

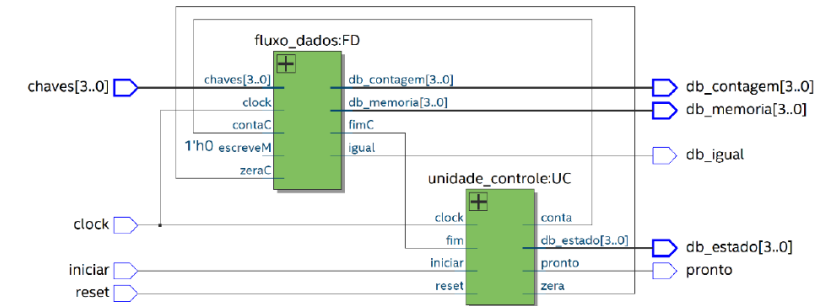
Experiência 3 – Projeto de uma Unidade de Controle

PCS3635 – Laboratório de
Projeto de Sistemas Digitais I
1º Quadrimestre de 2026

Experiência 3

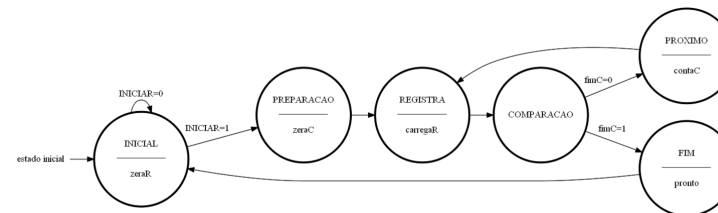
- Objetivos

- Projeto hierárquico de circuitos com fluxo de dados e unidade de controle;
- Projeto da unidade de controle de um circuito digital;
- Codificação de uma máquina de estados em Verilog;
- Síntese em uma placa FPGA



Algoritmo: sistema digital simples
entradas: iniciar, chaves
saídas: pronto
depuração: contagem, memória, estado, igual

```
1. {  
2.   while (verdadeiro) {  
3.     espera acionamento do sinal INICIAR  
4.     inicia circuito para condições iniciais  
5.     enquanto não atingir o último dado {  
6.       compara chaves de entrada com dados armazenados  
7.       incrementa contador interno  
8.     }  
9.     ativa saída PRONTO  
10.  }  
11. }
```

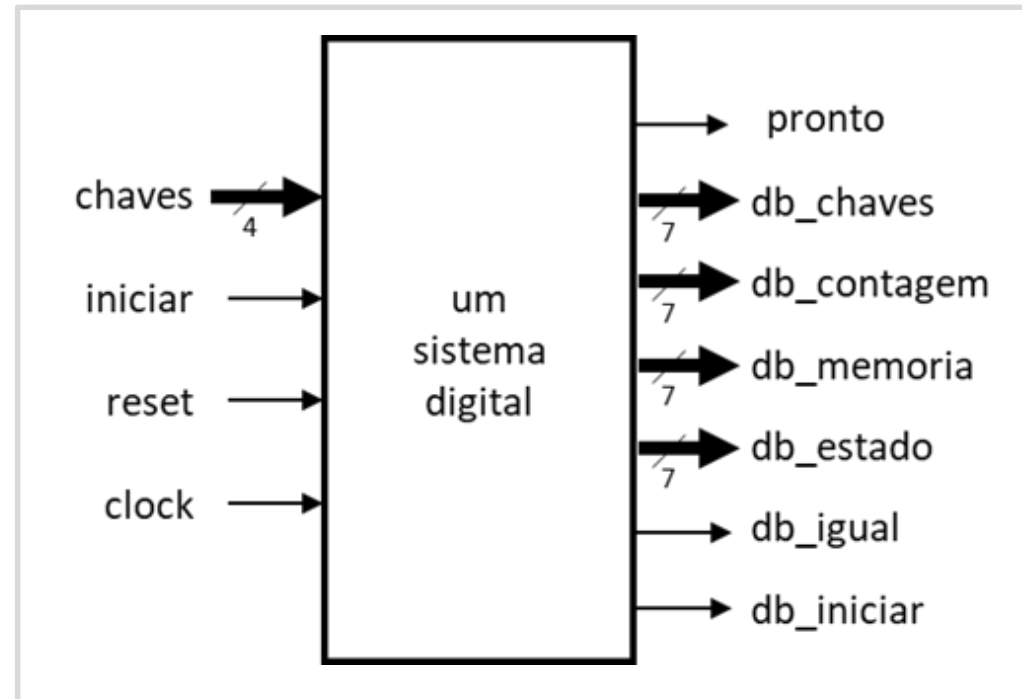


```
17 architecture fsm of unidade_controle is  
18   type t_estado is (A, B, C, D, E);  
19   signal E_atual, E_prox: t_estado;  
20 begin  
21   -- memoria de estado  
22   process (clock, reset)  
23   begin  
24     if reset='1' then  
25       E_atual <= A;  
26     elsif clock'event and clock = '1' then  
27       E_atual <= E_prox;  
28     end if;  
29   end process;  
30  
31   -- logica de proximo estado  
32   E_prox <=  
33     A when E_atual=A and iniciar='0' else  
34     B when E_atual=A and iniciar='1' else  
35     C when E_atual=B else  
36     D when E_atual=C and fim='0' else  
37     E when E_atual=C and fim='1' else  
38     C when E_atual=D else  
39     A when E_atual=E else  
40     A;  
41  
42   -- logica de saida (maquina de Moore)  
43   with E_atual select  
44     zera <= '0' when A | C | D | E,  
45           '1' when B,  
46           '0' when others;
```

Experiência 3

- Especificação do Projeto da Experiência:

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



Experiência 3

- Desenvolvimento do Projeto da Experiência:
 1. Sistema Digital = Fluxo de Dados + Unidade de Controle
 2. Fluxo de Dados baseado no projeto da aula passada
 3. Unidade de Controle = máquina de estados
 4. Modificação proposta (Desafio)

