

# Experiência 4 – Desenvolvimento de Projeto de Circuitos em FPGA

1º Quadrimestre de 2025

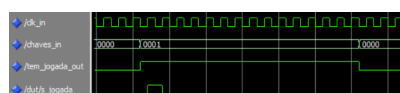
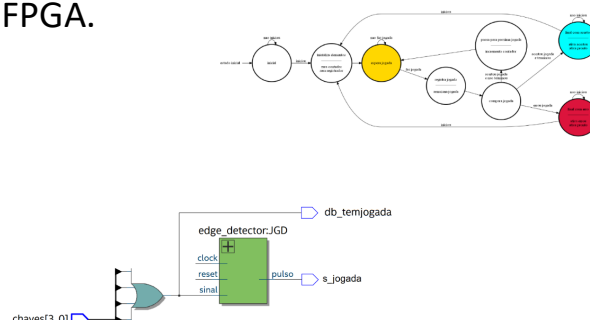
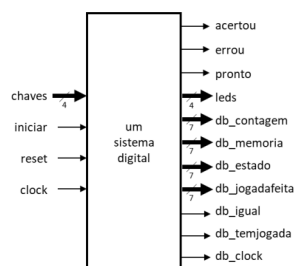
LabDig / ETM - 2025

1

## Experiência 4

### • Objetivos:

- Projeto de circuitos digitais com sinal periódico como *clock*;
- Interface de circuitos digitais com elementos externos de entrada de dados;
- Síntese em uma placa FPGA.



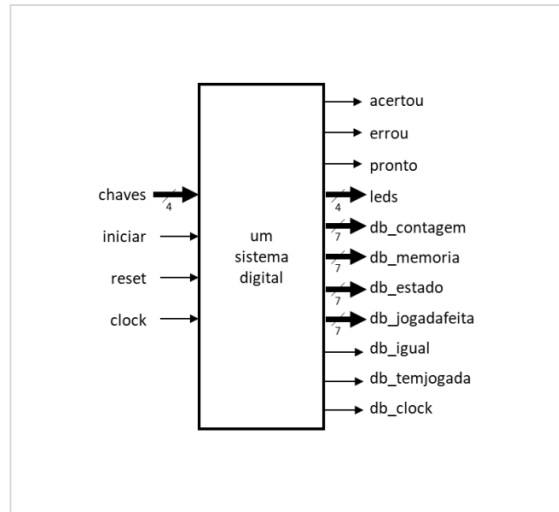
LabDig / ETM - 2025

2

# Experiência 4

- Especificação do Projeto da Experiência:

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



# Experiência 4

- Desenvolvimento do Projeto da Experiência:

1. Sistema Digital = Fluxo de Dados + Unidade de Controle
2. Circuito baseado no projeto da Atividade 3 da aula passada
3. Fluxo de Dados = introduzir interface com sinais externos
4. Unidade de Controle = máquina de estados ajustada para tratar sinais externos e sinal de clock periódico

# Atividade 1

## Experiência 4

- Contexto do projeto: <https://youtu.be/XwF9WDp-i-g>



Fonte: Youtube (duração: 2:49 min)

Brinquedo: Genius  
Modo #1 - O Desafio de Genius



# Experiência 4

## • Atividade 1 – Projeto Lógico

### Descrição de funcionamento:

1. Um conjunto de 16 dados de 4 bits 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.
2. Depois de iniciar seu funcionamento, o circuito deve inicialmente aguardar o acionamento de uma das chaves de entrada. A ocorrência desse acionamento deve ser indicada pela saída de depuração `db_temjogada`.
3. Em seguida, deve-se comparar o conteúdo das chaves de entrada (sinal `chaves`) 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 endereçamento da memória no próximo dado da memória. 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 da Unidade de Controle vigente em determinado instante do funcionamento do sistema digital. Essas três saídas (`db_contagem`, `db_memoria` e `db_estado`) devem ser projetadas para serem exibidas em *displays* de sete segmentos.
4. O ciclo de 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 interrompido, e o sinal de saída `errou` deve ser ativado. Ao final da operação, o sinal de saída `pronto` também deve ser ativado.
5. Depois disso, o circuito deve aguardar o próximo acionamento de iniciar. Essas saídas devem permanecer ativadas até o reinício da operação.

