ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Graduação em Engenharia de Computação

PCS3635 - Laboratório Digital 1

Experiência 2 - Circuito Digital em VHDL

Bancada A3

Professor Reginaldo Arakaki

José Lucas de Melo Costa NUSP: 07335100

Lucas de Menezes Cavalcante NUSP: 10770180

São Paulo - SP

2020

1 Resumo

Este relatório retrata as observações e os dados relativos à experiência 2: Circuito Digital em VHDL. Tal atividade abrange o processo de concepção, criação, simulação e síntese de um circuito digital. Nesse sentido, a abordagem se inicia com a análise de códigos em VHDL, sua implementação no software Intel Quartus Prime, assim como a elaboração de planos de teste e formas de onda para a realização de simulações.

Ainda neste ambiente de desenvolvimento, a experiência prossegue para a síntese do projeto na placa FPGA DE0-CV.

2 Objetivo

Essa experiência tem como objetivo proporcionar o estudo e a análise do processo de criação e descrição de um projeto, compilação, simulação e síntese do sistema desenvolvido. Dessa forma, irá envolver a linguagem de descrição de hardware VHDL, utilizando o software Quartus, e a síntese se dará sobre uma placa de FPGA.

3 Planejamento

Roteiro da experiência:

1. Utilizar um dos computadores para abrir os códigos vhdl dos contadores
2. Analisar novamente os códigos, identificando as funcionalidades
3. Abrir o programa Quartus no computador do laboratório e no computador pessoal
4. Lembrar de configurar para a placa DE0-CV, terceira opção na tabela:
    a. Family: Cyclone V
    b. Device: All
    c. Package: Any
    d. Pin Count: 484
    e. Core speed grade: 7
5. Lembrar de compilar o código antes de ir para a simulação
6. Na simulação lembrar de resetar o script

Linguagem de descrição de hardware "Very High Speed Integrated Circuits"-VHDL:

Ao longo desta experiência, utilizaremos a linguagem vhdl para analisar e descrever dispositivos. Percebe-se que duas abordagens foram utilizadas para descrever os contadores da atividade: abordagem comportamental e estrutural. Vale a pena ressaltar a diferença entre tais modos de descrever um hardware. Do ponto de vista comportamental, descreveremos um sistema tendo como base suas entradas e descrevendo as saídas esperadas. Essa forma de descrição irá abstrair detalhes específicos da estrutura interna do componente e, por esse motivos, é comum utilizar **process.**

Já a descrição estrutural é mais próxima de um sistema físico, na medida em que descreverá o comportamento de um componente conectando suas entradas e saídas, evidenciando essas interconexões.

Atividade 1

Nesta atividade apresentamos duas descrições comentadas em vhdl dos contadores de 4 e 8 bits. Nota-se que a primeira descrição é comportamental, enquanto a segunda é estrutural. Os comentários foram realizados para que possamos identificar os sinais corretamente na hora da experiência.

Contador 163

Imagem 1: Código VHDL da arquitetura do contador 163 com comentários

```
29  architecture comportamental of contador_163 is
30    signal IQ: integer range 0 to 15;                    -- o código trabalha com um inteiro IQ como sinal, que
31  begin                                                  -- depois será convertido e enviado à saida Q
32
33    process (clock,clr,ld,ent,enp,IQ)
34    begin
35
36      if clock'event and clock='1' then
37        if clr='0' then   IQ <= 0;                       -- IQ é sempre zerado com clr = 0
38        elsif ld='0' then IQ <= to_integer(unsigned(D)); -- IQ recebe o valor especificado em D
39        elsif ent='1' and enp='1' then                   -- condição para o circuito contar
40          if IQ=15 then   IQ <= 0;                       -- se IQ terminou de contar, o próximo valor é 0
41          else            IQ <= IQ + 1;                  -- se não, acrescentar 1 normalmente
42          end if;
43        else              IQ <= IQ;                      -- para ENP ou ENT em 0, nada acontece
44        end if;
45      end if;
46
47      if IQ=15 and ENT='1' then rco <= '1';              -- RCO ativo indica o final da contagem
48      else                      rco <= '0';
49      end if;
50
51      Q <= std_logic_vector(to_unsigned(IQ, Q'length));  -- Q finalmente recebe o valor convertido do inteiro IQ
52
53    end process;
54  end comportamental;
```

Perguntas:

1. A saída Q deve variar de 0 a 15. Quais linhas de código VHDL confirmam este intervalo de valores?

A linha 30, que garante que o sinal de contagem inteiro IQ está contido nesse intervalo; a linha 37, que passa o 0 como início no caso que clr é ativo; e a linha 40, que define $IQ = 15$ como fim da contagem, e $IQ = 0$ como início. Já a linha 41 faz com que o contador avance de 0 até 15.

