

Projeto Base do Jogo do Desafio da Memória

Versão 2025

OBJETIVOS

Após a conclusão desta experiência, os seguintes tópicos devem ser conhecidos pelos alunos:

- Projeto de um sistema digital a partir de sua especificação;
- Revisão de código de componentes;
- Projeto de fluxo de dados e de unidade de controle de circuitos digitais;
- Síntese em uma placa FPGA usando Quartus Prime.

RESUMO

Esta experiência tem como objetivo o início do desenvolvimento do circuito digital do **Jogo do Desafio da Memória** usando Verilog em uma placa FPGA. A experiência usa como base um fluxo de dados anterior, e a unidade de controle deve ser projetada tendo como base uma especificação do funcionamento do jogo.

1. PARTE EXPERIMENTAL

A parte experimental desta experiência visa iniciar o desenvolvimento do circuito base do projeto da disciplina, o **Jogo do Desafio da Memória**. O projeto deve ser sintetizado na placa DE0-CV com o FPGA Cyclone V 5CEBA4F23C7N. Posteriormente, deve-se executar um estudo detalhado de seu funcionamento por meio da execução de um plano de testes minucioso.

Esta experiência será dividida em algumas atividades, de tal forma a orientar o desenvolvimento do projeto de forma incremental.

1.1. Atividade 1 – Projeto Lógico do Jogo Base do Desafio da Memória

Nesta primeira atividade, será desenvolvido o projeto lógico do Jogo Base do Desafio da Memória, tendo como referências uma descrição do funcionamento do Jogo e os circuitos desenvolvidos nas experiências anteriores. Tais projetos das experiências anteriores devem ser revisados e ajustados para a especificação do projeto desta experiência (refatoração de código).

- a) Considere um sistema digital com a interface externa de sinais conforme diagrama de blocos da Figura 1. O circuito digital deve implementar um jogo similar ao Modo 1 dos brinquedos Genius [4] e Simon [7], denominado “O Desafio de Genius”. Neste jogo, o jogador deve repetir a sequência de sinais que o jogo apresenta nos LEDs de saída. Ao término de cada sequência, a sequência anterior é apresentada com uma jogada a mais. Uma jogada é realizada com o acionamento de um dos quatro botões de entrada.

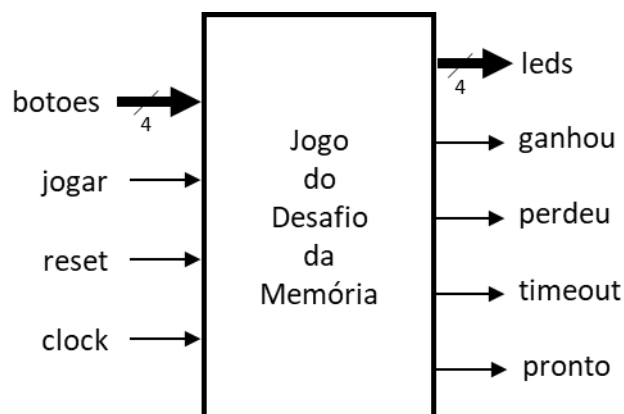


Figura 1: Diagrama de Blocos da Interface Externa de Sinais do Circuito para a Atividade 1

Na descrição prévia e no diagrama de blocos da Figura 1, não estão especificados **sinais adicionais de depuração**. Cabe ao grupo definir quais sinais devem ser acrescentados à interface de forma a garantir a realização eficiente de atividades de teste e de depuração e, depois, uma demonstração efetiva do circuito ao professor. Ao longo do enunciado da experiência, algumas sugestões são apresentadas com base em saídas de depuração já utilizadas em experiências anteriores.

O **funcionamento do sistema digital** da experiência deve seguir as instruções de jogo subsequentes (Figura 2).

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.

Figura 2: Funcionamento do Jogo do Desafio da Memória (Modo 1)

DICAS DE PROJETO:

O conjunto de jogadas é armazenado em uma **memória de jogadas** interna ao circuito da experiência. Cada jogada realizada pelo jogador deve ser armazenada pelo circuito em um **registrador**, e a sequência de execução das jogadas e de rodadas deve ser controlada por **contadores**. Uma jogada será avaliada como correta ou incorreta usando um **comparador**.

