

**Laboratório Digital I**

**Relatório da experiência 2**

Professores:  
 Paulo Sergio Cugnasca

Edson Midorikawa

Integrantes da T3BB3:

Arthur Pires da Fonseca - 10773096

Lucas Lopes de Paula Junior - 9344880

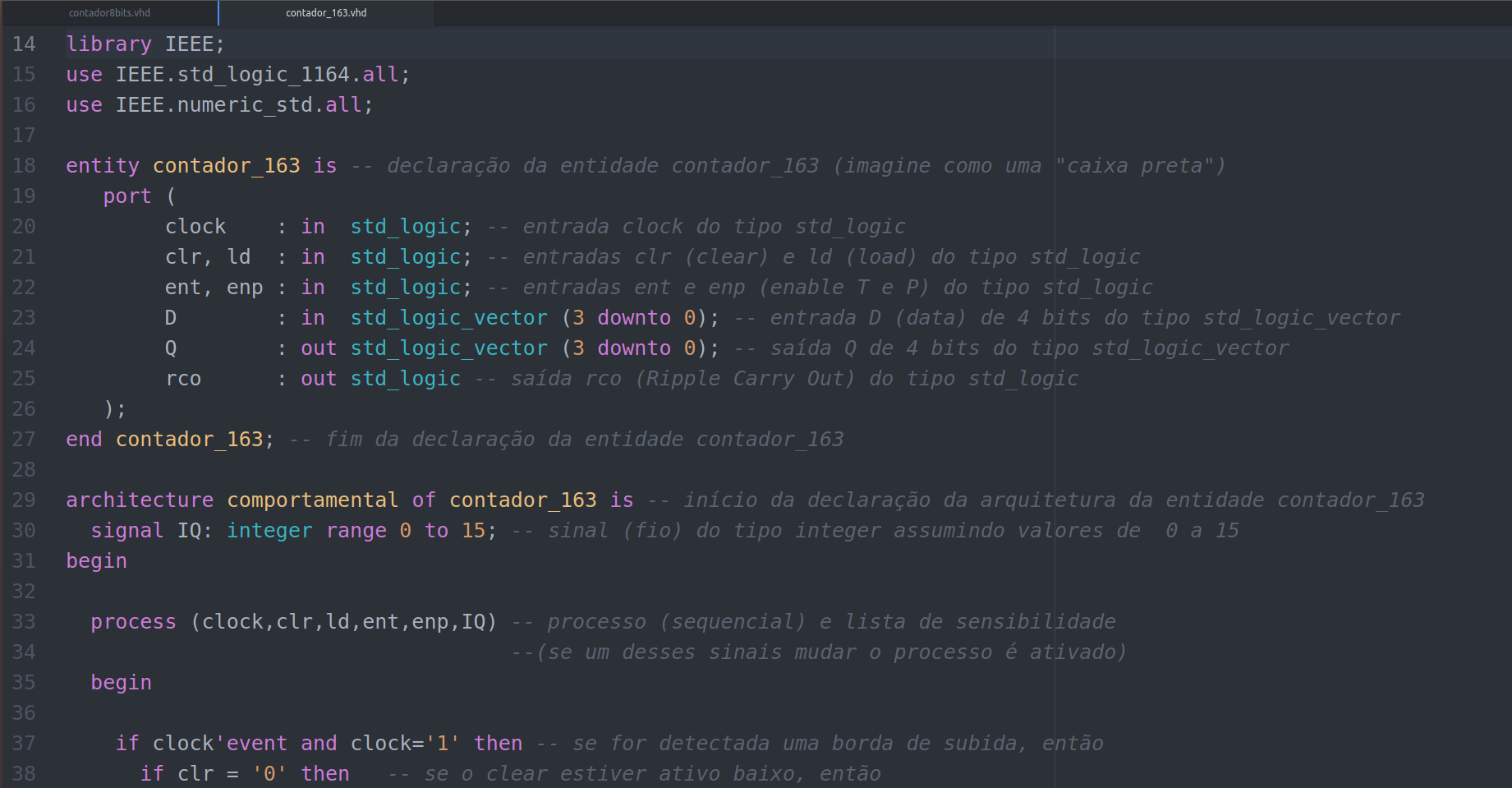
**Introdução**

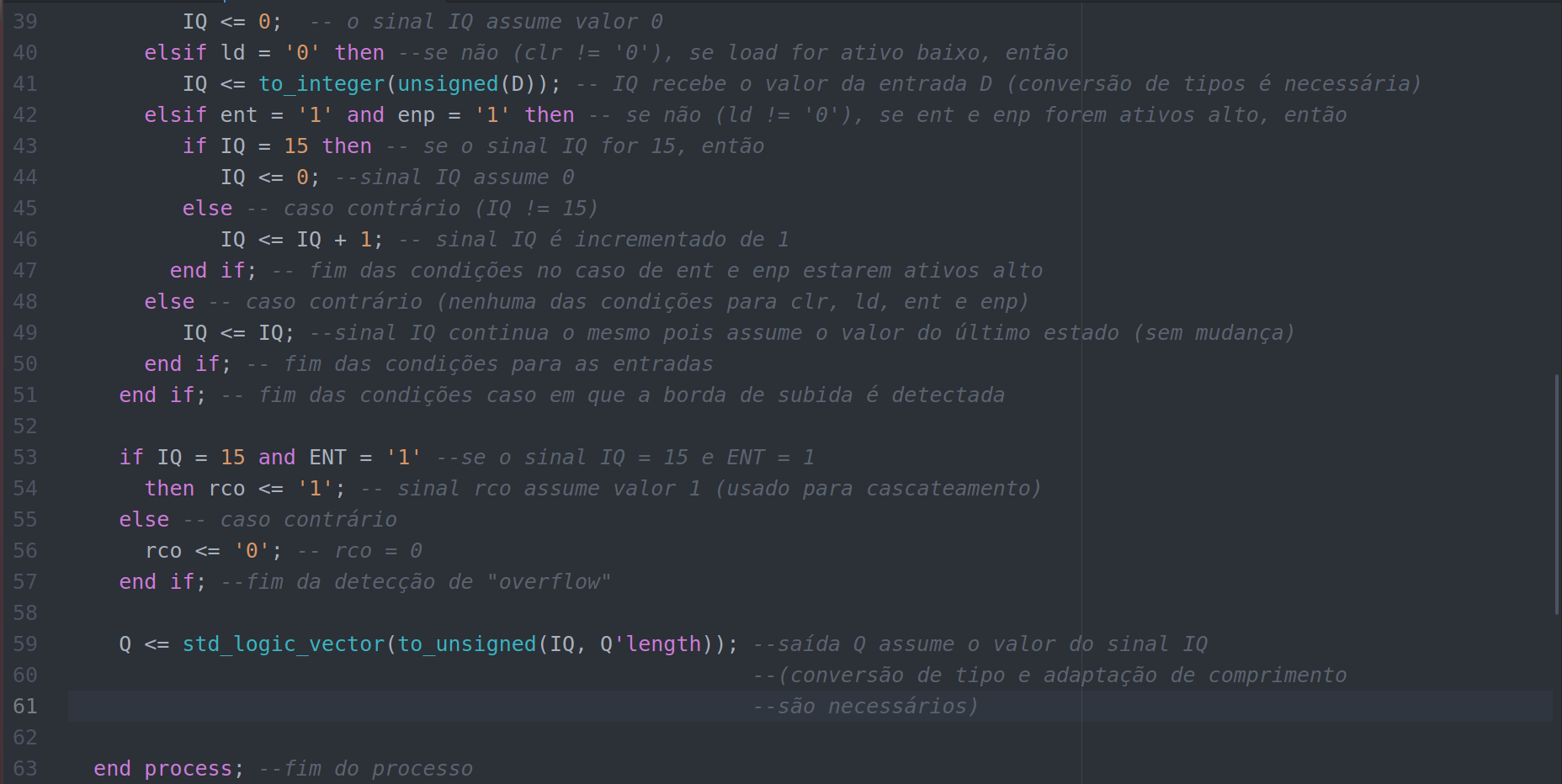
Neste experimento, faremos a simulação de um circuito contador de 8 bits usando uma placa FPGA (Altera DE0-CV) para implementar os circuitos lógicos descritos em VHDL fornecidos de antemão à dupla.