2. O sinal de CLEAR é síncrono e ativo em baixo. Quais linhas de código VHDL confirmam esta característica?

Linha 33: sinal síncrono dentro do process; linha 37: realiza sua função (reiniciar IQ para 0) com sinal 0. Outra evidência de que o sinal CLEAR é síncrono consiste no fato de que tal variável é utilizada dentro do escopo de um *if* que depende do sinal do *clock*. Além

disso, o *if* é uma estrutura sequencial, não sendo permitida em declarações concorrentes (assíncronas).

3. Este componente é sensível a borda de subida do sinal de clock. Quais linhas de código VHDL confirmam esta característica?

Linha 36, com os comandos *clock'event* and *clock='1'*. A utilização de *'event* indica que houve uma mudança no sinal do clock, ou seja, uma borda. Ao fazer *clock = 1* estamos especificando que se trata de uma borda de subida: se o sinal mudou e após a mudança ele está em 1 então a transição foi de 0 para 1.

4. Os sinais ENT e RCO devem ser usados para cascateamento de contadores. Quais linhas de código VHDL confirmam esta característica?

Linhas 47 a 49. Nestas linhas podemos perceber que o sinal *rco* representa, de fato, o *ripple-carry-out*, pois é ativado quando o contador está em seu último número. Nota-se também que este sinal é controlado pelo enable *ent*, que deve, portanto, ser usado no cascateamento de contadores.

Contador de 8 bits

Imagem 2: Código VHDL da arquitetura do contador de 8 bits com comentários

```
architecture estrutural of contador8bits is

  component contador_163           -- contador 163 instanciado como componente
    port (
          clock    : in  std_logic;
          clr, ld  : in  std_logic;
          ent, enp : in  std_logic;
          D        : in  std_logic_vector (3 downto 0);
          Q        : out std_logic_vector (3 downto 0);
          rco      : out std_logic
    );
  end component;

  signal s_rco : std_logic;                          -- sinal que conecta o RCO do
                                                     -- menos significativo ao ENT
                                                     -- do mais significativo
  signal s_Q   : std_logic_vector  (7 downto 0);     -- saída

begin

  CONT1: contador_163 port map ( clock=>clock,       -- contador menos significativo
                                 clr=>zera,
                                 ld=>'1',
                                 ent=>'1',           -- ENT sempre 1
                                 enp=>conta,
                                 D=>"1111",
                                 Q=>s_Q(3 downto 0), -- sinal de saída
                                 rco=>s_rco          -- sinal de término de contagem
                               );

  CONT2: contador_163 port map ( clock=>clock,       -- contador mais significativo
                                 clr=>zera,
                                 ld=>'1',
                                 ent=>s_rco,         -- ENT apenas ativo quando o
                                                     -- outro termina a contagem
                                 enp=>conta,
                                 D=>"1111",
                                 Q=>s_Q(7 downto 4), -- sinal de saida
                                 rco=>rco            -- término da contagem completa
                               );

  Q <= s_Q;

end estrutural;
```

1. A saída Q do contador de 8 bits deve variar de 0 a 255. Como isto pode ser confirmado pelo grupo?

Considerando que o contador de 8 bits cria duas instâncias do contador 163 já estudado, podemos inferir que a contagem com o dobro de bits variará de $0\ a\ 2^8 - 1,$ ao invés de $0\ a\ 2^4 - 1.$

2. O contador de 8 bits é composto pelo cascateamento de dois contadores 74163. Qual componente interno da descrição se refere ao dígito hexadecimal mais significativo (CONT1 ou CONT2)?

É o CONT2. Neste componente, podemos perceber que o dígito mais significativo está sendo representado internamente pelo sinal s_Q na posição 7 (*s_Q(7)*).

3. Os sinais ENT e RCO são usados para cascateamento dos contadores. Quais linhas de código VHDL estrutural mostram esta ligação de sinais? Quais sinais internos VHDL são usados neste cascateamento?