A lógica do jogo deve prever a realização de até 16 **rodadas**, de tal forma que a i-ésima rodada do jogo contemple a realização e a verificação de uma sequência de “i” jogadas, começando na primeira jogada e indo até a i-ésima jogada. Em outras palavras, o ciclo de espera pela jogada, o armazenamento da jogada, a comparação com a jogada da memória e o reposicionamento da memória deve seguir o seguinte padrão conforme a evolução das rodadas do jogo: na primeira rodada, somente o primeiro dado deve ser verificado; na segunda, os dois primeiros dados; na terceira, os três primeiros dados, e assim por diante até que, na décima sexta rodada, todos os 16 dados da memória sejam verificados.

A apresentação das jogadas na saída *leds* deve seguir a seguinte temporização: a saída *led* correspondente a cada jogada deve ser ativada por 500 ms e depois ser desativada por 500 ms, passando, em seguida para a próxima jogada.

- b) O funcionamento do Jogo Base do Desafio da Memória pode ser descrito pelo pseudocódigo da Figura 3. Alterações notáveis em relação às experiências anteriores são salientadas em **azul**.

Dica: Notar que o pseudocódigo da Figura 3 não prevê o tratamento do sinal *reset*. Este deve ser feito conforme especificado no item “a)”.

```

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.      aumenta o tamanho da sequência de jogadas
24.      vai para próxima rodada (reiniciar sequência)
25.
26.      // determina resultado do jogo
27.      se (atingiu o final do jogo acertando todas as jogadas) {
28.        então { // ganhou o jogo
29.          ativa saídas ganhou e pronto
30.          espera acionamento do sinal jogar
31.          reinicia o jogo
32.        }
33.      }
34.      se (houve jogada errada) {
35.        então { // perdeu o jogo
36.          ativa saídas perdeu e pronto
37.          espera acionamento do sinal jogar
38.          reinicia o jogo
39.        }
40.      }
41.      se (houve timeout) {
42.        então { // perdeu o jogo por timeout
43.          ativa saídas perdeu, timeout e pronto
44.          espera acionamento do sinal jogar
45.          reinicia o jogo
46.        }
47.      }
48.    }
49.  }

```

Figura 3: Pseudocódigo do Jogo Base do Desafio da Memória da Experiência 6

- c) A partir da especificação fornecida do Jogo do Desafio da Memória, desenvolva um **diagrama de alto nível** que descreve o funcionamento do circuito digital. Use comandos escritos e testes de condição em linguagem natural.
- d) Documente os elementos do **Fluxo de Dados** na forma de um diagrama de blocos.
- e) Considerando os sinais de controle e de condição dos elementos do Fluxo de Dados, elabore o **diagrama de transição de estados da Unidade de Controle**.

DICAS DE PROJETO:

O fluxo de dados apresentado deve seguir a mesma estrutura das experiências anteriores (p.ex. o fluxo de dados pode incluir componentes como um contador para controlar a rodada do jogo, um comparador para comparar o índice da jogada atual com o valor da rodada do jogo (verificar o final da sequência atual) e um registrador para armazenar a saída de dados da memória. Também deve ser incluído um contador para implementar a função de temporização, usada para medir o período de apresentação dos dados nos LEDs da saída *leds* (0,5s ou 500ms).

Os componentes das experiências anteriores devem ser revisados (**refatoração de código**) de forma a serem aproveitados para a experiência.

A apresentação das saídas *leds* das jogadas pode ser controlada pelo registrador de dados da memória da seguinte forma: o dado pode ser apresentado em *leds* carregando a saída da memória, e os LEDs podem ser apagados com a função *clear* do componente.

A unidade de controle deve gerar os sinais de controle dos elementos do Fluxo de Dados para implementar a sequência de operações do Jogo Base do Desafio da Memória descrita no pseudocódigo da Figura 3.

- f) Implemente o circuito da experiência em Verilog. Ao descrever o módulo topo (componente principal) do circuito, adote a interface (mínima) da Figura 4 para o nome do módulo e os sinais principais de entrada e saída.