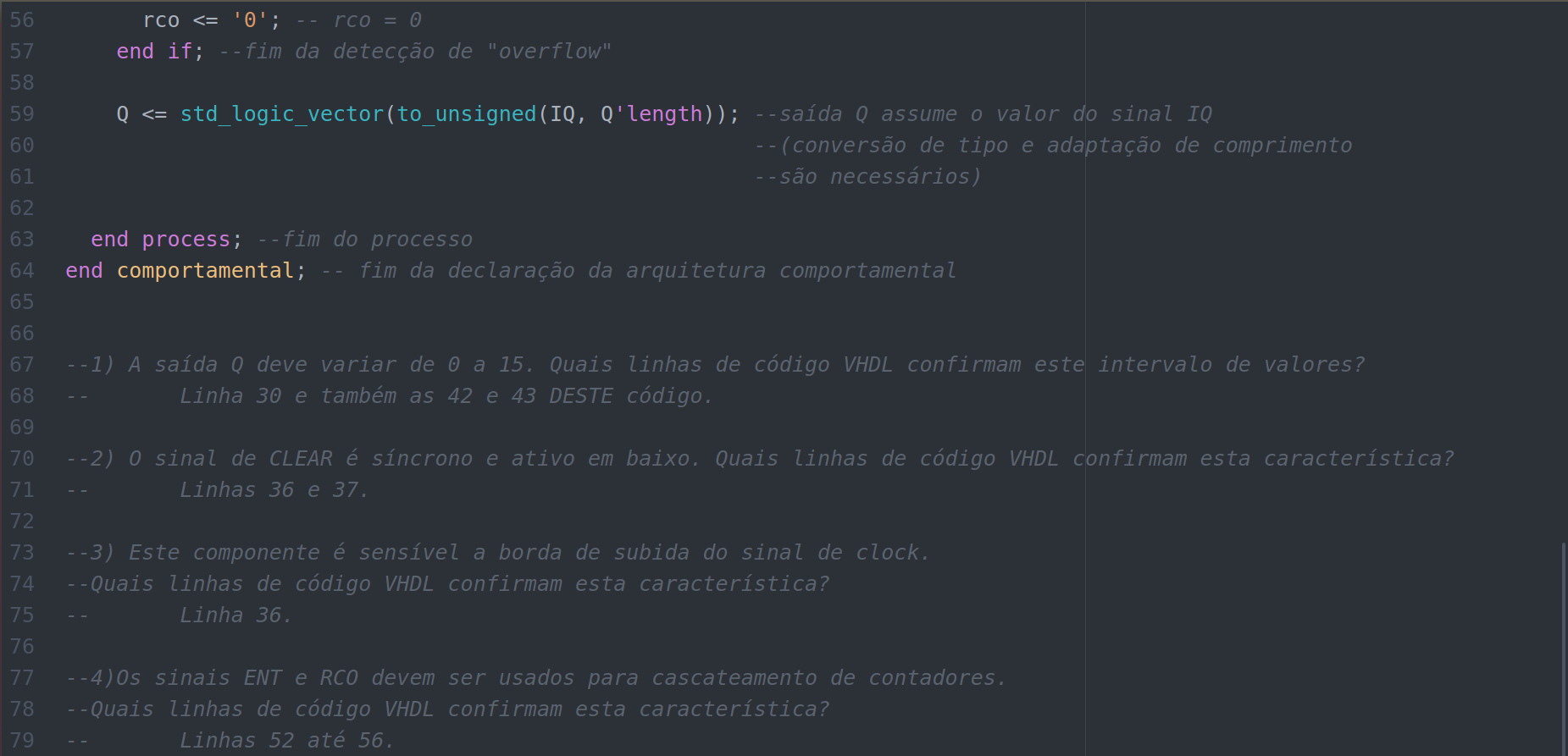
**Atividade 1 - Estudo de descrições VHDL**

Foram analisadas duas descrições VHDL, os arquivos contador\_163.vhd e contador8bits.vhd. Cada linha do código deles foi comentada, a fim de descrever qual a respectiva função.

