



**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS**

**INGENIERIA EN COMUNICACIONES Y ELECTRONICA**

**Circuitos Digitales 1**

**Tarea 02**

**Sumador 6 bits comportamental en VHDL**

Nombre: Aguilar Rodríguez Carlos Adolfo

Código: 215860049

Fecha: 17 de Febrero del 2018

Profesor: Chávez Martínez Ehecatl Joel

## Contenido

Marco Teórico .....	( 3 )
Sumador 6 bits .....	( 4 )
-Código vhdl .....	( 4 )
-Código pat .....	( 5 )
-Código ioc .....	( 5 )
-Diagrama Esquemático .....	( 6 )
-Diagrama caja negra .....	( 7 )
-Simulación .....	( 7 )
-Plano real .....	( 8 )
-Tabla de resultados .....	( 9 )

- **Marco Teórico**

- Diseñar en VHDL un sumador de 6 bits con 3 entradas y 2 salidas:

Dos de las entradas son de 6 bits y una tercera entrada solo de 1 bit

Una de las salidas guardara los 6 bits menos significativos del resultado de la operación

La otra salida será destinada para el bit más significativo en caso de que el resultado de la operación arroje 7 bits (bit de acarreo).

Las operaciones de los bits serán en binario pero en la simulación estarán expresados en hexadecimales

También se hace uso de señales para representar las conexiones internas del circuito estas no se ven en la simulación, ni en el código de patrones de estímulos ni en el código de entradas y salidas. Solo se puede ver y manejar por medio del archivo VHDL el cual contiene el código del hardware;

La arquitectura de la suma hace un llamado del componente sumador el cual realizara la suma bit a bit de cada una de las entradas

- Código VHDL

```
--librerias
library ieee;
use ieee.std_logic_1164.all;

--Entidad
entity sum6bc is
port(
    A,B: in std_logic_vector(5 downto 0);
    cin: in std_logic;
    Sum: out std_logic_vector(5 downto 0);
    Cout: out std_logic);
end sum6bc;

architecture suma of sum6bc is
    signal resultado: std_logic_vector(6 downto 0);

    component full_adder
    port(a,b,cin: in std_logic;
        sum,cout: out std_logic);
    end component;

    signal aca: std_logic_vector (5 downto 0);
    signal cins: std_logic;
begin
    x0:full_adder port map(
        a => A(0),
        b => B(0),
        cin => cins,
        sum => Sum(0),
        cout => aca(0));

    x1:full_adder port map(
        a => A(1),
        b => B(1),
        cin => cins,
        sum => Sum(1),
        cout => aca(1));

    x2:full_adder port map(
        a => A(2),
        b => B(2),
        cin => cins,
        sum => Sum(2),
        cout => aca(2));

    x3:full_adder port map(
        a => A(3),
        b => B(3),
        cin => cins,
        sum => Sum(3),
        cout => aca(3));

    x4:full_adder port map(
        a => A(4),
        b => B(4),
        cin => cins,
        sum => Sum(4),
        cout => aca(4));

    x5:full_adder port map(
        a => A(5),
        b => B(5),
        cin => cins,
        sum => Sum(5),
        cout => aca(5));
    cins <= cin;
end suma;
```

- Codigo ioc

```

TOP(
  (IOPIN a(0).0);
  (IOPIN a(1).0);
  (IOPIN a(2).0);
  (IOPIN a(3).0);
  (IOPIN a(4).0);
  (IOPIN a(5).0);
)

LEFT(
  (IOPIN b(0).0);
  (IOPIN b(1).0);
  (IOPIN b(2).0);
  (IOPIN b(3).0);
  (IOPIN b(4).0);
  (IOPIN b(5).0);
)

RIGHT(
  (IOPIN cin.0);
  (IOPIN cout.0);
)

BOTTOM(
  (IOPIN sum(0).0 );
  (IOPIN sum(1).0 );
  (IOPIN sum(2).0 );
  (IOPIN sum(3).0 );
  (IOPIN sum(4).0 );
  (IOPIN sum(5).0 );)

```

- Código pat

```

-- declaracion de puertos
in a    (5 downto 0)X;
in b    (5 downto 0)X;
in      cin B;
out sum (5 downto 0)X;
out cout      B;

```

```

-- fin de puertos
-- fuente de alimentacion

```

```

in vdd B;
in vss B;

