativados. Ao desvirar uma carta, será exibido o símbolo que é previamente designado para o display.

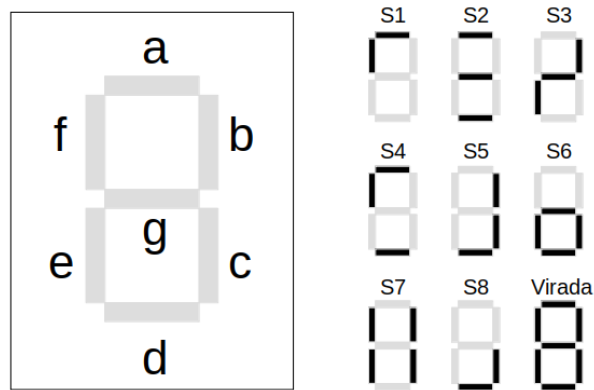


Fig 1 Display de 7 segmentos e suas representações de símbolos do jogo.

Por fim, destacamos que os blocos Cartas Jogo e Cartas Viradas são componentes de memória.

Esses sinais são interpretados pelo compilador VHDL como grupo de registradores.

A representação completa das comunicações entre os componentes pode ser visto no diagrama de blocos presente na Figura 2.

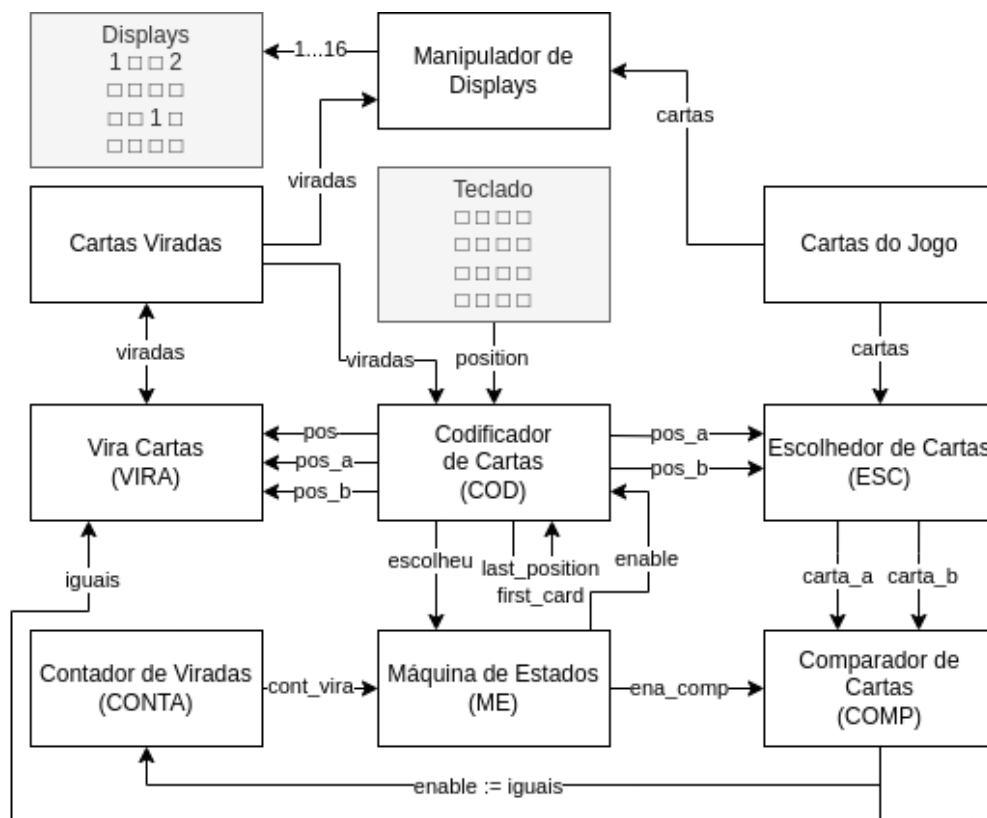


Fig 2 Diagrama de blocos representando a comunicação entre os componentes do sistema.

