TALLER

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.

\_\_\_\_\_\_\_\_\_\_

D-------| |

| |

RST-----| |\_\_\_\_\_\_\_Dout

| |

CLK-----|> |

|\_\_\_\_\_\_\_\_\_\_|

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

3. Convertir a instrucciones de bajo nivel.

int x=0;

int y =8;

int z = 1;

y=x+3;

z=z+3;

x=(x-z)+(3+y);

4. Usar el ld, y st.

a[4]= a[2]+x;

y[0] = y[40]+13;

5. Convertir a lenguaje de máquina.

a.

int main(){

int i =3; p=2;

return i+3;

}

b.

int main(){

int p=3; x=1; z=4;

int w=0;

w=(p+40)+(x-z);

return 0;

}

6. Inicializar las siguientes variables negativas usando OR.

n=-12,

a=-11,

b=-14

**1)** **--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**

**proccess (CLK) is**

**begin**

**if rising\_edge(CLK) then**

**count<=count + 1;**

**end if;**

**end process;**

**Dout <= count;**

**end behavioral;**

**3)ADD %go,0 ,%L0**

**ADD %go ,8,%L1**

**ADD %go ,1,%L2**

**ADD %L0,3,%L4**

**ADD %L2,3,%L5**

**SUB %LO,%L5,%L2**

**ADD 3,%L4,%L1**

**ADD %L1,%L2,%L1**

**4) A)LD[%L0+(2\*4)],%L2**

**ADD%L2,%L1,%L3**

**ST %L3,[%Oo+(4\*4)]**

**B)LD[%L0+(4\*40)],%L2**

**ADD[%L2,13],%L1**

**ST %L1,[%Oo+(4\*0)]**

5) OP=OPERACION BINARIO

RD=BINARIO VER TABLA LC**[0]-LC[7] =R[16]-R[23]**

RS=REGISTRO 1

I=INMEDIATO

UNSAMED=SE USA CUANDO HAY INEMDIAO

1) add %go,3,%lo

OP RD OP3 RS1 I

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10000 | 000000 | 00000 | 1 | 00000000 | 00011 |

SACAMOS EL HEXADECIMAL

A 0 0 0 2 0 0 3

2) Add %go,2,%l1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10001 | 000000 | 00000 | 1 | 00000000 | 00010 |

SACAMOS HEXADECIMAL

A 2 0 0 2 0 0 2

B)

5.1) add %go,3,%lo

OP RD OP3 RS1 I

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10000 | 000000 | 00000 | 1 | 00000000 | 00011 |

SACAMOS EL HEXADECIMAL

A 0 0 0 2 0 0 3

5.2) Add %go,2,%l1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10001 | 000000 | 00000 | 1 | 00000000 | 00001 |

SACAMOS HEXADECIMAL

A 1 0 0 2 0 0 2

5.3) Add %go,4,%l2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10010 | 000000 | 00000 | 1 | 00000000 | 00100 |

SACAMOS HEXADECIMAL

A 2 0 0 2 0 0 4

5.4) Add %go,0,%l3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 10011 | 000000 | 00000 | 1 | 00000000 | 00000 |

SACAMOS HEXADECIMAL

A A 0 0 2 0 0 0

6)01100

\_\_\_+1

01101

%go,-12,%LO

B)

a=-11,

01011

\_\_\_+1

01100

%GO,-11,%L2

C)

b=-14

01110

\_\_\_+1

01111

%GO ,-14,%L3