# Experiência 4

## • Atividade 1 – Projeto Lógico

### Pseudocódigo:

**Algoritmo:** sistema digital simples modificado

**entradas:** *iniciar*, *chaves*

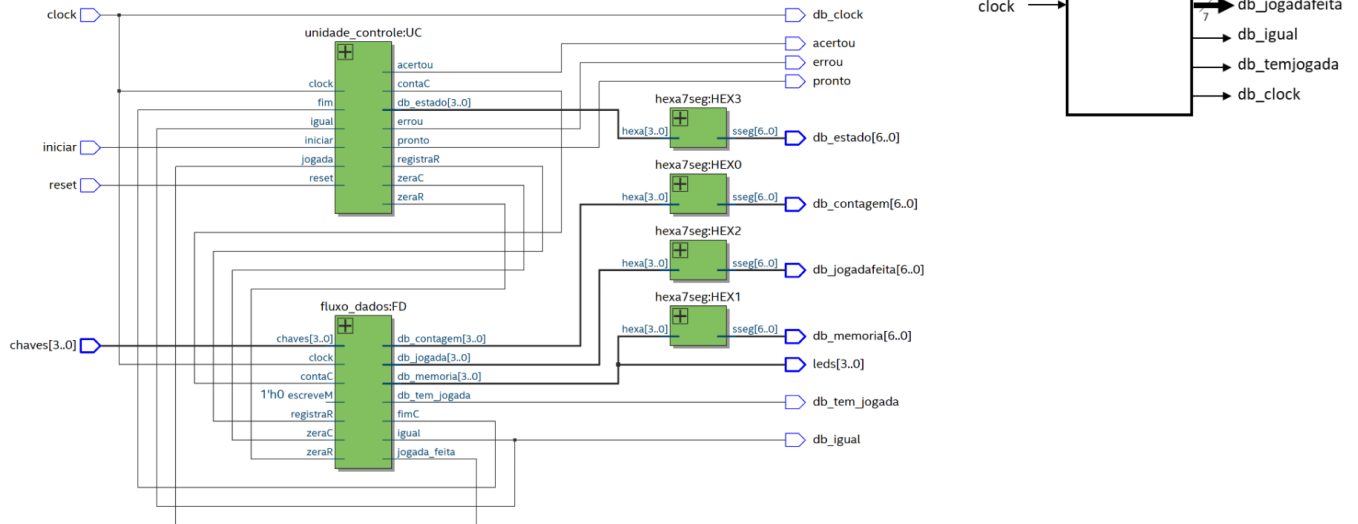
**saídas:** *acertou*, *errou*, *pronto*, *leds*

**depuração:** *contagem*, *memória*, *estado*, *jogadafeita*, *igual*, *temjogada*

```
1. {
2.   while (verdadeiro) {
3.     espera acionamento do sinal iniciar
4.     inicia circuito com condições iniciais
5.     while (acertou dado e não atingiu o último dado) {
6.       espera acionamento das chaves (jogada)
7.       registra entrada da jogada
8.       compara jogada realizada com dados armazenados
9.       incrementa contador interno
10.    }
11.    ativa acertou se acertou todos os dados da memória
12.    ativa errou se errou um dado
13.    ativa saída pronto
14.  }
15. }
```