2.1 Lógica dos componentes

Inicialmente, destacamos a lógica necessária para o funcionamento do D7. A partir de uma tabela verdade (Tabela 1), conseguimos obter os valores binários que foram utilizados no código VHDL deste componente.²

Table 1 Tabela verdade do componente DD

Input	Símbolo	a	b	c	d	e	f	g
000	S1	1	0	0	0	0	1	0
001	S2	1	0	0	1	0	0	1
010	S3	0	1	0	0	1	0	1
011	S4	1	0	0	1	0	1	0
100	S5	0	1	1	1	0	0	0
101	S6	0	0	1	1	1	0	1
110	S7	0	1	1	0	1	1	0
111	S8	0	0	1	1	0	0	0
-	Virada	1	1	1	1	1	1	1

Note que o display exibirá o símbolo “Virada” quando o seu sinal de *enable* estiver desativado.

Ademais, dada a posição que o jogador quer desvirar a carta, o COD verifica se essa é a primeira ou segunda carta que o jogador está selecionando. Se é a primeira carta a ser selecionada, então o valor da posição é armazenado em *pos_a*; se é a segunda, *pos_b*. Os valores de posição são considerados como números naturais por praticidade. Caso que *pos_a* ou *pos_b* estiver com valor igual à 0 isso significa que a respectiva carta ainda não foi escolhida pelo jogador. Por fim, quando o jogador seleciona as duas cartas, o COD sinaliza para a ME através do sinal de *escolheu*.

O VIRA recebe os valores de *pos*, *pos_a* e *pos_b*, além da saída do COMP. O valor do *pos* é utilizado para indicar qual das cartas será desvirada. Dessa forma, é possível, sequencialmente,

desvirar as cartas que o jogador seleciona. Quando o jogador seleciona as duas cartas, o VIRA irá considerar o resultado obtido no COMP para decidir se as cartas irão se manter ou não desviradas.

A ME, por sua vez, implementado como uma máquina de Moore, gerencia o seu estado atual/futuro com os seguintes valores: ESPERA, COMPARACAO e FIM. De acordo com a saída do COD, o ME realiza o processamento e mudança de estado. Caso que o estado é de ESPERA, o próximo estado será de COMPARACAO se o jogador escolheu as duas cartas e ainda não foram desviradas todas as cartas do jogo. No estado de COMPARACAO, o COD é desativado e o COMP ativado, permitindo, assim, a comparação das cartas escolhidas e impedindo que o usuário sobrescreva a sua escolha. Por fim, caso que todas as cartas são desviradas, então o próximo estado da ME é FIM (nesse estado o COD e COMP são desativados).

O ESC possui acesso ao sinal das cartas do jogo. Dessa forma, através dos sinais de posição do COD, o ESC disponibiliza para o COMP os valores das cartas selecionadas. A saída do COMP é usada como *enable* do CONTA, acionando-o apenas quando o jogador escolher duas cartas iguais. Consideramos aqui a contagem de pares de cartas viradas, i.e. como o jogo é de 16 cartas, o valor 8 no CONTA significa 8 pares desvirados. Esse valor mantido pelo CONTA é disponibilizado para a ME.

Por fim, um código de integração instancia os componentes supracitados mapeando as portas de entrada necessárias para que a comunicação entre os componentes seja realizada. O mesmo tempo de *clock* foi utilizado para todos os componentes.

3 Simulações

Afim de verificar uma certa corretude do sistema, foram realizadas simulações. Para tal, utilizamos o software *ModelSim* pois este nos permitiu configurar de maneira simples e organizada os sinais

para cada entrada sensível do sistema. Inicialmente, realizamos a simulação de uma escolha de duas cartas iguais (Figura 3). Nessa primeira simulação é possível observar que o processo de escolha das cartas é sequencial e a cada escolha a carta é exibida no painel de displays. Como as cartas são iguais então essas cartas foram mantidas desviradas e a quantidade de pares aumentou em uma unidade.