NOTA: outras interfaces de depuração podem ser adicionadas a critério da equipe, mas a nomenclatura e as interfaces mínimas definidas na Figura 4 devem ser obrigatoriamente adotadas.

```
module jogo_desafio_memoria (
    input    clock,
    input    reset,
    input    jogar,
    input    [3:0] botoes,
    output   [3:0] leds,
    output   ganhou,
    output   perdeu,
    output   pronto,
    output   timeout

    // acrescentar saidas de depuracao
);
```

Figura 4: Interface Mínima de Sinais da Módulo Topo `jogo_desafio_memoria`

- g) Elabore um **Plano de Testes** para estudar o funcionamento do circuito de tal forma a considerar diversos cenários de teste (por exemplo, jogo com vencedor, perda do jogo com erro na 5ª jogada, jogar duas vezes consecutivamente, etc.) com um conjunto mínimo de casos de teste que ateste o funcionamento básico do Jogo do Desafio da Memória. Para cada cenário do Plano de Testes vislumbrado pelo grupo, gere uma tabela seguindo a estrutura da Tabela 1 e inclua, em cada etapa, os sinais de entrada necessários para executar cada operação e os resultados esperados.

Nota Importante: Cada cenário de teste deve ser executado **separadamente**. Deve haver uma tabela com os testes específicos para cada cenário exercitado.

Tabela 1: Modelo de Tabela para cada Cenário do Plano de Testes

Cenário #i – Descrição do Cenário				
#	Operação	Sinais de Entrada	Resultado Esperado	Resultado Observado
c.i.	Condições Iniciais			
1				
2				
3				
...				
n				

- h) Execute **simulações do circuito** com o ModelSim usando o Plano de Testes elaborado para testar o funcionamento do seu circuito. Submeta os arquivos fonte e os *testbenches* usados junto com o Planejamento (exp6_TxByy_modelsim.zip). **DICA:** elabore um arquivo de *testbench* (bancada de testes) para cada cenário de teste.
- i) Crie um projeto no Intel Quartus Prime referente ao circuito da Atividade 1.
- j) Documente o projeto do circuito da experiência, acrescentando diagramas gerados com as ferramentas **RTL Viewer** e **State Machine Viewer** do Intel Quartus Prime. Faça anotações nas figuras e inclua-os no Planejamento.
- k) Elabore um **projeto no Digital** e execute uma simulação de funcionamento do circuito da experiência para verificar o funcionamento dinâmico do projeto. Verifique o funcionamento e documente. Submeta o arquivo DIG (exp6_TxByy_atividade1_digital.dig) junto com o Planejamento.

1.2. Atividade 2 – Implementação e Síntese do Sistema Digital

Esta atividade tem como objetivo a implementação e a síntese do circuito projetado na Atividade 1 na placa FPGA DE0-CV. Em seguida, esse circuito deve ser estudado, e seu comportamento deve ser validado conforme projetado.

- l) Sintetize o projeto do circuito no FPGA Cyclone V 5CEBA4F23C7N da placa DE0-CV. Para isso, adote a designação de pinos da placa DE0-CV da Tabela 2 e complete-a com os dados faltantes no Planejamento – incluindo os sinais de depuração definidos no projeto do grupo.

Tabela 2: Designação de Pinos para a Atividade 2

	Sinal	Pino na Placa DE0-CV	Pino no FPGA	Analog Discovery
entradas	CLOCK	GPIO_0_D1		Patterns – Clock – 1 KHz StaticIO – LED – DIO0
	RESET	GPIO_0_D3		StaticIO – Button 0/1 – DIO1
	JOGAR	GPIO_0_D5		StaticIO – Button 0/1 – DIO2
	BOTOES(0)	GPIO_0_D7		StaticIO – Button 0/1 – DIO4
	BOTOES(1)	GPIO_0_D9		StaticIO – Button 0/1 – DIO5
	BOTOES(2)	GPIO_0_D11		StaticIO – Button 0/1 – DIO6
	BOTOES(3)	GPIO_0_D13		StaticIO – Button 0/1 – DIO7
saídas	GANHOU	GPIO_1_D1		StaticIO – LED – DIO8
	PERDEU	GPIO_1_D3		StaticIO – LED – DIO9
	PRONTO	GPIO_1_D5		StaticIO – LED – DIO10
	TIMEOUT	GPIO_1_D7		StaticIO – LED – DIO11
	LEDS(0)	GPIO_1_D9		StaticIO – LED – DIO12
	LEDS(1)	GPIO_1_D11		StaticIO – LED – DIO13
	LEDS(2)	GPIO_1_D13		StaticIO – LED – DIO14
depuração (sugestão)	db_clock	Led LEDR0		-
	db_tem_jogada	Led LEDR1		-
	db_chavesIgualMemoria	Led LEDR2		-
	db_enderecoIgualSequencia	Led LEDR3		-
	db_fimS	Led LEDR4		-
	db_contagem	Display HEX0		-
	db_memoria	Display HEX1		-
	db_jogadafeita	Display HEX2		-
	db_sequencia	Display HEX3		-
	db_estado	Display HEX4		-