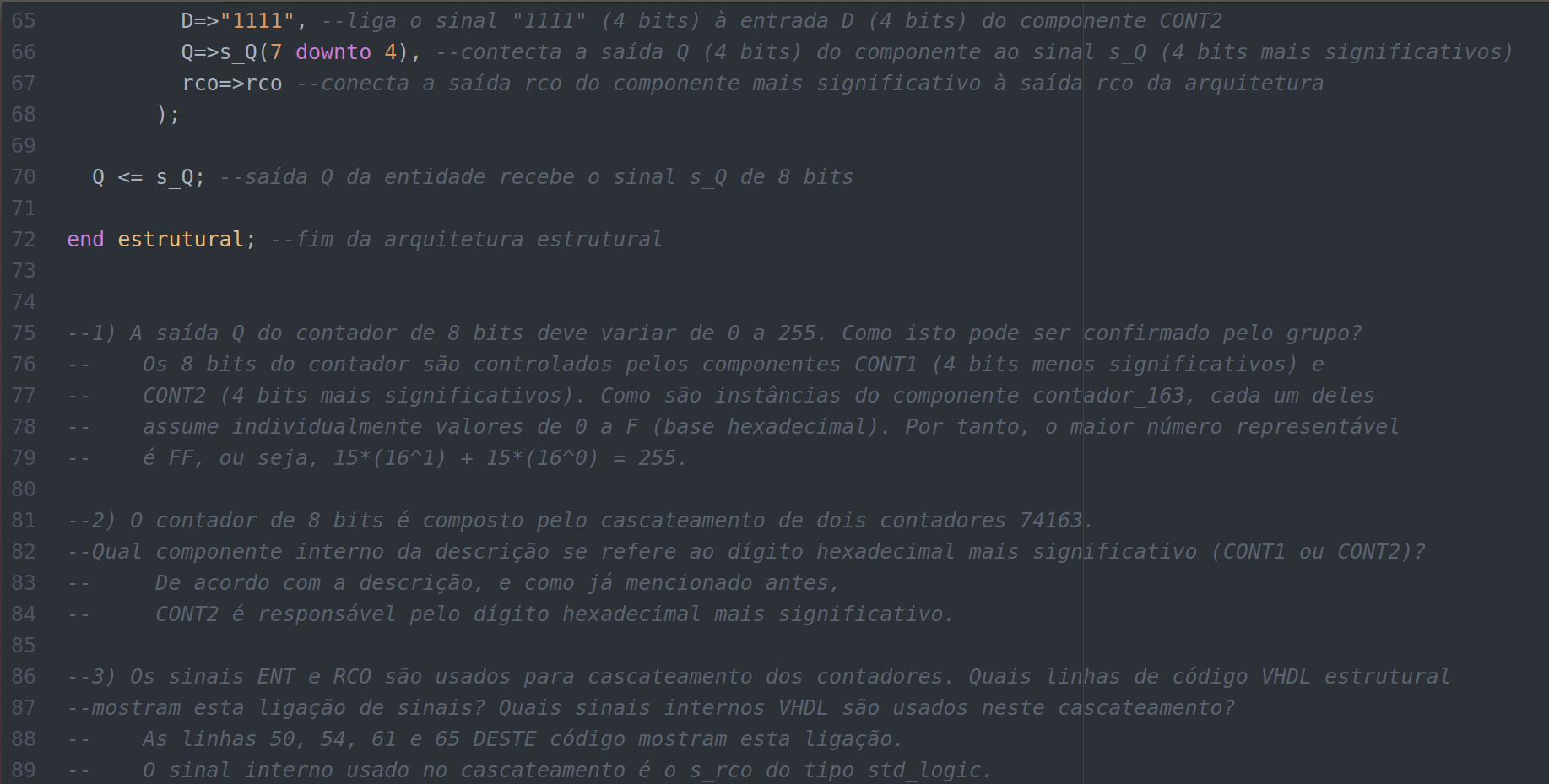
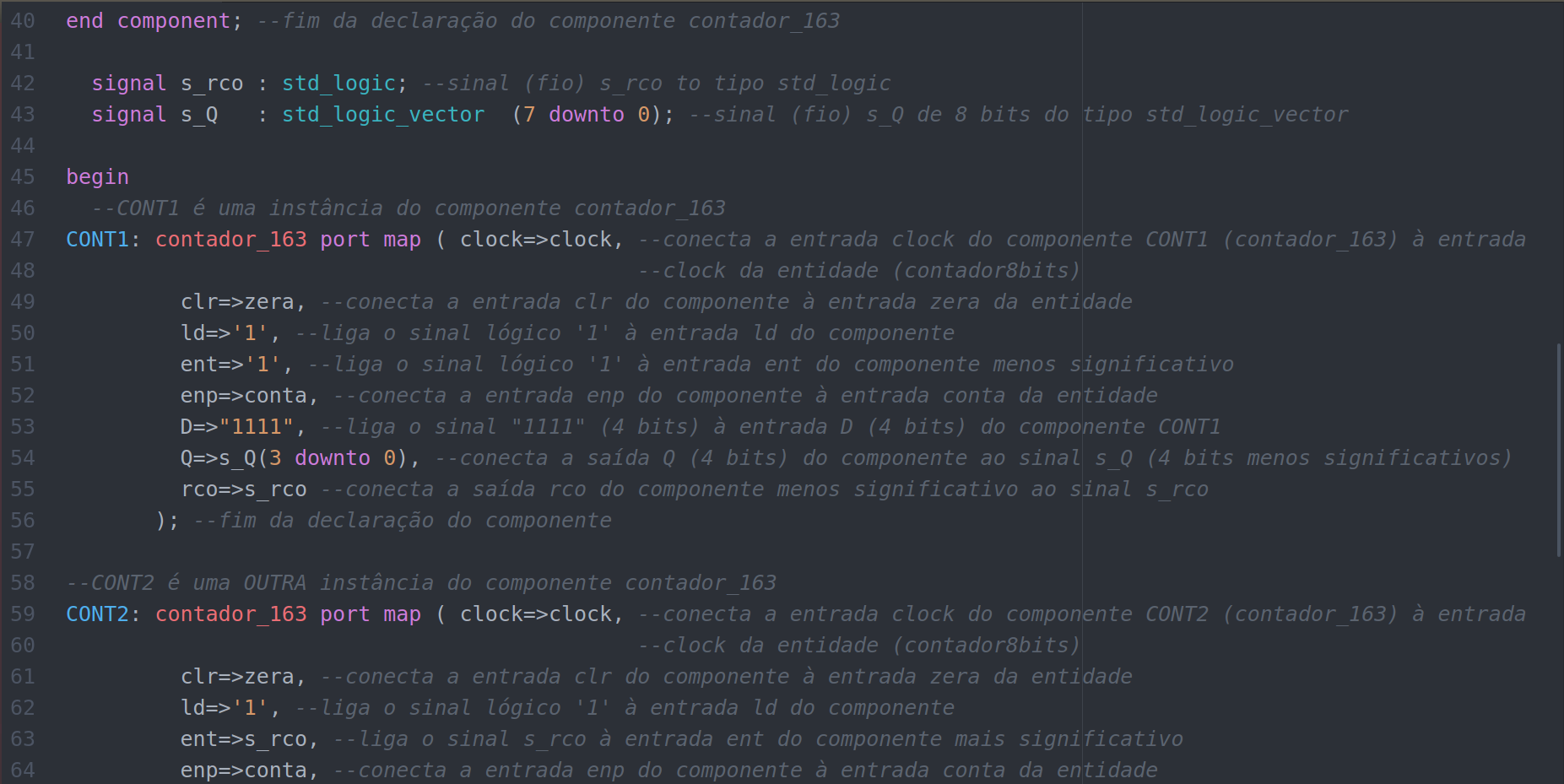
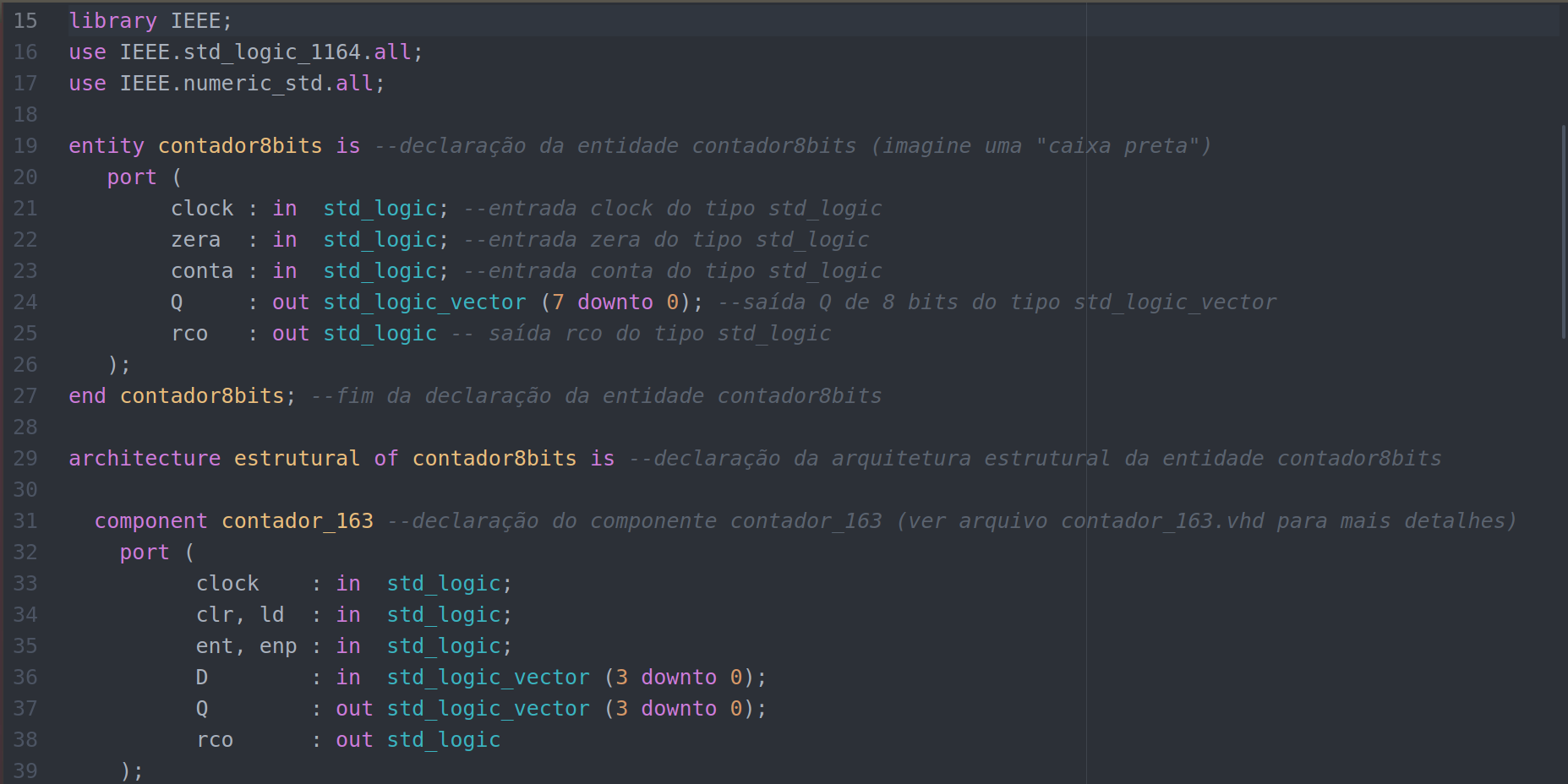
**contador\_163.vhd**





