



Escola Politécnica da Universidade de São Paulo  
Departamento de Engenharia de Computação e Sistemas Digitais  
PCS 3635 – Laboratório Digital 1  
Turma 2 - Professor Edson Toshimi Midorikawa  
Bancada A3

# Experiência 2

Um fluxo de dados simples

**Nome Completo**

Igor Pontes Tresolavy  
Thiago Antici de Souza Rodrigues

**nºUSP**

12553646  
12551412

**São Paulo**  
**2023/01/15**

# Introdução

Esta experiência consiste no primeiro contato com FPGAs e ferramentas de simulação e síntese de descrições de hardware em VHDL da disciplina de Laboratório Digital I.

Partindo dos circuitos integrados vistos na primeira experiência e de descrições em VHDL de seus funcionamentos, um fluxo de dados é confeccionado (também em VHDL), simulado (através do uso do ModelSim/QuartaSim, para fim de testes) e sintetizado em FPGA (utilizando o Quartus).

## Descrição do Projeto

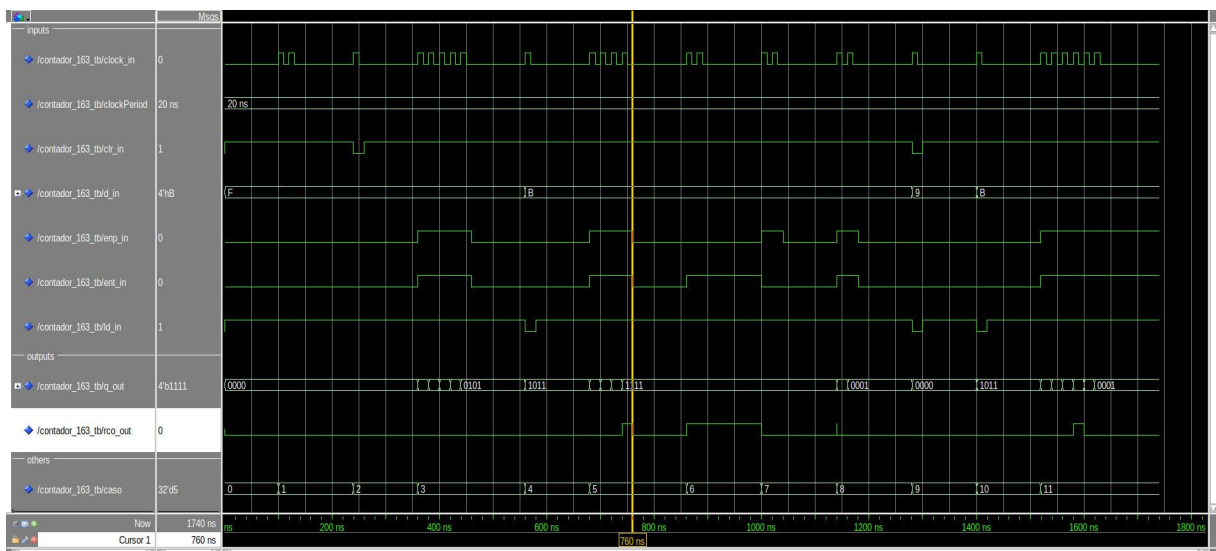
### 1.1.1. Componente contador\_163

Estudando o código-fonte do componente contador\_163.vhd, concluiu-se que seu funcionamento é idêntico ao do circuito integrado 74ALS163 (ou seu equivalente CMOS, 74HC163), visto na primeira experiência dessa disciplina.

Em seguida, executou-se o testbench e notou-se que, no caso de teste 10, a saída esperada não condiz com a obtida em simulação. No entanto, a mesma inconsistência está presente na tabela 1 da apostila da experiência 2, o que indica erro na confecção da mesma. Caso se leve em consideração o sinal D=1011, a saída do caso 10 de teste, (e, por conseguinte, o caso 11, com Q=0001 e RCO=0) estão condizentes com o funcionamento correto do contador.

