UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E SISTEMAS DIGITAIS
PCS3635 - LABORATÓRIO DIGITAL I

PLANEJAMENTO DA EXPERIÊNCIA 4

Felipe Luis Korbes - NUSP: 13682893
Henrique Eduardo dos Santos de Souza - NUSP: 13679972
João Felipe de Souza Melo - NUSP: 13682913
Turma: 5
Professor: Reginaldo Arakaki
Data da experiência: 31/janeiro/2024

São Paulo
2024

1. Introdução..... 2
2. Atividades Pré-laboratório..... 3
2.1. Projeto lógico de um sistema digital..... 3
2.1.1. Adaptação do Projeto Lógico do Fluxo de Dados..... 3
2.1.2 Estudo do Projeto Lógico de uma Unidade de Controle..... 8
2.1.3. Projeto de um Sistema Digital..... 9
3. Planejamento da Aula Prática..... 17
3.1. Montagem e programação da FPGA..... 17
3.2 Testes e sinais de depuração..... 18
3.3 Desafio..... 18
3.3.1 Testes..... 19
3.3.2 Sintetização..... 31

1. Introdução

A experiência tem como objetivo a familiarização com diversos aspectos do design de circuitos digitais. Será projetada uma unidade de controle de um circuito digital simples usando Verilog, seguindo uma estrutura desenvolvida a partir de um pseudocódigo fornecido. Além disso, a experiência abrange o projeto hierárquico de circuitos digitais, codificação de máquinas de estados em Verilog e a síntese do projeto em uma placa FPGA usando o software Intel Quartus Prime. O enfoque central é integrar essa unidade de controle a um fluxo de dados, proveniente do circuito abordado na Experiência 3, para criar um sistema digital completo.

O circuito montado na experiência realizará as seguintes especificações:
"*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 que é realizado com o acionamento do sinal de entrada iniciar. 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 e deve indicar o resultado da comparação na saída de depuração db_igual. O conteúdo armazenado das chaves é apresentado na saída de depuração db_chaves. Em seguida, o contador interno deve ser incrementado para posicionar o endereçamento da memória para permitir o acesso ao 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 vigente da Unidade de Controle em determinado instante do funcionamento do sistema digital1 . O ciclo de comparação e reposicionamento da memória deve prosseguir até que todos os 16 dados sejam verificados. Ao final da operação, o sinal de saída pronto deve ser acionado por um período de clock. Depois disso, o circuito deve voltar para o estado inicial para aguardar a próxima ativação de iniciar*".

2. Atividades Pré-laboratório

2.1. Projeto lógico de um sistema digital

2.1.1. Adaptação do Projeto Lógico do Fluxo de Dados

Abaixo estão os diagramas de blocos da experiência 3 e da experiência 4 respectivamente.

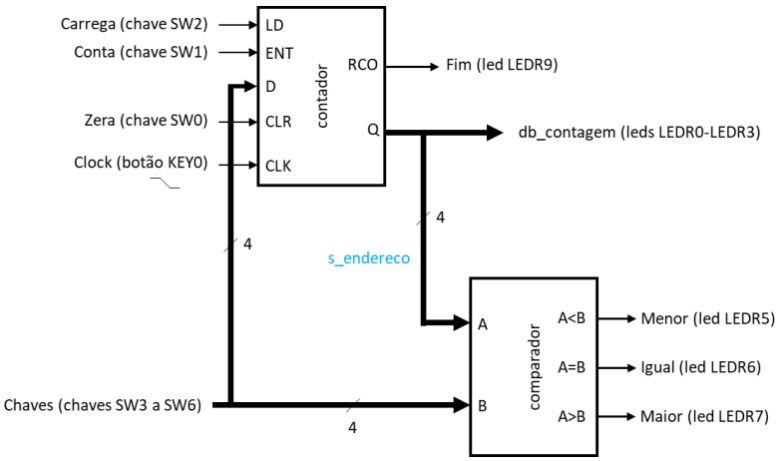

Figura 01 - Diagrama de Blocos das Experiência 3

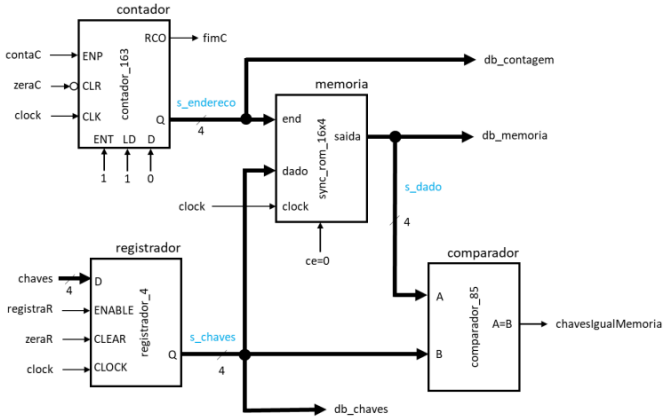

Figura 02 - Diagrama de blocos da Experiência 4

É possível observar que na experiência 4 usa-se como base o circuito da experiência 3 com algumas modificações. Existem 2 novos blocos, o bloco de memória e o bloco registrador. O bloco memória tem 4 entradas, end e dado de 4 bits e uma entrada *ce* que sempre recebe nível lógico LOW além do clock e há 1 saída de dados de 4 bits. O bloco registrador tem 4 entradas, a entrada *D* de 4 bits e as entradas *ENABLE*, *CLEAR* e *CLOCK* de 1 bit e uma saída de dados *Q* de 4 bits. O bloco memória está posicionado entre o contador e o comparador, onde a entrada *end* recebe a saída *Q* do contador e a saída da memória é a entrada *A* do

comparador. O registrador tem sua saída *Q* na entrada dado do bloco memória e na segunda entrada B do comparador.

Foram fornecidos os arquivos *registrador_4.v* e *sync_rom_16x4.v*. O arquivo *registrador_4.v* tem todas as entradas do bloco, ou seja, *clock*, *clear*, *enable*, *D* e *Q* do arquivo verilog são, respectivamente, *CLOCK*, *CLEAR*, *ENABLE*, *D* e *Q* do seu bloco. Há em seu módulo um bloco *always* que é sensível à borda de subida do *clock* e do sinal de entrada *clear*. Caso haja alguma borda de subida do *clock* ou de *clear*, então é comparado se o sinal *CLEAR* é ativo, caso seja então sua saída de dados recebe zero, esses sinais ilustram a operação de *CLEAR* do registrador. Caso haja a borda de subida de *clock* e *clear* esteja em nível lógico LOW e *enable* esteja em nível lógico HIGH, então a saída de dado recebe o dado de entrada, esses sinais ilustram a operação de LOAD.

O arquivo *sync_rom_16x4.v* é o módulo que representa o bloco da memória. Ele tem as mesmas entradas da sua representação por blocos com exceção à entrada *'ce'*, portanto, *'clock'*, *'address'* e *'data_out'* do arquivo verilog são *'clock'*, *'end'* e *'saida'* do bloco memória respectivamente. Há um always neste módulo sensível à borda de subida do *clock*, e ainda dentro desse bloco há um case para a entrada 'address' , onde a memória já está programada para alguns valores. Conectou-se os módulos de acordo com o fluxo de dados da figura 02 e foi feito os seguintes testes:

#	Operação	Sinais de Controle	Resultado Esperado	Veredito
ci	condições iniciais (entradas desativadas)	clock=0, zeraC=0, contaC=0, zeraR=0, registraR=0, chaves=0000	chavesIgualMemoria=0, fimC=0, db_contagem=0000, db_memoria=0001, db_chaves=0000	✔
1	Zerar contador e registrador (manter as outras desativadas)	zeraC=1, zeraR=1, clock↑	chavesIgualMemoria=0, fimC=0, db_contagem=0000, db_memoria=0001, db_chaves=0000	✔
2	Verificar saídas com chaves=0001 (manter outras entradas desativadas)	zeraC=0, contaC=0, zeraR=0, registraR=0, chaves=0001, clock↑	chavesIgualMemoria=0, fimC=0, db_contagem=0000, db_memoria=0001, db_chaves=0000	✔
3	Registrar chaves com chaves=0001 (manter outras entradas desativadas)	chaves=0001, registrarR=1, clock ↑	chavesIgualMemoria=1, fimC=0, db_contagem=0000, db_memoria=0001, db_chaves=0001	✔
4	Verificar saídas com	chaves=0001, clock ↑	chavesIgualMemoria=1,	✔

	chaves=0001 (manter outras entradas desativadas)		fimC=0, db_contagem=0000, db_memoria=0001, db_chaves=0001	
5	Incrementar contador (manter outras entradas desativadas)	contaC=1, clock ↑	chavesIgualMemoria=0, fimC=0, db_contagem=0001, db_memoria=0010, db_chaves=0001	✔
6	Registrar chaves com chaves=0010 (manter outras entradas desativadas)	chaves=0010 clock ↑	chavesIgualMemoria=1, fimC=0, db_contagem=0001, db_memoria=0010, db_chaves=0010	✔
7	Verificar saídas com chaves=0010 (manter outras entradas desativadas)	chaves=0010 clock ↑	chavesIgualMemoria=1, fimC=0, db_contagem=0001, db_memoria=0010, db_chaves=0010	✔
8	Incrementar contador (manter outras entradas desativadas)	contaC=1, clock ↑	chavesIgualMemoria=0, fimC=0, db_contagem=0010, db_memoria=0100, db_chaves=0010	✔
9	Registrar chaves com chaves=1000 (manter outras entradas desativadas)	chaves=1000 clock ↑	chavesIgualMemoria=0, fimC=0, db_contagem=0010, db_memoria=0100, db_chaves=1000	✔
10	Verificar saídas com chaves=1000 (manter outras entradas desativadas)	chaves=1000 clock ↑	chavesIgualMemoria=0, fimC=0, db_contagem=0010, db_memoria=0100, db_chaves=1000	✔
11	Incrementar contador até final da contagem	contaC=1 clock↑ (13x)	chavesIgualMemoria=0, fimC=1, db_contagem=1111, db_memoria=0100, db_chaves=1000	✔

