

# Projeto de um Jogo de Sequências de Jogadas

Versão 2025

## OBJETIVOS

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

- Projeto de circuitos digitais com sinal periódico como clock
- 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 de um Jogo de Sequência de Jogadas usando Verilog em uma placa FPGA. A experiência usa como base um fluxo de dados proposto 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 digital de um Jogo de Sequência de Jogadas. O projeto deve ser sintetizado para a placa FPGA DE0-CV. Posteriormente, deve-se executar um estudo detalhado de seu funcionamento, através da execução de um plano de testes minucioso. Esta experiência será dividida em algumas atividades, onde o projeto será desenvolvido em algumas etapas, de forma incremental.

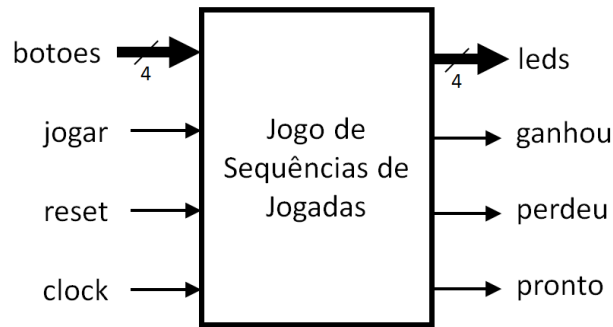
### 1.1. Atividade 1 – Projeto Lógico do Jogo de Sequência de Jogadas

Nesta primeira atividade, iremos desenvolver o projeto lógico do projeto do Jogo de Sequência de Jogadas, tendo como base os resultados dos estudos do fluxo de dados e da unidade de controle das últimas experiências. Os projetos das experiências anteriores devem ser revisados e ajustados para a especificação do projeto.

- a) Considere um sistema digital com a interface externa de sinais conforme diagrama de blocos da Figura 1. O **funcionamento do sistema digital** da experiência deve seguir a descrição subsequente. As alterações introduzidas em relação à experiência anterior estão destacadas em **azul**.

"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 `jogar`**. Depois de iniciar seu funcionamento, o circuito deve **aguardar uma jogada com o acionamento de um dos botões de entrada (sinal `botões`)**. Quando a jogada for realizada, o conteúdo dos botões deve ser armazenado pelo circuito e apresentado nos `leds` de saída. Em seguida, deve-se comparar o conteúdo armazenado da entrada de jogada com o respectivo dado da memória, gerando o resultado da comparação.

Se a comparação for igual, o contador interno deve ser ajustado para posicionar o endereçamento da memória para o próximo dado da memória. **O ciclo de registro de jogada, comparação e reposicionamento da memória deve seguir o seguinte padrão: inicialmente, somente o primeiro dado deve ser verificado. Em seguida, os dois primeiros dados. Daí, os três primeiros dados e, assim por diante, até que todos os 16 dados sejam verificados. Ou seja, a cada rodada, uma sequência de jogadas é verificada, iniciando com 1 elemento apenas, depois 2 elementos, e assim por diante, até chegar a uma sequência com todos os 16 elementos.** A cada etapa desse ciclo, o resultado da comparação entre a jogada efetuada e o conteúdo da memória determina a continuação ou o término do funcionamento do circuito. Quando a jogada efetuada for diferente do conteúdo da memória, o circuito encerra imediatamente o ciclo de operação e ativa a **saída `perdeu`**. No caso de o jogador acertar todas as sequências de jogadas até o conteúdo total da memória, a **saída `ganhou`** é ativada. Ao final da operação, independentemente do resultado do jogo, o sinal de saída `pronto` também deve ser ativado. Essas três saídas devem permanecer ativas até o reinício de operação do circuito, que deve ser realizado quando ocorrer a próxima ativação da **entrada `jogar`**".



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

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.

- b) Considere a seguinte descrição do funcionamento do Jogo de Sequências de Jogadas na forma de um pseudocódigo.