- m) Submeta o arquivo QAR do projeto da experiência preparado para sua execução no Laboratório Digital (exp6_TxByy-ativ2-inicial.qar) junto com o Planejamento.

- n) No Laboratório Digital, programe o projeto sintetizado na placa FPGA DE0-CV.
- o) Configure as entradas e saídas nas ferramentas do Analog Discovery conforme a figura 5 abaixo.

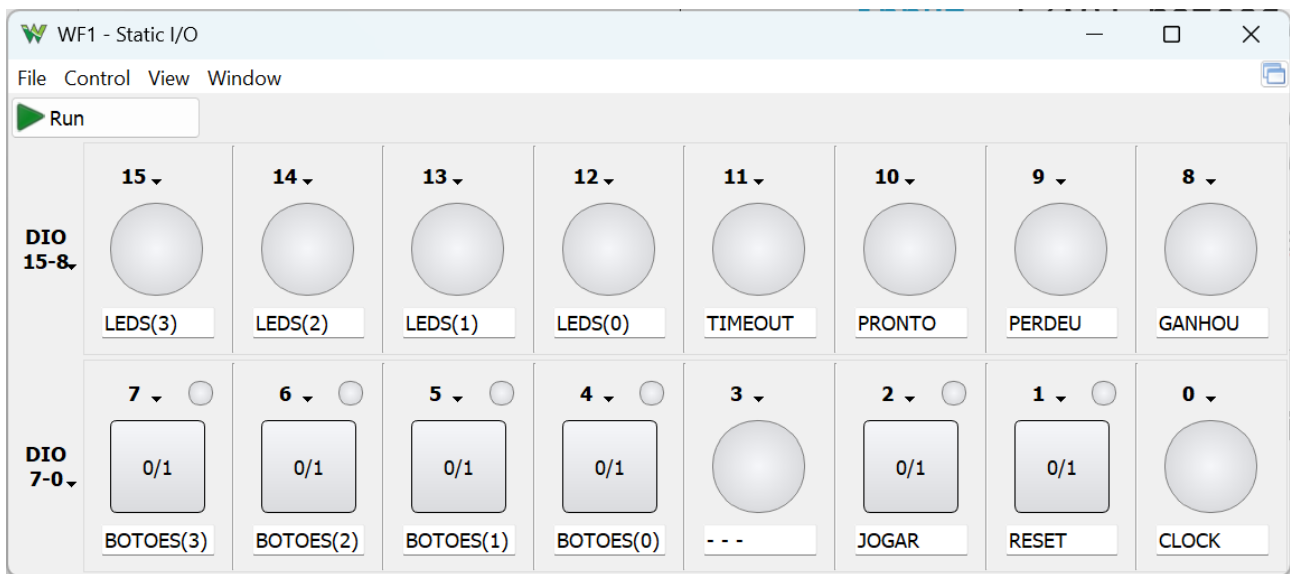


Figura 5: Sinais da experiência na ferramenta StaticIO do Analog Discovery.

- p) Ajuste também o Analog Discovery para gerar um sinal de *clock* com frequência de 1 KHz e execute o acionamento das sequências de sinais de entrada dos cenários do Plano de Testes do circuito. Anote os resultados experimentais obtidos para cada cenário.
- q) Analise os resultados obtidos e elabore um resumo do funcionamento do circuito estudado.
- r) Submeta o arquivo QAR final desta atividade (`exp6_TxByy-atividade2-final.qar`) junto com o Relato.