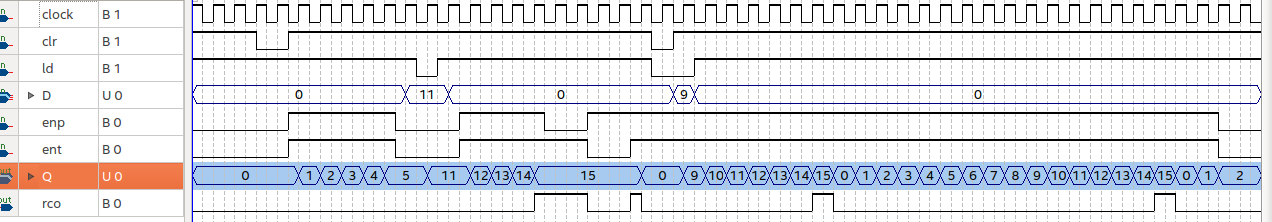


**contador8bits.vhd**

****

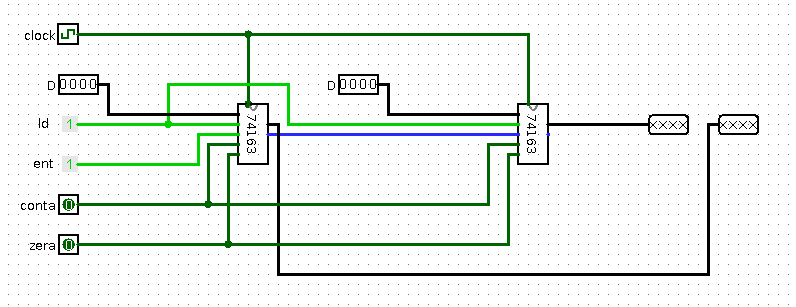
**Atividade 2 - Simulação de circuitos em VHDL**

Usando o *software* Intel Quartus Prime, simulamos o comportamento do circuito descrito pelo arquivo contador\_163.vhd, os resultados são mostrados abaixo.



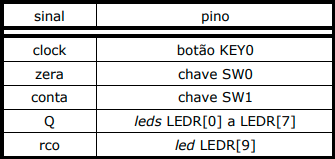
A simulação representou adequadamente a descrição VHDL, o sinal RCO apenas é ativado quando a saída Q vale 15 e o sinal ENT é 1.

Resumo de funcionamento do circuito: O contador de 8 bits é feito a partir de 2 contadores 74163 de 4 bits, que ficam cascateados. Cada contador por si só afere valores de 0 a 15 (decimal) e quando cascateados em dupla de 0 a 255. O efeito da cascata possibilita que um contador controle os 4 bits menos significativos e o outro os 4 mais significativos. O contador de 8 bits é síncrono e sensível à borda de subida do clock, assim como seus componentes de 4 bits do qual é formado.