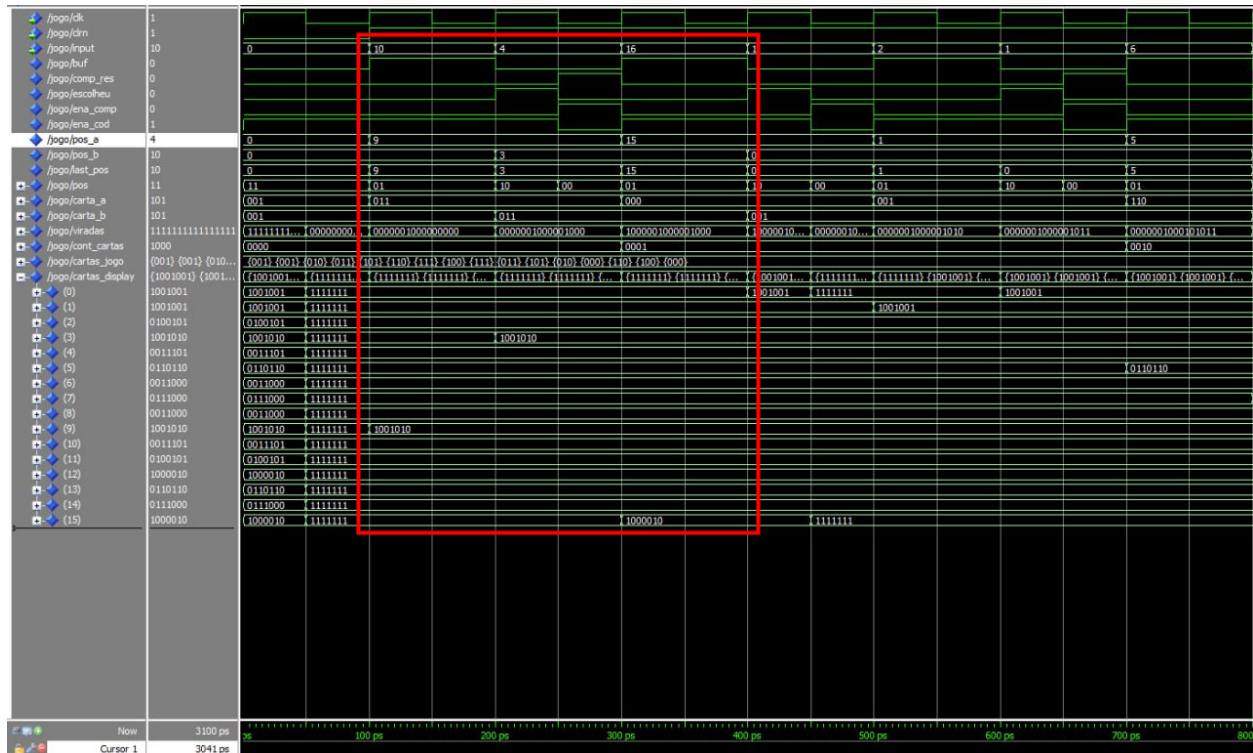


Fig 3 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após escolher duas cartas iguais.

Em seguida, fora realizada uma simulação de escolha de cartas diferentes (Figura 4). Inicialmente, nota-se que os displays mantiveram desviradas as cartas selecionadas na primeira simulação. Ademais, apesar da exibição das cartas selecionadas, percebe-se que as mesmas são viradas novamente pois não coincidem. Por fim, o CONTA manteve a quantidade de pares desvirados.

