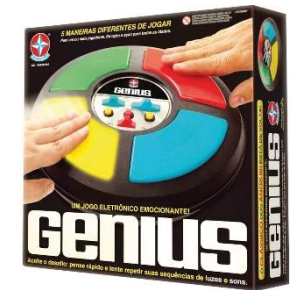


# Experiência 5 – Projeto do Jogo de Sequências de Jogadas

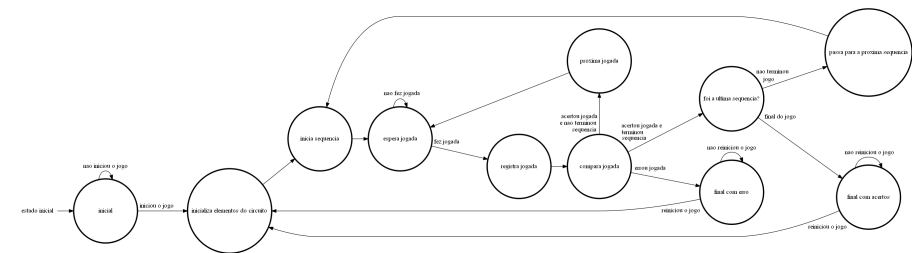
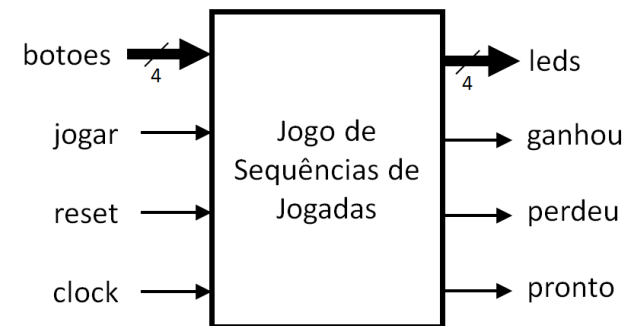
PCS3635 – Laboratório Digital I  
1º Quadrimestre de 2025

# Experiência 5



- Objetivos
  - Projeto de um sistema digital a partir de sua especificação;
  - Revisão de código de componentes (refatoração de código);
  - Projeto de fluxo de dados e de unidade de controle de circuitos digitais;
  - Síntese para uma placa FPGA

```
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
);
```

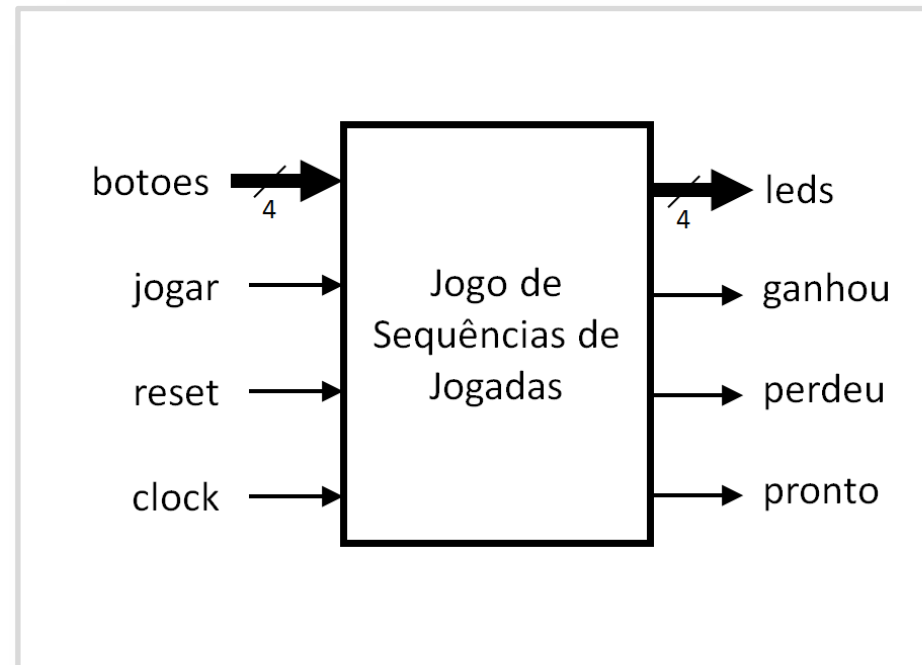


# Experiência 5



- Especificação do Projeto da Experiência:

Projetar o circuito  
ao lado baseado em  
uma descrição usando  
Verilog e sintetizar  
em placa FPGA



# Experiência 5

- Desenvolvimento do Projeto da Experiência:
  1. Sistema Digital = Fluxo de Dados + Unidade de Controle
  2. Circuito baseado no projetos das experiências passadas (refatoração de código)
  3. Fluxo de Dados = introduzir elementos adicionais
  4. Unidade de Controle = máquina de estados baseado na descrição do Jogo

# **ATIVIDADE 1**

# Experiência 5

- Atividade 1 – Projeto Lógico

## Descrição de funcionamento

“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 botoes)*. 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 deste 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, independente 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. Depois disso, o circuito deve aguardar a próxima ativação de *jogar*”.