**Atividade 3 - Simulação e síntese do Circuito Contador de 8 bits**

Esta é a relação entre os sinais da descrição VHDL e os componentes da placa FPGA que usaremos na experiência:

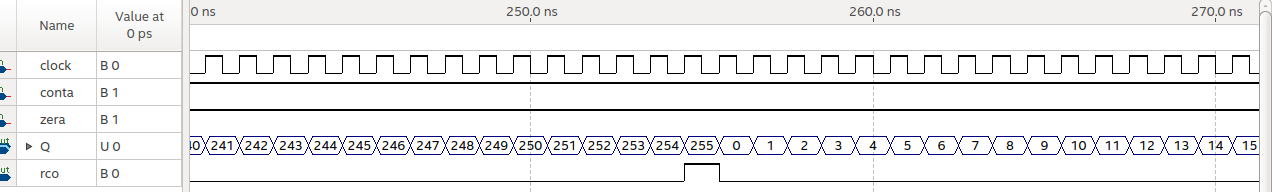


**Plano de testes**

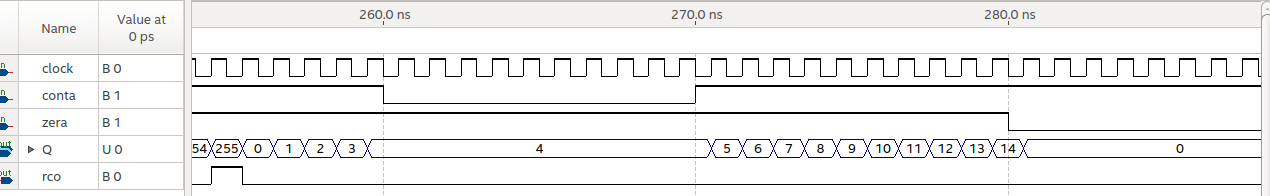
|  |  |  |
| --- | --- | --- |
| **Teste** | **Passos** | **Resultado observado** |
| zerar saída Q | ~CLR (L), CLK | todos os LEDs se apagaram |
| contar de 0 a 15 | ~CLR (H), ~LOAD (H), ENP (H), ENT (H), | configuração dos LEDs -> 00001111 |
|  | 15 \* CLK |  |
| contar de 15 a 20 | ~CLR (H), ~LOAD (H), ENP (H), ENT (H), | configuração dos LEDs -> 00010100 |
|  | 5 \* CLK |  |
| desativar ENP | ~ENP(L), CLK | nenhuma mudança nos LEDs |
| desativar ENT | ~ENT(L), CLK | todos os LEDs se apagaram |
| reativar ENP e ENT | ~ENP(H), ENT(H), CLK | configuração dos LEDs -> 00000001 |
| contar até 255 | ~CLR (H), ~LOAD (H), ENP (H), ENT (H), | todos os LEDs se ligaram |
|  | 5 \* CLK |  |
| contar mais uma vez | CLK | todos os LEDs se apagaram |

Com os resultados observados, podemos concluir que o funcionamento do circuito emulado pelo FPGA está funcionando da forma esperada.

A execução do plano de testes resultou nas seguintes formas de onda:



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



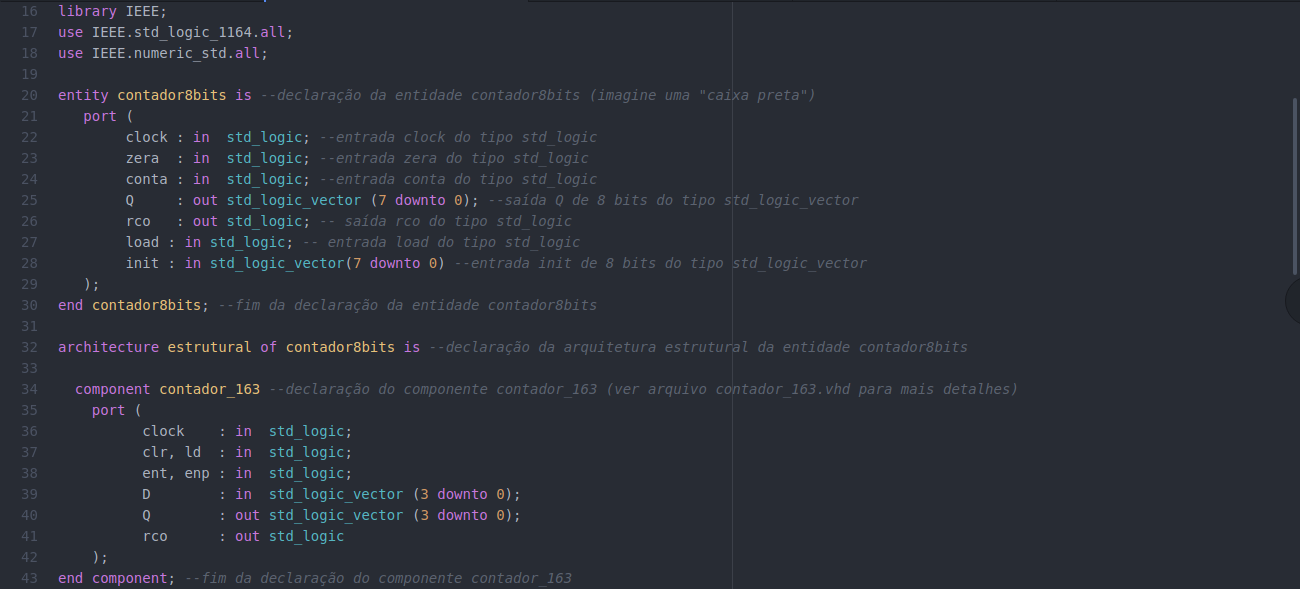
**Atividade 4 - Desafio**

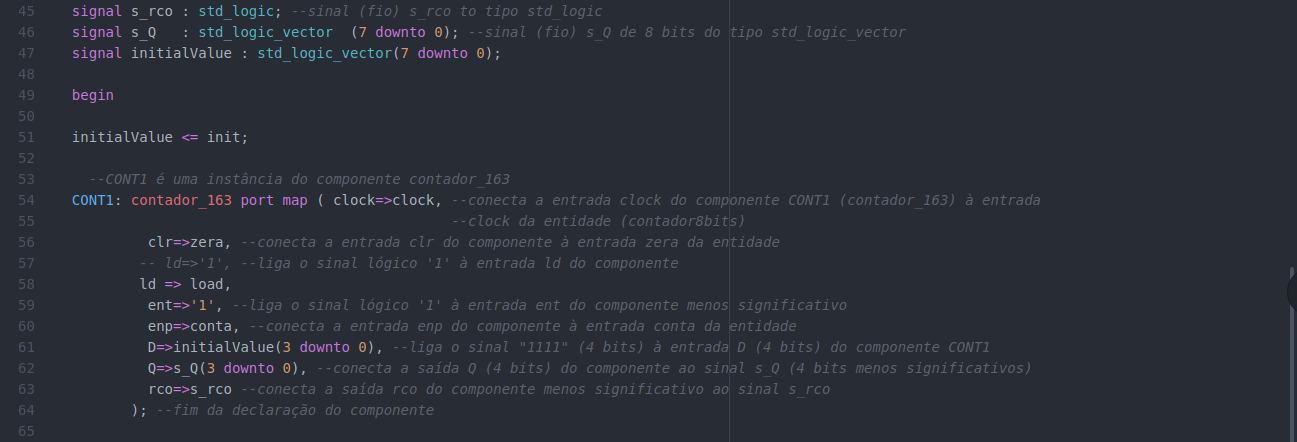
Foi solicitada uma modificação no circuito da experiência: criar um arquivo VHDL capaz de descrever um circuito contador capaz de ligar os *displays* da placa FPGA.