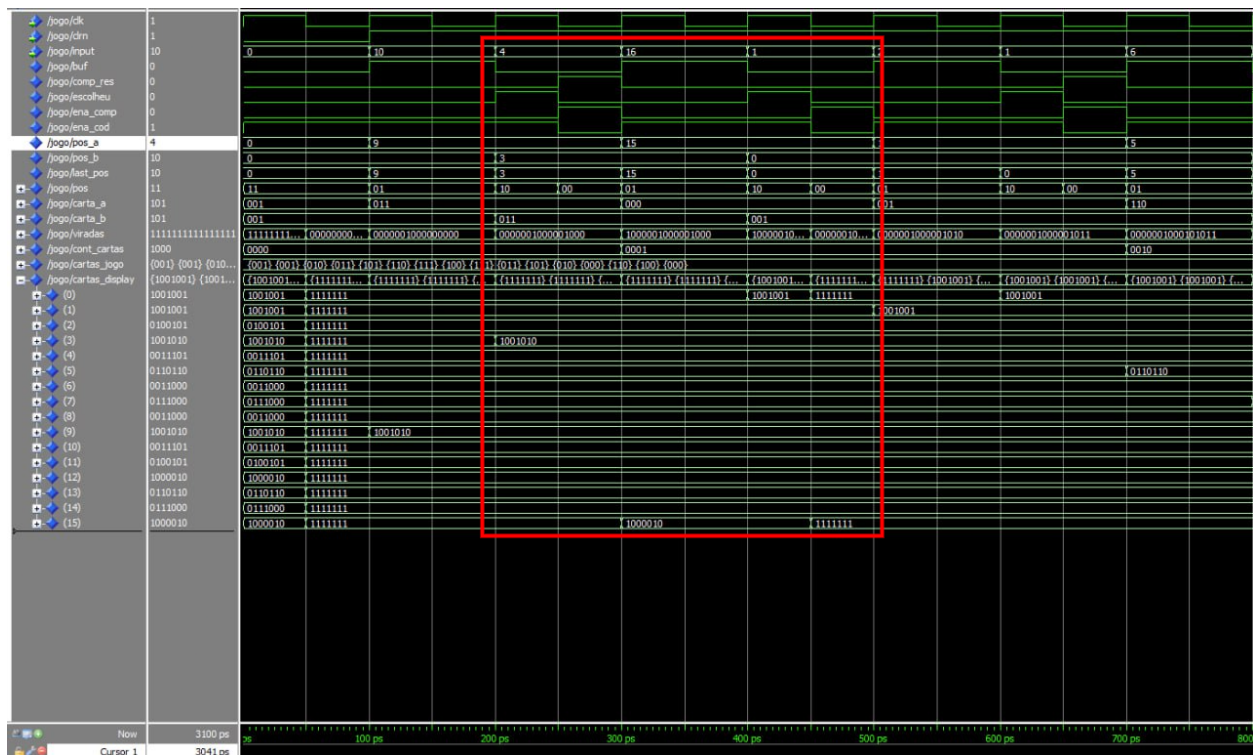
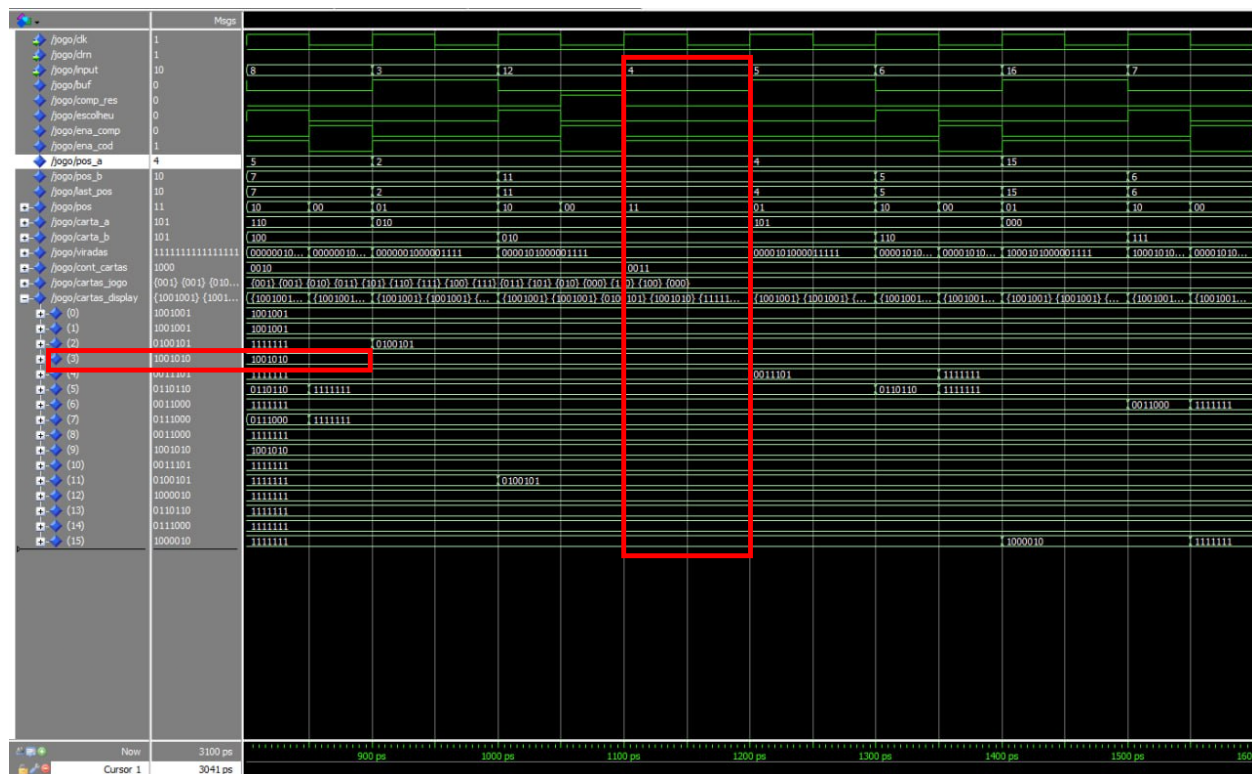


Fig 4 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após escolher duas cartas diferentes.