Experiência 3

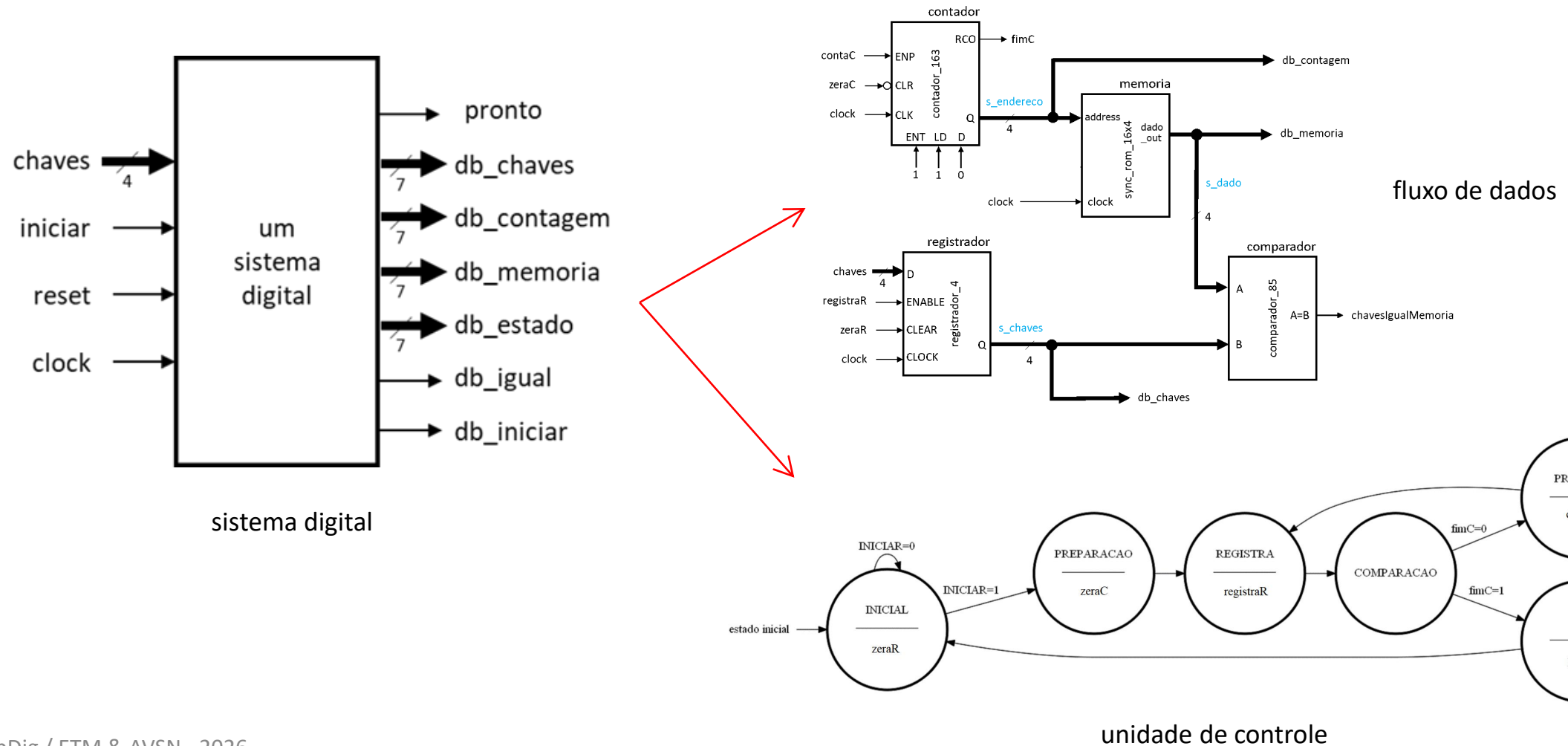
- Projeto Lógico de um Sistema Digital

Descrição de funcionamento:

1. 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.
2. Depois do acionamento do sinal `reset`, o circuito deve aguardar o início de sua operação até o acionamento do sinal de entrada `iniciar` – replicado na saída de depuração `db_inicial`.
3. Depois de iniciar seu funcionamento, o circuito deve armazenar o valor das chaves de entrada (sinal `chaves`) e depois comparar o conteúdo armazenado das chaves com o respectivo dado da memória.
4. A saída de depuração `db_igual` deve indicar o resultado da comparação.
5. Em seguida, o contador interno deve ser incrementado para posicionar o endereçamento da memória para permitir o acesso ao próximo dado da memória.
6. 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 vigente da Unidade de Controle em determinado instante do funcionamento do sistema digital .
7. O ciclo de comparação e reposicionamento da memória deve prosseguir até que todos os 16 dados sejam verificados.
8. Ao final da operação, o sinal de saída `pronto` deve ser acionado por um período de `clock`.
9. Depois disso, o circuito deve voltar para o estado inicial para aguardar a próxima ativação de `iniciar`.

Experiência 3

- Projeto Lógico de um Sistema Digital

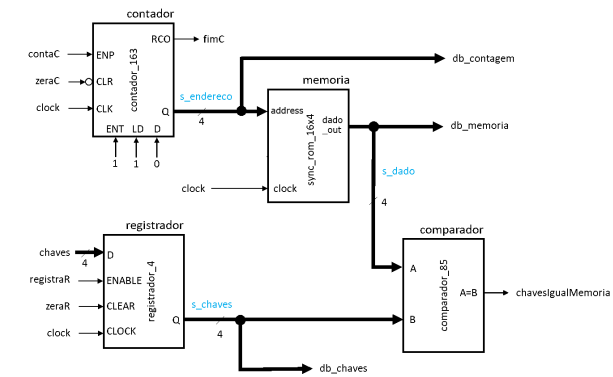


Experiência 3

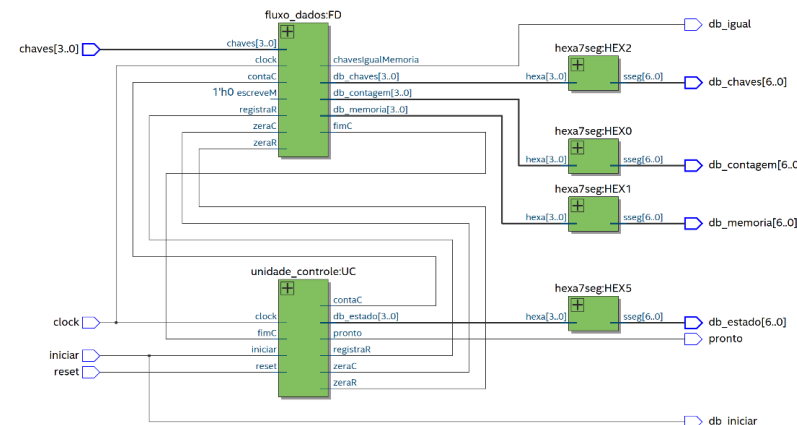
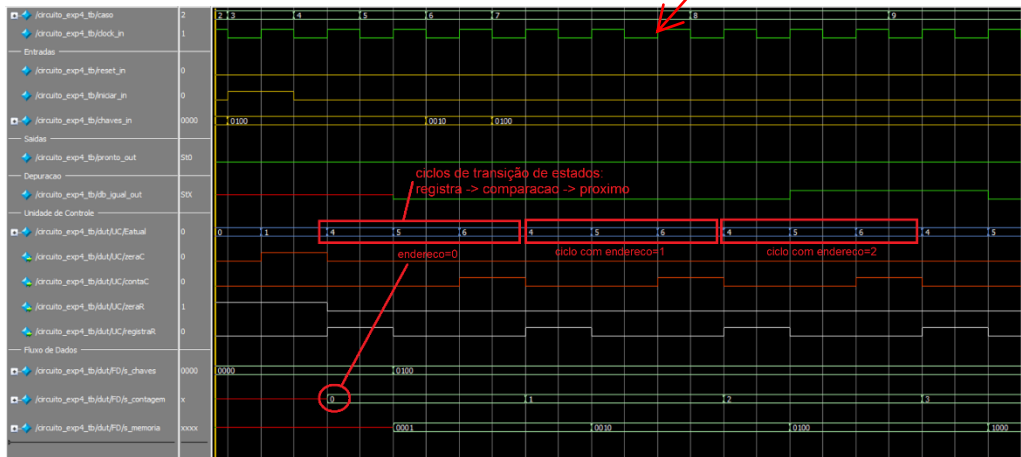
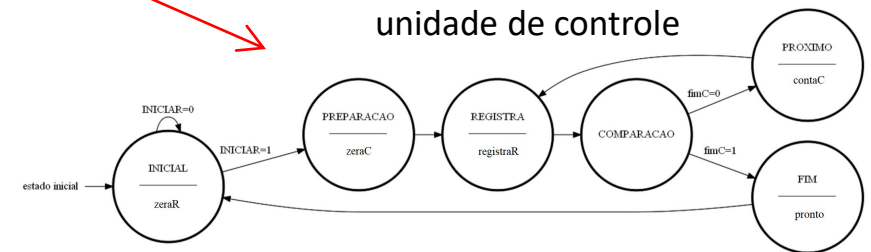
• Atividade 1 – Circuito da experiência

1. Adaptação do fluxo de dados
2. Estudo da unidade de controle
3. Projeto do circuito da experiência
4. Simulação com ModelSim

fluxo de dados



unidade de controle



sistema digital
(descrição estrutural)