Teste	Descrição	Sinais de Entradas	Saídas esperadas
condições iniciais	Todas as entradas inativadas	CLR=1, LD=1, D=0000, ENP=0, ENT=0, CLOCK=0	Q=0000, RCO=0
1	Aciona clock 2x com entradas inativadas	CLR=1, LD=1, D=0000, ENP=0, ENT=0, acionar CLOCK 2 vezes	Q=0000, RCO=0
2	Aciona clock (somente) com clear ativado	CLR=0, LD=1, D=0000, ENP=0, ENT=0, acionar CLOCK	Q=0000, RCO=0
3	Aciona clock 5x com sinais de enable ativados	CLR=1, LD=1, D=0000, ENP=1, ENT=1, acionar CLOCK 5 vezes	Q=0101, RCO=0
4	Aciona clock com load ativado e dado=1011	CLR=1, LD=0, D=1011, ENP=0, ENT=0, acionar CLOCK	Q=1011, RCO=0
5	Aciona clock 4x com sinais de enable ativados	CLR=1, LD=1, D=0000, ENP=1, ENT=1, acionar CLOCK 4 vezes	Q=1111, RCO=1
6	Aciona clock 2x com enp=0 e ent=1	CLR=1, LD=1, D=0000, ENP=0, ENT=1, acionar CLOCK 2 vezes	Q=1111, RCO=1
7	Aciona clock 2x com enp=1 e ent=0	CLR=1, LD=1, D=0000, ENP=1, ENT=0, acionar CLOCK 2 vezes	Q=1111, RCO=0
8	Aciona clock 2x com sinais de enable ativados	CLR=1, LD=1, D=0000, ENP=1, ENT=1, acionar CLOCK 2 vezes	Q=0001, RCO=0
9	Aciona clock com clear e load ativados e dado=1001	CLR=0, LD=0, D=1001, ENP=1, ENT=1, acionar CLOCK	Q=0000, RCO=0
10	Aciona clock com load ativado e dado=1011	CLR=1, LD=0, D=1001, ENP=1, ENT=1, acionar CLOCK	Q=1001, RCO=0
11	Aciona clock 6x com sinais de enable ativados	CLR=1, LD=1, D=1001, ENP=1, ENT=1, acionar CLOCK 6 vezes	Q=1111, RCO=1

Tabela 1 da apostila - plano de testes para o componente “contador\_163”



Forma de onda do *testbench* do componente contador\_163

**nota:** foi necessário incluir um comando que utiliza o sinal “caso” (um *assert caso = 0*) em todos os *testbenches* que possuíam esse sinal, para que fosse possível traçar sua forma de onda (caso contrário, o sinal não aparece na aba de *Objects* do QuestaSim).

### 1.1.2. Componente comparador\_85

Analogamente ao caso do componente anterior, a descrição VHDL do componente comparador\_85 se comporta identicamente ao comparador 7485, cujo circuito integrado também foi utilizado na primeira experiência da disciplina.