Tabela 01 - Plano de Testes para o Circuito do Fluxo de Dados

Durante a simulação, as formas de onda e os sinais do ModelSim durante o teste 5 e o teste 8 não bateram com as da Tabela 01. Após detectar esse erro percebeu-se que o erro estava no atraso para a memória atualizar o valor após o contador ter incrementado seu valor, que acontece somente na próxima borda de clock. Para corrigir esse erro, foi adicionado um delay no testbench para o teste 5 e o teste 8. Abaixo estão as figuras dos testes:

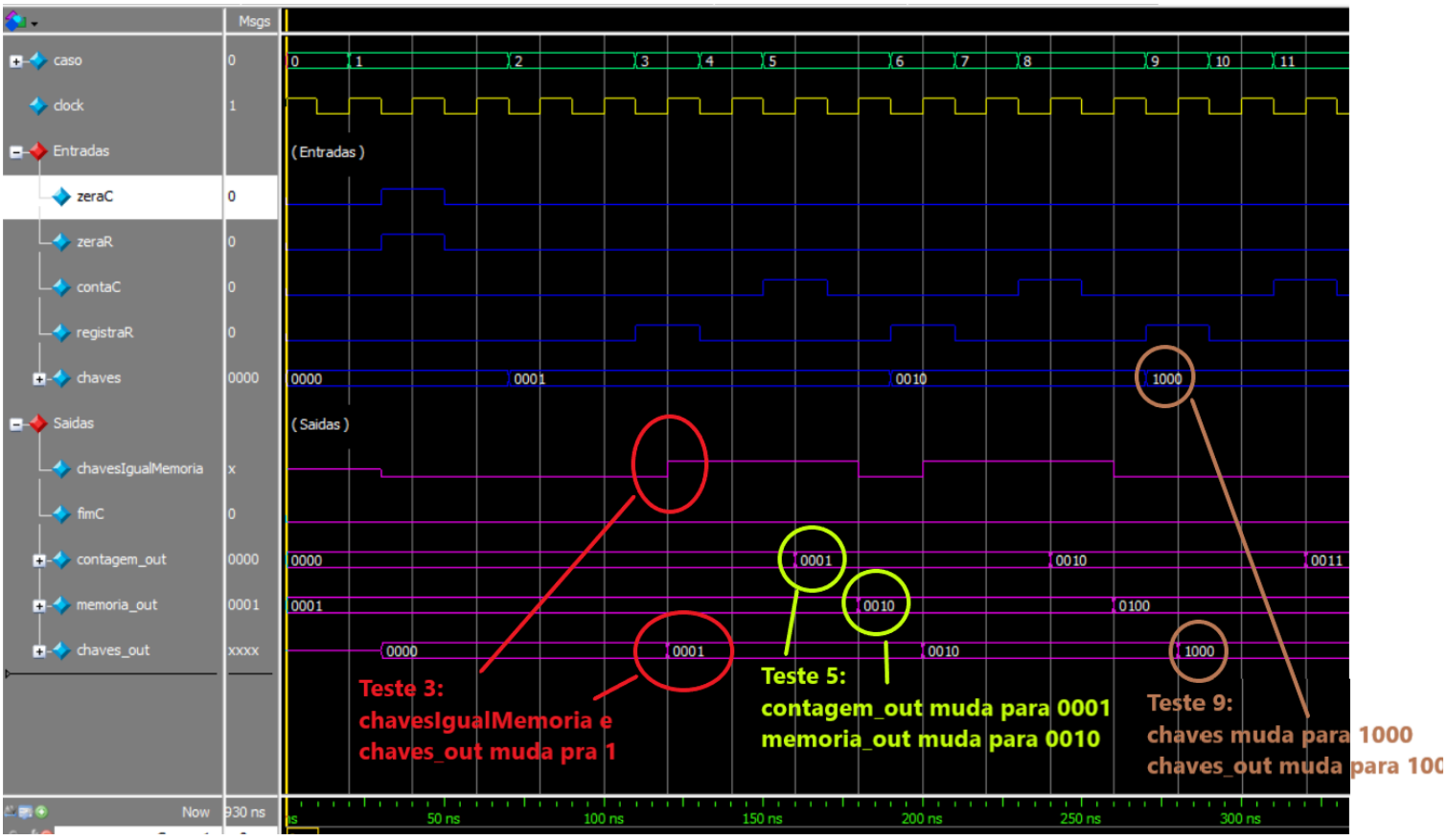

Figura 03: Ondas do fluxo de dados

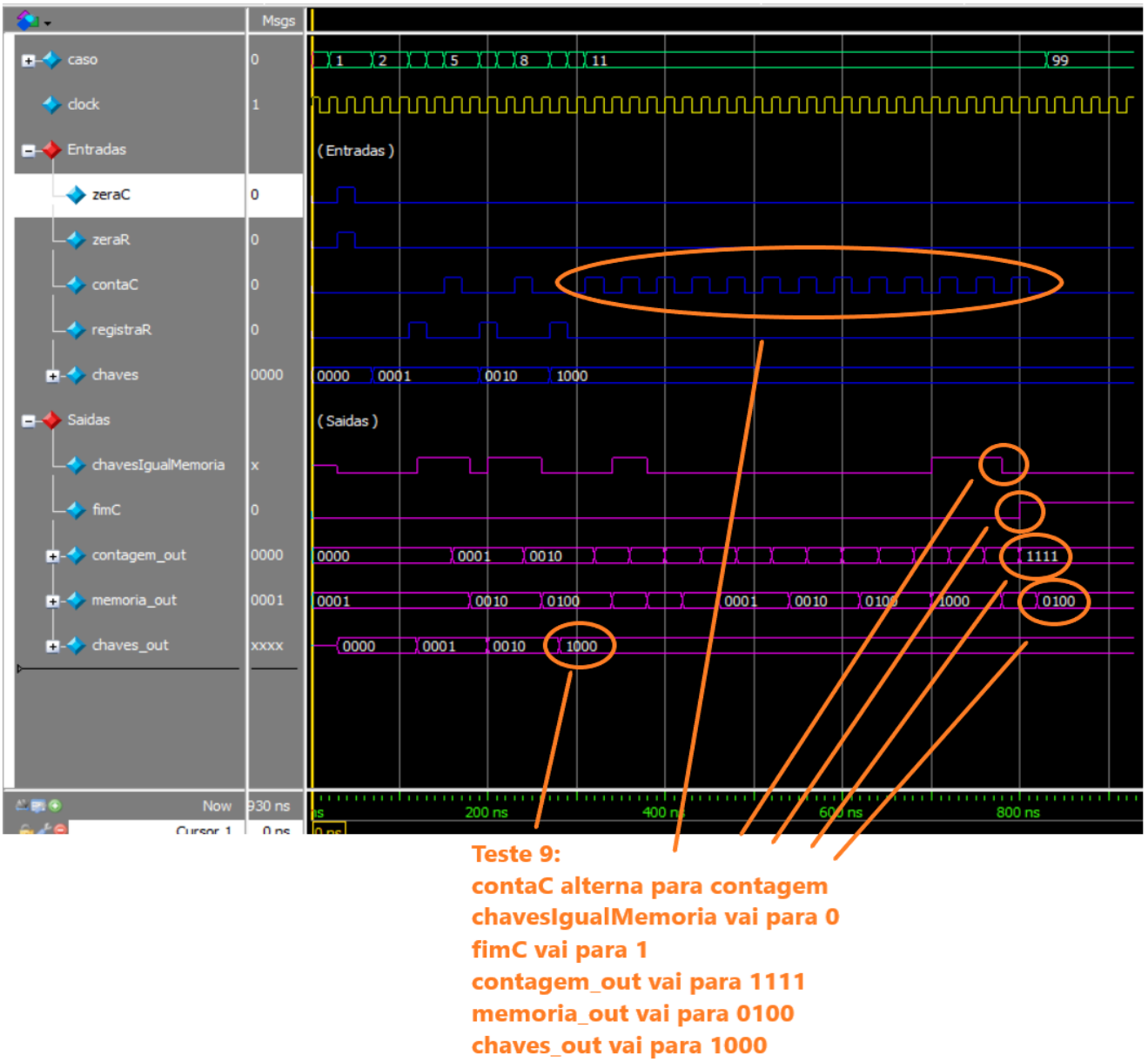

Figura 04: Continuação das ondas do fluxo de dados

2.1.2 Estudo do Projeto Lógico de uma Unidade de Controle

Foi disponibilizado o seguinte diagrama de transição de alto nível para a unidade de controle que satisfaz as condições supracitadas na Introdução:

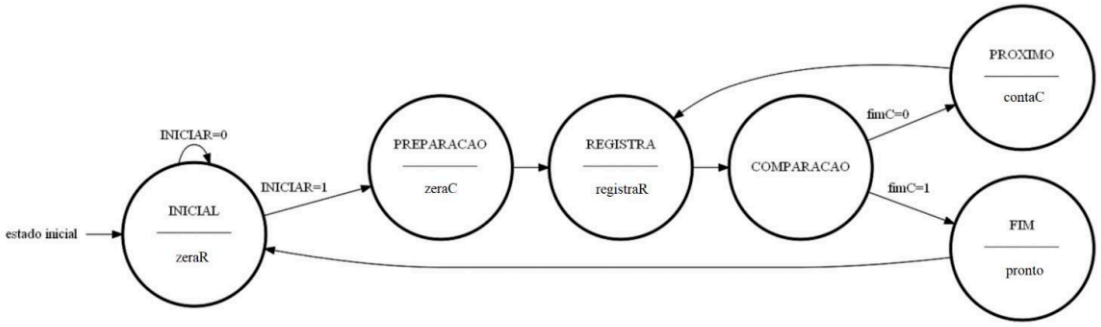

Figura 05 - Diagrama de Transição de alto nível

Também foi disponibilizado o arquivo *esp4_unidade_controle.v* onde o código da unidade de controle encontra-se. O módulo tem como entrada o *clock*, *reset*, *iniciar* e *fimC* e tem como saída *zeraC*, *contaC*, *zeraR*, *registraR*, *pronto* e um sinal de (4 bits) depuração chamado *db_estado*. As variáveis *Eatual* e *Eprox* são responsáveis por representar o estado atual e o próximo estado, respectivamente, na memória de estado. Para implementar a memória de estado, é utilizado um bloco *always @(posedge clock or posedge reset)* que atualiza *Eatual* com o valor de *Eprox* no flanco de subida do *clock* quando não há *reset*. Em caso de *reset*, *Eatual* é inicializado como "*inicial*".

A lógica de próximo estado é implementada por meio de um bloco *always* @* que incorpora uma estrutura *case*. As condições para transição de estados são determinadas com base no estado atual (*Eatual*) e nas condições de entrada (*iniciar* e *fimC*). A lógica de saída é definida utilizando um bloco *always* @*. As saídas (*zeraC*, *contaC*, *zeraR*, *registraR*, *pronto*, e *db_estado*) são atribuídas condicionalmente de acordo com o estado atual (*Eatual*). A saída de depuração db_estado apresenta uma representação binária do estado atual, como descrito funcionalmente.

2.1.3. Projeto de um Sistema Digital

A seguir, foi elaborado o projeto do circuito, consolidando todos os módulos mencionados até o momento. O diagrama de blocos desse circuito está representado na Figura 06. O bloco correspondente à unidade de controle destaca apenas o seu próprio módulo, ao passo que o bloco relacionado ao fluxo de dados abrange o contador, a memória, o registrador e o comparador.

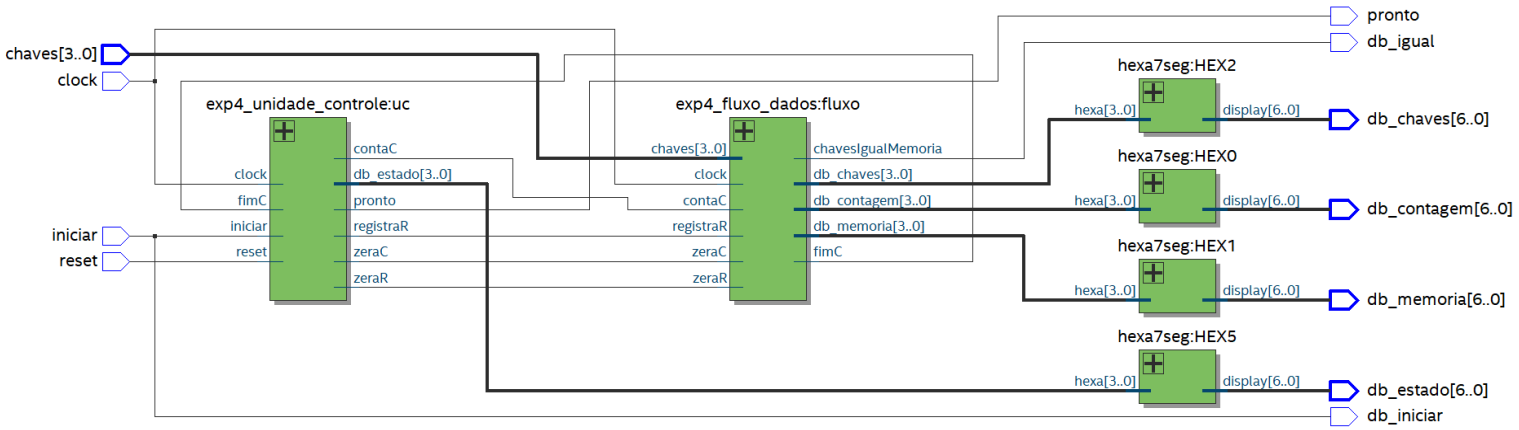

Figura 06 - Diagrama de blocos do circuito completo

Posteriormente, foram elaborados os planos de teste conforme a Tabela 02, utilizando o testbench fornecido. No entanto, é importante destacar que o testbench inicial não correspondia à Tabela de testes 02 para alguns testes específicos, mais especificamente no caso dos ciclos de clock. Diante disso, foram realizadas pequenas adaptações no testbench para alinhar os resultados dos testes observados pelo ModelSim de acordo com os da tabela.

O teste transcorreu conforme planejado, com a contagem sendo executada durante os testes e concluída conforme previsto no teste 15. Entre os testes 7 e 14, a máquina de estados variou entre os estados de "registra", "comparação" e "próximo" diversas vezes, enquanto o contador incrementa seu valor, refletindo nas alterações na saída da memória. Abaixo está a tabela:

#	Operação	Sinais de Controle	Resultado Esperado	Veredito
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) pronto=0 db_igual=0 db_iniciar=1 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0001	
4	Mantém chaves em 0100 e acionar clock 1x	chaves=0100 iniciar=0 clock=↑	(muda para estado registra) pronto=0 db_igual=0 db_iniciar=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0100	
5	Mantém chaves em 0100 e acionar clock 1x	chaves=0100 iniciar=0 clock=↑	(muda para estado comparação) pronto=0 db_igual=0 db_iniciar=0 db_contagem=0000 db_memoria=0001 db_chaves=0100 db_estado=0101	
6	Mantém chaves em 0100 e acionar clock 1x	chaves=0100 clock=↑	(muda para estado próximo) pronto=0 db_igual=0 db_iniciar=0 db_contagem=0000 db_memoria=0001 db_chaves=0100 db_estado=0110	
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)	(passa 2x pelos estados registra, comparação e próximo) pronto=0 db_igual varia 0-1 db_iniciar=0 db_contagem varia 0110-0111 db_memoria=0001 db_chaves=0001 db_estado=0100-0101-0110	
11	Ajustar chaves para 0010 e acionar clock 6x	chaves=0010 clock=↑(6x)	(passa 2x pelos estados registra, comparação e próximo) pronto=0 db_igual=1 db_iniciar=0 db_contagem varia 1000-1001 db_memoria=0010 db_chaves=0010 db_estado varia 0100-0101-0110	
12	Ajustar chaves para 0100 e acionar clock 6x	chaves=0100 clock=↑(6x)	(passa 2x pelos estados registra, comparação e próximo) pronto=0 db_igual=1 db_iniciar=0 db_contagem varia 1010-1011 db_memoria=0100 db_chaves=0100 db_estado varia 0100-0101-0110	
13	Ajustar chaves para 1000 e acionar clock 6x	chaves=1000 clock=↑(6x)	(passa 2x pelos estados registra, comparação e próximo) pronto=0 db_igual=1 db_iniciar=0 db_contagem varia 1100-1101 db_memoria=1000 db_chaves=1000 db_estado varia 0100-0101-0110	

14	Ajustar chaves para 0000 e acionar clock 3x	chaves=0001 clock=↑(3x)	(passa pelos estados registra, comparação e próximo) pronto=0 db_igual=1 db_iniciar=0 db_contagem=1110 db_memoria=0001 db_chaves=0001 db_estado varia 0100-0101-0110	
15	Mantém chaves em 0000 e acionar clock 3x	chaves=0010 clock=↑(3x)	(passa pelo estado fim) pronto=1 pronto=1 db_igual=0 db_iniciar=0 db_contagem=1111 db_memoria=0100 db_chaves=0010 db_estado varia 0100-0101-1100	
16	Mantém chaves em 0000 e acionar clock	chaves=0000 clock=↑	(termina no estado inicial) pronto=0 db_igual=0 db_iniciar=0 db_contagem=1111 db_memoria=0100 db_chaves=0000 db_estado varia 0	

Tabela 02 - Testes do circuito da experiência

A coluna ‘Veredito’ será preenchida durante a aula prática. Havia alguns erros no arquivo da testbench fornecido no moodle, como inconsistência no número de clocks ou chaves com valores inconsistentes em relação à tabela 02 que foram corrigidos. Após a realização e verificação dos testes, conforme detalhado na tabela 02 e por meio das formas de onda, foram efetuadas anotações diretas nas próprias ondas. Estas anotações destacam alguns sinais de importância particular, os quais podem ser visualizados nas Figuras 07 e 08.