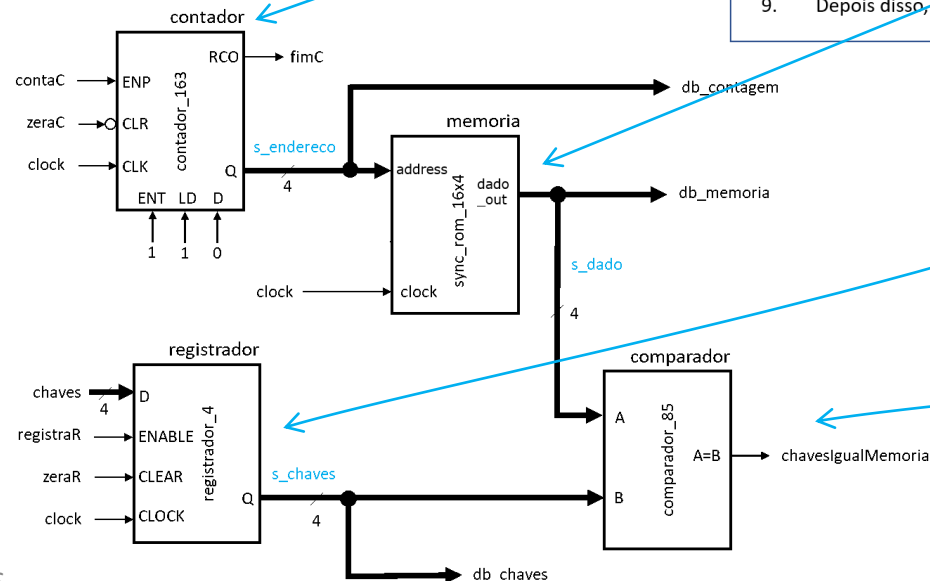
Experiência 3

• Atividade 1 – Projeto Lógico de um Sistema Digital

– Desenvolvimento do Fluxo de dados

1. 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**.
2. Depois do acionamento do sinal **reset**, o circuito deve aguardar o início de sua operação até o acionamento do sinal de entrada iniciar.
3. Depois de iniciar seu funcionamento, o **circuito deve armazenar o valor das chaves de entrada (sinal chaves)** e depois **comparar o conteúdo armazenado das chaves com o respectivo dado da memória**.
4. A saída de depuração **db_igual** deve indicar o resultado da comparação.
5. Em seguida, **o contador interno deve ser incrementado para posicionar o endereçamento da memória para permitir o acesso ao próximo dado da memória**.
6. 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 vigente da Unidade de Controle em determinado instante do funcionamento do sistema digital.
7. O ciclo de comparação e reposicionamento da memória deve prosseguir até que todos os 16 dados sejam verificados.
8. Ao final da operação, o sinal de saída pronto deve ser acionado por um período de clock.
9. Depois disso, o circuito deve voltar para o estado inicial para aguardar a próxima ativação de iniciar.

componentes
registrador_n e
sync_rom_16x4
fornecidos



Experiência 3

- Atividade 1 – Projeto Lógico de um Sistema Digital

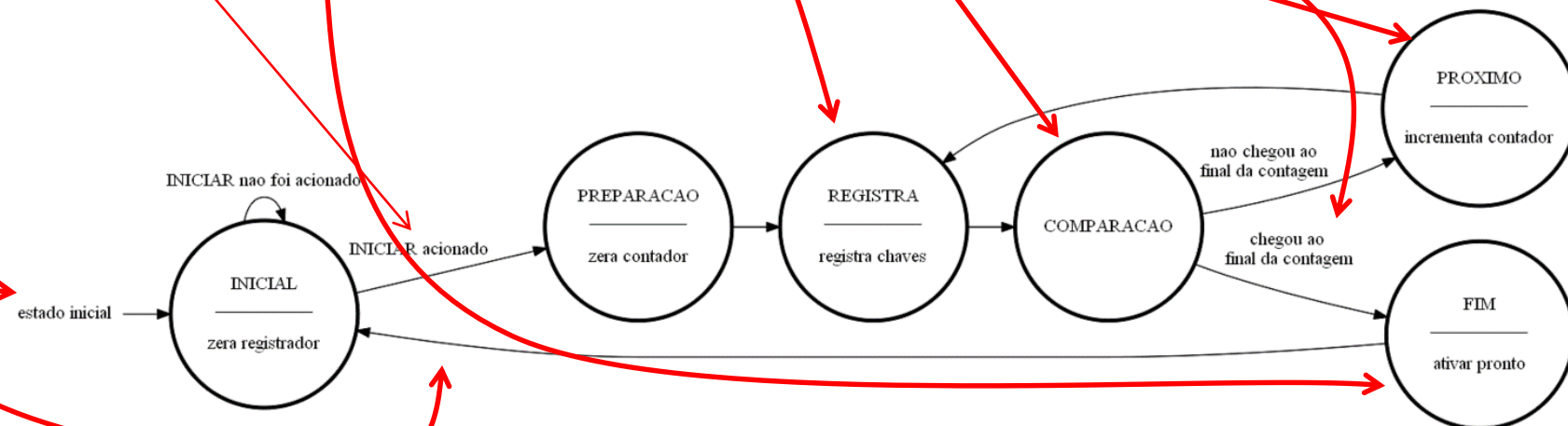
Desenvolvimento da Unidade de Controle

1. 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.
2. **Depois do acionamento do sinal `reset`, o circuito deve aguardar o início de sua operação até o acionamento do sinal de entrada `iniciar`** – replicado na saída de depuração `db_inicial`.
3. **Depois de iniciar seu funcionamento**, o circuito deve armazenar o valor das chaves de entrada (sinal `chaves`) e depois comparar o conteúdo armazenado das chaves com o respectivo dado da memória.
4. A saída de depuração `db_igual` deve indicar o resultado da comparação.
5. **Em seguida**, o contador interno deve ser incrementado para posicionar o endereçamento da memória para permitir o acesso ao próximo dado da memória.
6. 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 vigente da Unidade de Controle em determinado instante do funcionamento do sistema digital .
7. O ciclo de comparação e reposicionamento da memória **deve prosseguir até que todos os 16 dados sejam verificados**.
8. **Ao final da operação, o sinal de saída `pronto` deve ser acionado por um período de `clock`.**
9. **Depois disso, o circuito deve voltar para o estado inicial** para aguardar a próxima ativação de `iniciar`.

Experiência 3

• Atividade 1 – Projeto Lógico de um Sistema Digital

1. 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.
2. **Depois do acionamento do sinal reset, o circuito deve aguardar o início de sua operação até o acionamento do sinal de entrada iniciar** – replicado na saída de depuração db_inicial.
3. **Depois de iniciar seu funcionamento, o circuito deve armazenar o valor das chaves de entrada (sinal chaves) e depois comparar o conteúdo armazenado das chaves com o respectivo dado da memória.**
4. A saída de depuração db_igual deve indicar o resultado da comparação.
5. **Em seguida, o contador interno deve ser incrementado para posicionar o endereçamento da memória para permitir o acesso ao próximo dado da memória.**
6. 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 vigente da Unidade de Controle em determinado instante do funcionamento do sistema digital.
7. O ciclo de comparação e reposicionamento da memória **deve prosseguir até que todos os 16 dados sejam verificados.**
8. **Ao final da operação, o sinal de saída pronto deve ser acionado por um período de clock.**
9. **Depois disso, o circuito deve voltar para o estado inicial** para aguardar a próxima ativação de iniciar.