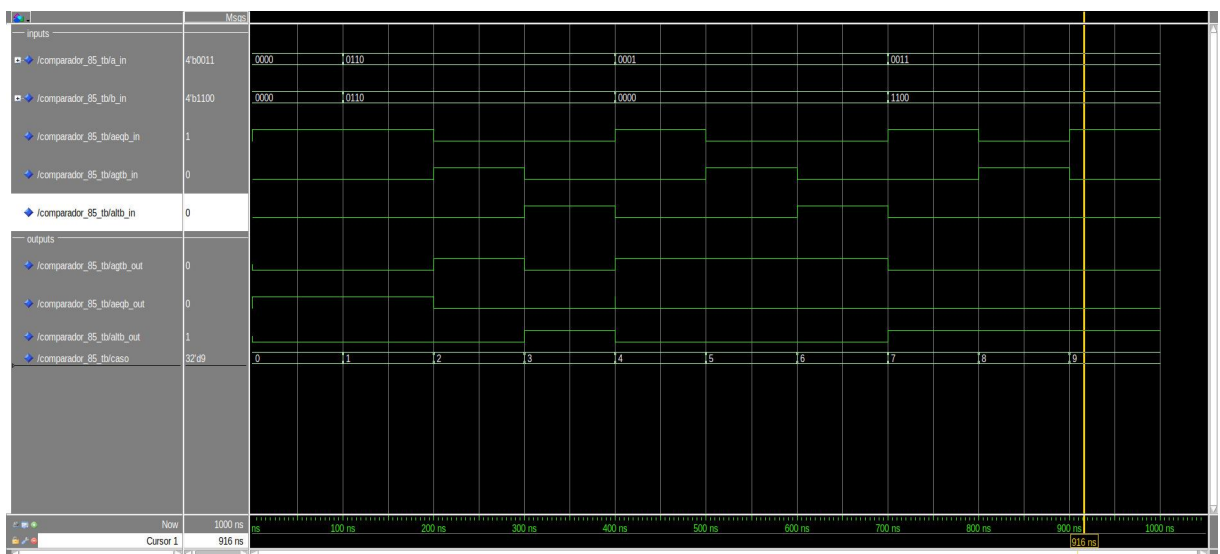
Teste	Sinais de Entradas	Saídas esperadas
condições iniciais	A>Bin=0, A=Bin=1, A<Bin=0, A=0000, B=0000	A>Bout=0, A=Bin=1, A<Bout=0
1	A>Bin=0, A=Bin=1, A<Bin=0, A=0110, B=0110	A>Bout=0, A=Bin=1, A<Bout=0
2	A>Bin=1, A=Bin=0, A<Bin=0, A=0110, B=0110	A>Bout=1, A=Bin=0, A<Bout=0
3	A>Bin=0, A=Bin=0, A<Bin=1, A=0110, B=0110	A>Bout=0, A=Bin=0, A<Bout=1
4	A>Bin=0, A=Bin=1, A<Bin=0, A=0001, B=0000	A>Bout=?, A=Bin=?, A<Bout=?
5	A>Bin=1, A=Bin=0, A<Bin=0, A=0001, B=0000	A>Bout=?, A=Bin=?, A<Bout=?
6	A>Bin=0, A=Bin=0, A<Bin=1, A=0001, B=0000	A>Bout=?, A=Bin=?, A<Bout=?
7	A>Bin=0, A=Bin=1, A<Bin=0, A=0011, B=1100	A>Bout=?, A=Bin=?, A<Bout=?
8	A>Bin=1, A=Bin=0, A<Bin=0, A=0011, B=1100	A>Bout=?, A=Bin=?, A<Bout=?
9	A>Bin=0, A=Bin=1, A<Bin=0, A=0011, B=1100	A>Bout=?, A=Bin=?, A<Bout=?

Tabela 2 da apostila - plano de testes para o componente “comparador\_85”

Utilizando o testbench dado, completou-se a tabela 2 da apostila da seguinte maneira:

Teste	Entradas	Saídas
4	A>Bin=0, A=Bin=1, A<Bin=0, A=0001, B=0000	A>Bout=1, A=Bout=0, A<Bout=0
5	A>Bin=1, A=Bin=0, A<Bin=0, A=0001, B=0000	A>Bout=1, A=Bout=0, A<Bout=0

6	A>Bin=0, A=Bin=0, A<Bin=1, A=0001, B=0000	A>Bout=1, A=Bout=0, A<Bout=0
7	A>Bin=0, A=Bin=1, A<Bin=0, A=0011, B=1100	A>Bout=0, A=Bout=0, A<Bout=1
8	A>Bin=1, A=Bin=0, A<Bin=0, A=0011, B=1100	A>Bout=0, A=Bout=0, A<Bout=1
9	A>Bin=0, A=Bin=1, A<Bin=0, A=0011, B=1100	A>Bout=0, A=Bout=0, A<Bout=1



