

Circuitos/Sistemas Integrados Digitais (CID/SID)

Ano Lectivo 2002/03

1ª Trabalho de Laboratório

Paulo Flores, Horácio Neto

1 Objectivos

Pretende-se com este trabalho de laboratório introduzir e familiarizar os alunos com as ferramentas de CAD e placas de desenvolvimento da Xilinx que serão usadas na cadeira de CID/SID.

2 Introdução

Neste laboratório vão ser implementados dois circuitos de exemplo:

- 1. Circuito para colocar, de forma "aleatória", o *led* da placa acesso ou apagado, cada vez que se carrega no interruptor.
- 2. Circuito divisor de relógio para que, em conjunto com o circuito anterior, se possa controlar se o *led* fica acesso ou apagado.

3 Projecto do 1º Circuito

O sistema de CAD utilizado para descrever e sintetizar o circuito é o WebPACK - Project Navigator da Xilinx. Para começar a correr este programa *click* no ícone **Project Navigator**. Os principais passos para criar um circuito descrito em VHDL são:

3.1 Criação do Projecto

- 1. Seleccione o menu File \rightarrow New Project...
- 2. Na janela que aparece escolha um localização para o projecto numa directoria do disco local (C:\tmp) e depois o nome que quer dar ao projecto (proj1). Em seguida certifique-se que a propriedades do dispositivo têm os seguintes valores:

Property Name	Value
Device Family	Spartan2
Device	xc2s200
Package	pq208
Speed Grade	-5
Design Flow	XST VHDL

3.2 Criação/Adição de Circuitos

- 1. Para adicionar um novo circuito ao projecto seleccione a opção $\mathsf{Project} \to \mathsf{New} \mathsf{Source}...$
- 2. No 1º passo (New) da janela que se abre escolha o tipo de ficheiro como VHDL Module para a descrição do circuito em VHDL ou Schematic para descrição do circuito através de esquemas. Note-se que num dado projecto podem existir ficheiros de vários tipos. Introduza o nome do ficheiro a criar (cir1), que deve de ser igual ao nome da entidade VHDL que vai ser descrita. Seleccione Next > para passar ao passo seguinte.
- 3. No 2º passo (Define VHDL Source) verifique se o nomes da entidade (Entity Name) e da arquitectura (Architecture Name) do componente que vai descrever em VHDL estão correctos. Introduza o nome dos portos de entrada (botao e clk) e saída (led) do circuito, as respectivas direcções (in, out ou inout). Para o portos que são vectores (buses) defina também qual a posição do bit mais e menos significativo.

Note que estes dados apenas vão ser utilizados no *template* de VHDL que será apresentado quando da criação do circuito pela primeira vez, podendo depois ser completamente alterados através do editor de texto.

Seleccione Next > para passar ao passo seguinte.

- 4. No 3º passo (**New Source Information**) certifique-se que todos os dados introduzido estão correctos e seleccione Finish para terminar a criação do novo ficheiro VHDL.
- 5. A janela criada do editor de texto permite alterar/completar o código VHDL do circuito. Complete a descrição do circuito para obter a seguinte especificação VDHL:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH. all;
use IEEE.STD_LOGIC_UNSIGNED. all;
entity cirl is
  port ( botao : in
                      std_logic ;
         led
                : out std_logic;
          clk
                : in
                      std_logic );
end cir1;
architecture Behavioral of cirl is
  signal d, q : std_logic;
begin
  process (clk, botao)
  begin -- process
    if clk' event and clk = '1' then
      if botao = '1' then
        q \ll d;
      end if;
    end if;
  end process;
  d \leq not q;
  led \leq = q;
end Behavioral;
```

6. Após introduzir a descrição VHDL grave-a com File \rightarrow Save.

3.3 Síntese do Circuito

1. A janela do lado esquerdo (**Processes for Current Source**) apresenta os processos e operações que podem ser aplicadas ao circuito seleccionado. Para sintetizar o circuito faça um duplo *click* em Synthesize. Se não ocorrerem erros durante o processo de síntese um sinal de executado aparece a verde. Se existirem erros ou avisos aparece uma cruz a vermelho ou uma exclamação a amarelo, respectivamente. Neste caso deve ser verificado o relatório do processo de síntese na janela inferior, onde estão indicados os erros com uma marca vermelha e os avisos com uma marca amarela. Se se fizer um duplo *click* nessa marca é indicada no código VHDL a linha do respectivo erro ou aviso.

3.4 Simulação Comportamental do Circuito

A simulação do circuito só deve de ser feita depois de se garantir que a respectiva descrição VHDL não apresenta erros e pode ser sintetizável pela ferramenta de síntese.

3.4.1 Criação da Bancada de Teste

Para realizar a simulação do circuito é conveniente criar uma uma bancada de teste (*testbench*) escrita em VHDL. Esta é a forma mais simples e normalizada de descrever os estímulos de simulação e verificar o comportamento do circuito, pois consiste na utilização da própria linguagem VHDL. Assim, deve ser criada um entidade de teste cuja arquitectura instancia o componente a testar e gera o estímulos necessários à sua verificação.

- 1. Para adicionar uma bancada de teste ao projecto seleccione $\mathsf{Project} \to \mathsf{Add} \ \mathsf{Source}...$
- 2. No 1º passo (**New**) da janela que se abre escolha o tipo de ficheiro como VHDL TestBench. Introduza o nome da bancada de teste a criar, em geral será igual ao nome do circuito seguido de _tb. Seleccione Next > para passar ao passo seguinte.
- 3. No 2^a passo (**Select**) seleccione o nome do circuito (entidade) para o qual se vai criar a bancada de teste. Seleccione Next > para passar ao passo seguinte.
- 4. No 3^a passo (**New Source Information**) certifique-se que todos os dados introduzido estão correctos e seleccione Finish para terminar a criação da nova bancada de teste em VHDL.
- 5. Na janela do editor de texto deve ser completada a bancada de teste do circuito com as sequências de teste necessárias para verificar o correcto funcionamento do circuito. Complete a descrição da bancada de teste do circuito para obter a seguinte especificação VHDL¹:

¹Nota: o ficheiro com a especificação VHDL pode ser escrito utilizando qualquer editor de texto e depois importado para o Project Navigator. Para isso, seleccione a opção Project \rightarrow Add Source... ou Project \rightarrow Add Copy of Source... e depois importe o ficheiro de acordo com o tipo de VHDL descrito: Module, Package ou TestBench.