Em terceiro lugar, realizamos uma simulação para verificar o comportamento do sistema ao jogador tentar desvirar uma carta que já está desvirada. Nota-se que, apesar de o *input* mudar seu valor, nada mais é afetado por essa mudança. Isso ocorre devido a carta já encontrar-se desvirada, logo a seleção será desconsiderada. O resultado pode ser visualizado na Figura 5.



Ademais, realizamos a simulação de um jogo em seu estado inicial (Figura 6). Note que durante a inicialização do jogo, o sinal “clrn” encontra-se habilitado por um curto período de tempo. Nesse tempo, o CONTA reinicia a quantidade de pares de cartas desviradas e todas as cartas do jogo são exibidas nos displays e em seguida são viradas.

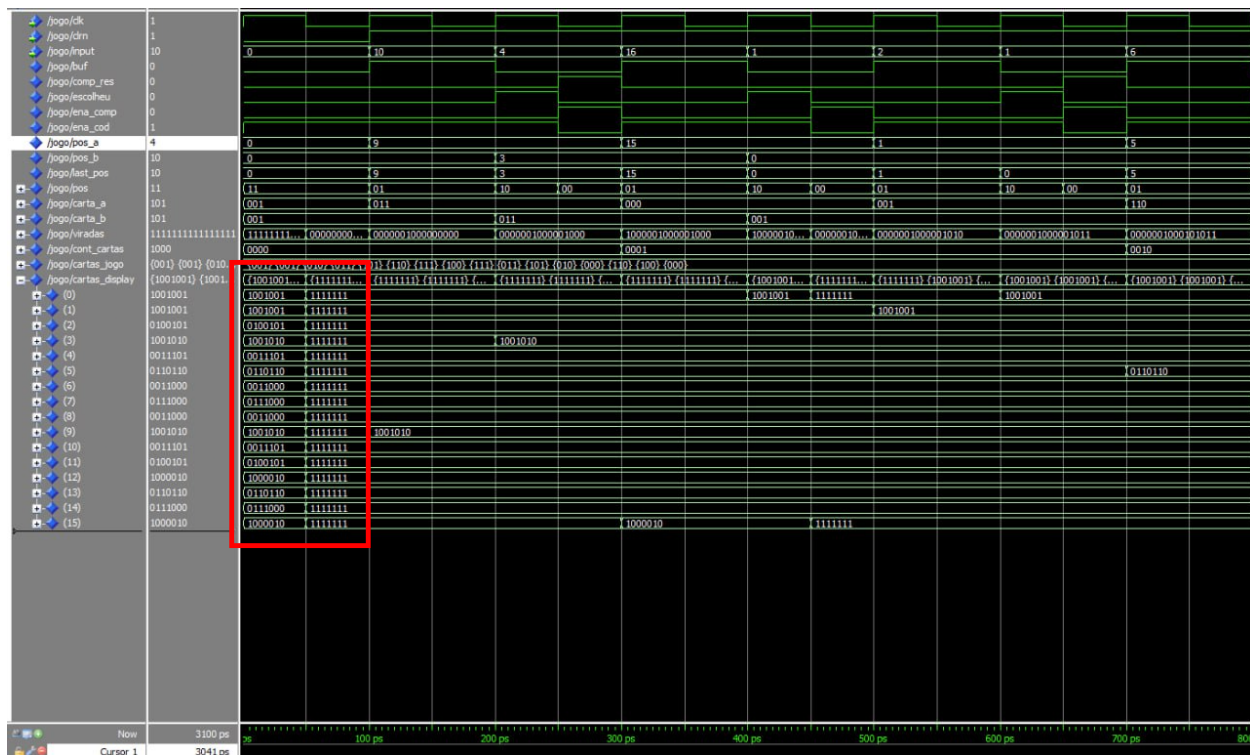


Fig 6 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais no início do jogo.

Por fim, realizamos a simulação de um jogo que obteve todas as cartas desviradas (Figura 7). Note que após o contador alcançar o valor de 8 pares de cartas desviradas, apesar do jogador tentar desvirar mais uma carta mesmo após o término do jogo, nada acontece.

