

Curso de Verilog - Dia 5

Caio Rodrigo, Jorge Reis, Manuel Adahil

Introdução

Implementar um código em Verilog em FPGA exige algumas etapas prévias:

1. Configurar o projeto no Vivado para realizar a síntese e implementação para uma FPGA específica, ou seja, dessa vez a escolha da FPGA na criação do projeto é relevante;
2. Configurar o arquivo de Constraints (.xdc) no projeto para que o sintetizador possa ligar as entradas e saídas do módulo **top** em Verilog com os pinos da FPGA;
3. Criar um módulo **top** que faça a interface dos pinos da FPGA definidos no arquivo .xdc com os módulos do projeto.

Usaremos a FPGA **Artix-7 XC7A35T-1CPG236C** presente na placa Basys 3 da Digilent. Portanto, na criação do projeto na tela de **Default Part** será preciso agora definir a FPGA.

Depois é preciso importar o arquivo de **.xdc** para o projeto, isso pode ser feito pelo recurso **Add Files "+"** que foi usado para criar arquivos de design e simulação. O arquivo **.xdc** para a Basys 3 pode ser encontrado [aqui](#).

```
## Switches
set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
#set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
#set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
```

Exemplo de arquivo .xdc.

Repare que o arquivo .xdc vem com todas as definições comentadas. Para usar uma definição é preciso descomentá-la. Na definição é informado ao sintetizador que pino da FPGA vai ser ligado a que variável do código. Por exemplo, na imagem acima é possível ver que o pino **V17** está ligado à variável **sw[0]** e que o pino **V16** à variável **sw[1]**. Esses pinos, por sua vez, estão ligados os DIP switches mais à direita da placa.

É importante notar que as variáveis declaradas no .xdc estão na forma **sw[0]** e **sw[1]**, ou seja, elas estão declaradas na forma de um barramento **[1:0]sw**. Para mudar o nome da variável que será atrelada ao pino, é só renomear as referências à variável **sw** pelo nome desejado.

Outro ponto importante de um arquivo de constraints é que **somente poderão ser descomentadas as definições que serão utilizadas**. Caso uma definição seja descomentada não seja usada, o sintetizador irá reportar um erro.

Exemplo

Utilizando o .xdc da Basys 3 fornecido acima, descomente os DIP switches `sw[0]` e `[1]`. Além disso, descomente também os leds `[0]` e `[1]`. Agora vamos criar um módulo de design que será sintetizado e possuirá uma lógica combinacional.

```
1 module TOP (  
2     input  sw,  
3     output led  
4 );  
5  
6     assign led[0] = ~sw[0];  
7     assign led[1] = &sw;  
8  
9 endmodule
```

Agora sintetize e gere o bitstream para implementar o código na FPGA e verifique o funcionamento do módulo.

Prática 8 - FPGA

Gates

Utilizando os switches e os leds, reproduza o funcionamento das seguintes portas: **not**, **or**, **and**, **xor**, **nor**, **nand** e **nxor**. Cada porta, com exceção do not, possui duas entradas vindas dos switches e uma saída indo pros leds.

Counter

Crie um contador de 8 bits que conte a cada um **segundo** e exiba a saída nos leds. O contador precisa ter reset e enable, ambos ativos em alto e sendo controlados por switches.

Importante:

- É preciso reduzir a frequência do sinal de clock da FPGA no módulo para um segundo, isso pode ser feito utilizando um contador auxiliar;
- O contador só irá contar e resetar sincronamente quando o enable estiver ativo.

Display de 7 Segmentos

Utilize os quatro displays de 7 segmentos e os switches da seguinte forma:

- Cada display se associa a quatro switches, estas enviarão um valor em hexadecimal que será exibido no display;
- Organize os switches e displays de forma que os quatro switches mais à esquerda controlem o display mais à esquerda,

Dica: esquematize o problema e o divida em várias partes criando vários módulos, cada um resolvendo uma parte do mesmo.

Semáforo

Implemente a FSM de Moore da prática do semáforo em FPGA. Utilize os switches como entradas para a FSM e o display de 7 segmentos para mostrar o estado atual e as saídas.