```
-- VHDL Test Bench Created from source file cir1.vhd
-- Notes: This testbench has been automatically
-- generated using types std_logic and std_logic_vector
-- for the ports of the unit under test. Xilinx
-- recommends that these types always be used for the
-- top-level I/O of a design in order to guarantee that
-- the testbench will bind correctly to the
-- post-implementation simulation model.
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity testbench is
end testbench;
architecture behavior of testbench is
  component cir1
    port ( botao : in std_logic ;
             : in std_logic;
          clk
          led
              : out std_logic );
 end component;
  signal botao : std_logic;
  signal led : std_logic;
              : std_logic;
  signal clk
  signal clock : std_logic := '1'; -- inicializacao somente
                                  -- valida para simulacao
begin
  uut : cir1 port map(
    botao => botao,
    led
        => led,
    clk
         => clk);
-- *** Test Bench - User Defined Section ***
         <= not clock after 10 ns; -T=20 ns => f=50 MHz
 clock
  clk
          \leq = clock:
  tb : process
  begin
    '0' after 930 ns;
    wait for 1000 ns;
    assert false report "Fim da simulacao" severity warning;
    wait; -- will wait forever
 end process;
 - *** End Test Bench - User Defined Section ***
END;
```

6. Depois de gravada, a nova entidade VHDL (e respectivo ficheiro) aparece associada ao circuito para qual o teste foi gerado.

3.4.2 Simulação da Bancada de Teste

- Tendo a bancada de teste que se deseja simular seleccionada na janela Sources in Project aparecem os tipos de simulação possíveis que podem ser realizados com o simulador ModelSim. Para realizar a simulação comportamental faça um duplo *click* em Simulate Behavioral VHDL Model. Note que, o circuito que se vai simular é a bancada de teste que por sua vez instancia o circuito o qual se pretende verificar o funcionamento.
- Se não houver erros aparecem várias janelas do simulador. Verifique o resultado da simulação na janela wave default. Para ver toda a simulação faça View → Zoom → Zoom Full. Tente perceber porque é que saída led fica com o valor constante U (Undefined).
- 3. Para re-simular o circuito com a saída q do *flip-flop* inicializada a um dado valor comece por re-inicializar a simulação seleccionando na janela ModelSim XE a seguinte opção Simulate → Run → Restart... e depois seleccionando Restart na janela que aparece.
- 4. Na janela **Structure** seleccione o circuito onde o sinal **q** está definido, ou seja, seleccione o circuito uut:cir1.
- 5. Na janela **Signals** aparecem todos os sinais do respectivo circuito. Seleccione o sinal q e depois a opção do menu Edit \rightarrow Force... para forçar um dado valor no sinal.
- 6. Na janela **Force Selected Signals** escolha o valor a forçar (por exemplo 1) e o tipo (Kind) como Deposit.
- 7. Para simular o circuito por 1 us altere o tempo de simulação de 100 para 1000000 (ou 1 us) no campo respectivo da janela **ModelSim XE**. Inicie a simulação seleccionado a primeira opção do sub-menu Simlate $\rightarrow \text{Run} \rightarrow$.
- 8. Utilizando o Zoom adequado verifique que as alterações de valor na saída led ocorrem simultaneamente com as transições do flanco ascendente do relógio clk.
- 9. Para sair do simulador seleccione a opção $\mathsf{File}\to\mathsf{Quit},$ voltado a aplicação do Project Navigator da Xilinx.