Forma de onda do *testbench* do componente comparador\_85

### 1.1.3. Componente ram\_16x4

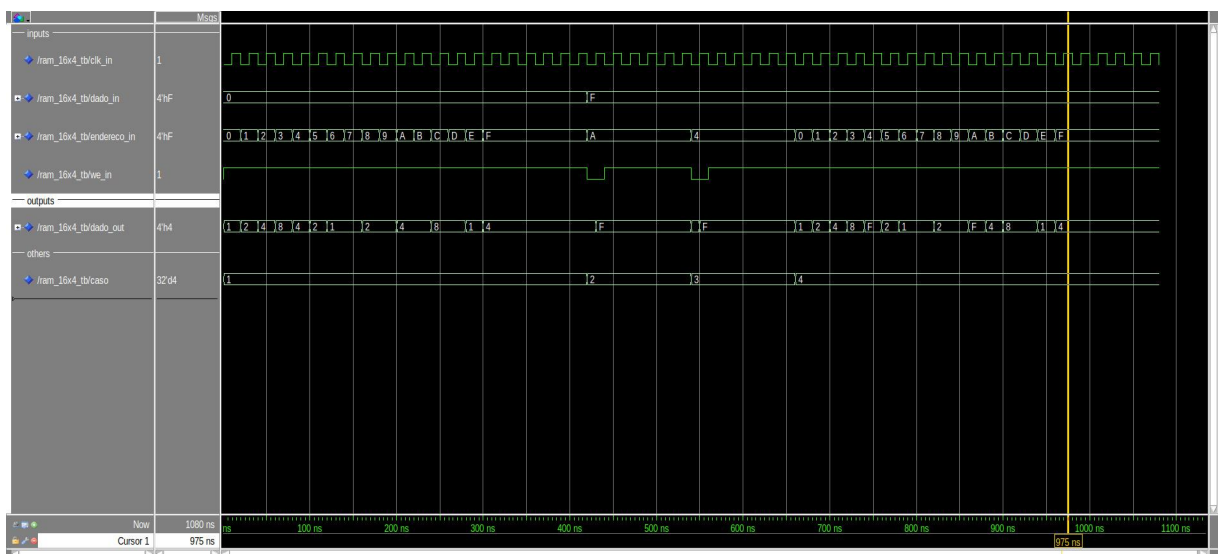
Teste	Descrição	Sinais de Entradas	Saída esperada
condições iniciais	Entradas inativadas com memória selecionada	clk=0, endereço=0000, dado_entrada=0000, we=1, ce=0	dado_saida=0001
1	Leitura de todas as posições da memória	endereço varia de 0000 até 1111	dado_saida conforme MIF
2	Escrita do dado 1111 na posição 10 da memória	dado=1111, endereço=1010, we=0, clk=↑	dado_saida=1111
3	Escrita do dado 1111 na posição 4 da memória	dado=1111, endereço=0100, we=0, clk=↑	?
4	Leitura de todas as posições da memória	endereço varia de 0000 até 1111	?

Tabela 3 da apostila - plano de testes do componente “ram\_16x4”

A partir do código-fonte do componente, concluiu-se que o mesmo se trata de uma memória ram com 4 bits de endereçamento e dados de 4 bits. Além disso, a memória possui os sinais de controle equivalentes a um “clock enable” e um “write enable”, ambos ativos baixos, que controlam a escrita de dados na memória (para uma operação de escrita, ambos devem estar em nível lógico 0 na borda de subida do clock).

Completou-se a tabela 3 da apostila da seguinte maneira:

Teste	Entradas	Saídas
3	dado=1111, endereco=0100, we=0, clk=↑	dado_saida=1111
4	endereco varia de 0000 até 1111	dado_saida conforme MIF, exceto pelos endereços 4 e 10, cujos valores condizem com o que foi escrito nos casos de teste anteriores (dado=1111)



Forma de onda do *testbench* do componente ram\_16x4

## 1.2. Atividade 2 - Projeto e Simulação de um Fluxo de Dados

Primeiramente, completou-se o código VHDL fornecido da maneira especificada abaixo:

- Componente ram\_16x4:

```
81 component ram_16x4 is
82   port (
83     clk          : in  std_logic;
84     endereco      : in  std_logic_vector(3 downto 0);
85     dado_entrada  : in  std_logic_vector(3 downto 0);
86     we           : in  std_logic;
87     ce           : in  std_logic;
88     dado_saida    : out std_logic_vector(3 downto 0)
89   );
90 end component;
```

- Interligação das portas de entrada do comparador com a saída da memória e os sinais das chaves:

```
111 comparador: comparador_85
112   port map (
113     i_A3 => s_dado(3),
114     i_B3 => chaves(3),
115     i_A2 => s_dado(2),
116     i_B2 => chaves(2),
117     i_A1 => s_dado(1),
118     i_B1 => chaves(1),
119     i_A0 => s_dado(0),
120     i_B0 => chaves(0),
121     i_AGTB => '0',
122     i_ALTB => '0',
123     i_AEQB => '1',
124     o_AGTB => open, -- saidas nao usadas
125     o_ALTB => open,
126     o_AEQB => igual
127   );
```