# Experiência 5

- Atividade 1 – Projeto Lógico

## Pseudocódigo

```
Algoritmo: Jogo de Sequências de Jogadas
entradas: jogar, botoes
saídas: leds, ganhou, perdeu, pronto
depuração (sugestão): contagem, memória, limite, estado, jogada,
enderecoIgualLimite, 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.       enquanto 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.
18.    se atingiu o final do jogo acertando todas as jogadas {
19.      então { // ganhou o jogo
20.        ativa saída GANHOU
21.        ativa saída PRONTO
22.        espera acionamento do sinal JOGAR
23.        reinicia o jogo
24.      }
25.    }
26.
27.    se jogada errada {
28.      então { // perdeu o jogo
29.        ativa saída PERDEU
30.        ativa saída PRONTO
31.        espera acionamento do sinal JOGAR
32.        reinicia o jogo
33.      }
34.    }
35.  }
36. }
```

# Experiência 5



- Atividade 1 – Projeto Lógico

## Algumas dicas de projeto

### Conceitos de Jogo, Rodada, Sequência, Jogada

- o Jogo pode ter até 16 rodadas (ou sequências de jogadas);
- a 1ª rodada tem 1 jogada (primeira sequência tem apenas 1 jogada);
- a 2ª rodada tem 2 jogadas (2ª sequência tem 2 jogadas);
- a  $n$ -ésima rodada tem  $n$  jogadas (sequência de  $n$  jogadas);
- a última rodada tem 16 jogadas (sequência de 16 jogadas).

← igual ao circuito da experiência 4

### Leds de saída

- mostram a sequência de jogadas realizadas.

Dúvidas?



# Experiência 5

- Atividade 1 – Projeto Lógico

## Pseudocódigo:

**Algoritmo:** Jogo de Sequências de Jogadas

**entradas:** jogar, botoes

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

**depuração** (sugestão): contagem, memória, limite, estado, jogada, endereçoIguarLimite, chavesIguarMemoria

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

Controla as Rodadas  
(sequências de 1 até 16 jogadas)

Controla a sequência  
de jogadas dentro de  
uma rodada  
(Jogada 1 até final da  
sequência)