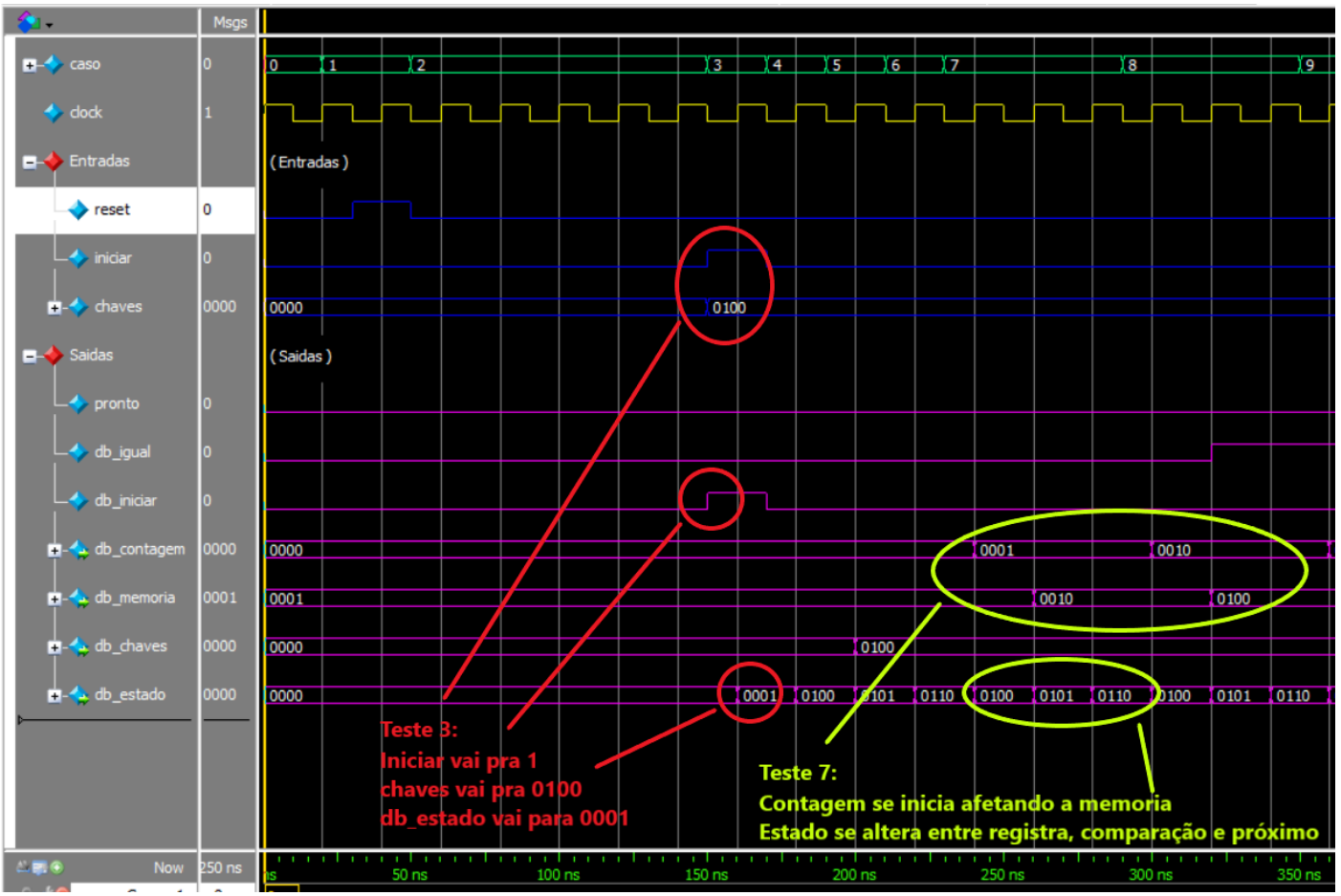

Figura 07 - Formas de ondas do testbench do circuito da experiência

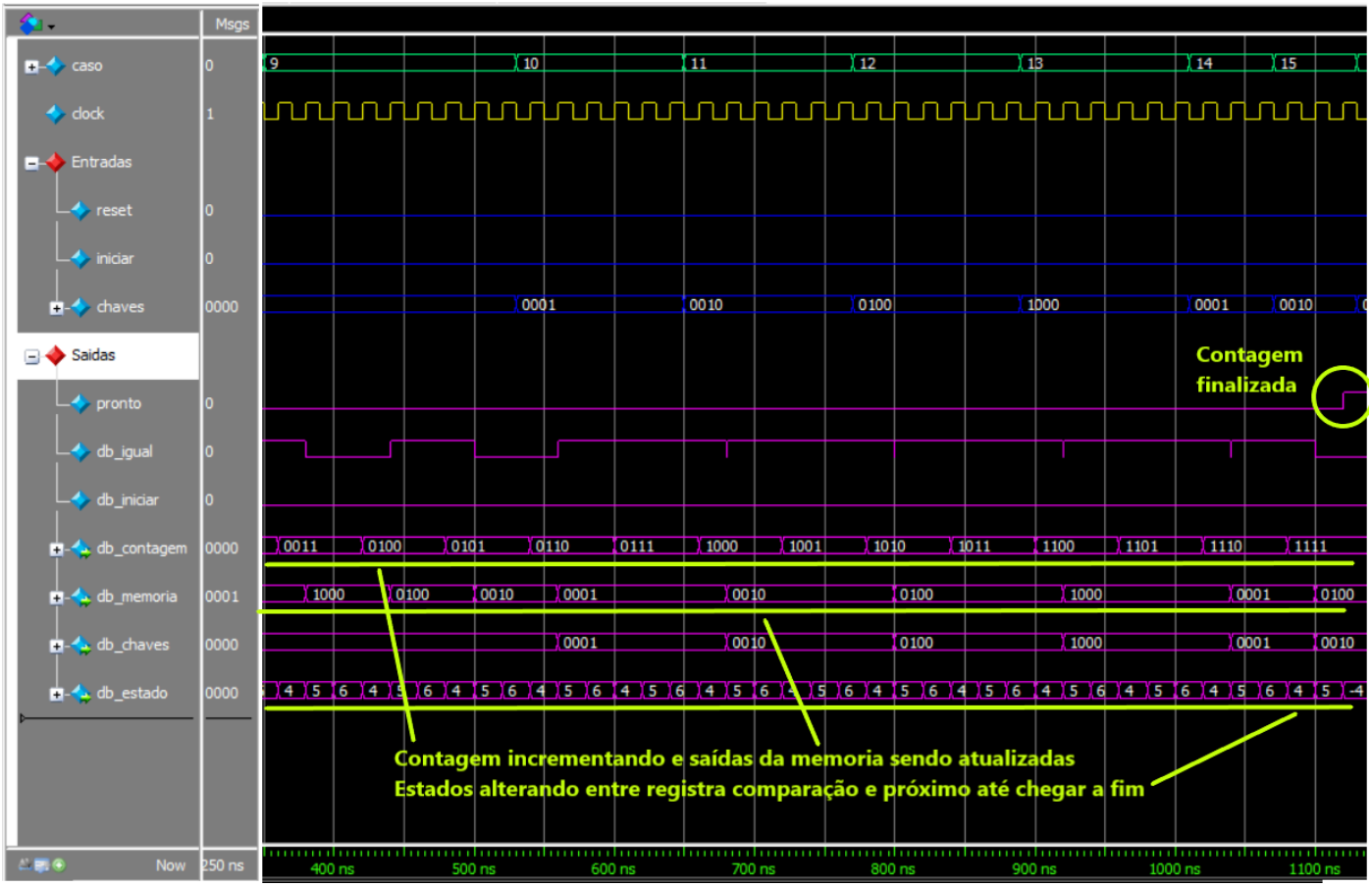

Figura 08 - Continuação das formas de onda do testbench do circuito da experiência

Na atribuição dos pinos na placa por meio do Quartus, foram incluídos 5 sinais adicionais de depuração: *db_zeraC, db_contaC, db_fimC, db_zeraR* e *db_registraR*. O diagrama de blocos atualizado, com a inclusão desses sinais extras, está apresentado na Figura 09. Adicionalmente, os sinais e seus respectivos pinos já designados no Quartus podem ser visualizados na Figura 10 que utilizou como base a tabela abaixo:

Sinal	Pino na Placa DEV0-CV	Pino na FPGA	Analog Discovery
CLOCK	GPIO_0_D0	PIN_N16 (pino 1 do GPIO 0)	StaticIO – Button 0/1
RESET	GPIO_0_D1	PIN_B16 (pino 2 do GPIO 0)	StaticIO – Button 0/1
INICIAR	chave SW0	PIN_U13	-
CHAVES(0)	chave SW1	PIN_V13	-

CHAVES(1)	chave SW2	PIN_T13	-
CHAVES(2)	chave SW3	PIN_T12	-
CHAVES(3)	chave SW4	PIN_AA15	-
PRONTO	led LEDR0	PIN_AA2	-
DB_IGUAL	led LEDR1	PIN_AA1	-
DB_INICIAR	led LEDR2	PIN_W2	-
DB_ZERAC	led LEDR4	PIN_N2	-
DB_CONTAC	led LEDR5	PIN_N1	-
DB_FIMC	led LEDR6	PIN_U2	-
DB_ZERAR	led LEDR8	PIN_L2	-
DB_REGISTRAR	led LEDR9	PIN_L1	-
DB_CONTAGEM	display HEX0	DB_CONTAGEM[0]=PIN_U21 DB_CONTAGEM[1]=PIN_V21 DB_CONTAGEM[2]=PIN_W22 DB_CONTAGEM[3]=PIN_W21 DB_CONTAGEM[4]=PIN_Y22 DB_CONTAGEM[5]=PIN_Y21 DB_CONTAGEM[6]=PIN_AA22	-
DB_MEMORIA	display HEX1	DB_MEMORIA[0]=PIN_AA20 DB_MEMORIA[1]=PIN_AB20 DB_MEMORIA[2]=PIN_AA19 DB_MEMORIA[3]=PIN_AA18 DB_MEMORIA[4]=PIN_AB18 DB_MEMORIA[5]=PIN_AA17 DB_MEMORIA[6]=PIN_U22	-
DB_CHAVES	display HEX2	DB_CHAVES[0]=PIN_Y19 DB_CHAVES[1]=PIN_AB17 DB_CHAVES[2]=PIN_AA10 DB_CHAVES[3]=PIN_Y14 DB_CHAVES[4]=PIN_V14 DB_CHAVES[5]=PIN_AB22 DB_CHAVES[6]=PIN_AB21	-
DB_ESTADO	display HEX5	DB_ESTADO[0] = PIN_N9 DB_ESTADO[1] = PIN_M8 DB_ESTADO[2] = PIN_T14 DB_ESTADO[3] = PIN_P14 DB_ESTADO[4] = PIN_C1 DB_ESTADO[5] = PIN_C2 DB_ESTADO[6] = PIN_W19	-