3.5 Implementação do Circuito

- 1. Para realizar a implementação do circuito para a FPGA da Xilinx seleccione o circuito a implementa (cir1) e faça um duplo *click* em Implementing Design. A ferramenta vai correr uma série de programas e apresentando diversos relatórios na janela inferior.
- Depois da ferramenta acabar, cada um desses relatórios pode ser analisado separadamente. Por exemplo, para listar os relatórios produzidos pela ferramenta de colocação e interligação (*place and route*) deve-se expandir a seguinte sequência de ramos (através de um *click* no sinal ⊞): Implementing Design → Place & Route

Para ver o relatório de interligação e colocação numa janela faça um duplo *click* em Place & Route Report. Nesse relatório poderá verificar, por exemplo, o número de *slices* da FPGA ocupados pelo circuito.

3.6 Simulação Temporal do Circuito

A simulação temporal do circuito tem como objectivo simular o circuito tendo em conta todos os atrasos existentes em cada um dos sinais do circuito. Estes atrasos resultam do mapeamento do circuito em portas lógicas de uma dada tecnologia, com atrasos específicos, e também dos atrasos introduzidos pelas interligações dessas portas (que pode ser muito significativo no caso de uso da tecnologia FPGA).

Para esta simulação pode, e deve, ser usada a mesma bancada de teste que foi utilizada na simulação comportamental. Pretende-se verificar que os atrasos introduzidos pela implementação do circuito numa dada tecnologia não vão alterar a sua funcionalidade.

3.6.1 Simulação Final da Bancada de Teste

- Seleccione a bancada de teste que deseja simular (cir1_tb.vhd) na janela Sources in Project. Para realizar a simulação temporal faça um duplo *click* em Simulate Post-Palce & VHDL Model.
- 2. Verifique se o circuito apresenta o comportamento adequado através do resultados apresentados nas janelas criadas pelo simulador.
- 3. Verifique também que agora não é necessário inicializar a saída **q** do *flip-flop*, pois esta é automaticamente inicializada pela FPGA.
- 4. Utilizando o *zoom* adequado observe que as alterações de valor na saída led deixaram de ocorrer simultaneamente com as transições do flanco ascendente do relógio clk. Existe um atraso (da ordem dos ps) que resulta do circuito estar a ser simulado com portas lógicas de uma dada tecnologia.
- 5. Para sair do simulador seleccione a opção File \rightarrow Quit, voltado a aplicação do Project Navigator da Xilinx.

3.7 Programação do FPGA

Após verificação do correcto funcionamento do circuito através da simulação temporal do circuito, este poderá ser implementado na FPGA.

3.7.1 Criação de um Circuito de Topo

- 1. Cada circuito a ser programado na FPGA deve ter todas as entradas e saídas globais associadas aos pinos correspondentes na FPGA e os buffers especiais de entrada e saída que poderão ser necessários. Assim, crie um circuito para a programação da FPGA que associa os sinais externos aos respectivos pinos, cria os buffers necessários e contém o circuito a implementar.
- 2. Seleccione Project \rightarrow Add Copy of Source. Seleccione o ficheiro VHDL fpga.vhd do disco C: ou A:. Escolha o tipo de ficheiro VHDL como VHDL Module.
- 3. Faça um duplo *click* sobre o ficheiro que acabou de adicionar ao projecto e verifique que o código VHDL está de acordo com a seguinte descrição:

```
library IEEE;
library UNISIM;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH. all;
use IEEE.STD_LOGIC_UNSIGNED. all;
use UNISIM. Vcomponents. all;
entity fpga is
  port ( btn_in : in std_logic;
         clk_in : in std_logic;
         led_out : out std_logic );
end fpga;
architecture Behavioral of fpga is
  attribute loc : string; -- para associar cada sinal externo
                           -- a um pino da FPGA
  attribute loc of btn_in : signal is "p77";
  attribute loc of clk_in : signal is "p80";
  attribute loc of led_out : signal is "p71";
  component cir1
    port ( botao : in std_logic ;
          clk : in std_logic;
          led
                : out std_logic);
  end component;
  signal clk_local : std_logic;
  signal btn_local : std_logic;
begin
  -- sinais que necessitam de buffers
  CLK_BUFFER : IBUFG
    port map (I => clk_in , O => clk_local);
  BTN_BUFFER : IBUFG
    port map (I => btn_in, O => btn_local);
  -- o circuito a implementar
  Inst_cir1 : cir1
    port map( botao => btn_local,
              led => led_out,
              clk => clk local );
end Behavioral;
```

- 4. Com o circuito de topo (fpga) seleccionado encontre na janela Processes for Current Source o processo Generate Programing File. Sobre este processo e com o botão da direita seleccione a opção Properties....
- 5. Na janela criada (**Process Properties**) escolha a entrada **Startup Options** e altere o valor da entrada **Start-Up Clock** de CCLK para **JTAG Clock**. Seleccione OK para fechar a janela.
- 6. Faça um duplo *click* em Generate Programing File para executar todos os processos de síntese sobre o circuito de topo, incluindo a geração do ficheiro para a programação da FPGA. Verifique os relatórios gerados por estes processos na janela inferior. Se não houver erros a FPGA da placa de desenvolvimento pode ser programada.

3.7.2 Configuração da FPGA com o Circuito

- 1. Ligue a placa com a FPGA ao computador utilizado a porta paralela. Alimente a placa da FPGA e verifique que o *switch* existente na placa está colocado na posição JTAG.
- 2. Expanda os ramos de Generate Programing File (através de um *click* no sinal ⊞). Com o rato sobre Configure Device (iMPACT) seleccione com o botão da direita a opção Properties....
- 3. Na janela **Process Properties**, altere o valor de Configuration Mode para Boundary Scan e seleccione o ficheiro .bit correspondente ao circuito de topo (fpga.bit) na propriedade Configuration Filename. Seleccione OK para fechar a janela.
- 4. Faça um duplo *click* em Configure Device (iMPACT) para iniciar o processo de configuração da FPGA.
- 5. Depois de ter sido estabelecida a comunicação com a placa através da porta paralela na janela criada (Untitle [Configuration Mode] iMPACT) aparece o ícone de uma FPGA da Xilinx. Certifique-se que sobre esse ícone se encontra o ficheiro .bit correcto para ser carregado (correspondente à entidade de topo).
- 6. Sobre o ícone da FPGA seleccione com o botão da direita a opção **Program**... e depois seleccione OK na janela **Program Options**. Após o carregar correctamente a configuração para FPGA esta comporta-se de imediato como uma implementação do circuito projectado.
- 7. Verifique se o circuito funciona como pretendido.

4 Arquivar o Projecto

Para guardar o projecto sobre a forma de um arquivo ZIP deve-se primeiro eliminar os ficheiros desnecessários do projecto seleccionado Project \rightarrow Cleaup Project Files e só depois arquivar o projecto com Projecto \rightarrow Archive. Na directoria onde foi criado o projecto é criado um ficheiro .zip com o mesmo nome do projecto e que pode ser copiado para *diskette*.

5 Projecto do 2º Circuito

Utilize os conhecimentos que adquiriu para projectar um circuito que utilizando um contador divida o relógio de 50 Mhz para se obter um período da ordem dos segundos.

Interponha esse circuito entre o pino de relógio da FPGA e a entrada de relógio do 1^a circuito. Para ter uma ideia de como fazer um divisor de relógio baseie o seu circuito na seguinte especificação VHDL:

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH. all;
use IEEE.STD_LOGIC_UNSIGNED. all;
entity clock_div is
  port ( clk_in : in
                        std_logic ;
         clk_out : out std_logic );
end clock_div;
architecture Behavioral of clock_div is
  signal counter : std_logic_vector (24 downto 0);
begin
  process (clk_in)
  begin
    if clk_in 'event and clk_in = '1' then
      counter \leq counter + 1;
    end if;
  end process;
  clk_out <= counter(counter'high);</pre>
end Behavioral;
```

6 Referências

Listam-se em seguida algumas formas de obter mais informação sobre síntese de circuitos usando este ambiente de síntese da Xilinx.

- O capítulo 6 do Xilinx Synthesis Technology (XST) User Guide que se encontra disponível na página da cadeira (em formato pdf).
- Consultar e utilizar os Language Templates que se encontram no ícone lâmpada da barra de ferramenta do Project Navigator. Em particular os *templates* de VHDL para síntese.
- Ler a documentação existente no menu de Help da ferramenta. Em Particular Help \rightarrow On-Line Documentation (em formato pdf).

\$Id: trab1.tex, v $1.5\ 2003/03/07\ 18:13:26\ {\rm pff}\ {\rm Exp}$