Resultado  
do jogo

# Experiência 5

- Atividade 1 – Projeto Lógico

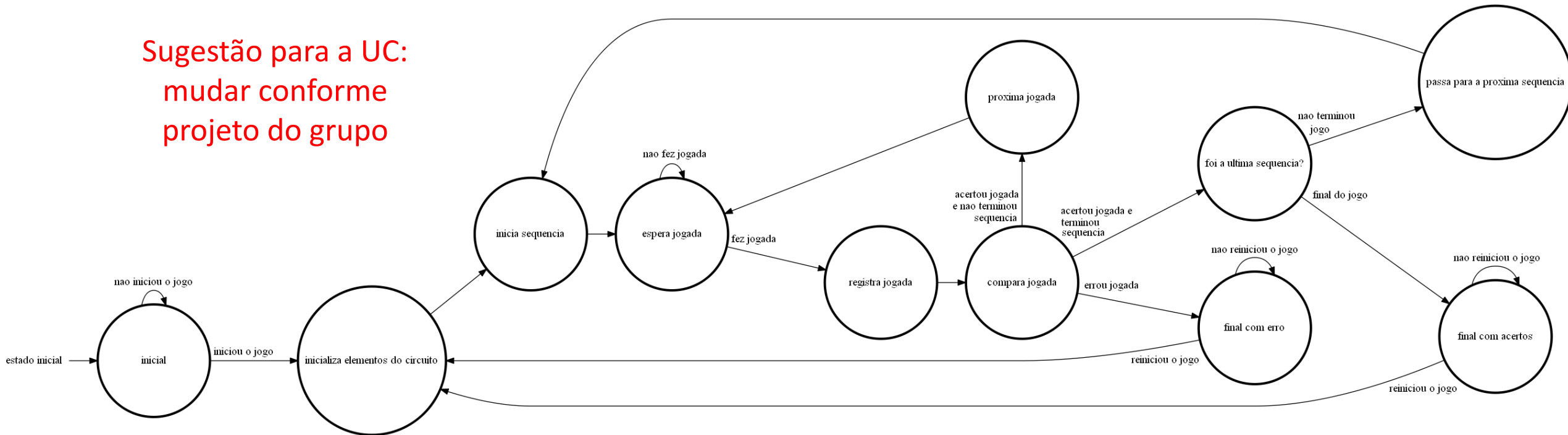
Pseudocódigo

```
Algoritmo: Jogo de Sequências de Jogadas
entradas: jogar, botoes
saídas: leds, ganhou, perdeu, pronto
depuração (sugestão): contagem, memória, limite, estado, jogada,
enderecoIgualLimite, chavesIgualMemoria

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

## Desenvolvimento do Sistema Digital

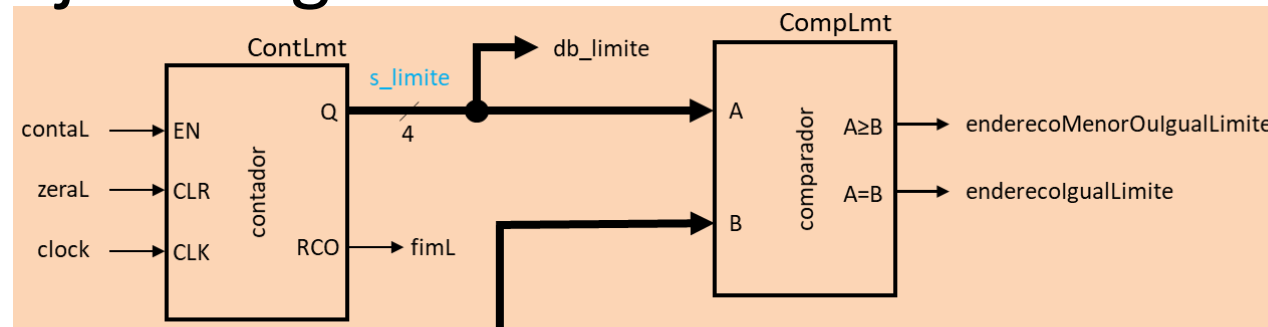
Sugestão para a UC:  
mudar conforme  
projeto do grupo



# Experiência 5

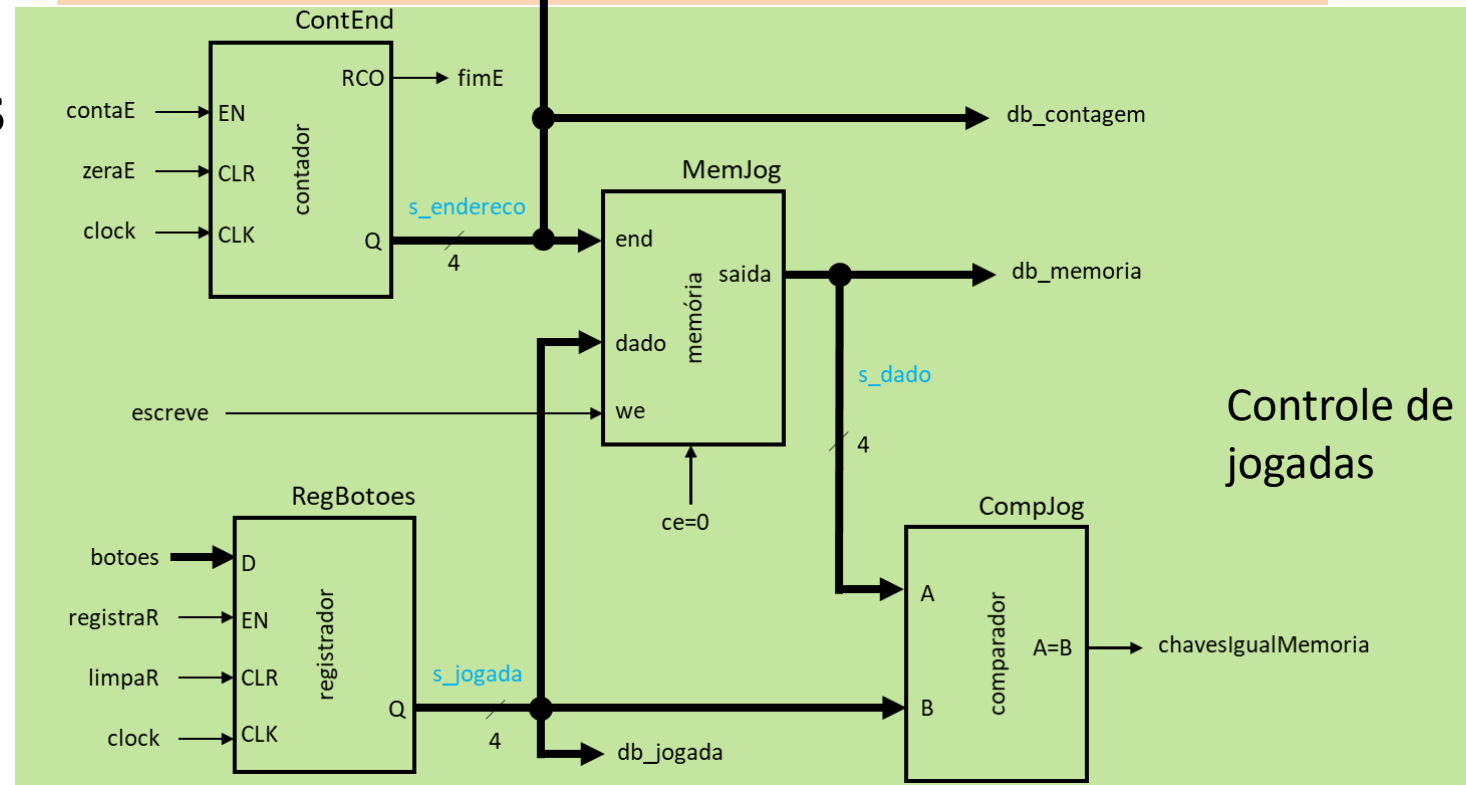