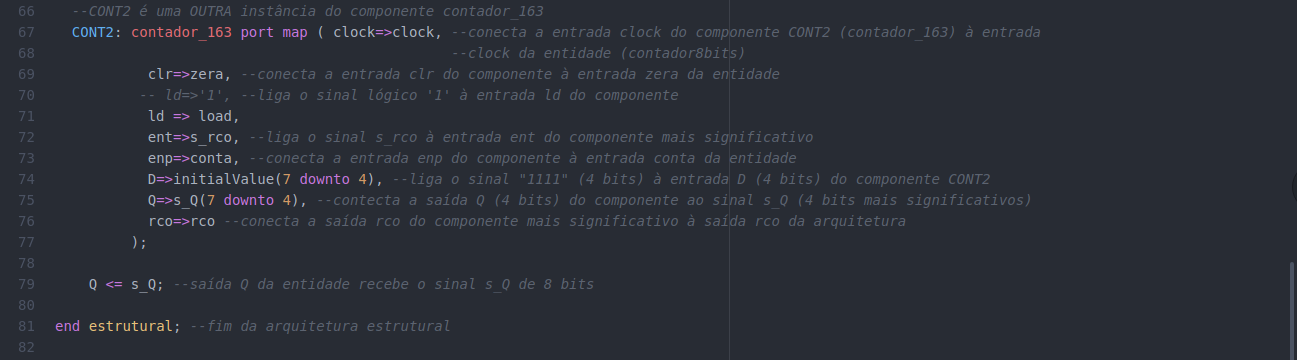
Criamos o arquivo exp2\_desafio.vhd para descrever o novo circuito, adicionamos duas entradas nele para permitir a inicialização de valores através de um sinal externo.

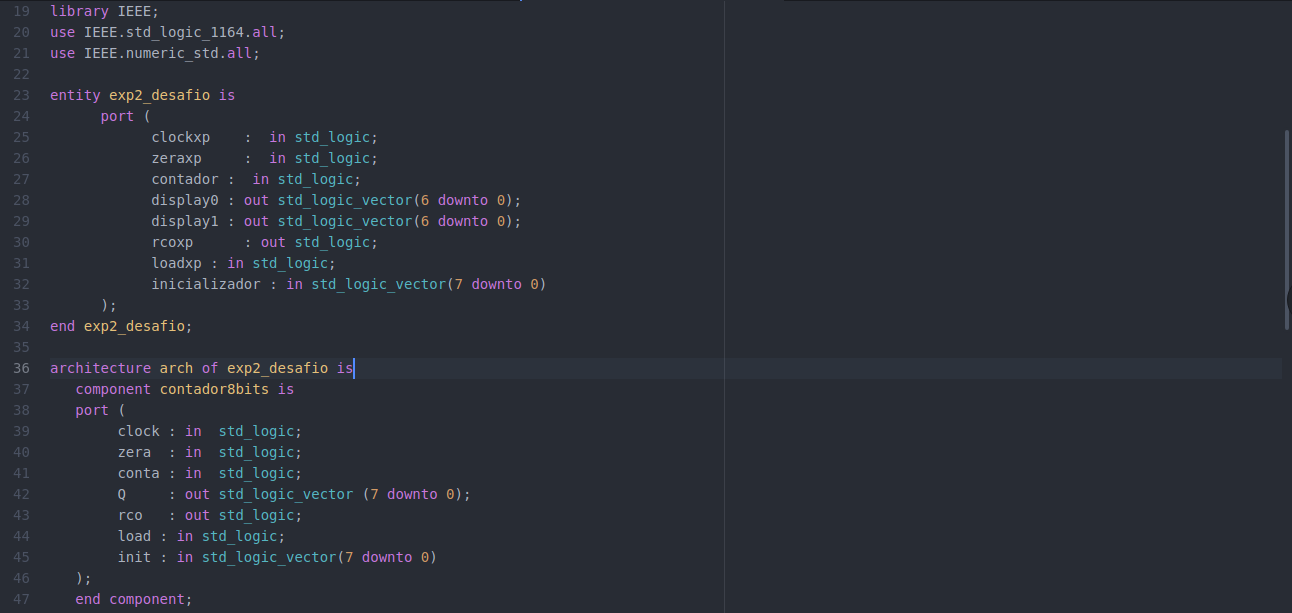
SImulamos tanto a contagem o como o carregamento dos valores no Intem Quartus Prime. Para isso foi necessária a modificação da entidade do VHDL contador8bits e do exp2\_desafio. As mudanças e as simulaçẽos são amostradas abaixo:

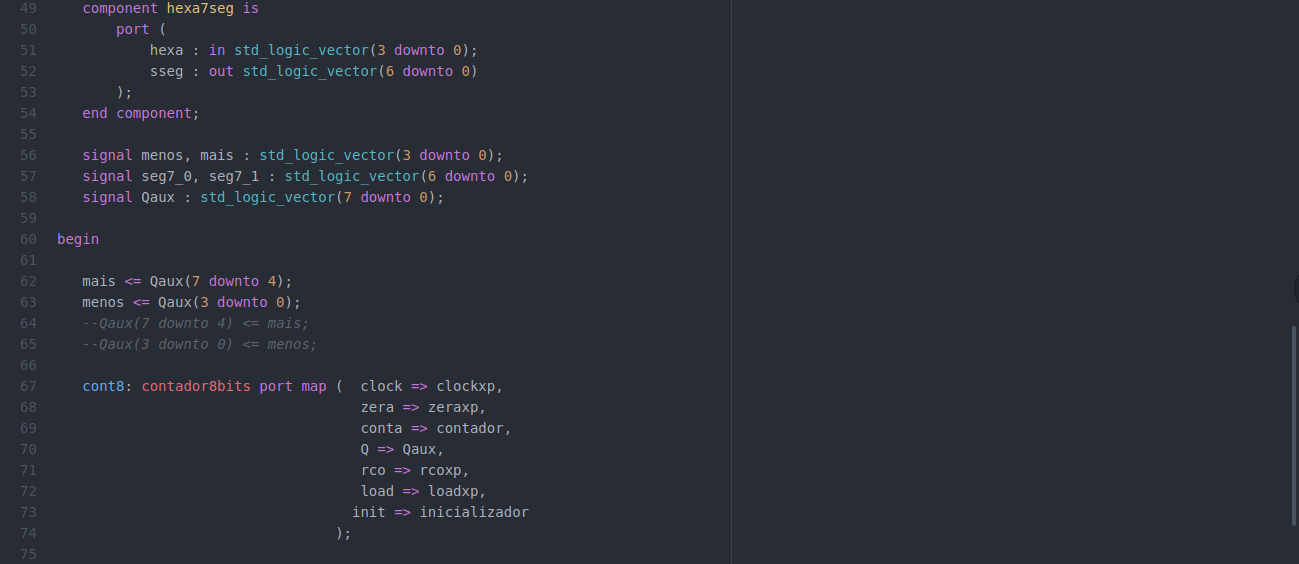
contador8bits.vhd (modificado)

