

Circuitos e Sistemas Integrados – Introdução ao Projeto VLSI

Prof. Fabian L. Cabrera & Prof. José L. Güntzel

e-mail: fabian.cabrera.r@gmail.com

Laboratório 7 – Fluxo de projeto baseado em standard cells

Forma de avaliação do Laboratório 7:

- Entregar um arquivo comprimido contendo os arquivos em meio eletrônico (subir arquivo .tar.gz no moodle). Tamanho máximo: 5MByte. Este arquivo deve conter uma estrutura parecida com o exemplo fornecido “lab7.tar.gz”. Também deve conter um relatório descrevendo cada etapa do projeto do sistema UAL-registrador, código VHDL, síntese lógica e física.
- Mostrar o layout e a simulação com a máxima frequência de relógio e incluindo todos os atrasos (pós-layout).

CrITÉrios de avaliação do Laboratório 7:

- A nota será atribuída conforme uma figura de mérito (FoM), onde $FoM = \text{área ocupada no layout} / \text{máxima frequência do sinal de relógio}$.
- O menor valor de FoM corresponderá à nota 10.
- Valores de FoM iguais entre diferentes trabalhos apresentados têm nota 6.

Especificações de projeto:

- Projetar um sistema UAL (Unidade aritmético lógica)-registrador de 32 bits como mostrado na Figura 1. As quatro operações que serão feitas pela UAL são: soma ($A+B$), subtração usando representação em complemento a 2 ($A-B$), deslocamento de B (1 bit para a esquerda) e deslocamento de B (1 bit para direita).

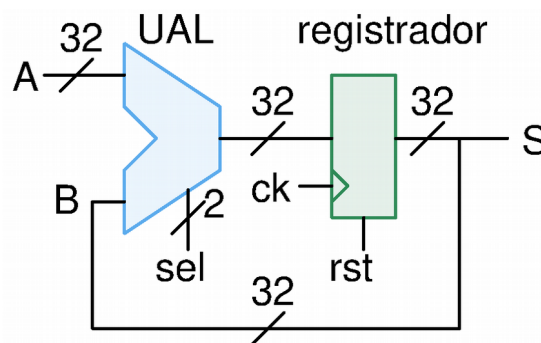


Figura 1: Diagrama de blocos do sistema especificado.

Forma de uso de este guia:

As instruções mostradas em seguida servem apenas como referência para desenvolver o exemplo proposto. Uma explicação detalhada será feita em sala de aula. Com base no

exemplo, o aluno deve construir a estrutura de arquivos para o circuito solicitado (UAL de 8 bits).

Instruções:

O fluxo de projeto segue os passos mostrados na Figura 2.

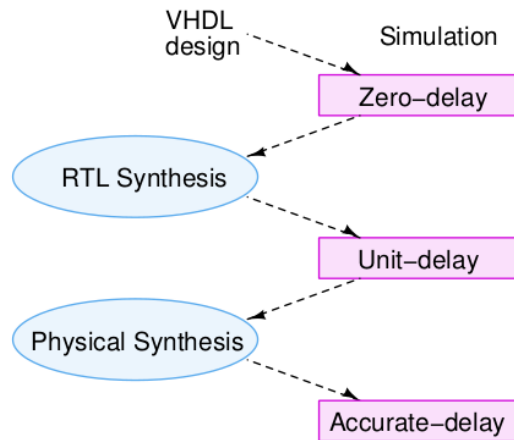


Figura 2: Resumo do fluxo de projeto.

1. Descarregar o arquivo lab7.tar.gz
2. Descomprimir o arquivo usando o comando:
`tar xvfz lab7.tar.gz`
3. Ao descomprimir, será criada uma pasta "lab7/contador10", esta pasta contém os arquivos necessários para implementar um contador de 4 bits (saída entre 0 e 9) com a estrutura mostrada em seguida:

```
|-- rtl
|   |-- contador.vhd      Código VHDL do circuito contador
|-- sim
|   |-- rtl
|   |   |-- file_list.f   Script de simulação 1 (zero-delay)
|   |-- synth
|   |   |-- file_list.f   Script de simulação 2 (unit-delay)
|   |-- sdf
|   |   |-- file_list.f   Script de simulação 3 (pós-layout)
|   |   |-- sdf_cmd.cmd   Configuração dos arquivos sdf
|   |-- tb
|   |   |-- tb_contador.vhd Test bench utilizado para as simulações
|-- constraint
|   |-- contador.sdc      Definição do sinal de relógio
|-- synthesis
|   |-- sintese.tcl       Script para realizar a síntese lógica com a ferramenta Genus
```

		-- physical	(síntese física com a ferramenta Innovus)
		-- 1.init.tcl	Carregar configuração e floorplan
		-- 2.power_plan.tcl	Criar pinos e linhas de alimentação
		-- 3.place_route.tcl	Colocar células e pinos no layout
		-- 4.nano_route.tcl	Fazer roteamento dos sinais
		-- 5.fillers_reports.tcl	Colocar fillers e contatos ao poço-N e substrato
		-- 6.netlist_sdf.tcl	Gerar arquivos de saída

4. Rodar a configuração das ferramentas:

```
cd
source /usr/eda/scripts/setup.digital
```

(cada vez que abrir um terminal)

5. Rodar a simulação 1 (zero-delay) do contador, esta simulação não considera atrasos.

```
cd lab7/contador10
cd sim/rtl
irun -f file_list.f
```

6. Síntese RTL

```
cd lab7/contador10
cd synthesis
genus -gui -legacy_ui -f sintese.tcl
```

```
report area
report timing
```

(verificar no reporte anterior se o circuito consegue operar na frequência de relógio especificada, ou seja, se o “slack time” é um número positivo).

7. Simulação pós-síntese (atraso unitário)

```
cd lab7/contador10
cd sim/synth
irun -f file_list.f
```

8. Síntese física

```
cd lab7/contador10
cd synthesis
```

```
innovus  
source physical/1.init.tcl  
source physical/2.power_plan.tcl  
source physical/3.place_route.tcl  
source physical/4.nano_route.tcl  
source physical/5.fillers_reports.tcl  
source physical/6.netlist_sdf.tcl
```

9. Verificar se há erros, isto pode ser feito nos menus:

Verify > Verify Geometry

Verify > Verify DRC

Verify > Verify Process Antenna

10. Simulação com atrasos de roteamento (pós-layout)

```
cd lab7/contador10  
cd synthesis  
ncsdfc contador.sdf
```

```
cd lab7/contador10  
cd sim/sdf  
irun -f file_list.f
```

Referências

[1] Matheus Moreira, Ricardo Guazzelli, Leonardo Rezende, Fernando Moraes. “Tutorial para síntese standard-cells utilizando Cadence”.

Disponível em:

https://www.inf.pucrs.br/~moraes/microel/cadence_design_flow_tutorial_padrao.pdf