

VHDL Concorrente

Problemas

Rafael Corsi Ferrão

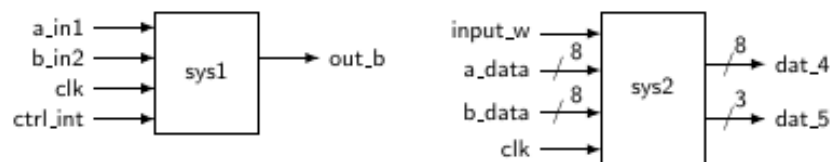
corsiferrao@gmail.com

4 de outubro de 2015

1 Entidade

1.1 Caixa Preta

Escreva um código em VHDL que descreva a entidade da Figura a seguir :



2 Operações lógicas

Crie uma entidade que possua 4 elementos de entrada (a,b,c,d) e quatro portas de saída (s0, s1, s2, s3) todas do tipo std_logic. Crie agora as seguintes operações lógicas em sua arquitetura:

- $S1 = a + \bar{b}$
- $S2 = a + \bar{b}.c + d$
- $S3 = (a + \bar{b}).(c + d)$
- $S4 = (a + \bar{b}).\overline{(c + a.d)}$

3 Conversões e Tipos de dados

A partir do código a seguir :

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;

ENTITY Conversoes IS
END Conversoes;

ARCHITECTURE bhv OF Conversoes IS

    -- Declarações sinais e constantes
    signal slv : std_logic_vector(7 downto 0);
    signal s   : signed(7 downto 0);
    signal us  : unsigned(7 downto 0);
    signal i   : integer range 0 to 12;

BEGIN

END bhv;
```

Torne possível através das conversões a atribuição dos sinais :

```
-- Signed
slv <= s;
us  <= s;
i   <= s

-- std_logic_vector
s   <= slv;
us  <= slv;
i   <= slv;

-- unsigned
slv <= us;
s   <= us;
i   <= usv;

-- integer
slv <= i;
s   <= i;
us  <= i;
```

4 Somador

Desenvolva uma entidade capaz de somar dois vetores e retornar o valor em uma terceira porta. Os vetores devem conseguir armazenar um dado de valor 0 à 256. Utilize

as chaves `sw[0]` até `sw[7]` como entrada ao vetor **A** e as chaves `sw[8]` até `sw[15]` como entrada para o vetor **B**. Exiba o resultado da soma nos led.

As portas a serem utilizadas são:

- Chaves : `sw[0]` até `sw[15]`
- LEDs: `led[0]` até `led[15]`

4.1 Verifique

- O tipo de dado devemos utilizar ?
- Como lidamos com o estouro (overflow) ?
- analise o RTL.
- implemente na FPGA, o resultado é o esperado ?

5 Mux

Mux é um seletor digital amplamente utilizado em projetos discretos. O Mux deve possuir três entradas (`in0`, `in1` e `in2`) to tipo **STD_LOGIC**, uma porta para de endereçamento de 2 bits (`addr`) do tipo **STD_LOGIC_VECTOR**, e um bit de saída (`dout`) do tipo **STD_LOGIC**.

1. Implemente um Mux utilizando o construtor **WITH SELECT**.
2. Implemente utilizando o construtor **WHEN ELSE**.

5.1 Verifique

- Em ambos os casos o RTL criado (Esquemático e síntese)
- qual a tecnologia escolhida pelo sintetizador para implementar a lógica ?

6 ALU - Unidade Lógica

Implemente uma ALU capaz de realizar as seguintes operações quatro diferentes operações:

- a) somar dois vetores;
- b) subtrair dois vetores;
- c) comparar qual é maior;

d) multiplicar um dos vetores por 2^n vezes.

Os vetores de entradas devem como no problema passado as chaves SW e a saída da ALU os LEDs. Como seletor das operações, usar os botões BTN :

- BTNU : implementa a soma de dois vetores
- BTNL: implementa a subtração de dois vetores
- BTNR: compara os vetores
- BRND: faz a multiplicação do vetor A por 2^n vezes, sendo n definido pelo vetor B.

6.1 Dica multiplicação por dois:

Sabemos que a multiplicação binária por 2 é equivalente a rotação do vetor para a esquerda, o mesmo acontece para a divisão (rotação para direita)

exemplo: $4 \cdot 2 = 8 = "0100" \ll 1 = "1000" = 8$

Para implementar em VHDL, utilize o comando **SSL**. Esse comando recebe como primeiro argumento um dado do tipo **UNSIGNED** e como segundo argumento um dado do tipo **INTEGER**.

6.2 Verifique

- Em ambos os casos o RTL criado (Esquemático e síntese)
- qual a tecnologia escolhida pelo sintetizador para implementar a lógica ?
- a ferramenta utilizou algum flip-flop ?