- Interligação entre as portas de entrada e saída da memória, e as entradas do comparador e saídas do contador:

```
129 -- memoria: entity work.ram_16x4 (ram_mif) -- usar esta linha para Intel Quartus
130 memoria: entity work.ram_16x4 (ram_modelsim) -- usar arquitetura para ModelSim
131   port map (
132     clk          => clock,
133     endereco      => s_endereco,
134     dado_entrada  => chaves,
135     we           => s_not_escreve, -- we ativo em baixo
136     ce           => '0',
137     dado_saida    => s_dado
138   );
```

- Finalmente, para os sinais de debug:

```
140 db_contagem <= s_endereco;
141 db_memoria  <= s_dado;
```

- Plano de testes:

#	Operação	Sinais de controle	Resultado esperado	Resultado observado
c.i.	<i>Condições iniciais</i>	clock=0 enable=0 reset=0 escreve=0 chaves=0000	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	
1	<i>Zerar contador e observar a saída da memória</i>	reset=1 clock ↑	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	
2	<i>Ajustar chaves para 0001</i>	chaves=0001	igual=1 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	
3	<i>Ajustar chaves para 1111</i>	<completar>	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	
4	<i>Incrementar endereço para 2 e ajustar chaves para 0100</i>	<completar>	igual=1 - fim=0 db_contagem=0010 (2) db_memoria=0100 (4)	
5	<i>Ajustar chaves para 1010 e acionar sinal "escreve" (escrita do dado 1010 no endereço 2 da memória)</i>	<completar>	igual=1 - fim=0 db_contagem=0010 (2) db_memoria=1010 (A)	
6	<i>Zerar contador e ajustar chaves para 1111</i>	<completar>	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	
7	<i>Incrementar endereço para 2 (verificar dado gravado)</i>	<completar>	igual=0 - fim=0 db_contagem=0010 (2) db_memoria=1010 (4)	
8	<i>Incrementar endereço até F</i>	<completar>	igual=0 - fim=1 db_contagem=1111 (F) db_memoria=0100 (4)	
9	<i>Ajustar chaves para 0100</i>	<completar>	igual=1 - fim=1 db_contagem=1111 (F) db_memoria=0100 (4)	

Tabela 4 da apostila - plano de testes para o circuito digital da atividade 2

Para verificar-se se o circuito integrado seguia as nossas especificações, completou-se o plano de testes da tabela 4 da apostila, como mostra a tabela a seguir:

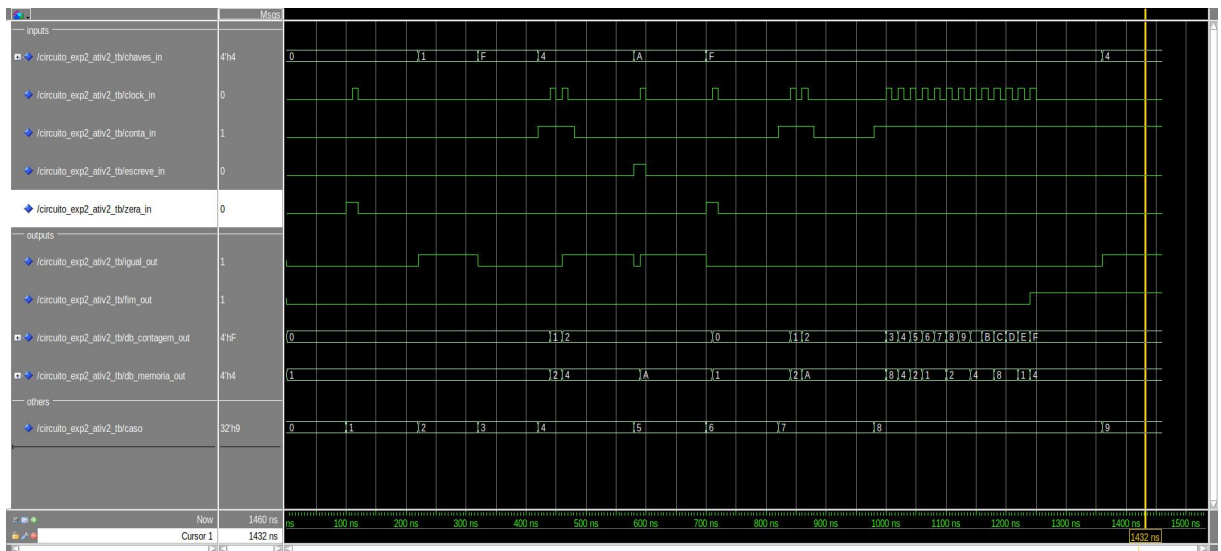
#	Operação	Sinais de controle	Resultado esperado	Resultado observado
c.i	Condições iniciais	clock=0 enable=0 reset=0 escreve=0 chaves=0000	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)
1	Zerar contador e observar a saída da memória	reset=1 clock ↑	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)
2	Ajustar chaves para 0001	chaves=0001	igual=1 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	igual=1 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)
3	Ajustar chaves para 1111	chaves=1111	igual=0 - fim=0 db_contagem=0000 (0)	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)