Tabela 03 - Designação de Pinos

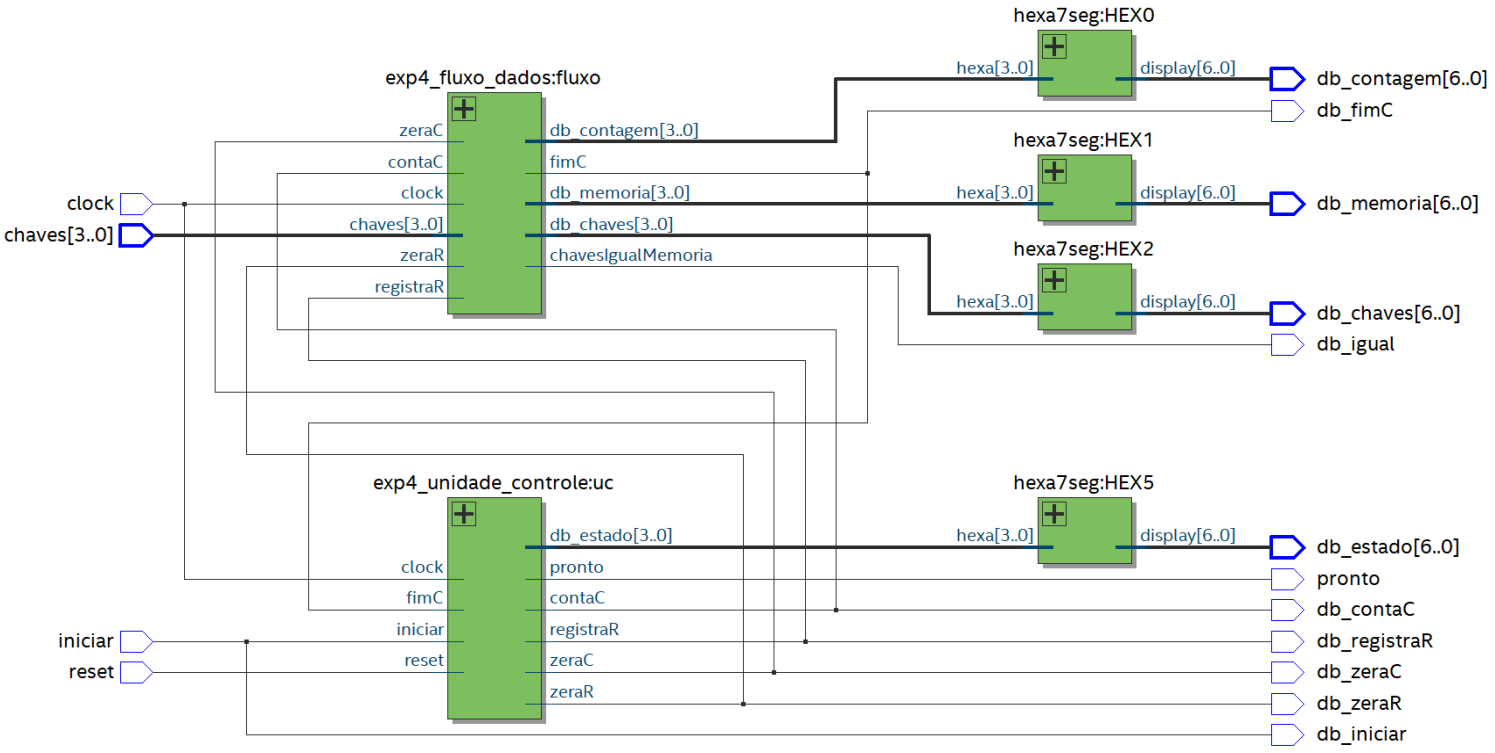

Figura 09 - Diagrama de bloco do circuito da experiência com sinais extras de depuração

chaves[3]	Input	PIN_AA15
clock	Input	PIN_N16
db_chaves[0]	Output	PIN_Y19
db_chaves[1]	Output	PIN_AB17
db_chaves[2]	Output	PIN_AA10
db_chaves[3]	Output	PIN_Y14
db_chaves[4]	Output	PIN_V14
db_chaves[5]	Output	PIN_AB22
db_chaves[6]	Output	PIN_AB21
db_contaC	Output	PIN_N1
db_contagem[0]	Output	PIN_U21
db_contagem[1]	Output	PIN_V21
db_contagem[2]	Output	PIN_W22
db_contagem[3]	Output	PIN_W21
db_contagem[4]	Output	PIN_Y22
db_contagem[5]	Output	PIN_Y21
db_contagem[6]	Output	PIN_AA22
db_estado[0]	Output	PIN_N9
db_estado[1]	Output	PIN_M8
db_estado[2]	Output	PIN_T14
db_estado[3]	Output	PIN_P14
db_estado[4]	Output	PIN_C1
db_estado[5]	Output	PIN_C2
db_estado[6]	Output	PIN_W19
db_fimC	Output	PIN_U2
db_igual	Output	PIN_AA1
db_igual	Output	PIN_AA1
db_iniciar	Output	PIN_W2
db_memoria[0]	Output	PIN_AA20
db_memoria[1]	Output	PIN_AB20
db_memoria[2]	Output	PIN_AA19
db_memoria[3]	Output	PIN_AA18
db_memoria[4]	Output	PIN_AB18
db_memoria[5]	Output	PIN_AA17
db_memoria[6]	Output	PIN_U22
db_registraR	Output	PIN_L1
db_zeraC	Output	PIN_N2
db_zeraR	Output	PIN_L2
iniciar	Input	PIN_U13
pronto	Output	PIN_AA2
reset	Input	PIN_B16

Figura 10 - Conexão dos pinos no Quartus

3. Planejamento da Aula Prática

3.1. Montagem e programação da FPGA

Como as etapas de preparação do projeto, desde a modelagem até a implementação em Verilog dos componentes e testes básicos de corretude no ModelSim, estão feitas desde já nesse planejamento, inicia-se a aula prática a partir do processo de síntese e programação do circuito na placa FPGA DE0-CV.

Assim, será feito o download dos arquivos pré-preparados para a experiência e serão colocados no diretório C:\Projetos\T5BB5\Exp4, no computador do laboratório. Em seguida, abri-se o projeto no Intel Quartus e programa-se a FPGA. Feito isso, conectar-se-á o fio 1 (cor esverdeada) e o fio 0 (cor rosa) do Analog Discovery 2 com os pinos PIN_D1 e PIN_D0 respectivamente, do GPIO 0 da FPGA além de conectar-se o fio W1 (cor preto) do Analog com o pino 12 do GPIO 0. Como segue a figura abaixo:

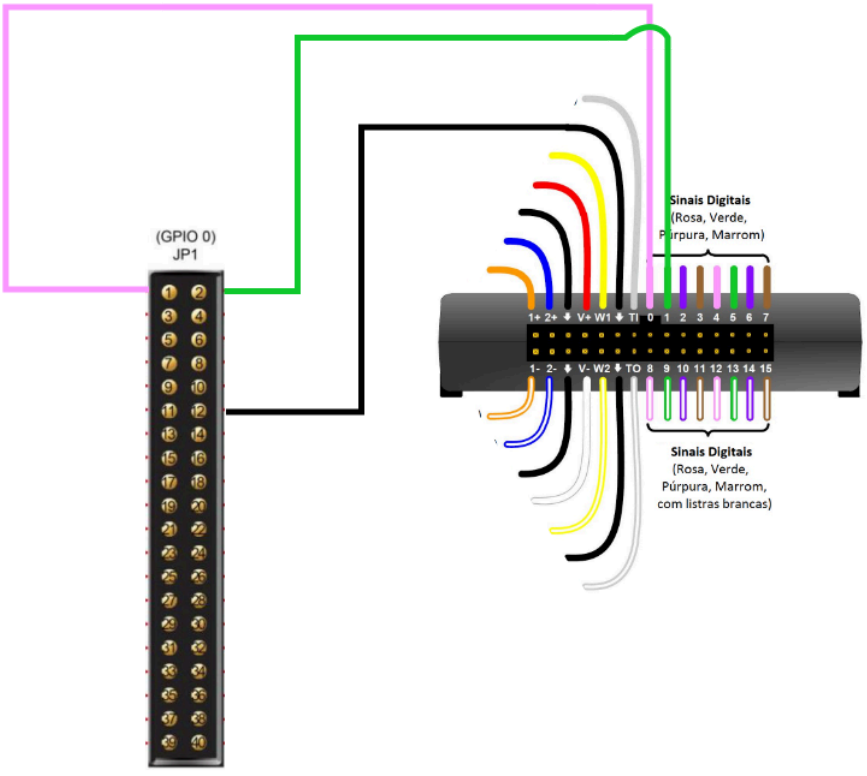

Figura 11 - Conexão da FPGA com o Analog Discovery 2

3.2 Testes e sinais de depuração

Após a montagem e programação da FPGA, procederá-se para a realização do plano de teste do circuito, tal qual da tabela 02, em que anotará-se os resultados experimentais obtidos, comparativamente com os esperados. Os testes serão realizados sequencialmente, na ordem em que aparecem e com os devidos acionamentos dos sinais de controle. Em caso de sucesso, ou seja, sinais esperados e obtidos iguais, registrará-se "*check*" na linha correspondente. Em caso de falha, o teste será interrompido e, a partir daí, inicia-se o processo de depuração dos códigos, para corrigir eventuais falhas de lógica. O processo de depuração seguirá o seguinte procedimento padrão:

(i) verificar consistência de entrada e saída dos módulos com o datapath;

(ii) avaliar lógica aplicada sobre sinal nos módulo em que ele atua como entrada ou saída;

(iii) testar alguns valores possíveis para alguns sinais, para estudar a reação do circuito - por exemplo, inverter um sinal.