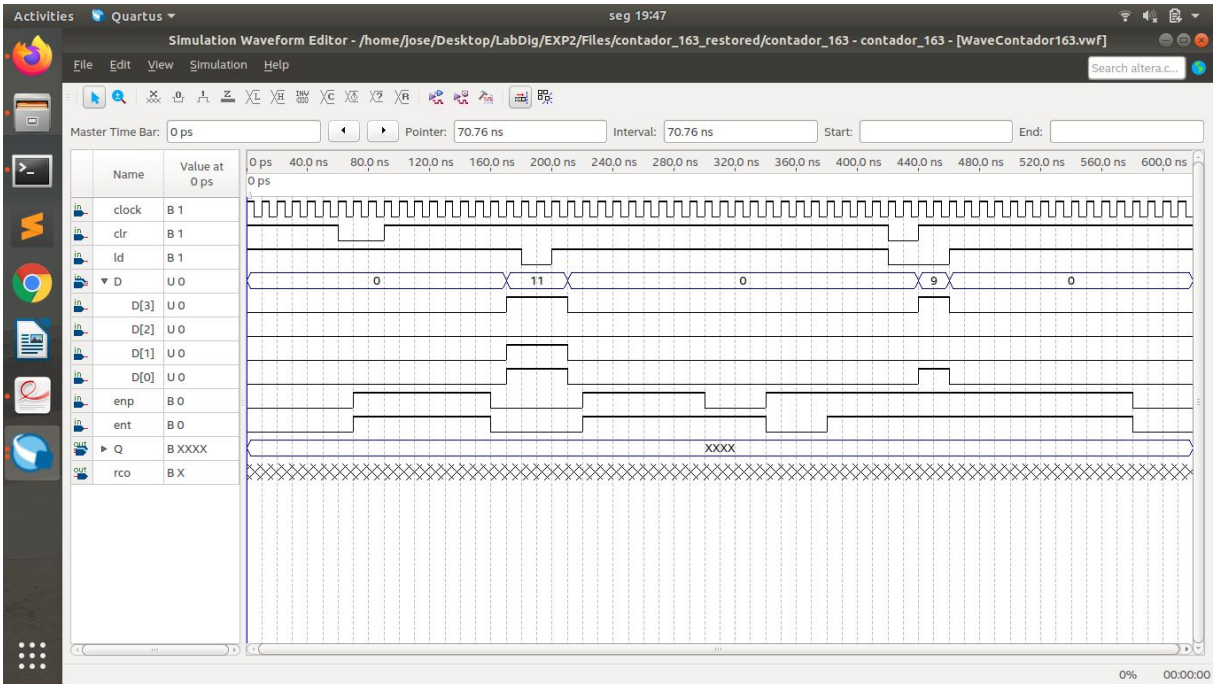
Baseando-se na imagem acima, temos a linha 42, que cria o sinal de conexão entre ENT e RCO; e as linhas 56 e 62, que definem esse sinal como saída RCO do contador menos significativo (CONT1) e entrada ENT do contador mais significativo (CONT2).

4 Parte Experimental

Atividade 2

Esta etapa trata da simulação do circuito de um contador (contador 74163) utilizando o programa Intel Quartus Prime. Nesta parte da experiência, geramos um arquivo de simulação para verificar se os resultados corroboram com as funcionalidades esperadas. Durante o planejamento, construímos o arquivo das formas de onda, no formato '*.vwf*', para facilitar a execução da experiência. Nota-se que é de certa forma trabalhoso colocar os sinais adequados na simulação.

Imagem 3: Formas de onda da simulação do contador 163



Atividade 3

Esta atividade envolve a simulação e síntese do projeto do contador de 8 bits. Durante o planejamento desenvolvemos um plano de teste para verificar se os requisitos funcionais do circuito foram atendidos. Para tanto, desenvolvemos as formas de onda, de forma semelhante ao que aconteceu na atividade 2.

Para desenvolver a forma de onda, utilizamos o Quartus, atribuindo os respectivos sinais que indicassem o plano de teste descrito abaixo. Analisando o ***datasheet*** do contador 74163, verificamos que a frequência de clock máxima de operação é de 25 MHz, o que corresponderia a um período de 40 ns.

Plano de Teste:

Apresentamos aqui um plano de testes para o contador de 8 bits para verificar se suas funcionalidades foram atendidas:

Requisitos funcionais:

- Realizar a contagem de 0 até 255, com o contador habilitado, ativando o sinal de clock
- Deve apresentar uma operação CLEAR, que limpa a saída do contador
- Deve apresentar um sinal de *enable* que habilita o contador
- Deve apresentar um sinal de *ripple-carry-out* que indicará quando o contador estiver em seu estado final
- Deve apresentar os bits correspondentes ao estado do contador, utilizando 8 LEDs.