			db_memoria=0001 (1)	
4	Incrementar endereço para 2 e ajustar chaves para 0100	chaves=0100 enable=1 2 x clock ↑ enable=0	igual=1 - fim=0 db_contagem=0010 (2) db_memoria=0100 (4)	igual=1 - fim=0 db_contagem=0010 (2) db_memoria=0100 (4)
5	Ajustar chaves para 1010 e acionar sinal "escreve" (escrita do dado 1010 no endereço 2 da memória)	chaves=1010 escreve=1 clock ↑ escreve=0	igual=1 - fim=0 db_contagem=0010 (2) db_memoria=1010 (A)	igual=1 - fim=0 db_contagem=0010 (2) db_memoria=1010 (A)
6	Zerar contador e ajustar chaves para 1111	chaves=1111 reset=1 clock ↑ reset=0	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)	igual=0 - fim=0 db_contagem=0000 (0) db_memoria=0001 (1)
7	Incrementar endereço para 2 (verificar dado gravado)	enable=1 2 x clock ↑ enable=0	igual=0 - fim=0 db_contagem=0010 (2) db_memoria=1010 (A)	igual=0 - fim=0 db_contagem=0010 (2) db_memoria=1010 (A)
8	Incrementar endereço até F	enable =1 13 x clock ↑	igual=0 - fim=1 db_contagem=1111 (F) db_memoria=0100 (4)	igual=0 - fim=1 db_contagem=1111 (F) db_memoria=0100 (4)
9	Ajustar chaves para 0100	chaves=0100	igual=1 - fim=1 db_contagem=1111 (F) db_memoria=0100 (4)	igual=1 - fim=1 db_contagem=1111 (F) db_memoria=0100 (4)

Como exposto acima, todos os resultados observados condizem com os esperados.

Abaixo, segue a forma de onda resultante da execução do plano de testes acima. Conclui-se que a codificação foi feita de maneira correta, pois a forma de onda segue os resultados esperados.





### 1.3. Atividade 3 - Síntese e Teste do Circuito na Placa FPGA

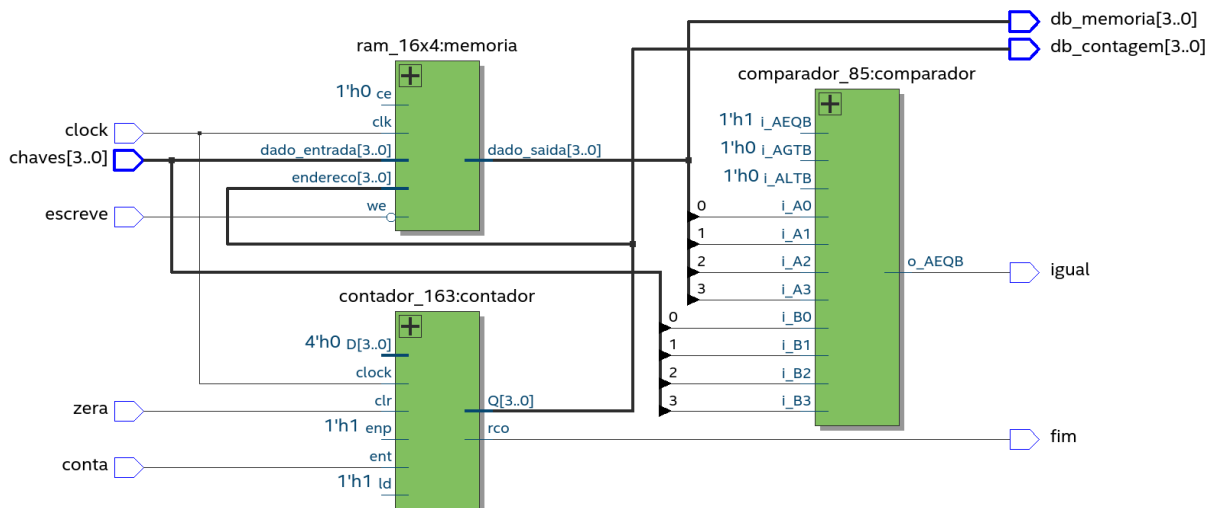
Após isso, com base em tutoriais encontrados no moodle da disciplina, procedeu-se com a criação de um projeto no Quartus e a compilação parcial (*Analysis & Synthesis*) do código fonte para a FPGA DE0-CV. Por conseguinte, realizou-se o planejamento dos pinos com o *Pin Planner*.