**Algoritmo:** Jogo de Sequências de Jogadas

**entradas:** jogar, botoes

**saídas:** leds, ganhou, perdeu, pronto

**depuração** (sugestão): contagem, memória, limite, estado, jogada, endereçoIgualLimite, chavesIgualMemoria

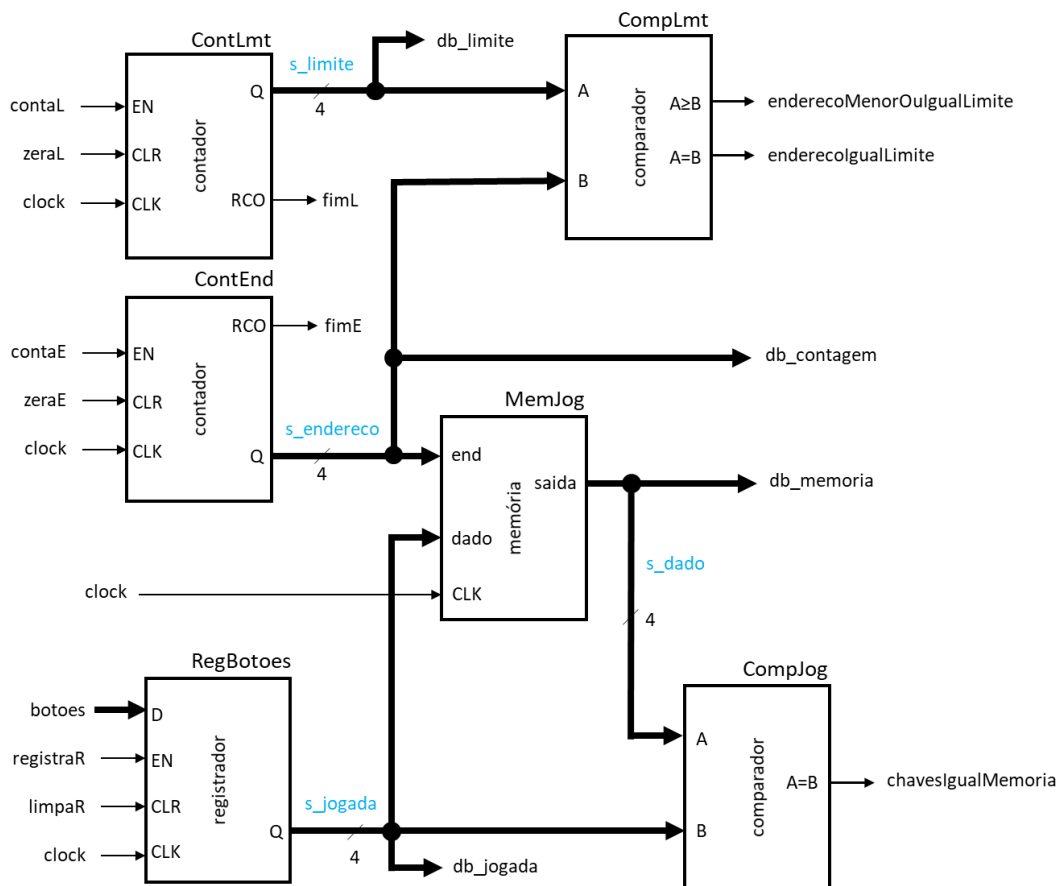
```

1. {
2.   while (verdadeiro) {
3.     espera acionamento do sinal JOGAR
4.     inicia circuito para condições iniciais e sequência inicial
5.
6.     enquanto não atingir final do jogo e não ocorrer uma jogada errada {
7.       reinicia sequência de jogadas
8.
9.       while não atingir jogada final da sequência atual e jogada foi correta {
10.        espera jogada
11.        compara jogada efetuada com jogada armazenada
12.        atualiza (incrementa) endereço da próxima jogada
13.      }
14.      aumenta o tamanho da sequência
15.      vai para próxima rodada (início da sequência)
16.
17.      se atingiu o final do jogo acertando todas as jogadas {
18.        então { // ganhou o jogo
19.          ativa saída GANHOU
20.          ativa saída PRONTO
21.          espera acionamento do sinal JOGAR
22.          reinicia o jogo
23.        }
24.      }
25.      se jogada errada {
26.        então { // perdeu o jogo
27.          ativa saída PERDEU
28.          ativa saída PRONTO
29.          espera acionamento do sinal JOGAR
30.          reinicia o jogo
31.        }
32.      }
33.    }
34.  }
  