# Experiência 4

## • Atividade 1 – Projeto Lógico



# Experiência 4

## • Atividade 1 – Projeto Lógico

### Desenvolvimento da Unidade de Controle

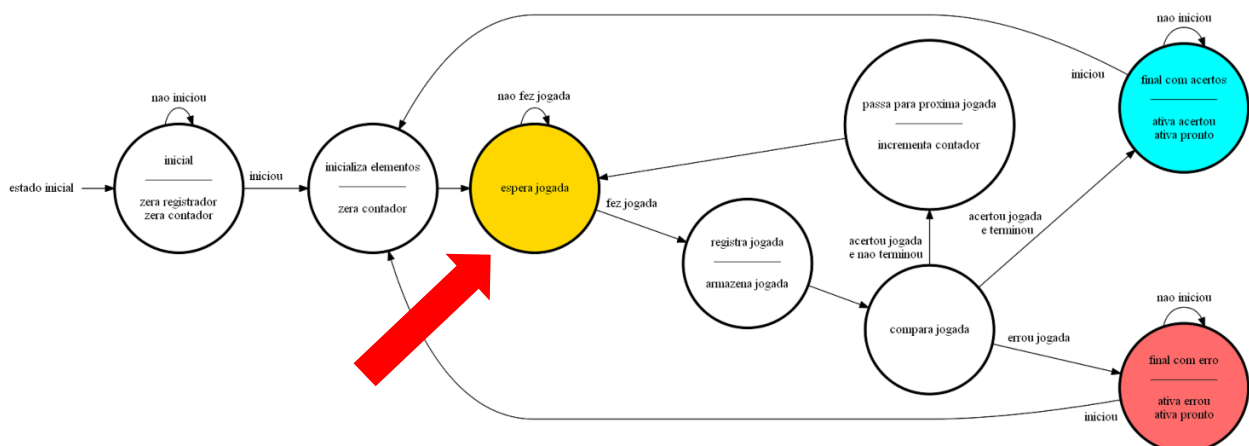
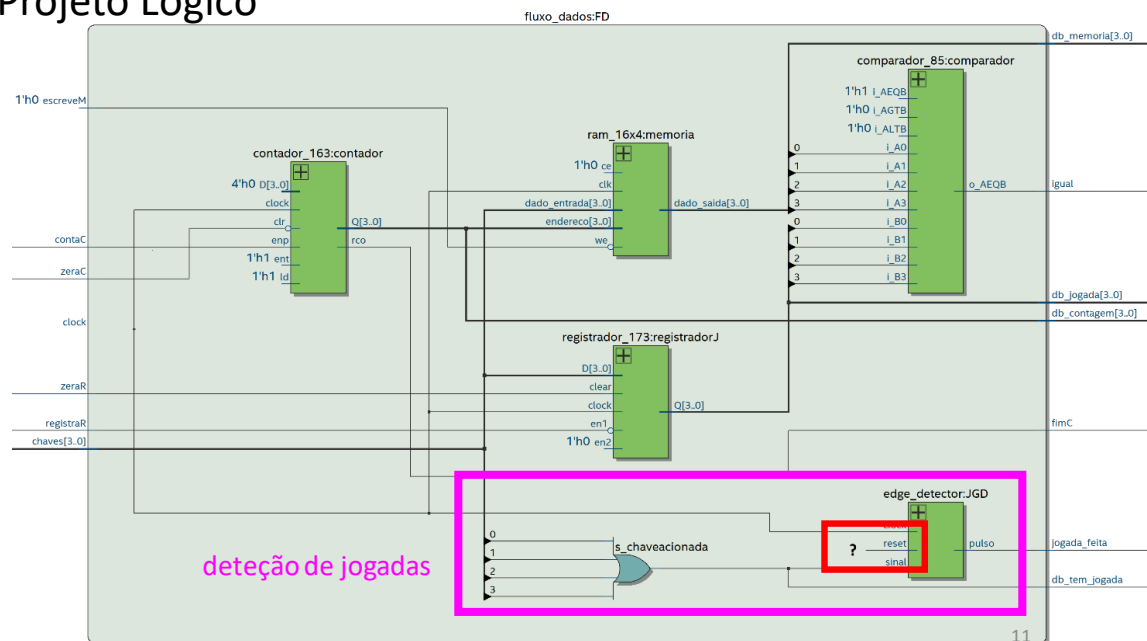