Sinal	Pino na Placa DE0-CV	Pino na FPGA
CLOCK	botão KEY0	PIN_U7
ZERA	botão KEY1	PIN_W9
CONTA	chave SW0	PIN_U13
CHAVE(0)	chave SW1	PIN_V13
CHAVE(1)	chave SW2	PIN_T13
CHAVE(2)	chave SW3	PIN_T12
CHAVE(3)	chave SW4	PIN_AA15
ESCREVE	chave SW5	PIN_AB15
DB_CONTAGEM(0)	Led LEDR0	PIN_AA2
DB_CONTAGEM(1)	Led LEDR1	PIN_AA1
DB_CONTAGEM(2)	Led LEDR2	PIN_W2
DB_CONTAGEM(3)	Led LEDR3	PIN_Y3
FIM	Led LEDR4	PIN_N2
DB_MEMORIA(0)	Led LEDR5	PIN_N1
DB_MEMORIA(1)	Led LEDR6	PIN_U2
DB_MEMORIA(2)	Led LEDR7	PIN_U1
DB_MEMORIA(3)	Led LEDR8	PIN_L2
IGUAL	Led LEDR9	PIN_L1

Tabela 5 da apostila - designação de pinos da FPGA DE-CV para o circuito digital da atividade 3

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
chaves[3]	Input	PIN_AA15	4A	B4A_NO	2.5 V (default)		12mA (default)	
chaves[2]	Input	PIN_T12	4A	B4A_NO	2.5 V (default)		12mA (default)	
chaves[1]	Input	PIN_T13	4A	B4A_NO	2.5 V (default)		12mA (default)	
chaves[0]	Input	PIN_V13	4A	B4A_NO	2.5 V (default)		12mA (default)	
clock	Input	PIN_U7	3A	B3A_NO	2.5 V (default)		12mA (default)	
conta	Input	PIN_U13	4A	B4A_NO	2.5 V (default)		12mA (default)	
db_contagem[3]	Output	PIN_Y3	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_contagem[2]	Output	PIN_W2	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_contagem[1]	Output	PIN_AA1	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_contagem[0]	Output	PIN_AA2	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_memoria[3]	Output	PIN_L2	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_memoria[2]	Output	PIN_U1	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_memoria[1]	Output	PIN_U2	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
db_memoria[0]	Output	PIN_N1	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
escreve	Input	PIN_AB15	4A	B4A_NO	2.5 V (default)		12mA (default)	
fim	Output	PIN_N2	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
igual	Output	PIN_L1	2A	B2A_NO	2.5 V (default)		12mA (default)	1 (default)
zera	Input	PIN_W9	3A	B3A_NO	2.5 V (default)		12mA (default)	

Ligação dos sinais da descrição nos pinos da FPGA



*Visão Register Transfer Level*

Finalmente, compilou-se por completo o projeto e gerou-se o arquivo *.qar*.

## Planejamento da Aula Prática

Na aula prática, planeja-se programar a FPGA disponibilizada no laboratório com a síntese do projeto e executar o plano de testes descrito na tabela 4 da apostila, exposta na seção anterior. Todo o experimento será documentado na forma de um relatório e entregue ao fim da aula, junto ao arquivo *.qar* gerado na etapa anterior. Caso haja tempo, pretende-se realizar, também, o desafio proposto na apostila.