- Atividade 1 – Projeto Lógico

Controle de sequências



## Fluxo de Dados

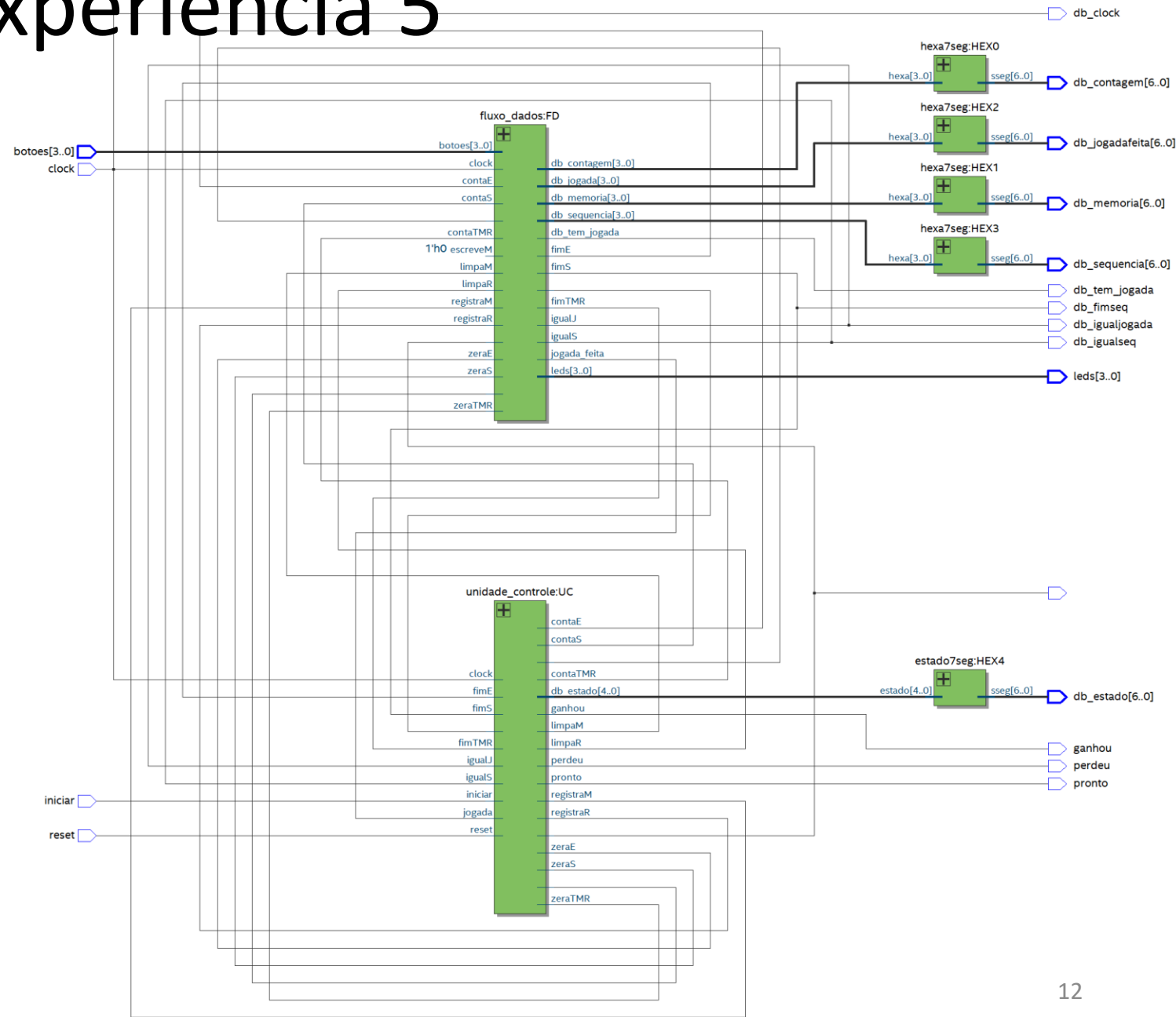
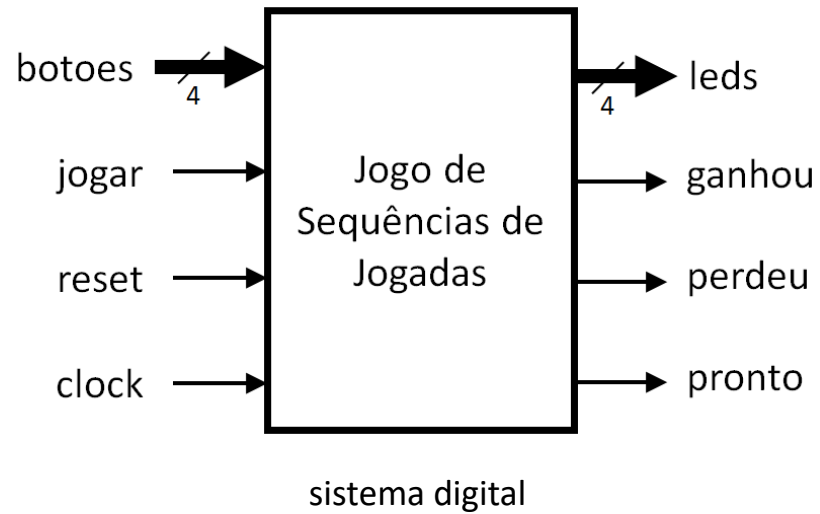
Sugestão para o FD:  
mudar conforme  
projeto do grupo



Controle de jogadas

# Experiência 5

- Atividade 1 – Projeto Lógico



# Experiência 5

- Atividade 1 – Projeto Lógico

## Plano de Teste:

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				

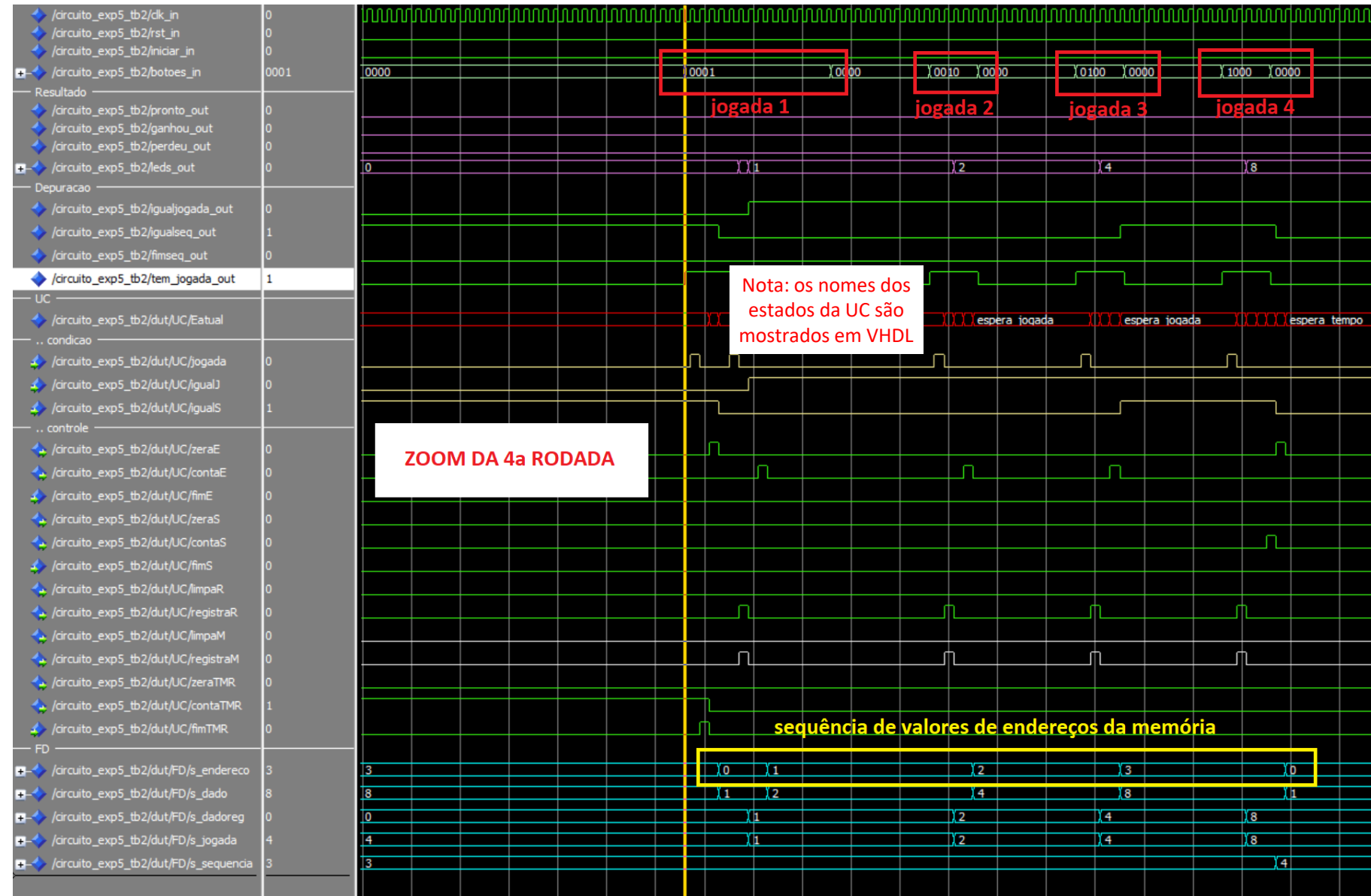
Desenvolver cenários variados (exemplos):

- Jogo com vencedor (acertos nas 16 rodadas)
- Perda do jogo com erro na 2ª jogada da 5ª rodada