Diagrama de transição de alto nível

# Experiência 4

## • Atividade 1 – Projeto Lógico

Desenvolvimento do  
Fluxo de dados

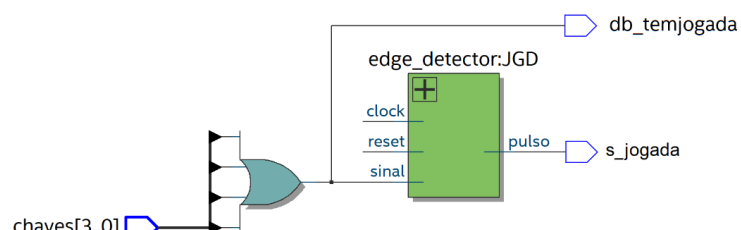


LabDig / ETM - 2025

# Experiência 4

## • Atividade 1 – Projeto Lógico

Desenvolvimento do Fluxo de dados



LabDig / ETM - 2025

# Experiência 4

## • Atividade 1 – Projeto Lógico (verificação de funcionamento)

### Plano de Teste

Cenário #1 – Acerto das 16 jogadas				
#	Operação	Sinais de entrada	Resultado esperado	Resultado observado
c.i.	Condições Iniciais			
1	"Resetar" circuito	acionar reset		
2	Aguardar alguns segundos			
3	Acionar sinal iniciar	acionar iniciar		
4	Acionar primeira entrada (jogada 1)	acionar chave(0)		
5	Acionar segunda entrada (jogada 2)	acionar chave(1)		
...	...	...	...	...
19	Acionar última entrada (jogada 16)	acionar chave(2)	saídas pronto e acertou ativadas	
Cenário #2 – Acerto das 3 primeiras jogadas e erro na 4ª jogada				
#	Operação	Sinais de entrada	Resultado esperado	Resultado observado
c.i.	Condições Iniciais			
1	"Resetar" circuito	acionar reset		
2	Aguardar alguns segundos			
3	Acionar sinal iniciar	acionar iniciar		
4	Acionar primeira entrada (jogada 1)	acionar chave(0)		
5	Acionar segunda entrada (jogada 2)	acionar chave(1)		
6	Acionar terceira entrada (jogada 3)	acionar chave(2)		
7	Acionar quarta entrada errada (jogada 4)	acionar chave(0)	saídas pronto e errou ativadas	