Experiência 3

- Atividade 1 – Projeto Lógico de um Sistema Digital
 - Sinais de controle e Sinais de condição para a Unidade de Controle

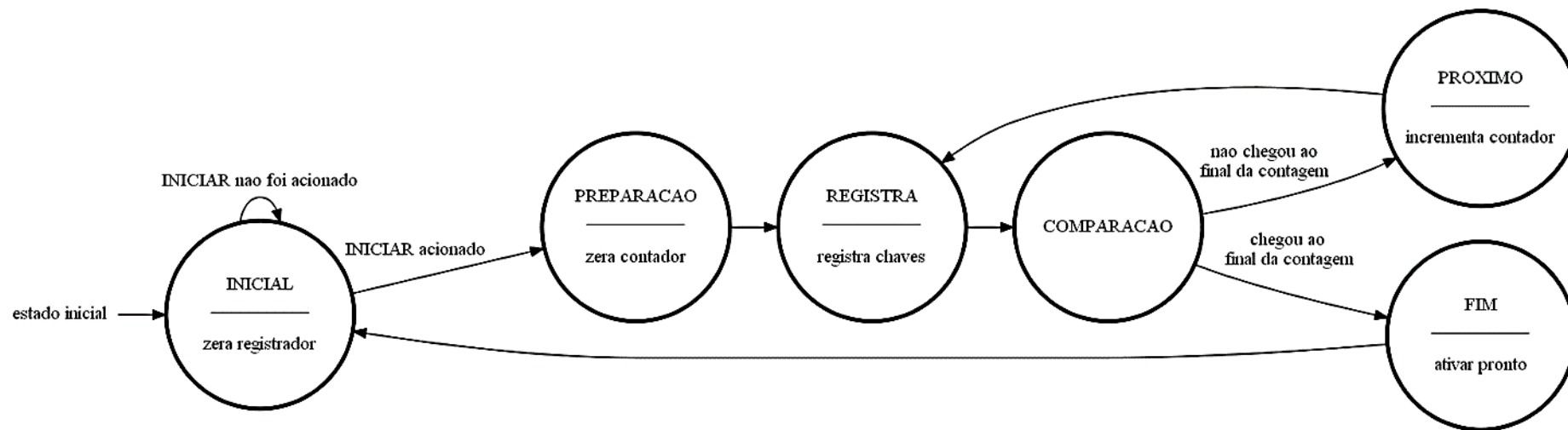


Diagrama de transição de alto nível

Experiência 3

- Atividade 1 – Projeto Lógico de um Sistema Digital
 - Sinais de controle e Sinais de condição para a Unidade de Controle

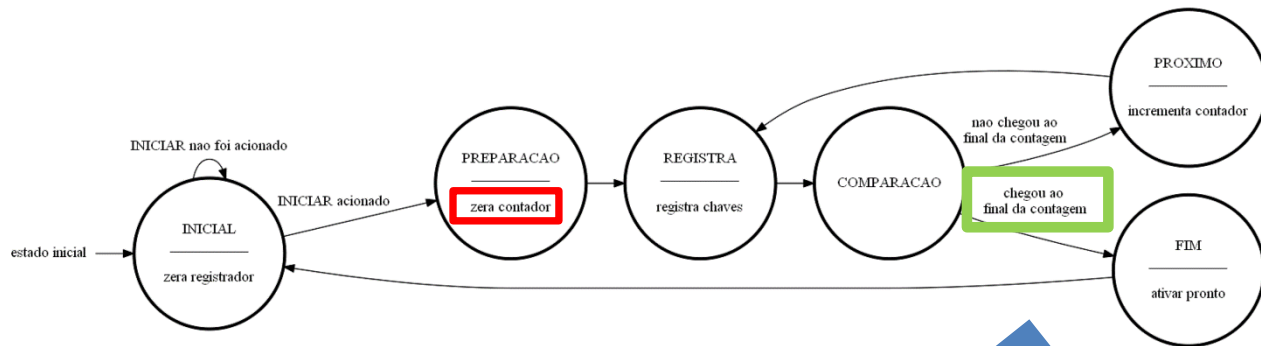
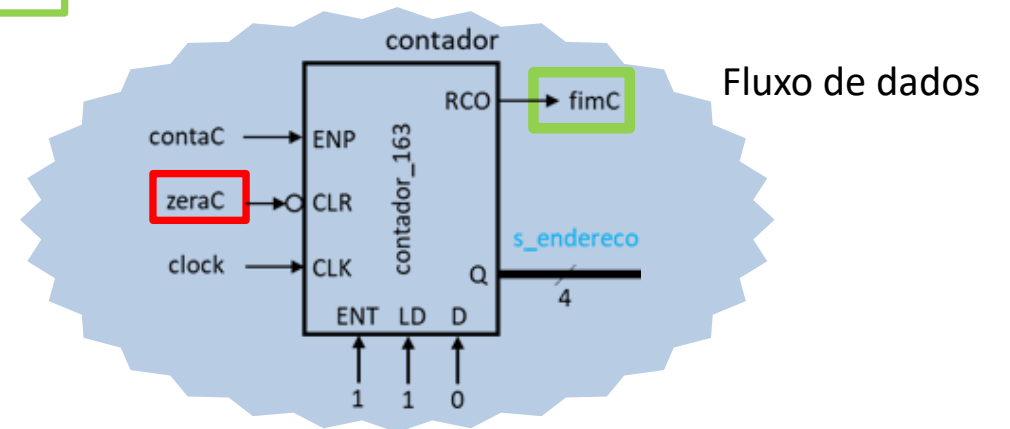


Diagrama de transição de alto nível



exemplos de sinais

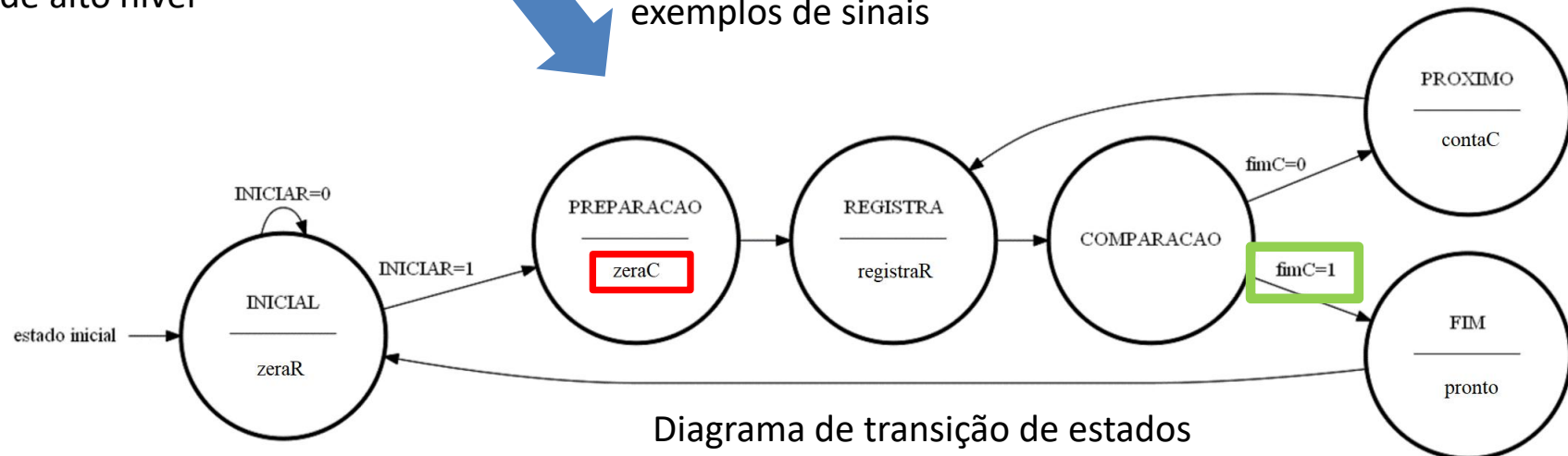
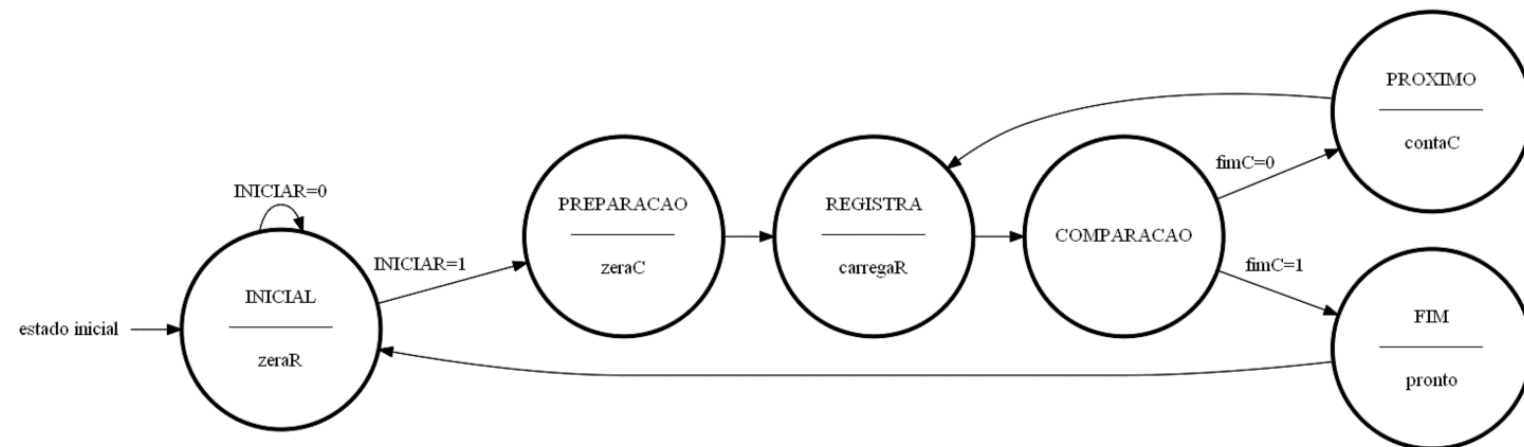