Tabela 1: Plano de teste contador de 8 bits

Função	Sequência de sinais	Objetivo do teste
zerar saída Q	acionar zera = 0 uma vez	Verificar a função de clear
contagem de 0 a 15	desativar zera = 1 ativar conta pressionar KEY0 15 vezes	Checar se a contagem está sendo realizada adequadamente no primeiro contador
avançar um bit	pressionar KEY0 1 vez	Verificar o cascateamento e checar o sinal do rco
desligar conta e continuar a contagem por 3 vezes	desativar conta pressionar KEY0 3 vezes	Testar o controle do sinal conta no circuito
com o contador desabilitado,	manter conta desativado e	Verificar a precedência do

zerar os bits de saída	ativar o sinal de zera	sinal de zera sobre o sinal de conta
Limpar o contador e contar até 255	ativar limpa apertar KEY0 uma vez desativar limpa apertar KEY0 255 vezes	Testando um ciclo completo do contador

Imagem 4: Formas de onda da simulação do contador de 8 bits

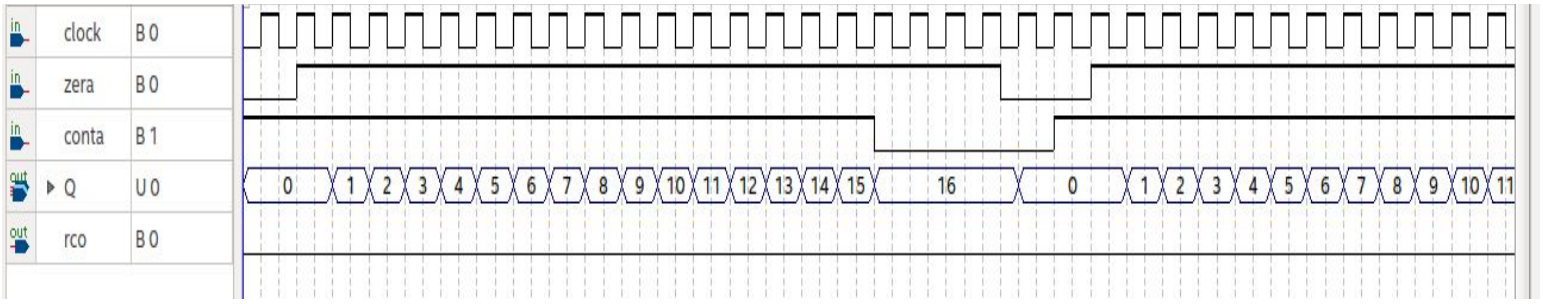
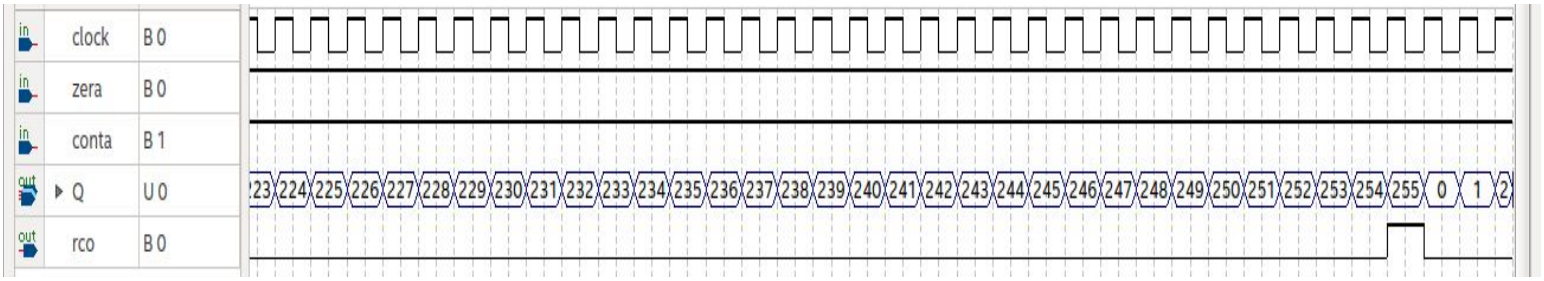


Imagem 4: Continuação das formas de onda do contador de 8 bits



6 Referências Bibliográficas

• ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. Tutorial para criação de circuitos digitais utilizando diagrama esquemático no Quartus Prime 16.1. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.