```

Repare que, na descrição de funcionamento do circuito, ele ainda não se comporta como o brinquedo que serve como base do Jogo do Desafio da Memória. Isso vai ser considerado na próxima experiência.

- c) A partir da especificação fornecida do Jogo do Desafio da Memória, desenvolver um **diagrama de transição de alto nível** que descreve o funcionamento do circuito digital. Documente no Planejamento.
- d) Identifique os principais elementos do **Fluxo de Dados** (mostrados no diagrama de blocos na Figura 2) e sua aplicação na implementação das operações do circuito da experiência. Documente no Planejamento.
- e) A partir dos sinais de controle dos elementos do Fluxo de Dados, elabore o **diagrama de transição de estados da Unidade de Controle**. Documente no Planejamento.

DICA: estude os elementos do Fluxo de Dados e suas interações. Verifique como as diferentes funcionalidades do circuito da experiência podem ser implementadas com o acionamento dos elementos internos (p.ex. detecção do final de uma sequência de jogadas).



**Figura 2: Diagrama de blocos do Fluxo de Dados do circuito para a atividade 1.**

- f) Implemente o circuito da experiência em Verilog. Adote a seguinte interface padrão mínima de sinais do componente principal do circuito da experiência. Documente no Planejamento.

```
module circuito_exp5 (
    input    clock,
    input    reset,
    input    jogar,
    input    [3:0] botoes,
    output   [3:0] leds,
    output   pronto,
    output   ganhou,
    output   perdeu
    // acrescentar saidas de depuracao
);
```

**Figura 3: Interface Mínima de Sinais do Módulo Topo circuito\_exp5**

- g) Elabore um **Plano de Testes** para estudar o funcionamento do circuito. Considere diversos cenários de teste (com vencedor, com erro na jogada, dois jogos consecutivos, etc.) com um conjunto mínimo de casos de teste com o funcionamento básico do Jogo de Sequências de Jogadas. Complete a tabela com os sinais de entrada e de controle necessários para executar cada operação e os resultados esperados. Cada cenário de teste deve ser verificado separadamente.
- h) Executar **simulações do circuito** com o ModelSim usando o Plano de Testes elaborado para testar o funcionamento do seu circuito. Anote as formas de onda resultantes no Planejamento.
- DICA:** elaborar um arquivo de *testbench* (bancada de testes) para cada cenário de teste.
- i) Criar um projeto no Intel Quartus Prime referente ao circuito da atividade 1. Documente o projeto elaborado no Planejamento. Mostre as saídas da ferramenta **RTL Viewer** para o Sistema Digital e para o Fluxo de Dados e a saída da ferramenta **State Machine Viewer** para a Unidade de Controle.

## 1.2. Atividade 2 – Implementação e Síntese do Sistema Digital

Esta atividade tem como objetivo a implementação do projeto em Verilog e sua síntese para a placa FPGA DE0-CV. Em seguida, este circuito deve ser estudado e seu comportamento deve ser validado conforme projetado.

- j) 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 1 e complete-a com os dados faltantes no Planejamento. Adicione os sinais de depuração definidos no projeto do grupo. Submeta o arquivo QAR do projeto da atividade (`exp5_TxByy-ativ2-inicial.qar`) junto com o Planejamento.