Além disso, cada etapa seguirá acompanhada de discussão conjunta e, possivelmente, anotações e tabelas em papel sulfite.

3.3 Desafio

O desafio proposto está na figura abaixo, onde está destacado as mudanças:

```
Algoritmo: sistema digital simples modificado
   entradas: iniciar, chaves
   saídas: pronto, acertou, errou
   depuração: contagem, memória, estado, igual
1.  {
2.      while (verdadeiro) {
3.          espera acionamento do sinal iniciar
4.          inicia circuito para condições iniciais
5.          while (não atingiu o último dado e acertou todos os dados) {
                // continua enquanto chaves=dado e não chegou no
                // final da memória
6.              compara chaves de entrada com dados armazenados
7.              incrementa contador interno
8.          }
9.          ativa acertou se acertou todos os dados da memória
10.         ativa errou se errou um dado
11.         ativa saída pronto
12.         }
13.  }
```

Figura 12 - Pseudocódigo do Sistema Digital do Desafio

Foi adicionado duas saídas do arquivo circuito_exp4.v, uma chamada acertou e outra chamada errou, assim, tornando-se o circuito_exp4_desafio.v . As únicas alterações significativas foram na unidade de controle, foi adicionada uma entrada chamada igual, que é a saída do comparador para o caso em que A=B além de 2 saídas chamadas acertou e errou.

Foi alterada a lógica de mudança de estado de comparação, onde só deve-se ir para o estado *fim* caso *fimC* seja HIGH OU *igual* seja LOW.

Foi adicionado mais duas linhas na lógica de sinais de saída, uma responsável por controlar a saída *acertou* e outra pela saída *errou. A* saída *acertou* deve ser HIGH quando o estado atual for fim e igual for HIGH. A saída errou deve ser HIGH quando o estado atual for fim e igual for LOW.

3.3.1 Testes

Após realizadas as alterações no código para atender às especificações do desafio, procedemos com a montagem do circuito no Digital para apurar os sinais. Os detalhes dos planos de testes executados estão documentados na Tabela 04 e 05. Os testes incluem um cenário em que todos os 16 números da memória foram corretamente acertados, bem como um teste em que houve um erro na posição 4 da memória respectivamente.

Cenário 1 - 16 Acertos					
#	**Operação**	**Sinais de controle**	**Resultado esperado**	**Resultado observado**	**Veredito**
c.i.	Condições iniciais	clock=0 reset=0 iniciar=0 chaves=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=0 db_contagem=0000 db_memoria=0000 db_chaves=0000 db_estado=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=0 db_contagem=0 db_memoria=0 db_chaves=0 db_estado=0	
1	Ativa reset	clock=↑ reset=1 iniciar=0	pronto=0 acertou=0 errou=0	pronto=0 acertou=0 errou=0	

		chaves=0000	db_igual=0 db_iniciar=0 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=1 db_registraR=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000	db_igual=0 db_iniciar=0 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=1 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000	
2	Iniciar	clock=↑ reset=0 iniciar=1 chaves=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=1 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=1 db_registraR=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0001	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=1 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=1 db_registraR=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0001	
3	Ajusta chave para 0001 e compara	clock=↑(x2) reset=0 iniciar=0 chaves=0001	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0000 db_memoria=0001 db_chaves=0001 db_estado varia 0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0000 db_memoria=0001 db_chaves=0001 db_estado varia 0004-0005	
4	Ajusta chave para 0010 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0010	pronto=0 acertou=0 pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0001 db_memoria=0010 db_chaves=0010 db_estado varia	pronto=0 acertou=0 pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0001 db_memoria=0010 db_chaves=0010 db_estado varia	

			0006-0004-0005	0006-0004-0005	
5	Ajusta chave para 0100 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0010 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0010 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	
6	Ajusta chave para 1000 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=1000	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0011 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0011 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	
7	Ajusta chave para 0100 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0100 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0100 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	
8	Ajusta chave para 0010 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0010	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0	

			db_registraR varia 0-1 db_contagem=0101 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	db_registraR varia 0-1 db_contagem=0101 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	
9	Ajusta chave para 0001 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0001	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0110 db_memoria=0001 db_chaves=0001 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0110 db_memoria=0001 db_chaves=0001 db_estado varia 0006-0004-0005	
10	Ajusta chave para 0001 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0001	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0111 db_memoria=0001 db_chaves=0001 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0111 db_memoria=0001 db_chaves=0001 db_estado varia 0006-0004-0005	
11	Ajusta chave para 0010 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0010	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1000 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1000 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	
12	Ajusta chave para 0010 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0010	pronto=0 acertou=0 errou=0 db_igual=1	pronto=0 acertou=0 errou=0 db_igual=1	

			db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1001 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1001 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	
13	Ajusta chave para 0100 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1010 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1010 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	
14	Ajusta chave para 0100 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1011 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1011 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	
15	Ajusta chave para 1000 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=1000	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1100 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1100 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	

16	Ajusta chave para 1000 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=1000	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1101 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1101 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	
17	Ajusta chave para 0001 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0001	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1110 db_memoria=0001 db_chaves=0001 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1110 db_memoria=0001 db_chaves=0001 db_estado varia 0006-0004-0005	
18	Ajusta chave para 0100 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1111 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=1111 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	
19	Fim, acertou e reset	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=1 acertou=1 errou=0 db_igual=1 db_iniciar=0 db_zeraC varia 0-1 db_contaC=0 db_fimC=1 db_zeraR varia 0-1 db_registraR=0 db_contagem varia	pronto=1 acertou=1 errou=0 db_igual=1 db_iniciar=0 db_zeraC varia 0-1 db_contaC=0 db_fimC=1 db_zeraR varia 0-1 db_registraR=0 db_contagem varia	

			1111-0000-0001 db_memoria varia 0100-0001 db_chaves varia 0100-0000 db_estado varia 1100-0000	1111-0000-0001 db_memoria varia 0100-0001 db_chaves varia 0100-0000 db_estado varia 1100-0000	

Tabela 04 - Teste do desafio para 16 acertos

Cenário 2 - Erro na posição 4 da memória					
#	**Operação**	**Sinais de controle**	**Resultado esperado**	**Resultado observado**	**Veredito**
c.i.	Condições iniciais	clock=0 reset=0 iniciar=0 chaves=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=0 db_contagem=0000 db_memoria=0000 db_chaves=0000 db_estado=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=0 db_contagem=0 db_memoria=0 db_chaves=0 db_estado=0	
1	Ativa reset	clock=↑ reset=1 iniciar=0 chaves=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=0 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=1 db_registraR=0 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=0 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=1 db_contagem=0000 db_memoria=0001 db_chaves=0000 db_estado=0000	
2	Iniciar	clock=↑ reset=0 iniciar=1 chaves=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=1 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=1 db_registraR=0 db_contagem=0000 db_memoria=0001 db_chaves=0000	pronto=0 acertou=0 errou=0 db_igual=0 db_iniciar=1 db_zeraC=1 db_contaC=0 db_fimC=0 db_zeraR=1 db_registraR=0 db_contagem=0000 db_memoria=0001 db_chaves=0000	

			db_estado=0001	db_estado=0001	
3	Ajusta chave para 0001 e compara	clock=↑(x2) reset=0 iniciar=0 chaves=0001	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0000 db_memoria=0001 db_chaves=0001 db_estado varia 0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0000 db_memoria=0001 db_chaves=0001 db_estado varia 0004-0005	
4	Ajusta chave para 0010 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0010	pronto=0 acertou=0 pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0001 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	pronto=0 acertou=0 pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0001 db_memoria=0010 db_chaves=0010 db_estado varia 0006-0004-0005	
5	Ajusta chave para 0100 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=0100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0010 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0010 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	
6	Ajusta chave para 1000 e compara	clock=↑(x3) reset=0 iniciar=0 chaves=1000	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0	

			db_zeraR=0 db_registraR varia 0-1 db_contagem=0011 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	db_zeraR=0 db_registraR varia 0-1 db_contagem=0011 db_memoria=1000 db_chaves=1000 db_estado varia 0006-0004-0005	
7	Ajusta chave para 0010 e compara e fim	clock=↑(x2) reset=0 iniciar=0 chaves=0010	pronto=1 acertou=0 errou=1 db_igual=0 db_iniciar=0 db_zeraC=0 db_contaC=0 db_fimC=0 db_zeraR=0 db_registraR=0 db_contagem=0100 db_memoria=0000 db_chaves=0010 db_estado varia 0005-1100	pronto=0 acertou=0 errou=0 db_igual=1 db_iniciar=0 db_zeraC=0 db_contaC varia 1-0 db_fimC=0 db_zeraR=0 db_registraR varia 0-1 db_contagem=0100 db_memoria=0100 db_chaves=0100 db_estado varia 0006-0004-0005	

Tabela 05 - Teste do desafio para erro na posição 4 de memória

Abaixo estão as figuras dos testes do desafio realizados no ModelSim, com as formas de ondas e resultados esperados de acordo com a tabela 05. A coluna ‘Veredito’ será preenchida durante a aula prática.