LabDig / ETM - 2025

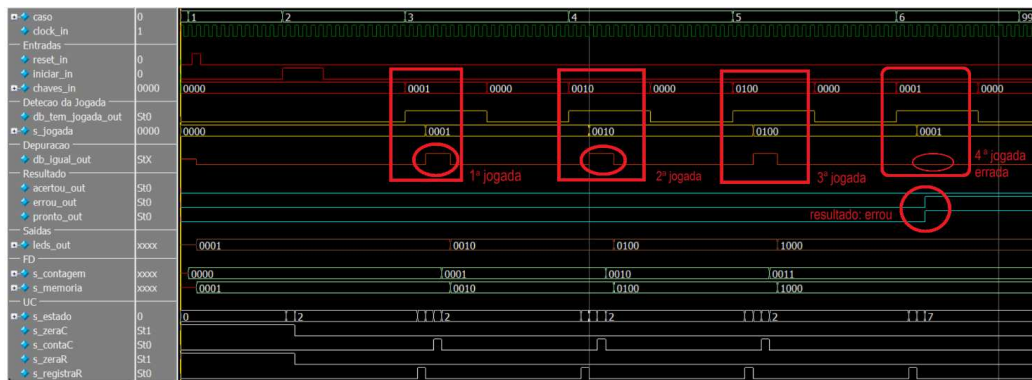
13

# Experiência 4

## • Atividade 1 – Projeto Lógico (verificação com ModelSim)

### Simulações com ModelSim

Fazer anotações nas formas de onda



LabDig / ETM - 2025

14

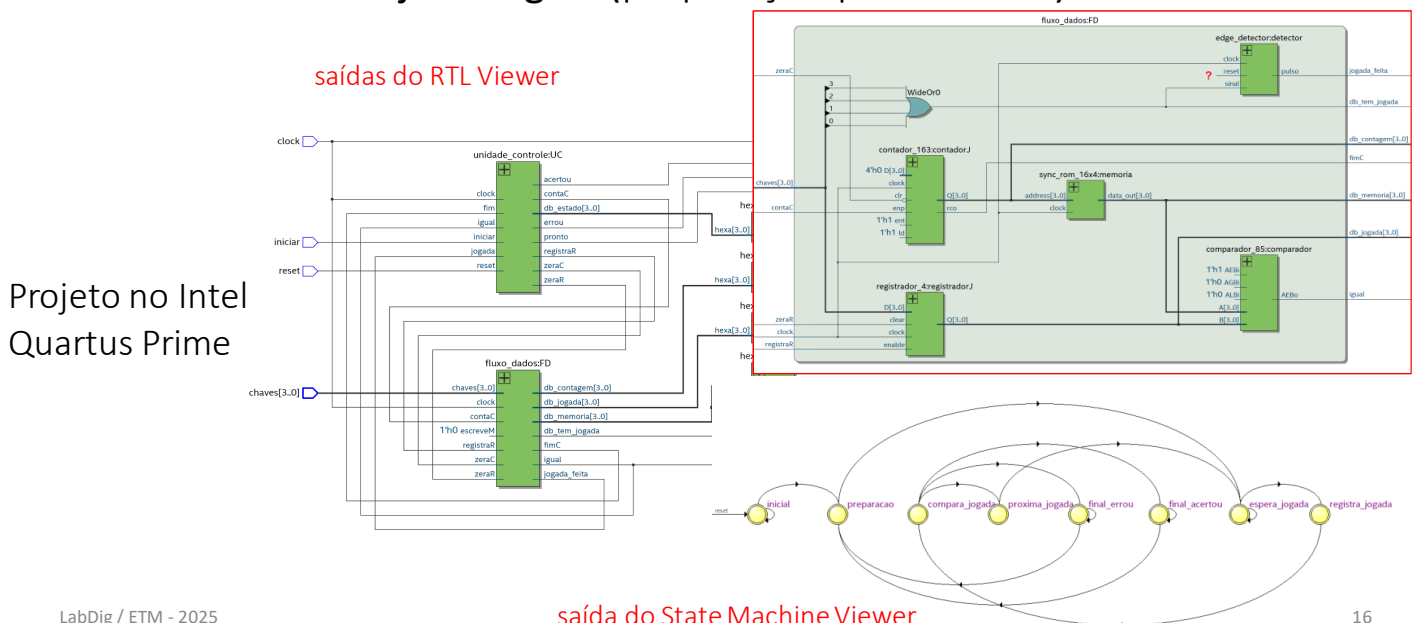
## Experiência 4

- **Atividade 1 – Projeto Lógico (preparação para síntese)**

- Projeto no Intel Quartus Prime
- Análise com RTL Viewer
- Análise com State Machine Viewer
- Designação de Pinos

## Experiência 4

- Atividade 1 – Projeto Lógico (preparação para síntese)





# Experiência 4

## • Atividade 1 – Projeto Lógico (preparação para síntese)

- Projeto no Intel Quartus Prime
- Análise com RTL Viewer
- Análise com State Machine Viewer
- Designação de Pinos

Tabela 2: Designação de Pinos para a Atividade 2

Sinal	Pino na Placa DE0-CV	Pino na FPGA	Analog Discovery
CLOCK	GPIO_0_D0		StaticIO – LED – DIO0 e Patterns – Clock – 1 kHz
RESET	GPIO_0_D1		StaticIO – Button 0/1 – DIO1
INICIAR	chave SW0		-
CHAVES(0)	GPIO_0_D11		StaticIO – Button 0/1 – DIO2
CHAVES(1)	GPIO_0_D13		StaticIO – Button 0/1 – DIO3
CHAVES(2)	GPIO_0_D15		StaticIO – Button 0/1 – DIO4
CHAVES(3)	GPIO_0_D17		StaticIO – Button 0/1 – DIO5
LEDS(0)	Led LEDR0		-
LEDS(1)	Led LEDR1		-
LEDS(2)	Led LEDR2		-
LEDS(3)	Led LEDR3		-
PRONTO	Led LEDR4		-
ACERTO	Led LEDR5		-
ERRO	Led LEDR6		-
DB_IGUAL	Led LEDR7		-
DB_CLOCK	Led LEDR8		-
DB_TEM_JOGADA	Led LEDR9		-
DB_CONTAGEM	Display HEX0		-
DB_MEMORIA	Display HEX1		-
DB_JOGADAFEITA	Display HEX2		-
DB_ESTADO	Display HEX5		-