Diagrama de transição de estados

Experiência 3

- Atividade 1 – Projeto Lógico de um Sistema Digital
 - Máquinas de estado em Verilog (`exp3_unidade_controle.v` – fornecido)

Estudar a tradução
do diagrama de
transição de estados
para o código Verilog

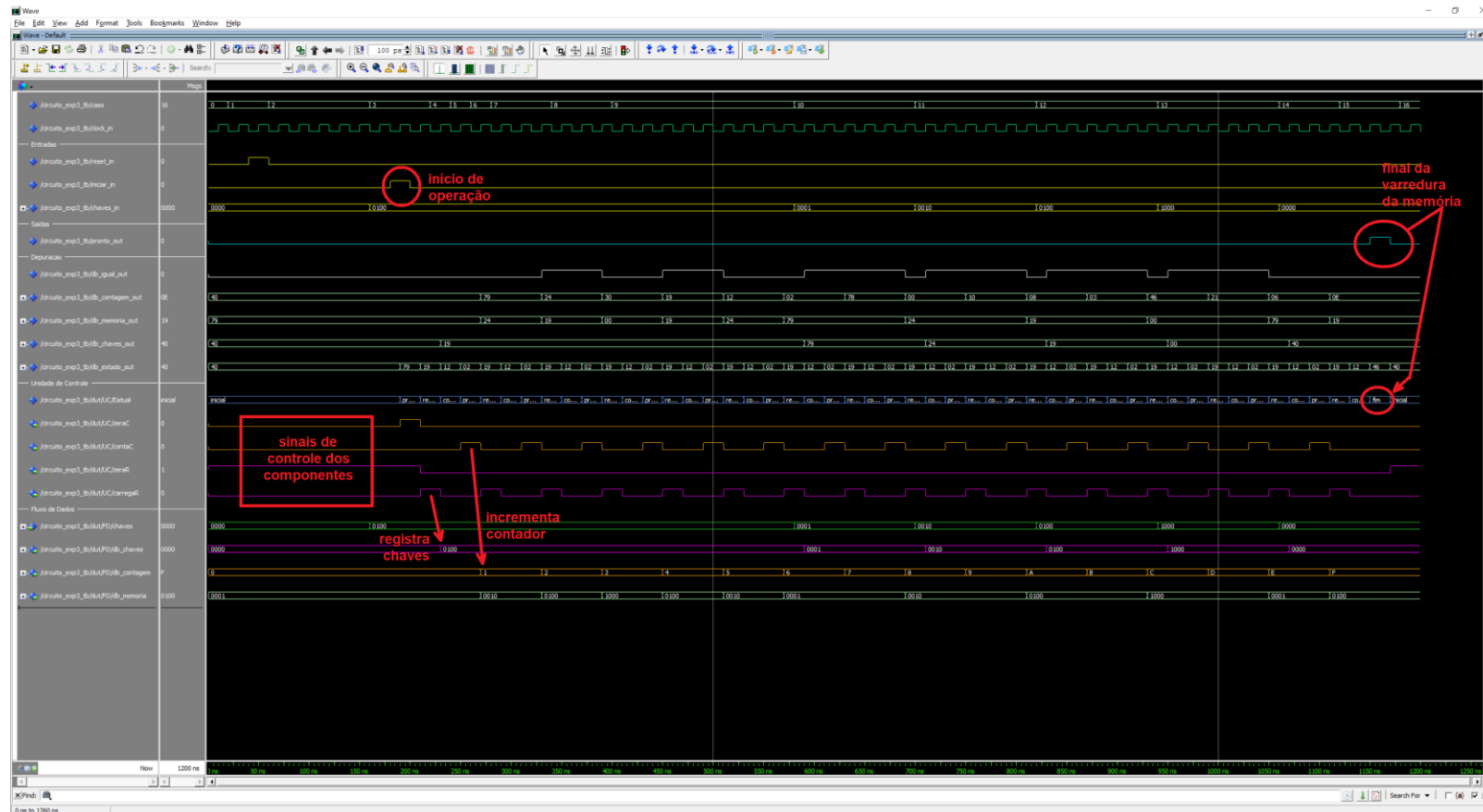


Inconsistências?

Experiência 3

- Simulação com **ModelSim** (`circuito_exp3`) com *testbench*

Executar
simulação
dos testes
contidos
no Plano
de Testes



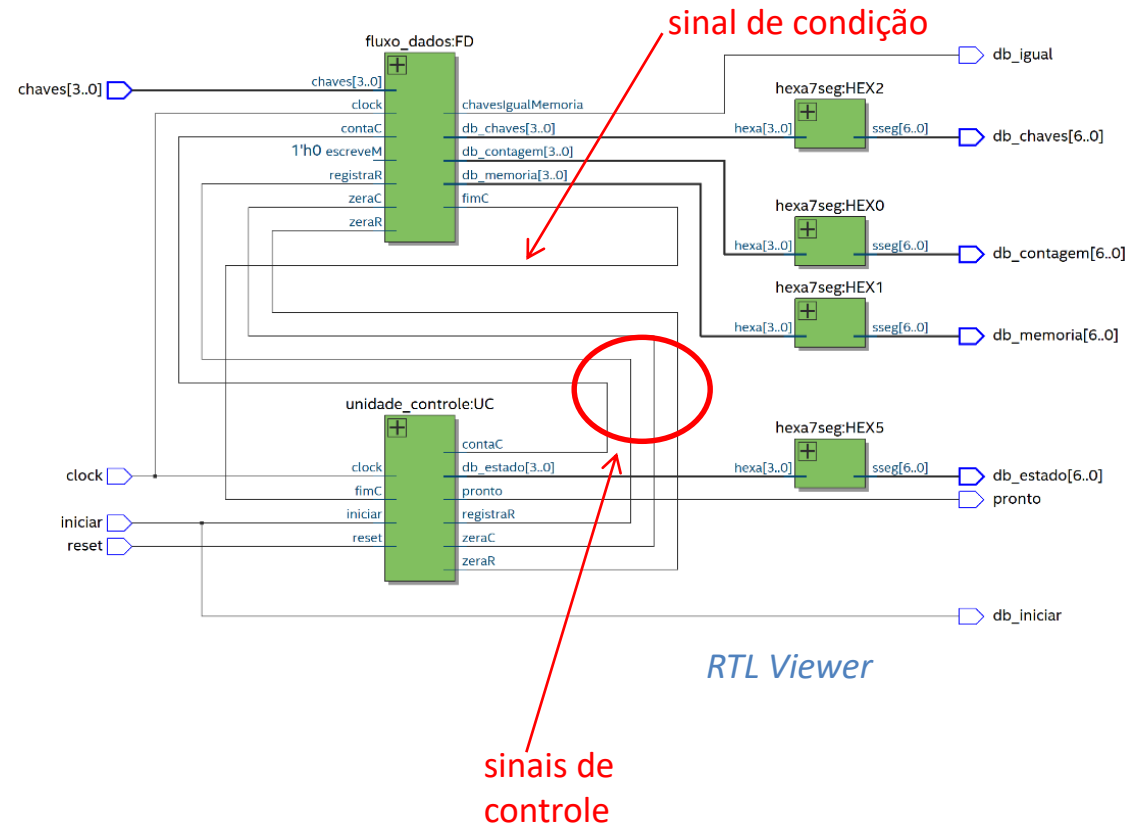
Inserir anotações nas
formas de onda no
Planejamento.