- Jogar 2x consecutivamente
- outros?

# Experiência 5

- Atividade 1 – Projeto Lógico

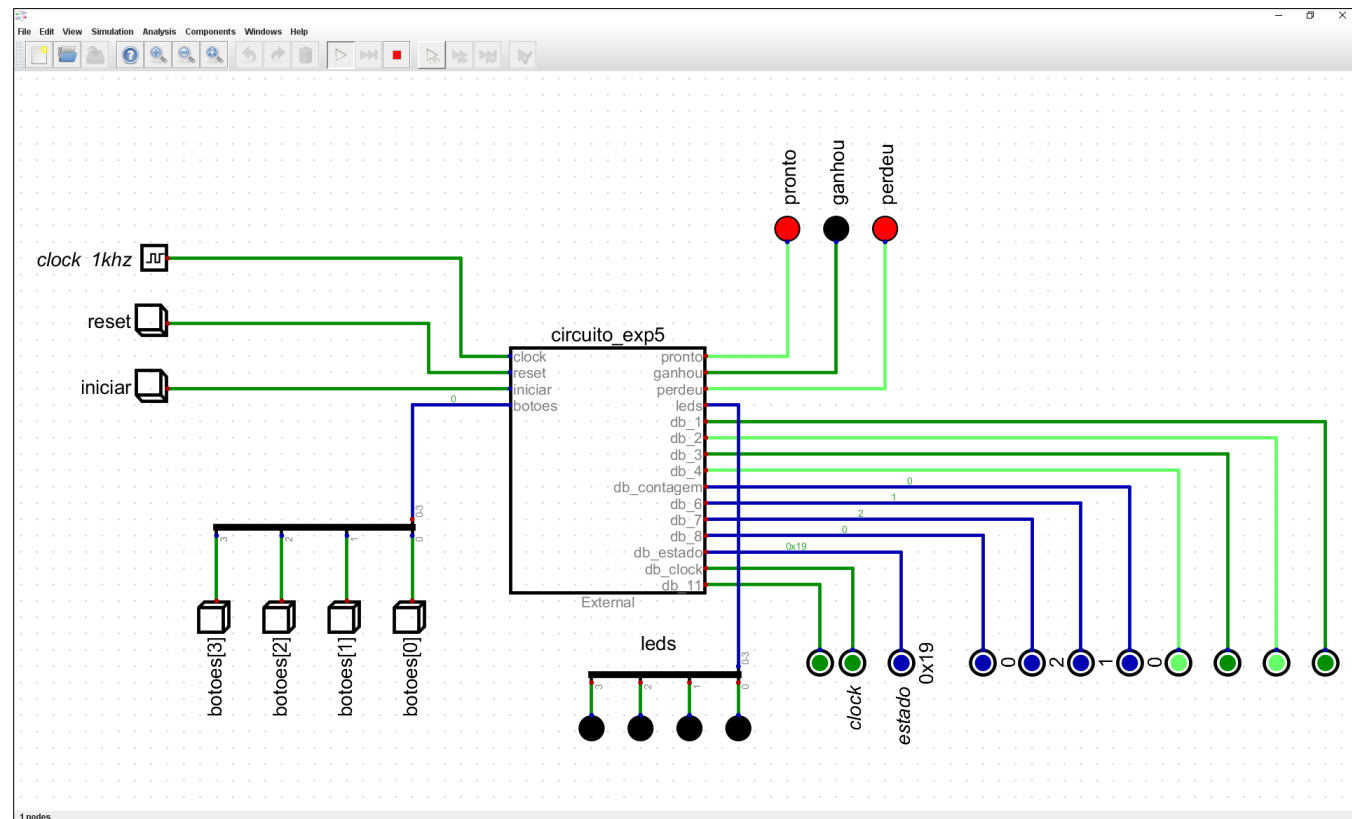
# Simulação com ModelSim



# Experiência 5

- Atividade 1 – Projeto Lógico (simulação com Digital)

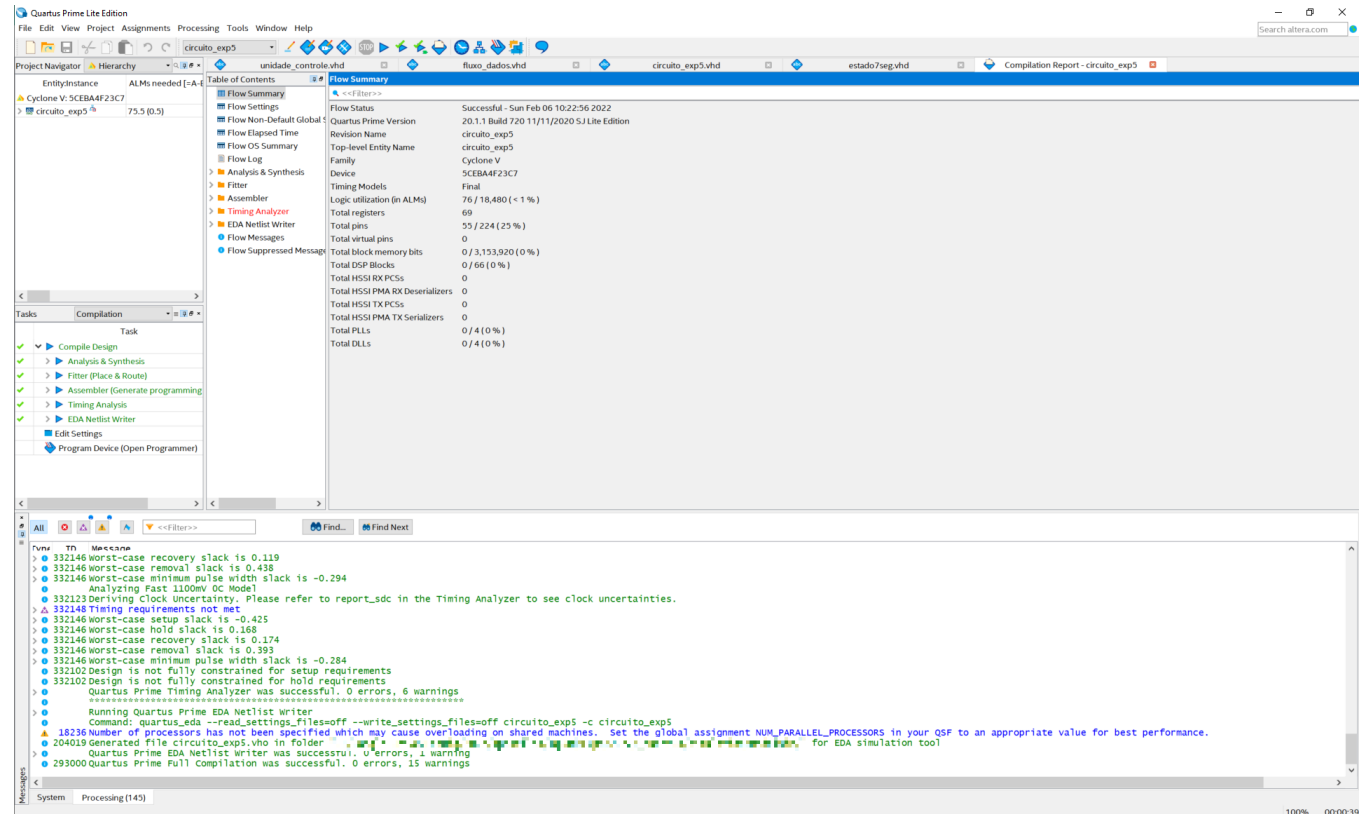
Sugestão:  
Simulação com  
Digital



# Experiência 5

- Atividade 1 – Projeto Lógico

Projeto no  
Intel Quartus Prime





# **ATIVIDADE 2**

# Experiência 5

- Atividade 2 –  
Síntese  
do circuito

Designação de pinos

**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		StaticIO – LED – DIO0 Patterns – Clock – 1KHz
	RESET	GPIO_0_D15		StaticIO – Button 0/1 – DIO1