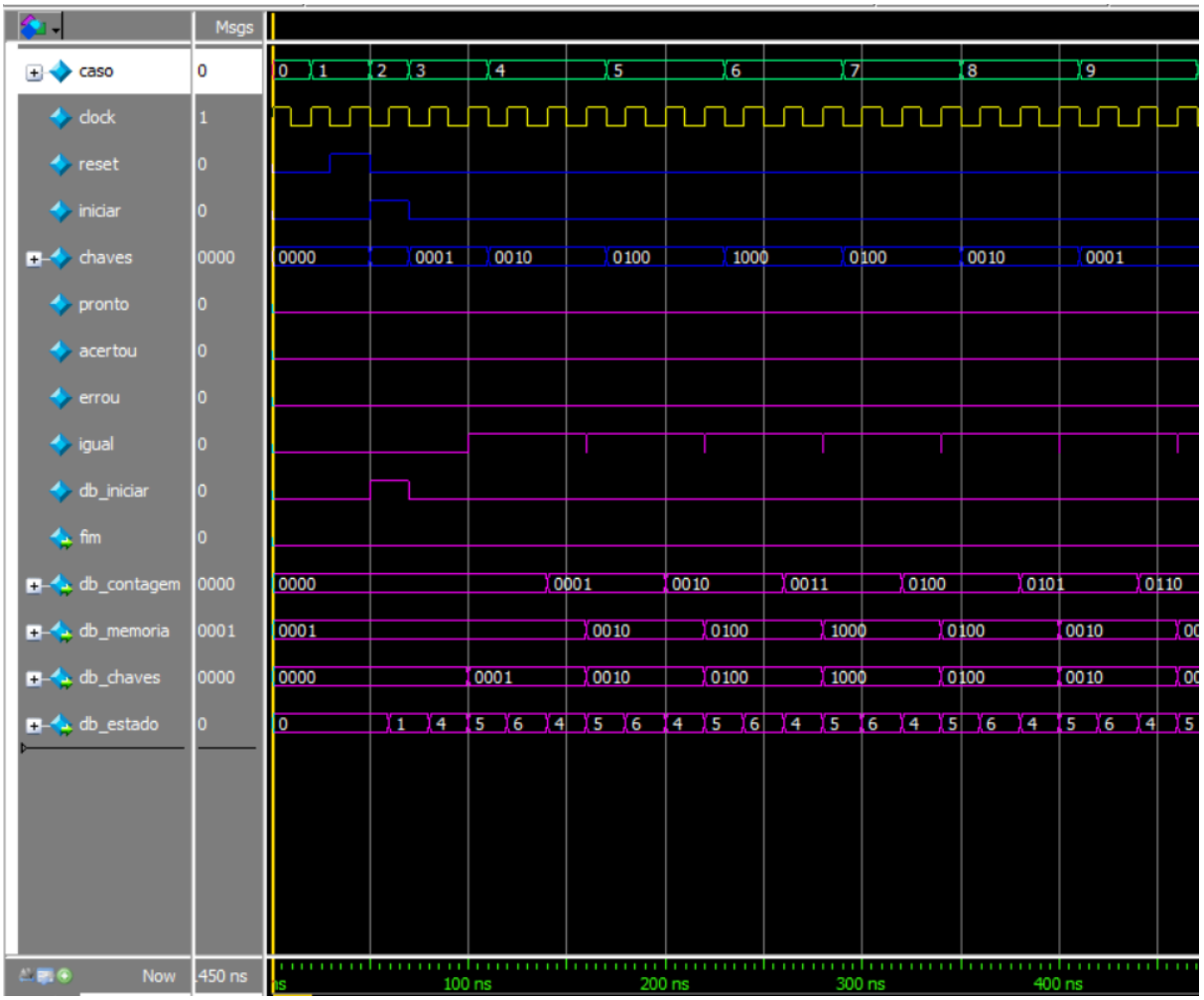

Figura 13 - Forma de onda do teste de 16 acertos

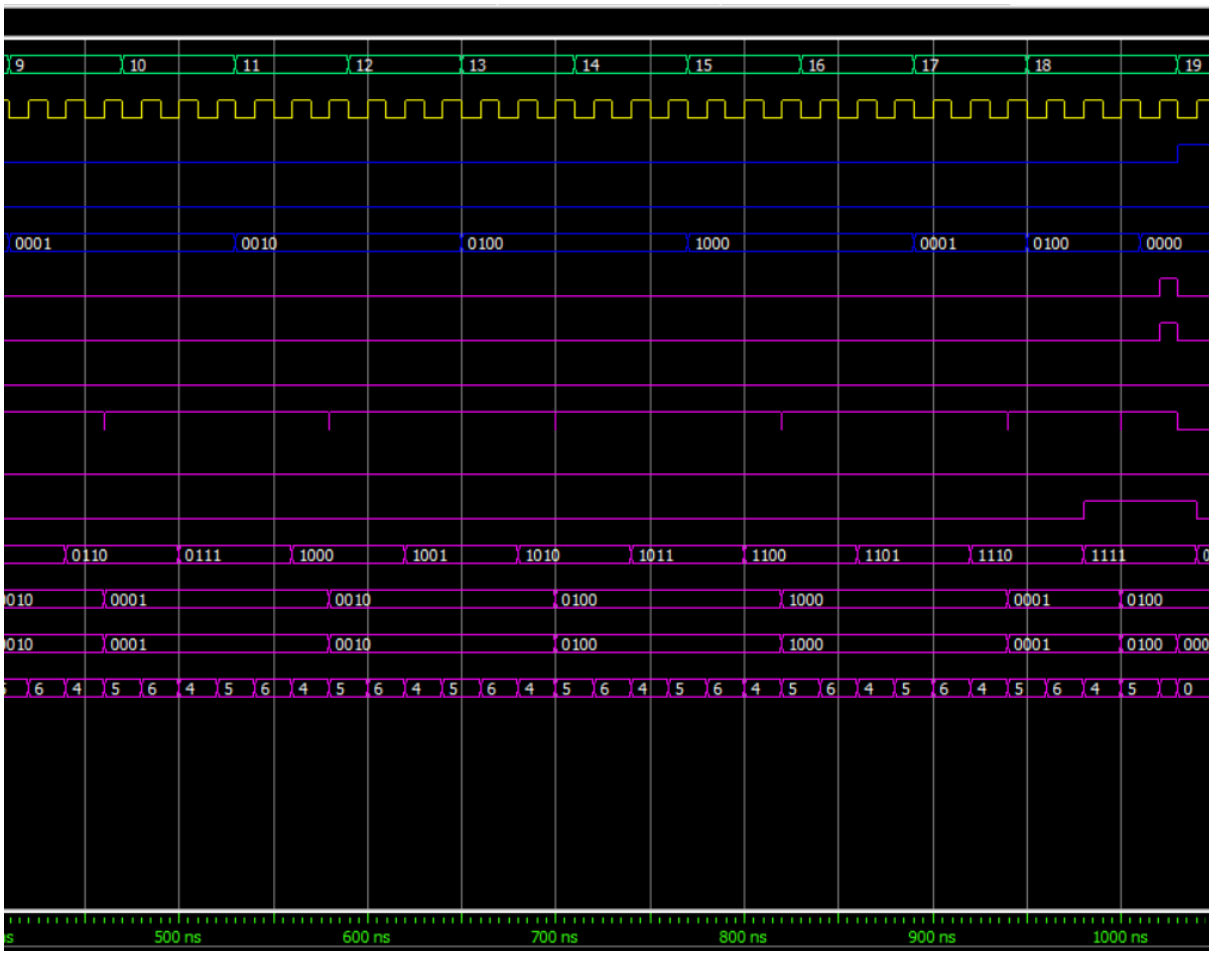

Figura 14 - Continuação da forma de onda do teste de 16 acertos

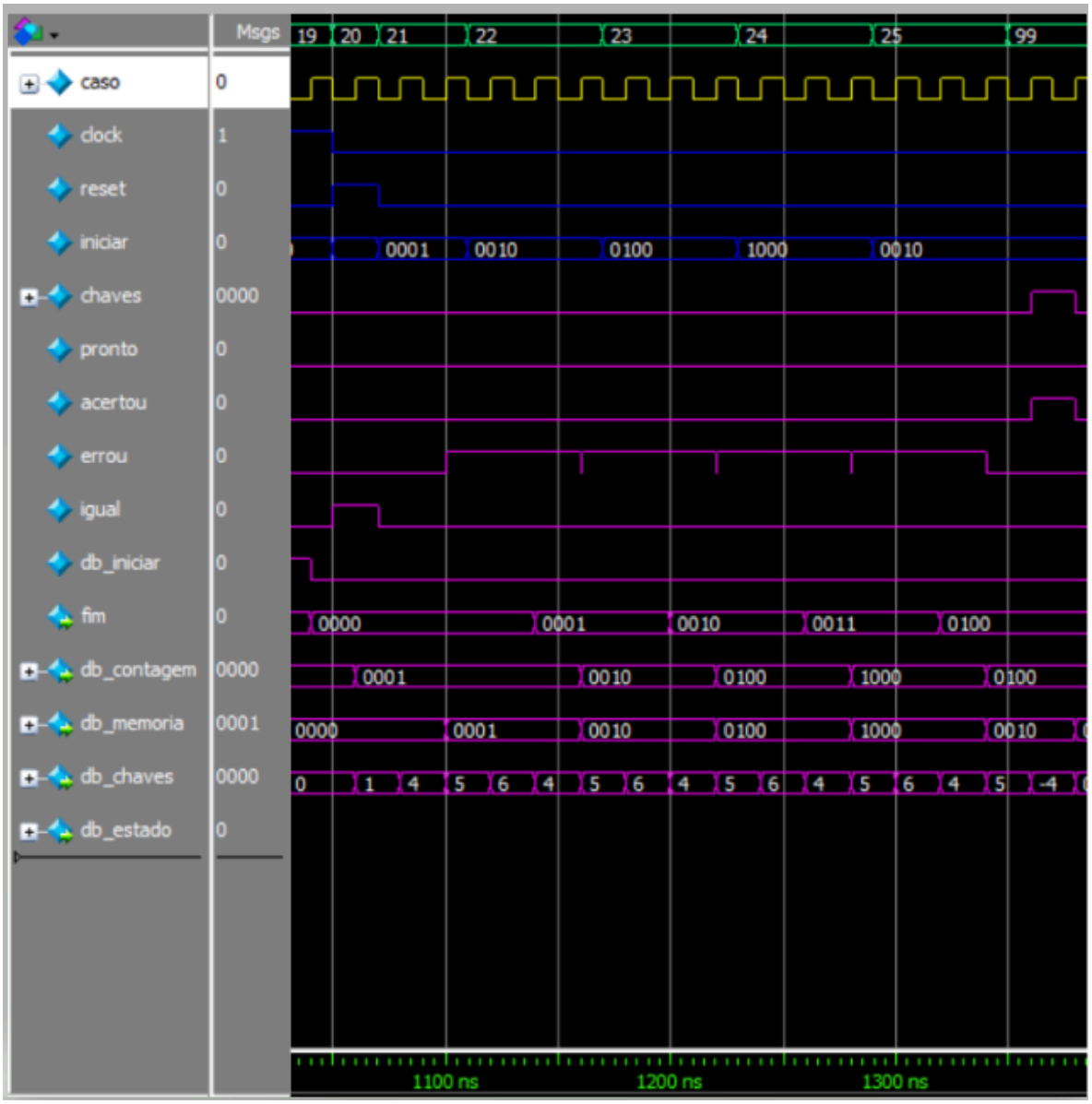

Figura 15 - Forma de onda do teste do erro da posição 4 de memória

3.3.2 Sintetização

Abaixo estão os pinos destinados para o desafio, tomou-se como base a tabela 3.

Sinal	Pino na Placa DEV0-CV	Pino na FPGA	Analog Discovery
CLOCK	GPIO_0_D0	PIN_N16 (pino 1 do GPIO 0)	StaticIO – Button 0/1
RESET	GPIO_0_D1	PIN_B16 (pino 2 do GPIO 0)	StaticIO – Button 0/1
INICIAR	chave SW0	PIN_U13	-
CHAVES(0)	chave SW1	PIN_V13	-
CHAVES(1)	chave SW2	PIN_T13	-
CHAVES(2)	chave SW3	PIN_T12	-
CHAVES(3)	chave SW4	PIN_AA15	-
PRONTO	led LEDR0	PIN_AA2	-
DB_IGUAL	led LEDR1	PIN_AA1	-
DB_INICIAR	led LEDR2	PIN_W2	-
ACERTOU	Led LEDR8	PIN_L2	-
ERROU	Led LEDR9	PIN_L1	-
DB_ZERAC	led LEDR4	PIN_N2	-
DB_CONTAC	led LEDR5	PIN_N1	-
DB_FIMC	led LEDR6	PIN_U2	-
DB_CONTAGEM	display HEX0	DB_CONTAGEM[0]=PIN_U21 DB_CONTAGEM[1]=PIN_V21 DB_CONTAGEM[2]=PIN_W22 DB_CONTAGEM[3]=PIN_W21 DB_CONTAGEM[4]=PIN_Y22 DB_CONTAGEM[5]=PIN_Y21 DB_CONTAGEM[6]=PIN_AA22	-