Experiência 3

- Projeto no Intel Quartus Prime

código Verilog

```
module circuito_exp3 (
    input clock,
    input reset,
    input iniciar,
    input [3:0] chaves,
    output pronto,
    output db_igual,
    output db_iniciar,
    output [6:0] db_contagem,
    output [6:0] db_memoria,
    output [6:0] db_chaves,
    output [6:0] db_estado
);
```

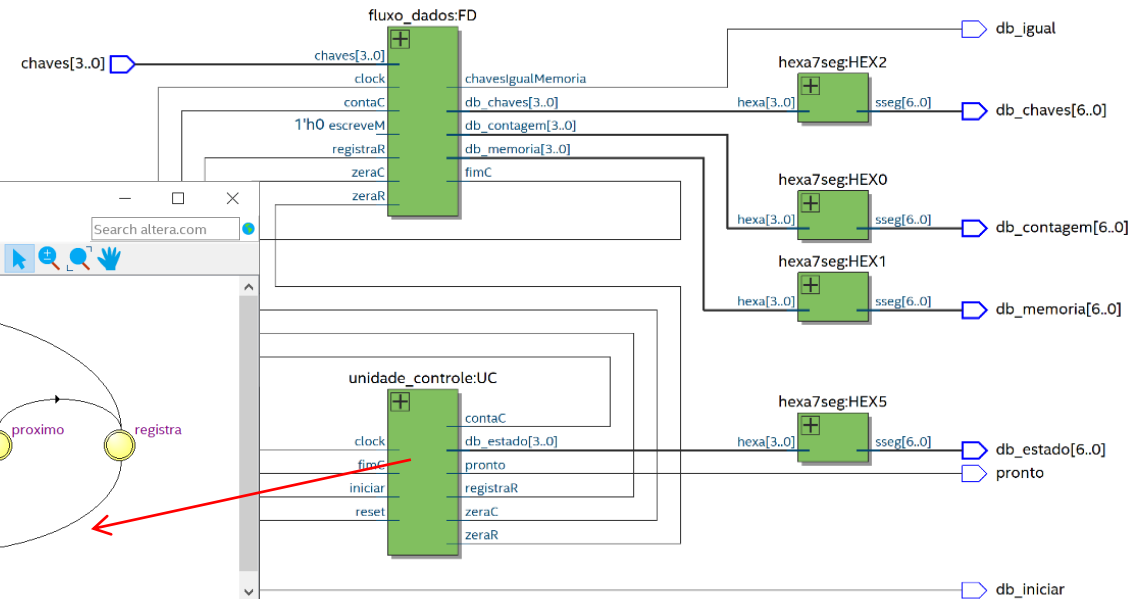
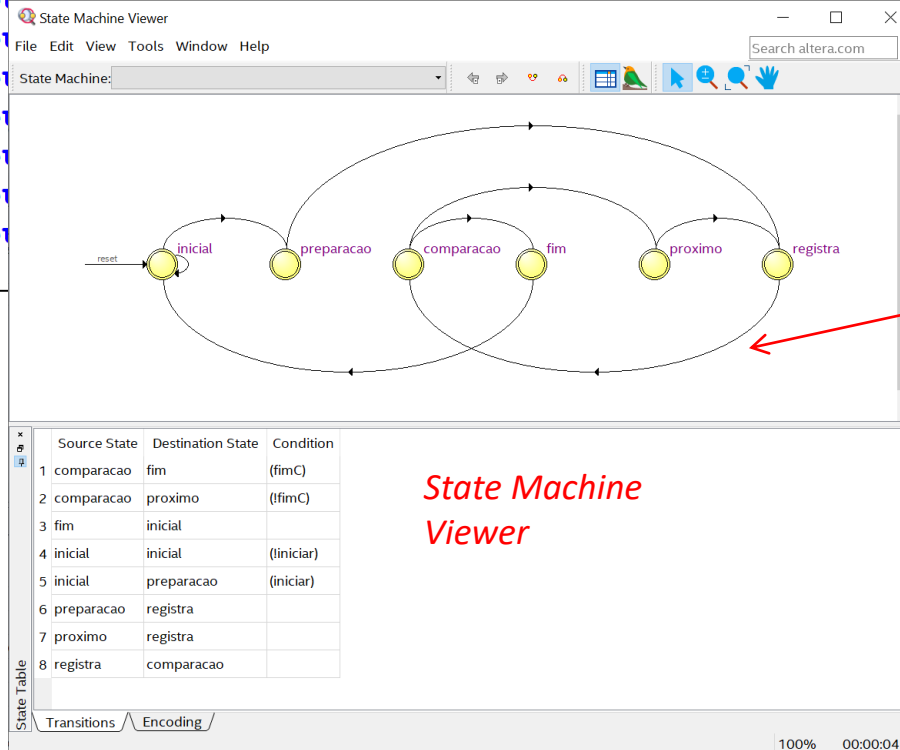


Experiência 3

- Projeto no Intel Quartus Prime

código Verilog

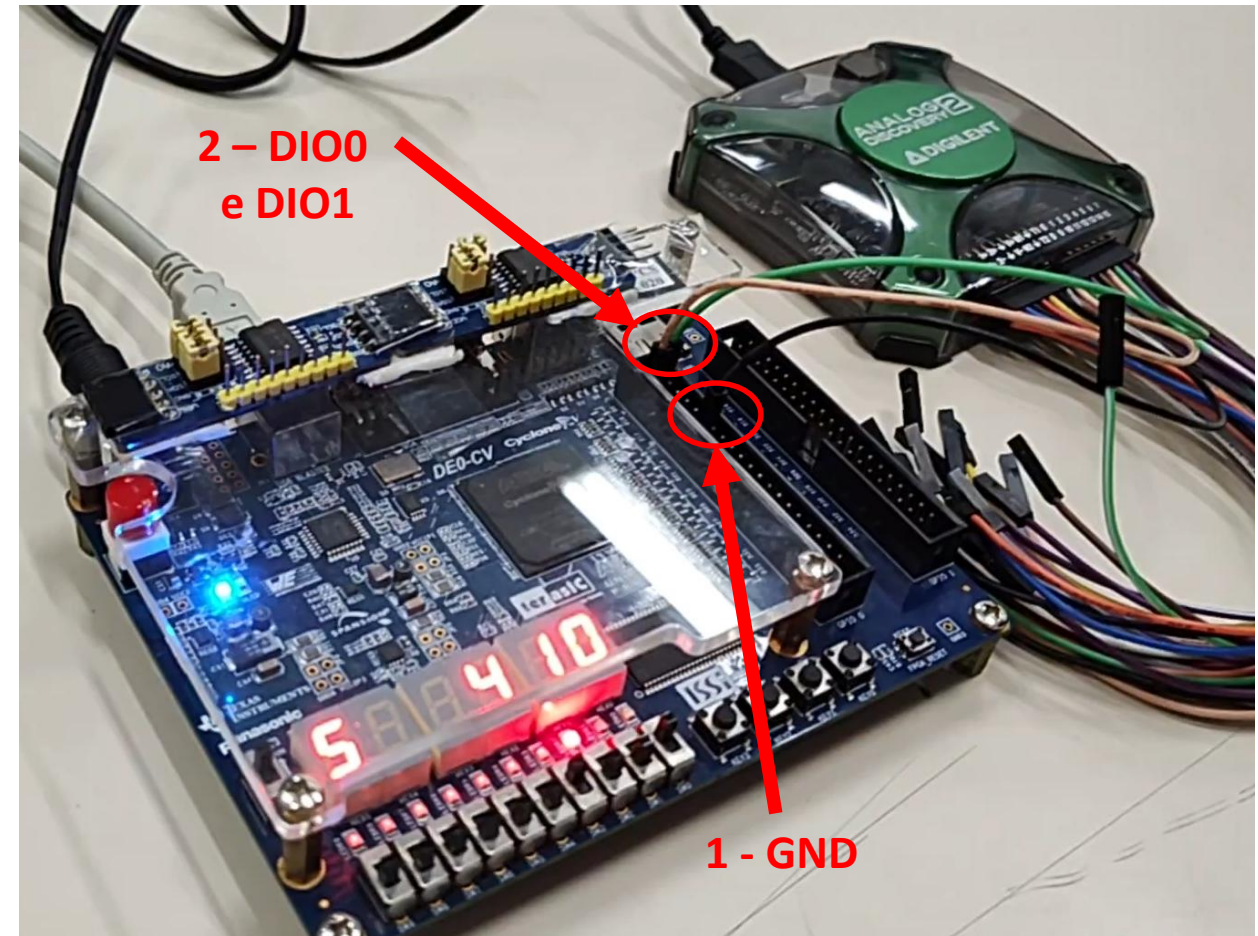
```
module circuito_exp3 (  
    input clock,  
    input reset,  
    input iniciar,  
    input [3:0] chaves,  
    output  
);
```



