

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
Cursos: Engenharia de Computação e Ciência da Computação

TRABALHO 2
Disciplina: INF1058 – Circuitos Digitais

Prof. Sergio Bampi
Turma B

TRABALHO PRÁTICO II

Descrição de um Somador Completo utilizando linguagem de descrição de *hardware*
(HDL)

1) OBJETIVO

O aluno deverá desenvolver a descrição em HDL de um somador completo com suas entradas e saídas registradas e verificar a correta operação da descrição através de simulação. A descrição do somador completo deve ser feita de modo hierárquico, isto é, deve ser descrito separadamente o Meio-Somador e o Somador Completo (conforme descrito na próxima seção).

O Somador Completo deverá trabalhar com números em Complemento de Dois. Deste modo as entradas/saída seguem a tabela abaixo.

Binário	Decimal
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

2) METODOLOGIA

O aluno deverá seguir os procedimentos descritos abaixo para realizar este trabalho.

1. Especificar a descrição **estrutural**, isto é, descrevendo com portas lógicas (AND, OR, etc), de um **Meio-Somador** (*Half Adder*). Salve esta descrição em um arquivo chamado *half_adder.vhd*
2. Especificar a descrição **estrutural** de um **Somador Completo** (*Full Adder*) que utiliza o **Meio-Somador** descrito na primeira etapa. Salve esta descrição em um arquivo chamado *full_adder.vhd*. **DICA:** Para instanciar um módulo descrito em outro arquivo utilize o comando **port map**.
3. Especificar a descrição **comportamental** de um **Flip-Flop tipo D** (FFD). Salve esta descrição em um arquivo chamado *ffd.vhd*.
4. Escrever um arquivo topo, chamado *somador4bits.vhd* que instanciará todos os módulos: Somador Completo e Flip-Flop tipo D. Como cada Somador Completo e o FFD tratam apenas 1 bit, são necessários 4 instâncias de cada um.
5. A entidade do *somador4bits.vhd* deve seguir o modelo abaixo

```
entity somador4bits is
port (
    clock   : in std_logic;
    reset   : in std_logic;
    Cin     : in std_logic;
    A       : in std_logic_vector(3 downto 0);
    B       : in std_logic_vector(3 downto 0);
    F       : out std_logic_vector(3 downto 0);
    Overflow : out std_logic
);
end somador4bits;
```

6. Interconecte o Somador Completo e os FFD utilizando sinais internos (**signal**).

3) ENTREGA

O aluno deverá entregar até dia **31 de maio** o relatório completo do trabalho. Neste devem estar, em arquivos separados, os itens abaixo em um **.ZIP** ou **.TAR.GZ** :

- Relatório em formato **.PDF** (Não será aceito relatórios em outro formato)
- Arquivos fonte deste trabalho, isto é, arquivos **.vhd** e arquivos utilizados para simulação.

O relatório escrito deve conter as informações detalhadas abaixo:

- Síntese e extração de resultados de utilização de recursos (número de LUTs, de registradores, de pinos de I/O etc).
- Frequência máxima de operação do Somador Completo.
- Criação do arquivo de teste Vector Waveform File (VWF) com as formas de onda e plotagem dos resultados. Mostre que a descrição funciona para os casos que há **Overflow** e **Propagação de carry**. Lembre-se de testar a soma entre números positivos e negativos.
- Plotagem do Netlist gerado pelo Quartus II.
- Resposta para a pergunta:
 - Quais vantagens o projetista tem ao escolher uma linguagem de descrição de *hardware*, como o VHDL, para implementar seu sistema ao invés de uma abordagem de esquemático com portas lógicas.
 - Qual a diferença, em VHDL, entre descrição estrutural e comportamental?