DICA: No Planejamento, prepare a realização desta Atividade e elabore um **Roteiro Experimental** de sua execução. Detalhe como os itens “n)” até “q)” deverão ser executados no Laboratório Digital e acrescente uma análise com possíveis causas de erros que podem ocorrer durante a execução dos testes e as ações que podem ser executadas para depurá-las e corrigi-las.

1.3. Atividade 3 – Desafio: Modificação do Sistema Digital

Esta atividade tem como objetivo praticar atividades de projeto de sistemas digitais. O circuito deve ser modificado a partir do projeto da Atividade 1.

- s) A especificação de uma modificação ao circuito da experiência será fornecida pelo professor.
- t) Projete a modificação do sistema digital e documente-a no relatório.
- u) Elabore um Plano de Testes para verificar o funcionamento do circuito, projetando-o para atestar o funcionamento das modificações introduzidas e a não regressão das demais funcionalidades não afetadas pela modificação (isto é, que as demais funcionalidades do circuito seguem inalteradas com a modificação). Para cada cenário do Plano de Testes vislumbrado pelo grupo, gere uma tabela seguindo a estrutura da Tabela 1 (ver Atividade 1) e inclua, em cada etapa, os sinais de entrada necessários para executar cada operação e os resultados esperados.
- v) Realize algumas **simulações do projeto** com o ModelSim executando os cenários do Plano de Testes do item anterior. Inclua as formas de onda obtidas na documentação da experiência. Submeta os arquivos usados com o ModelSim (exp6_TxByy_desafio_modelsim.zip) junto com o relato da experiência.
- w) Sintetize o projeto do circuito no FPGA Cyclone V 5CEBA4F23C7N da placa DE0-CV. Para isto, adote a mesma designação de pinos da Atividade 2 e complete-a com os dados faltantes. **Sinais adicionais de depuração** podem ser incluídos conforme a necessidade (consulte a designação de pinos disponível no e-Disciplinas e documente-a no relatório).
- x) Programe o projeto na placa DE0-CV.
- y) Execute o acionamento da sequência de sinais de entrada conforme Plano de Testes elaborado. Anote os resultados experimentais obtidos.
- z) Submeta o arquivo QAR final desta atividade (exp6_TxByy_desafio.qar) junto com o Relatório.

2. BIBLIOGRAFIA

- [1] ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. **Tutorial para criação de circuitos digitais em Verilog no Quartus Prime 20.1**. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2024.
- [2] ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. **Tutorial para criação de circuitos digitais hierárquicos em VHDL no Quartus Prime 16.1**. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.
- [3] ALTERA / Intel. **DE0-CV User Manual**. 2015.
- [4] ALTERA / Intel. **Quartus Prime Introduction Using Verilog Designs**. 2016.
- [5] ALTERA / Intel. **Quartus Prime Introduction to Simulation of Verilog Designs**. 2016.
- [6] ESTRELA. **Genius: manual do jogo**. Acessado em <https://www.estrela.com.br/central-de-manuais>.
- [7] HASBRO. **Simon instruction sheet**. 2013.
- [8] Ricardo Menotti, Ricardo dos Santos Ferreira. **Introdução à Lógica Digital com Verilog: uma abordagem prática**. Kindle. 2023
- [9] WAKERLY, John F. **Digital Design Principles & Practices**. 5th edition, Prentice Hall, 2018.

3. EQUIPAMENTOS NECESSÁRIOS

- 1 computador pessoal com os softwares Intel Quartus Prime e Intel ModelSim.
- 1 placa de desenvolvimento com FPGA DE0-CV com o dispositivo Cyclone V 5CEBA4F23C7N.
- 1 dispositivo Analog Discovery da Digilent ou equivalente.

Histórico de Revisões

E.T.M. / 2018 (versão inicial)
 E.T.M. & A.V.S.N / 2021 (revisão e adaptação para ensino remoto)
 E.T.M. & A.V.S.N / 2022 (revisão e adaptação do projeto)
 E.T.M. & A.V.S.N / 2025 (revisão e adaptação para Verilog)