Experiência 3

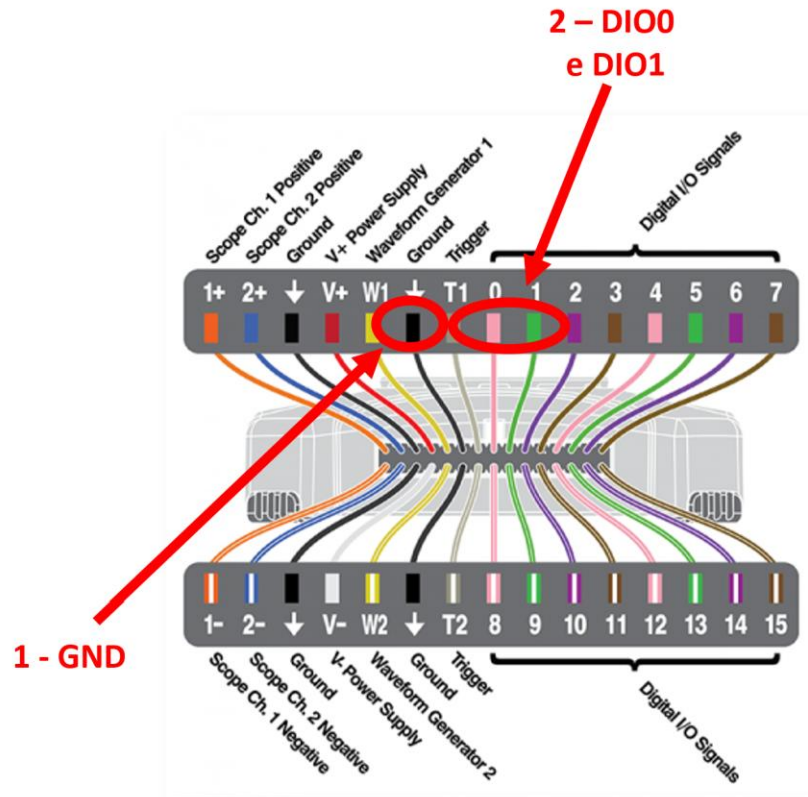
- Uso do **Analog Discovery** (ligação com a placa FPGA)

1. Ligar primeiro o fio **GND** (preto) no pino **GND** da **GPIO_0**
2. Ligação dos canais digitais **DIO0** (rosa) e **DIO1** (verde) nos pinos **GPIO_0_D0** e **GPIO_0_D1** da **GPIO_0**, respectivamente.

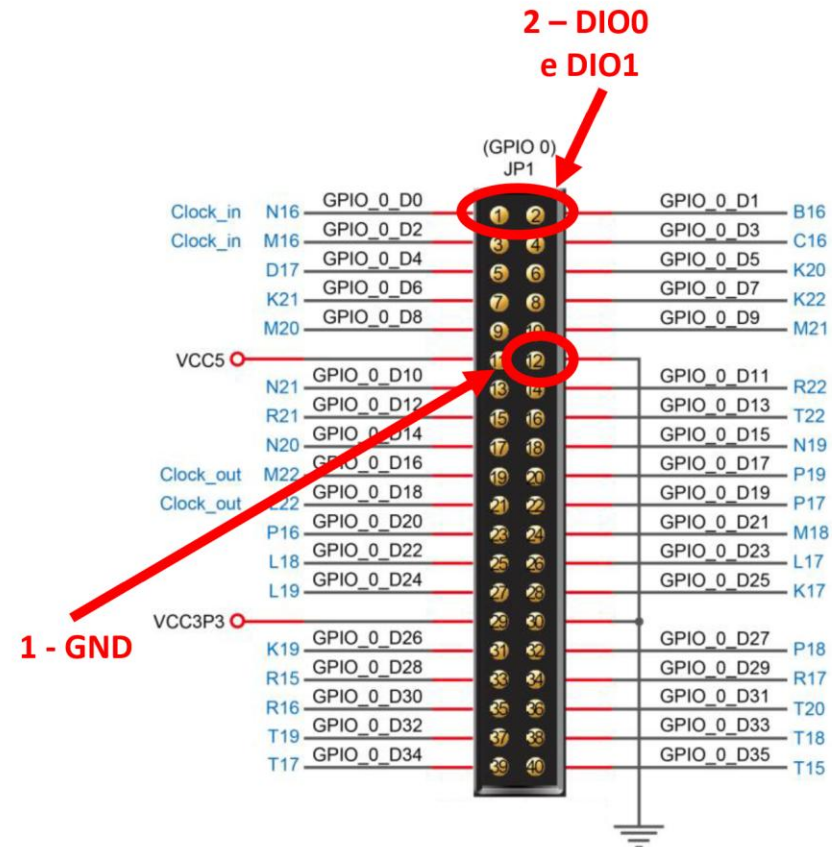


Experiência 3

- Uso do Analog Discovery (ligação com a placa FPGA)



Analog Discovery

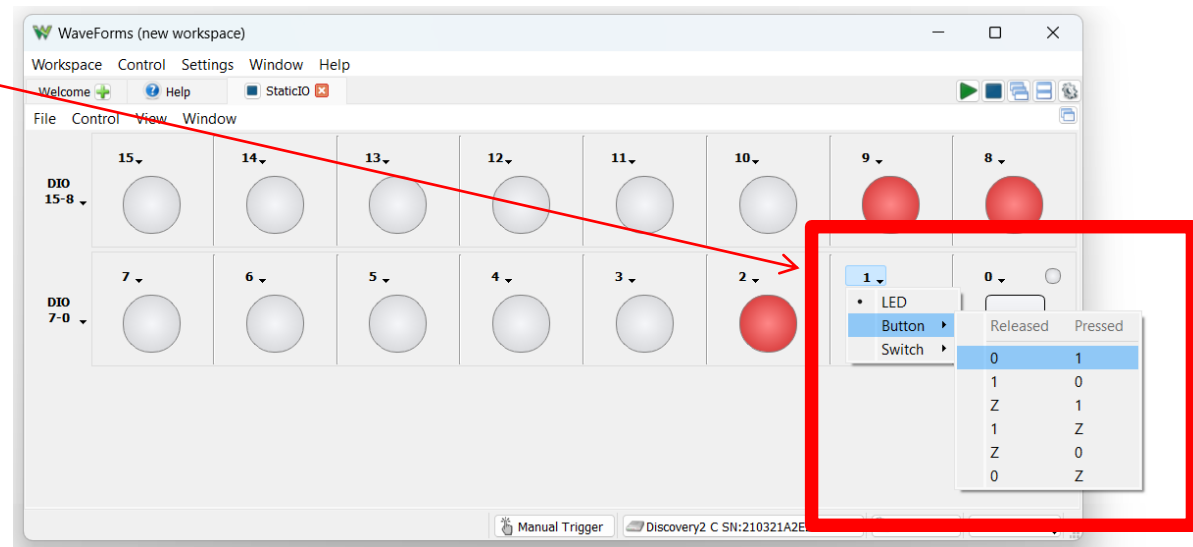
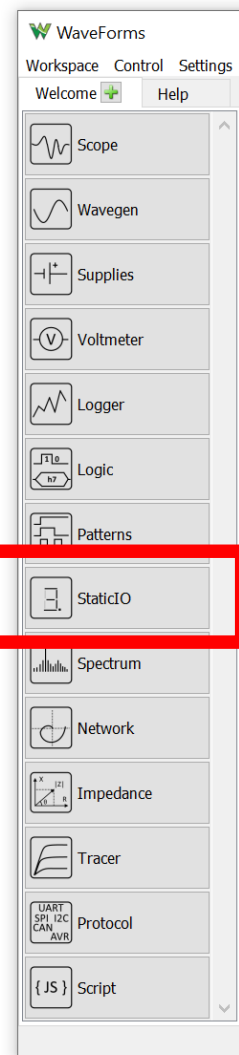