DB_MEMORIA	display HEX1	DB_MEMORIA[0]=PIN_AA20 DB_MEMORIA[1]=PIN_AB20 DB_MEMORIA[2]=PIN_AA19 DB_MEMORIA[3]=PIN_AA18 DB_MEMORIA[4]=PIN_AB18 DB_MEMORIA[5]=PIN_AA17 DB_MEMORIA[6]=PIN_U22	-
DB_CHAVES	display HEX2	DB_CHAVES[0]=PIN_Y19	-

		DB_CHAVES[1]=PIN_AB17 DB_CHAVES[2]=PIN_AA10 DB_CHAVES[3]=PIN_Y14 DB_CHAVES[4]=PIN_V14 DB_CHAVES[5]=PIN_AB22 DB_CHAVES[6]=PIN_AB21	
DB_ESTADO	display HEX5	DB_ESTADO[0] = PIN_N9 DB_ESTADO[1] = PIN_M8 DB_ESTADO[2] = PIN_T14 DB_ESTADO[3] = PIN_P14 DB_ESTADO[4] = PIN_C1 DB_ESTADO[5] = PIN_C2 DB_ESTADO[6] = PIN_W19	-

Tabela 06 - Designação de Pinos

O circuito sintetizado pelo quartus está abaixo:

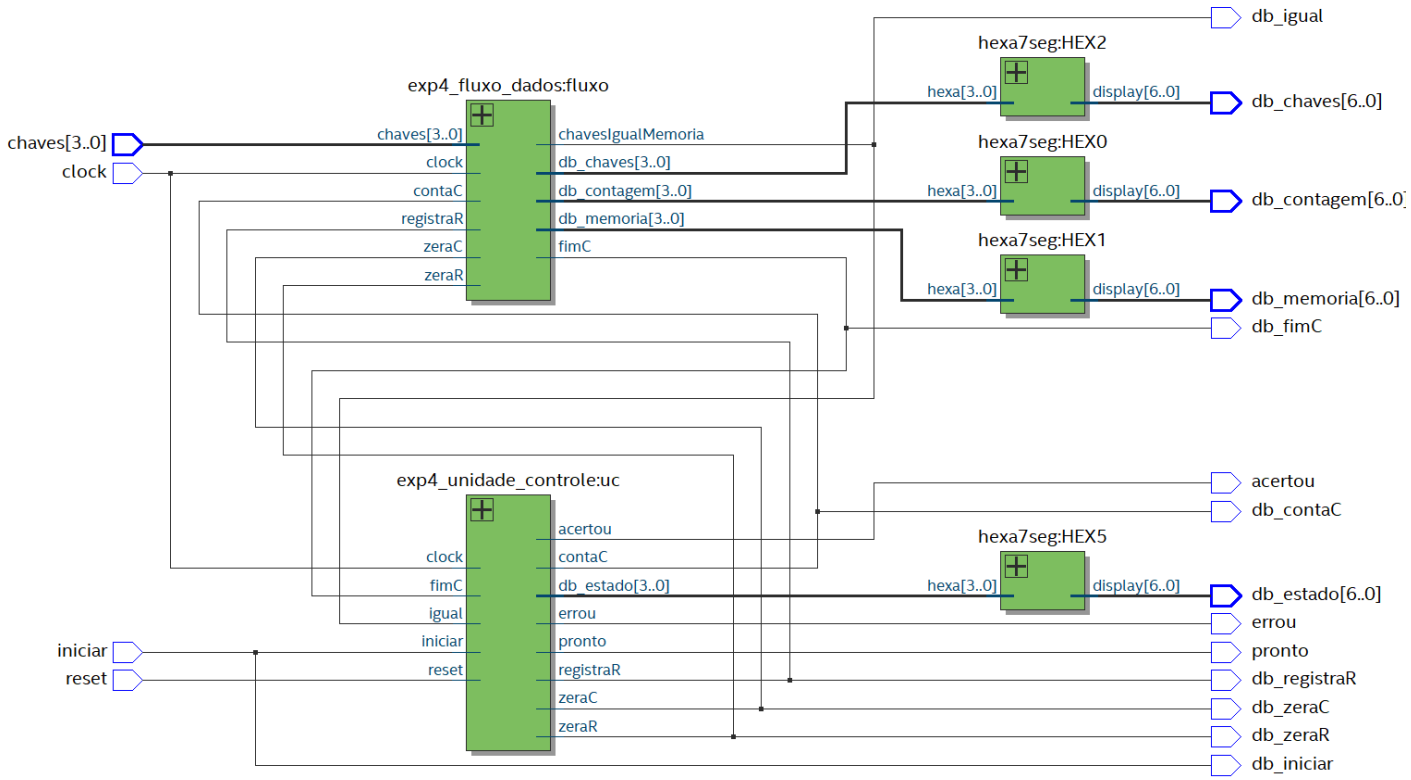

Figura 16 - Circuito do desafio sintetizado

Feita a sintetização e a programação da FPGA, basta conectar os pinos corretos na Analog Discovery 2. A montagem do circuito é análoga à figura 11. Os testes a serem realizados serão os testes das tabela 04 e tabela 05. Por questão de legibilidade, optou-se por não reescrever a tabela.