**Tabela 1: Designação de Pinos para a Atividade 2**

	Sinal	Pino na Placa DE0-CV	Pino no FPGA	Analog Discovery
entradas	CLOCK	GPIO_0_D13	completar	StaticIO – LED – DIO0 Patterns – Clock – 1KHz
	RESET	GPIO_0_D15	completar	StaticIO – Button 0/1 – DIO1
	INICIAR	GPIO_0_D17	completar	StaticIO – Button 0/1 – DIO2
	BOTOES(0)	GPIO_0_D19	completar	StaticIO – Button 0/1 – DIO3
	BOTOES(1)	GPIO_0_D21	completar	StaticIO – Button 0/1 – DIO4
	BOTOES(2)	GPIO_0_D23	completar	StaticIO – Button 0/1 – DIO5
	BOTOES(3)	GPIO_0_D25	completar	StaticIO – Button 0/1 – DIO6
saídas	LEDS(0)	Led LEDR0	completar	-
	LEDS(1)	Led LEDR1	completar	-
	LEDS(2)	Led LEDR2	completar	-
	LEDS(3)	Led LEDR3	completar	-
	PERDEU	Led LEDR7	completar	-
	GANHOU	Led LEDR8	completar	-
	PRONTO	Led LEDR9	completar	-
depuração (sugestão)	db_chavesIgualMemoria	Led LEDR4	completar	-
	db_clock	Led LEDR5	completar	-
	db_tem_jogada	Led LEDR6	completar	-
	db_contagem	Display HEX0	completar	-
	db_memoria	Display HEX1	completar	-
	db_jogada	Display HEX2	completar	-
	db_limite	Display HEX3	completar	-
	db_estado	Display HEX5	completar	-

- k) Programe o projeto sintetizado na placa DE0-CV na bancada do Laboratório Digital.
- l) 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.
- m) Ajuste os sinais de entrada usando o Analog Discovery. Adote uma frequência de clock de 1KHz.
- n) Analise os resultados obtidos e elabore um resumo do funcionamento do circuito estudado.
- o) Submeta o arquivo QAR final após a realização desta atividade (`exp5_TxByy-ativ2-final.qar`) junto com o Relatório.

### 1.3. Atividade 3 – Desafio: Modificação do Sistema Digital

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

- p) A especificação de uma modificação ao circuito da experiência será fornecida pelo professor.
- q) Projete a modificação do sistema digital e documente-a no relatório.
- r) Elabore um Plano de Testes para verificar o funcionamento do circuito. Considere ao menos dois cenários de teste para o circuito modificado. Para cada cenário vislumbrado pelo grupo, gere uma tabela com o plano de testes seguindo a estrutura da Tabela 1 e complete-a com os sinais de controle necessários para executar cada operação e os resultados esperados.

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

**Tabela 1: Modelo de Tabela para o Plano de Testes da Atividade 3**

Cenário #n – Descrição do Cenário				
#	Operação	Sinais de entrada	Resultado esperado	Resultado observado
c.i.	Condições Iniciais			
1				
2				
3				
...				
n				

- s) 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.
- t) Sintetize o projeto do circuito no FPGA Cyclone V 5CEBA4F23C7N. 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).
- u) Programe o projeto na placa DE0-CV.
- v) Execute o acionamento da sequência de sinais de entrada conforme Plano de Testes elaborado. Anote os resultados experimentais obtidos.
- w) Submeta o arquivo QAR final desta atividade (`exp5_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. **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] Ricardo Menotti, Ricardo dos Santos Ferreira. **Introdução à Lógica Digital com Verilog: uma abordagem prática.** Kindle. 2023
- [7] WAKERLY, John F. **Digital Design Principles & Practices.** 5<sup>th</sup> edition, Prentice Hall, 2018.

## 3. EQUIPAMENTOS NECESSÁRIOS

- 1 computador pessoal com o software Intel Quartus Prime e ModelSim.
- 1 placa de desenvolvimento FPGA DE0-CV da Altera 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. / 2025 (revisão e adaptação para Verilog)