GPIO_0 da placa DE0-CV

Experiência 3

- Uso do Analog Discovery (configuração)

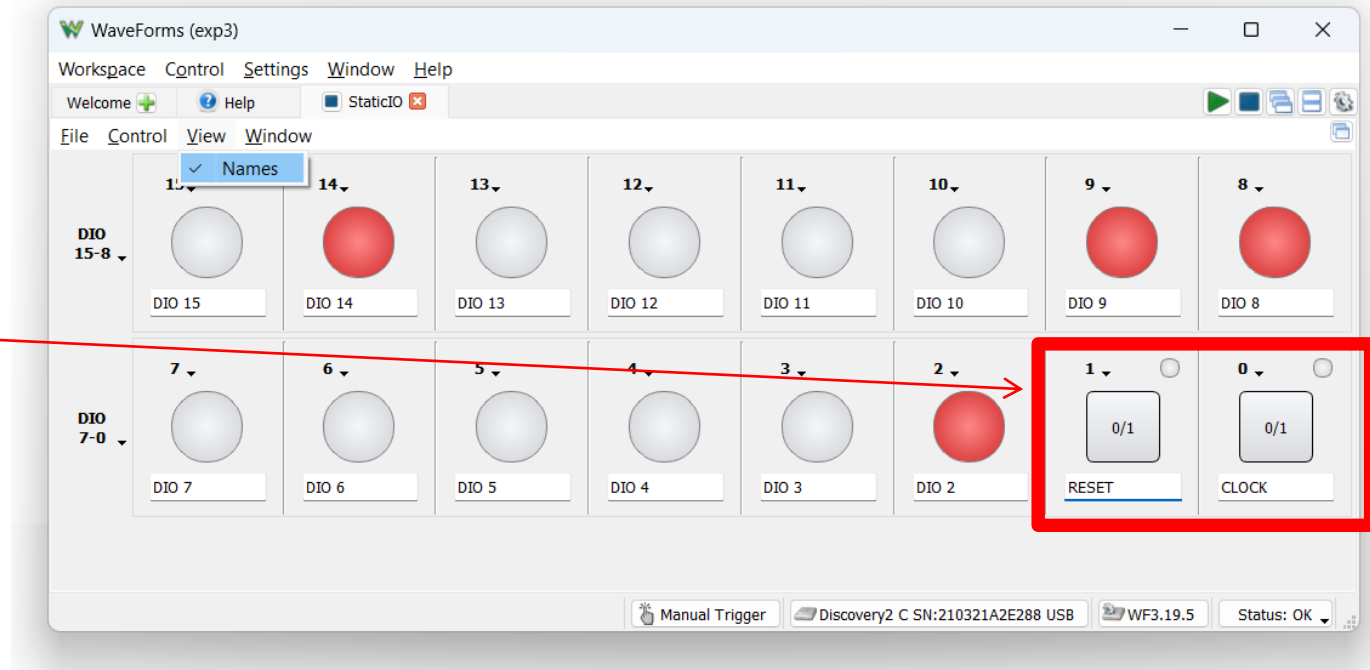
1. Software Waveforms
2. Ferramenta StaticIO
3. Configurar DIO0 e DIO1 como Button 0/1



Experiência 3

- Uso do Analog Discovery (acionamento dos sinais)

1. Software Waveforms
2. Ferramenta StaticIO
3. Configurar DIO0 e DIO1 como Button 0/1
4. Nomear DIO0 e DIO1



Experiência 3

• Plano de Teste (teste detalhado)

Teste do circuito na bancada

- Análise *clock a clock*;
- Acionamento das entradas;
- Monitoramento das saídas.

#	Operação	Sinais de controle	Resultado esperado
c.i.	Condições iniciais	clock=0 reset=0 iniciar=0 chaves=0000	pronto=0 db_igual=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000
1	"Resetar" circuito e observar a saída da memória	reset=1 clock ↑	pronto=0 db_igual=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000
2	Acionar sinal de clock 5 vezes com iniciar=0	reset=0 iniciar=0 clock ↑ (5x)	(permanece no estado inicial) pronto=0 db_igual=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000
3	Ajustar chaves para 0100, ativar iniciar=1 e acionar clock 1x	chaves=0100 iniciar=1 clock ↑	(muda para estado preparação) <completar todas as saídas>
4	Mantém chaves em 0100 e acionar clock 1x	chaves=0100 iniciar=0 clock ↑	(muda para estado registra) <completar todas as saídas>
5	Mantém chaves em 0100 e acionar clock 1x	chaves=0100 iniciar=0 clock ↑	(muda para estado comparação) db_igual=0 db_contagem=0000 db_memoria=0001 db_chaves=0100 <completar todas as saídas>
6	Mantém chaves em 0100 e acionar clock 1x	chaves=0100 clock ↑	(muda para estado próximo) <completar todas as saídas>
7	Mantém chaves em 0100 e acionar clock 3x	chaves=0100 clock ↑ (3x)	(passa pelos estados registra, comparação e próximo) pronto=0 db_igual=0 db_contagem=0001 db_memoria=0010 db_chaves=0100
8	Mantém chaves em 0100 e acionar clock 3x	chaves=0100 clock ↑ (3x)	(passa pelos estados registra, comparação e próximo) pronto=0 db_igual=1 db_contagem=0010 db_memoria=0100 db_chaves=0100
9	Mantém chaves em 0100 e acionar clock 9x	chaves=0100 clock ↑ (9x)	(passa 3x pelos estados registra, comparação e próximo) pronto=0 db_igual varia 0-1-0 db_contagem varia 0011-0100-0101 db_memoria varia 1000-0100-0010 db_chaves=0100
10	Ajustar chaves para 0001 e acionar clock 6x	chaves=0001 clock ↑ (6x)	<completar todas as saídas>
11	Ajustar chaves para 0010 e acionar clock 6x	chaves=0010 clock ↑ (6x)	<completar todas as saídas>
12	Ajustar chaves para 0100 e acionar clock 6x	chaves=0100 clock ↑ (6x)	<completar todas as saídas>
13	Ajustar chaves para 1000 e acionar clock 6x	chaves=1000 clock ↑ (6x)	<completar todas as saídas>
14	Ajustar chaves para 0000 e acionar clock 3x	chaves=0001 clock ↑ (3x)	<completar todas as saídas>
15	Mantém chaves em 0000 e acionar clock 3x	chaves=0010 clock ↑ (3x)	(passa pelo estado fim) pronto=1 <completar todas as saídas>
16	Mantém chaves em 0000 e acionar clock	chaves=0000 clock ↑	(termina no estado inicial) <completar todas as saídas>