TALLER Sebastian Serna 1088283998

Este taller consiste en mejorar las habilidades de los conceptos aprendidos en clase.

1. Implemente en VHDL un FlipFlop de 1bit, con entradas D, Reset, CLK, y una salida Dout.

----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 21:34:57 08/21/2016

-- Design Name:

-- Module Name: FlipFLop - Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 - File Created

-- Additional Comments:

--

----------------------------------------------------------------------------------

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

-- Uncomment the following library declaration if using

-- arithmetic functions with Signed or Unsigned values

--use IEEE.NUMERIC\_STD.ALL;

-- Uncomment the following library declaration if instantiating

-- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity FlipFLop is

Port ( D : in STD\_LOGIC;

RST : in STD\_LOGIC;

CLK : in STD\_LOGIC;

Dout : out STD\_LOGIC);

end FlipFLop;

architecture Behavioral of FlipFLop is

signal count : STD\_LOGIC;

begin

process (CLK) is

begin

if rising\_edge (CLK) then

count <= count + 1;

end if;

end process;

Dout <= count;

end Behavioral;

2. Escriba los 4 principios de diseño de hardware aprendidos en clase.

• Primer Principio del diseño de Hardware: La simplicidad favorece la regularidad.

• Segundo Principio del diseño de Hardware: Entre más pequeño más rápido.

• Tercer Principio del diseño de Hardware: Hacer el caso común más rápido.

3. Convertir a instrucciones de bajo nivel.

int x=0; ---→ add %g0, 0, %L1

int y =8; --→ add %g0, 8, %L2

int z = 1; ---→ add %g0, 1, %L3

y=x+3;---→ add %L1,3,%O0

z=z+3;---→ add %L2,3,%L2

x=(x-z)+(3+y);----→ sub %L1,%L2,%L2

add 3,%L3,%L3

add %L2,%L3,%L1

4. Usar el ld, y st.

a[4]= a[2]+x;--→ ld [%L2+(2\*4)],%L4

add %L4, %L3, %L3

st %L3,[%L2+(4\*4)]

y[0] = y[40]+13;--→ ld [%L2+(40\*4)],%L3

add %L3,13,%L4

st %L4,[%L2+(0\*4)]

5. Convertir a lenguaje de máquina.

a.

int main(){

int i =3; p=2; --→ add %g0,3,%L0(1); add %g0,2,%L1 (2)

return i+3;

}

Formato 3 (1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10000 | 000000 | 00000 | 1 | 0000000000011 |

Formato 3 (2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10000 | 000000 | 00000 | 1 | 0000000000010 |

b.

int main(){

int p=3; x=1; z=4;-→ add %g0,3,%L1(1); add %g0,1,%L2 (2); add %g0,4,%L3 (3)

int w=0; ---→ add %g0, 0, %L4 (4);

w=(p+40)+(x-z);add %L1,40,%L1 (5)

sup %L2,%L3,%L2 (6)

add %L1,%L2,%L4 (7)

return 0;

}

Formato 3 (1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10001 | 000000 | 00000 | 1 | 0000000000011 |

Formato 3 (2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10010 | 000000 | 00000 | 1 | 0000000000001 |

Formato 3 (3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10011 | 000000 | 00000 | 1 | 0000000000100 |

Formato 3 (4)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10100 | 000000 | 00000 | 1 | 0000000000000 |

Formato 3 (5)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | 10001 | 000000 | 10001 | 1 | 0000000101000 |

Formato 3 (6)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10010 | 000100 | 10010 | 0 | 00000000 | 10011 |

Formato 3 (7)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10100 | 000000 | 10001 | 0 | 00000000 | 10100 |

6. Inicializar las siguientes variables negativas usando OR.

n=-12,-→ OR %g0,-12,%O1

a=-11,-→ OR %g0, -11, %O2

b=-14--→ OR %g0, -14, %O3