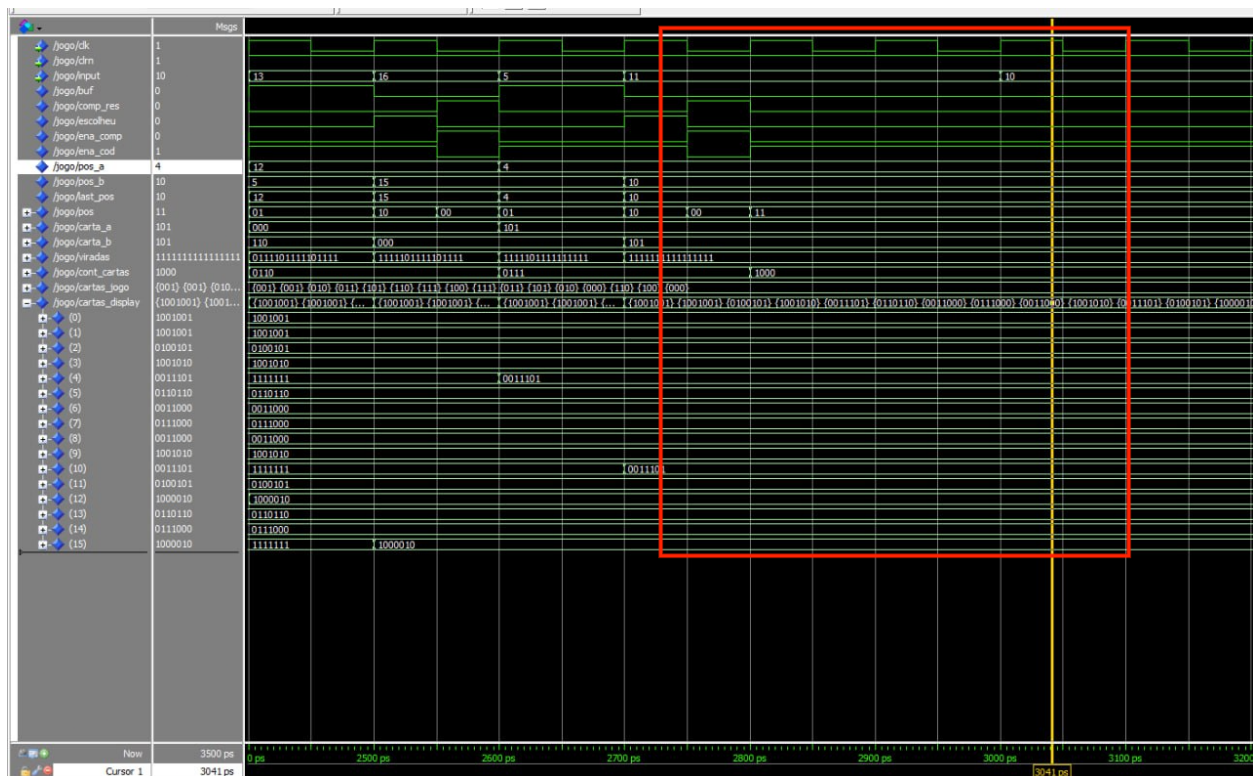


Fig 7 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após o estado FIM.

4 Conclusão

A modularização dos componentes se mostrou altamente eficaz, permitindo uma fácil manutenção e reutilização em outros projetos. Além disso, o software *ModelSim* permitiu a realização de testes de maneira simples e configurável que confirmaram, mesmo que empiricamente, a corretude do sistema do elevador.

Acknowledgments

Reconhecemos aqui o Dr. Prof. Márcio Eduardo Kreutz pela sua orientação e disposição na resolução dos problemas enfrentados na produção do sistema do elevador e na produção deste artigo.

References

- 1 V. Srinivasan, S. Radhakrishnan, and R. Vemuri, “Hardware software partitioning with integrated hardware design space exploration,” in *Proceedings Design, Automation and Test in Europe*, 28–35, IEEE (1998).
- 2 I. Lourenço, “Jogo da memória.” <https://github.com/isaacmsl/circuitos-jogo-memoria>. Acessado: December 6, 2023.

List of Figures

- 1 Display de 7 segmentos e suas representações de símbolos do jogo.
- 2 Diagrama de blocos representando a comunicação entre os componentes do sistema.
- 3 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após escolher duas cartas iguais.
- 4 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após escolher duas cartas diferentes.
- 5 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após escolher uma carta que já se encontra desvirada.
- 6 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais no início do jogo.
- 7 Gráfico de sinais da plataforma *ModelSim* representando o comportamento dos sinais do jogo após o estado FIM.