

PROJETO 5

Projeto de Circuitos Integrados Digitais (10/2013)

O objetivo deste projeto é gerar a partir de uma descrição de alto nível o *layout* de um circuito.

1. Vamos utilizar agora o programa **LeonardoSpectrum** (**Windows**). Para poder empregar as bibliotecas da AMS_0.35 μm devemos ter alguns arquivos (c35_CORELIB.syn; c35_CORELIB_3B.syn; etc.) no diretório C:\MGC\LeoSpec\LS2005a_82\lib. Verifique se eles já estão lá; caso não estejam, copie os arquivos do diretório Mentor/Lib (*pen drive*) para aquele diretório. Ao abrir o Leonardo configure o *working directory* para sua área de trabalho. Neste diretório serão colocados os resultados do que fizer.
2. Vamos sintetizar inicialmente o circuito descrito no arquivo **uart.vhd** do diretório Demo (C:\MGC\LeoSpec\LS2005a_82\Demo) que descreve um *universal asynchronous receiver/transmitter*. Abra este arquivo no Leonardo (**INPUT - Open files**) e então execute o comando **INPUT - Read**. Observe que ao ser feita a leitura já é realizada uma primeira síntese.
3. Abra o arquivo texto dentro do Leonardo e dê uma olhada no conteúdo do **uart.vhd** (*click* duas vezes em cima do nome do arquivo). Faça alguma modificação no arquivo de forma a causar erro (por exemplo, troque “ENTITY **uart** IS” por “ENTITY **art** IS”). Salve, feche o texto e execute novamente o **INPUT - Read**. Veja as mensagens de erro. Caso abra novamente o arquivo texto aparecerão indicações dos erro.
4. Selecione a tecnologia para AMS-C35_CORELIB (**Technology**) e otimize o circuito (**Optimize-Optimize**).
5. Veja os esquemáticos que foram gerados (observe o esquemático com a opção *multipages* ou não). Qual é a diferença entre o esquemático associado a EXEMPLAR_XTR e o esquemático associado a EXEMPLAR.
6. Verifique o *Critical Path* do esquemático sintetizado e mapeado na tecnologia da AMS. O que significa este *critical path* e como é calculado?
7. Refaça a otimização alterando as opções de objetivo, área, velocidade, etc. e verifique os resultados. Variando as opções, minimize o *Critical Path*. Qual o valor final obtido e qual a frequência máxima de operação que o circuito sintetizado pode atingir? Apresente no relatório a figura do caminho crítico encontrado.
8. Vamos gerar agora um arquivo de saída no formato Verilog. O Verilog é uma linguagem de descrição de *hardware* usada com C.I.s (mais usada que o VHDL na indústria). Este formato servirá de interface para passarmos os resultados para o *Design Architecture*. No Leonardo, depois de fazer a síntese, vá ao menu **Output**, e configure para gerar Verilog.

De o nome que deseja e lembre-se que o resultado será colocado no diretório de trabalho que escolheu.

9. Verifique o arquivo de saída com um editor de texto e tente compreender a descrição que está feita (ao menos ter uma idéia).
10. Gere arquivos Verilog para os circuitos **uart.vhd** e **priority_encoder.vhd**. Transfira os arquivos Verilog para o Linux (no sistema Linux, os arquivos do Windows podem ser vistos no diretório /windows). Agora devemos convertê-los para gerar um *layout*. Abra o **IcStudio** (em algum projeto que já usou ou em um novo). Nele dê o comando *import verilog*.

Configure:

Output library: onde quer colocar

Verilog netlist: o arquivo gerado

Name map file:

local/tools/dkit/ams_3.70_mgc/mentor/c35/verilogin_cellmapfiles/c35b4_digital.cellmap

Execute. Deve ser criado tanto o esquemático como o símbolo do circuito **uart** e do circuito **priority.encoder**. Verifique ambos. No esquemático da **uart** há mais de uma página e para o **Check Schematic** passar corretamente as duas páginas do esquemático devem estar abertas.

11. A partir do esquemático (utilize o *viewpoint*) faça a geração do *layout* da **uart**. No roteamento utilize para V_{DD} e V_{SS} apenas linhas mais largas do que 1,8 μm . Para conseguir isso utilize, dentro do menu **ARoute**, a edição dos **net Classes**.

Procure nesse circuito fazer o roteamento mas observe que devido ao tamanho, completá-lo é bastante trabalhoso. Após o roteamento passe o LVS e o DRC.

Obs.1: Para se posicionar todas as células da **uart** é necessário se ter as duas páginas do esquemático abertas no **ICStation**. Para isso utilize o comando **\$open_sheet()**.

Posteriormente selecione todas as células de cada esquemático e faça o seu *placement*.

Obs.2: Para terminar *layout* a sugestão é seguir os passos:

- ✓ Faça inicialmente o roteamento das linhas de alimentação (V_{DD} e V_{SS});
- ✓ Execute, sem colocar os *ports*, o roteamento automático do restante dos sinais. Deixe os metais configurados para serem utilizados em apenas uma direção;
- ✓ Quando o número de ligações não feitas estiver em torno de 30, altere a configuração para permitir que os metais sejam utilizados nas duas direções;
- ✓ Agora selecione uma linha de cada vez e mande executar o roteamento automático. Caso a ferramenta não consiga executar um roteamento, apague as ligações que estão

- ✓ Quando conseguiu realizar todas as ligações execute o LVS (coloque nele a opção “ignorar os ports”);
- ✓ Após o LVS dar resultado correto acrescente os *ports* e termine o roteamento;
- ✓ Passe o DRC e corrija os erros (muitos);
- ✓ Termine o *layout* passando o LVS (agora considerando os *ports*).

12. Considere o circuito da Figura 1 (circuito *prescaler*).

- ✓ os sinais A, B e C definem o estado da máquina (ex.: o estado 000 é quando $A=0$, $B=0$ e $C=0$);
- ✓ esta máquina tem como entrada o sinal MC que define se o circuito divide o *clock* por 4, $MC=1$, ou por 5, $MC=0$;
- ✓ a saída é o sinal A.



13. Verifique que a máquina de estados representada pelo diagrama que traçou funciona dividindo o sinal de *clock* por 4 ou 5, de acordo com o valor de *MC*.
14. Descreva a máquina de estados em VHDL. Utilize o manual de VHDL do *software* Leonardo (LeonardoSpectrum HDL Synthesis Manual (D:\MGC\LeoSpec\LS2005a_82\doc\

leospec_hdl.pdf) para ver modelos de descrição para máquinas de estado. Apresente o VHDL no relatório.

15. Abra e leia o circuito VHDL, corrija os possíveis erros e otimize. Verifique o circuito gerado e compare com o esquemático da Figura 1.
16. Feche o Leonardo e abra novamente. Leia o VHDL, mas tome cuidado para utilizar como **Encoding Style** a opção **Binary**. Otimize e veja se o resultado melhorou. Otimize o circuito para obter o menor *Critical Path*. Qual é o valor encontrado? Apresente o esquemático obtido.
17. Sintetize o circuito utilizando a biblioteca da AMS.
18. Faça as verificações com DRC e LVS e apresente o *layout* final.
19. Faça a extração do circuito com **R+C+CC** e determine a máxima velocidade que o circuito atinge (teste para MC = "1" e para MC= "0"). Compare o resultado com o encontrado no item 17.

**"A quem muito foi dado, muito será cobrado."
(Lc. 12, 48)**