## Atividade 2

## Experiência 4

### • Atividade 2 – Síntese do circuito

Designação de pinos

Sinais vindos do  
Analog Discovery

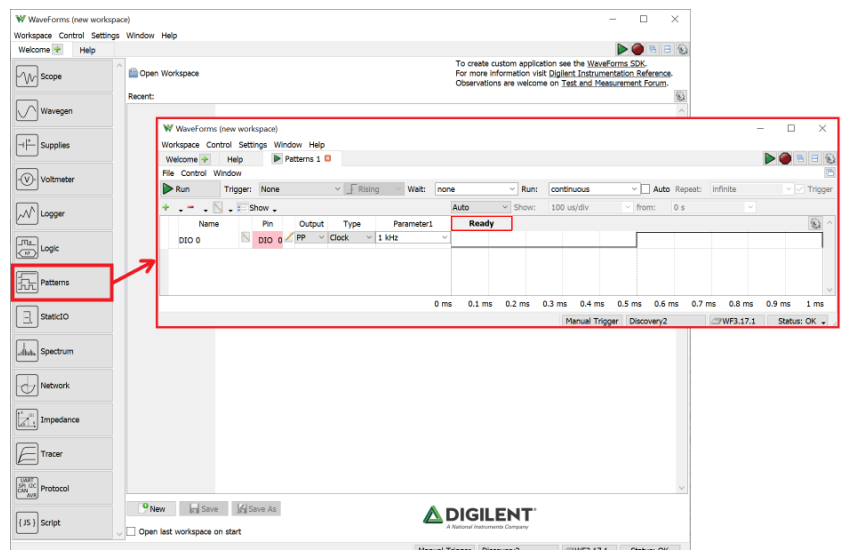
Sinal	Pino na Placa DE0-CV	Pino na FPGA	Analog Discovery
CLOCK	GPIO_0_D0		StaticIO – LED – DIO0 ou Patterns – Clock – 1 kHz
RESET	GPIO_0_D1		StaticIO – Button 0/1 – DIO1
INICIAR	chave SW0		-
CHAVES(0)	GPIO_0_D11		StaticIO – Button 0/1 – DIO2
CHAVES(1)	GPIO_0_D13		StaticIO – Button 0/1 – DIO3
CHAVES(2)	GPIO_0_D15		StaticIO – Button 0/1 – DIO4
CHAVES(3)	GPIO_0_D17		StaticIO – Button 0/1 – DIO5
LEDS(0)	Led LEDR0		-
LEDS(1)	Led LEDR1		-
LEDS(2)	Led LEDR2		-
LEDS(3)	Led LEDR3		-
PRONTO	Led LEDR4		-
ACERTO	Led LEDR5		-
ERRO	Led LEDR6		-
DB_IGUAL	Led LEDR7		-
DB_CLOCK	Led LEDR8		-
DB_TEM_JOGADA	Led LEDR9		-
DB_CONTAGEM	Display HEX0		-
DB_MEMORIA	Display HEX1		-
DB_JOGADAFEITA	Display HEX2		-
DB_ESTADO	Display HEX5		-

## Experiência 4

### • Atividade 2 – Implementação do Circuito

Testes na placa FPGA (**DICAS**)

- ❑ Ferramenta **Patterns** do Waveforms (Analog Discovery)
  1. Adicionar sinal DIO0
  2. Configurar como tipo Clock
  3. Configurar parâmetro para 1KHz



# Experiência 4

## • 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 de códigos  
para db\_estado

LabDig / ETM - 2025

Cenário #1 – Acerto das 16 jogadas				
#	Operação	Sinais de entrada	Resultado esperado	Resultado observado
c.i.	Condições Iniciais			
1	"Resetar" circuito	acionar reset		
2	Aguardar alguns segundos			
3	Acionar sinal iniciar	acionar iniciar		
4	Acionar primeira entrada (jogada 1)	acionar chave(0)		
5	Acionar segunda entrada (jogada 2)	acionar chave(1)		
...	...	...	...	...
19	Acionar última entrada (jogada 16)	acionar chave(2)	saídas pronto e acertou ativadas	

Cenário #2 – Acerto das 3 primeiras jogadas e erro na 4ª jogada				
#	Operação	Sinais de entrada	Resultado esperado	Resultado observado
c.i.	Condições Iniciais			
1	"Resetar" circuito	acionar reset		
2	Aguardar alguns segundos			
3	Acionar sinal iniciar	acionar iniciar		
4	Acionar primeira entrada (jogada 1)	acionar chave(0)		
5	Acionar segunda entrada (jogada 2)	acionar chave(1)		
6	Acionar terceira entrada (jogada 3)	acionar chave(2)		
7	Acionar quarta entrada errada (jogada 4)	acionar chave(0)	saídas pronto e errou ativadas	

21

# Experiência 4

## • Atividade 2 – Implementação do Circuito

### Testes na placa FPGA (**DICAS**)

1. Na aplicação do Plano de Testes, acompanhar a sequência de sinais de saída e sinais de depuração (p.ex. db\_clock, db\_estado).
2. Sinal de *clock* (canal digital **DIO0** do Analog Discovery):
  - Alterar frequência => 1KHz (operação do circuito), 1Hz (testes) (ferramenta **Patterns** do Waveforms), ou
  - Em caso de erro, na depuração usar *clock* manual => **DIO0 como botão** (ferramenta **StaticIO** do Waveforms)
    - lembrar de fechar ou parar ferramenta **Patterns**.

LabDig / ETM - 2025

22

# Atividade 3

## Experiência 4

### • Atividade 3 – Desafio: Modificação ao Sistema Digital

1. Especificação a ser apresentada pelo professor.
2. Projetar a modificação e plano de testes.
3. Sintetizar o projeto no placa FPGA.
4. Testar o funcionamento do circuito.
5. Documentar projeto e funcionamento no Relatório.

DICA: usar componente  
`contador_m`  
(parâmetro *generic*)