	INICIAR	GPIO_0_D17		StaticIO – Button 0/1 – DIO2
	BOTOES(0)	GPIO_0_D19		StaticIO – Button 0/1 – DIO3
	BOTOES(1)	GPIO_0_D21		StaticIO – Button 0/1 – DIO4
	BOTOES(2)	GPIO_0_D23		StaticIO – Button 0/1 – DIO5
	BOTOES(3)	GPIO_0_D25		StaticIO – Button 0/1 – DIO6
saídas	LEDS(0)	Led LEDR0		-
	LEDS(1)	Led LEDR1		-
	LEDS(2)	Led LEDR2		-
	LEDS(3)	Led LEDR3		-
	PERDEU	Led LEDR7		-
	GANHOU	Led LEDR8		-
	PRONTO	Led LEDR9		-
depuração (sugestão)	db_chavesIgualMemoria	Led LEDR4		-
	db_clock	Led LEDR5		-
	db_tem_jogada	Led LEDR6		-
	db_contagem	Display HEX0		-
	db_memoria	Display HEX1		-
	db_jogada	Display HEX2		-
	db_limite	Display HEX3		-
	db_estado	Display HEX5		-

# Experiência 5

- Atividade 2 – Implementação do Circuito

## Testes na placa FPGA

- Aplicar o Plano de Teste proposto para o circuito.
- Acionar as entradas do circuito.
- Acompanhar saídas mapeadas em *displays* de 7 segmentos e *leds*.

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				

# Experiência 5

- Atividade 2 – Implementação do Circuito

Depuração na Placa FPGA (sugestão)

**Passo 1** – verificação do funcionamento geral do circuito  
(rodadas 1 a 16, jogadas da rodada, etapas do jogo);

**Passo 2** – verificação de cada etapa do jogo  
(processamento de sequências, processamento de jogadas)

**Passo 3** – verificação do Jogo

Acompanhar dos sinais  
de depuração do circuito  
(acrescentar outros se necessário)

# **ATIVIDADE 3**

# Experiência 5

- Atividade 3 – Desafio: Modificação ao Sistema Digital
  1. Especificação a ser apresentada pelo professor.
  2. Projetar a modificação.
  3. Sintetizar o projeto no placa FPGA.
  4. Testar o funcionamento do circuito.
  5. Documentar projeto e funcionamento no Relatório.