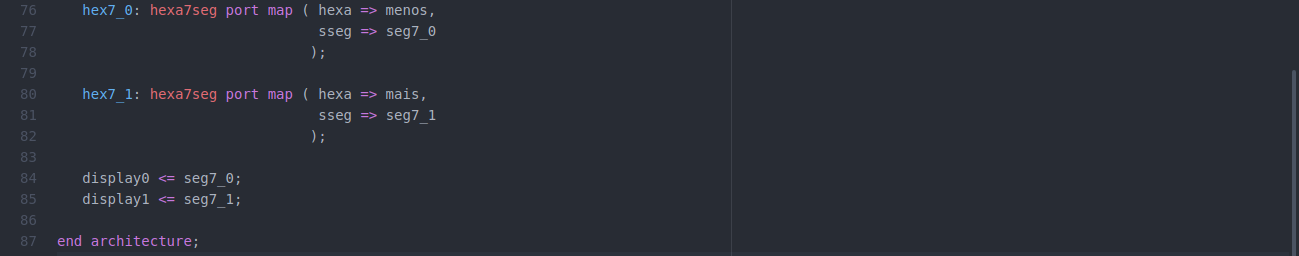


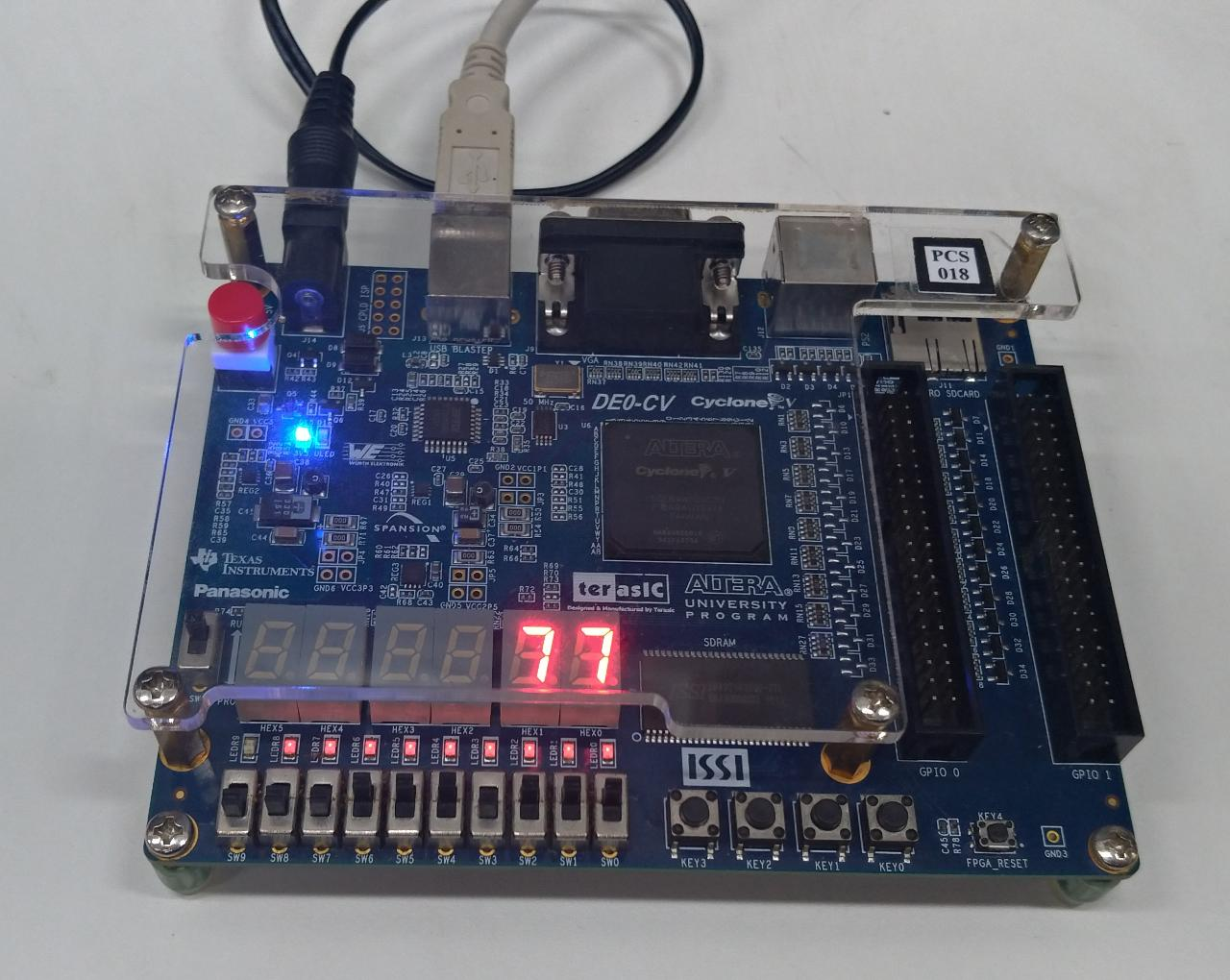




exp2\_desafio.vhd  








Simulações

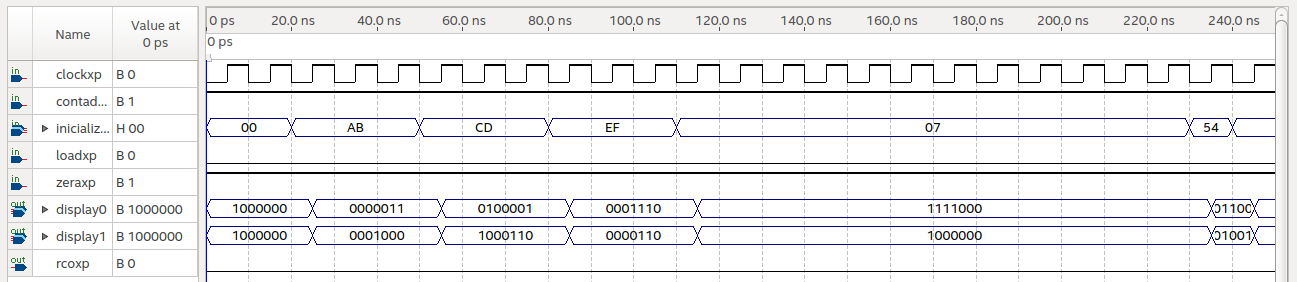


Imagem: carregamento de valores

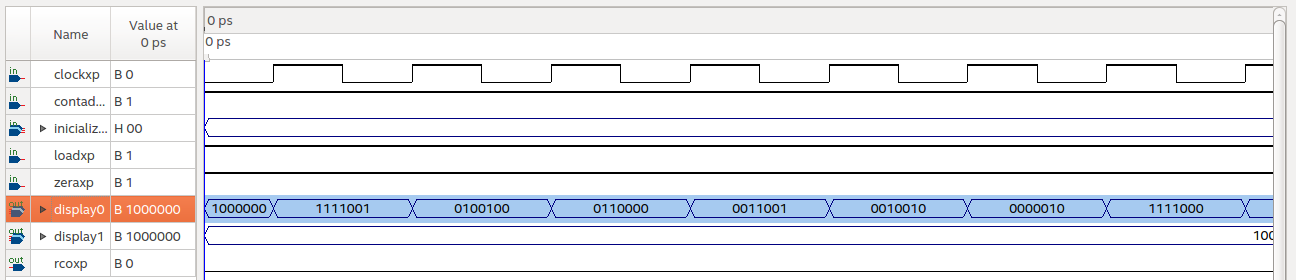


Imagem: contagem de valores (display menos significativo em evidência)

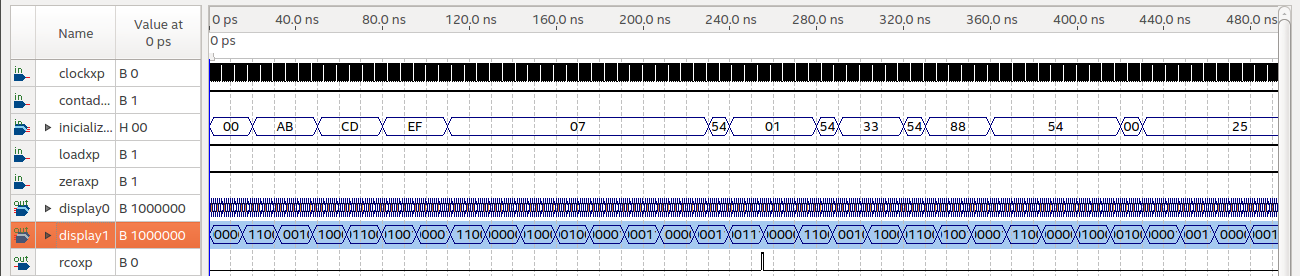


Imagem: contagem de valores (display mais significativo em evidência)

Mudamos um pouco a seleção de pinos da atividade anterior e adicionamos uma comunicação com os sinais de carregamento:

|  |  |  |
| --- | --- | --- |
| **sinal** | **nome** | **número do pino** |
| rcoxp | LEDR[9] | PIN\_L1 |
| display0[0] | HEX00 | PIN\_U21 |
| display0[1] | HEX01 | PIN\_V21 |
| display0[2] | HEX02 | PIN\_W22 |
| display0[3] | HEX03 | PIN\_W21 |
| display0[4] | HEX04 | PIN\_Y22 |
| display0[5] | HEX05 | PIN\_Y21 |
| display0[6] | HEX06 | PIN\_AA22 |
| display1[0] | HEX10 | PIN\_AA20 |
| display1[1] | HEX11 | PIN\_AB20 |
| display1[2] | HEX12 | PIN\_AA19 |
| display1[3] | HEX13 | PIN\_AA18 |
| display1[4] | HEX14 | PIN\_AB18 |
| display1[5] | HEX15 | PIN\_AA17 |
| display1[6] | HEX16 | PIN\_U22 |
| clockxp | KEY0 | PIN\_U7 |
| zera | KEY1 | PIN\_W9 |
| inicializador[0] | SW0 | PIN\_U13 |
| inicializador[1] | SW1 | PIN\_V13 |
| inicializador[2] | SW2 | PIN\_T13 |
| inicializador[3] | SW3 | PIN\_T12 |
| inicializador[4] | SW4 | PIN\_AA15 |
| inicializador[5] | SW5 | PIN\_AB15 |
| inicializador[6] | SW6 | PIN\_AA14 |
| inicializador[7] | SW7 | PIN\_AA13 |
| loadxp | SW8 | PIN\_AB13 |
| contador | SW9 | PIN\_AB12 |

|  |  |  |
| --- | --- | --- |
| **Teste** | **Passos** | **Resultado observado** |
| zerar saída | ~zera (L), clockxp | Display com valor 0x00 |
| contar de 0 a 15 | ~zera (H), ~loadxp (H),  contador (H), 15 \* clockxp | Display com valor 0x0F |
| contar de 15 a 20 | ~zera (H), ~loadxp (H), contador (H), 5 \* clockxp | Display com valor 0x14 |
| inicializar valor 0xF7 | ~loadxp (L) ~inicializador = (LLLL\_LHHH),  clockxp | Display com valor 0x07 |
| contar até 255 | ~zera (H), ~load (H), contador (H),  255 \* clockxp | Display com valor 0xFF |
| contar mais uma vez | clockxp | Display com valor 0x00 |

**Após o experimento**

**Resultados alcançados**

Conseguimos entender como funciona o 74163 e como configurá-lo afim de formar um contador com maior capacidade à partir do cascateamento de 2 ou mais deles. O desafio nos proporcionou a oportunidade de implementar o hardware na placa FPGA de forma ímpar, modificando algumas coisas como fora mostrado neste relatório. Tudo funcionou como o esperado e em concordância com as simulações.

**Pontos positivos**

Do nosso ponto de vista estamos entendendo, na prática, como funciona o projeto e implementação de hardware, bem como as etapas de simulação, teste e depuração, além de é claro, estudar o funcionamento e operação de CIs e placas FPGAs.

**Lições aprendidas**

A prática, organização e planejamento para se projetar um circuito digital (ou hardware) são coisas que andam juntas e vão além da técnica/teoria sobre como um componente funciona ou como combiná-los a fim de projetar outras coisas à partir destes.  
 *“Se o seu esforço é pequeno você provavelmente não está focando na OPORTUNIDADE. Você provavelmente está focando na OBRIGAÇÃO.”*