```

```

begin

```

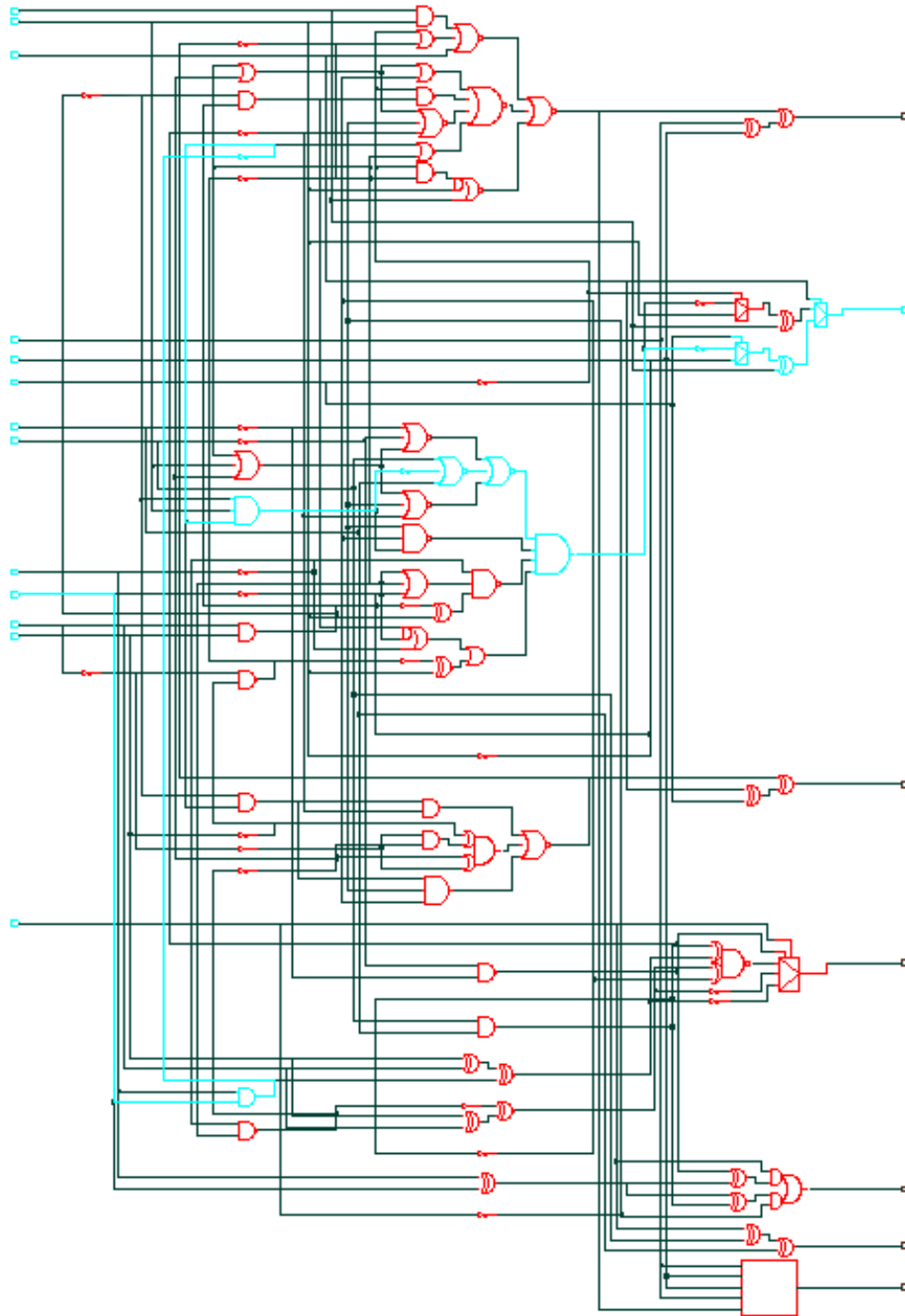
--	a	b	cin	sum	cout	vdd	vss
<0ns>:	00	00	0	?**	?*	1	0;
<+50ns>:	01	00	1	?**	?*	1	0;
<+50ns>:	02	01	0	?**	?*	1	0;
<+50ns>:	03	02	0	?**	?*	1	0;
<+50ns>:	05	03	0	?**	?*	1	0;
<+50ns>:	08	05	0	?**	?*	1	0;
<+50ns>:	0D	08	0	?**	?*	1	0;
<+50ns>:	15	0D	0	?**	?*	1	0;
<+50ns>:	22	15	0	?**	?*	1	0;
<+50ns>:	37	22	0	?**	?*	1	0;
<+50ns>:	59	37	0	?**	?*	1	0;

```

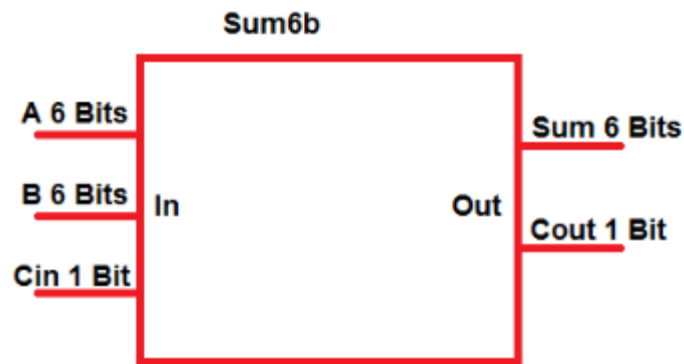
end;

```

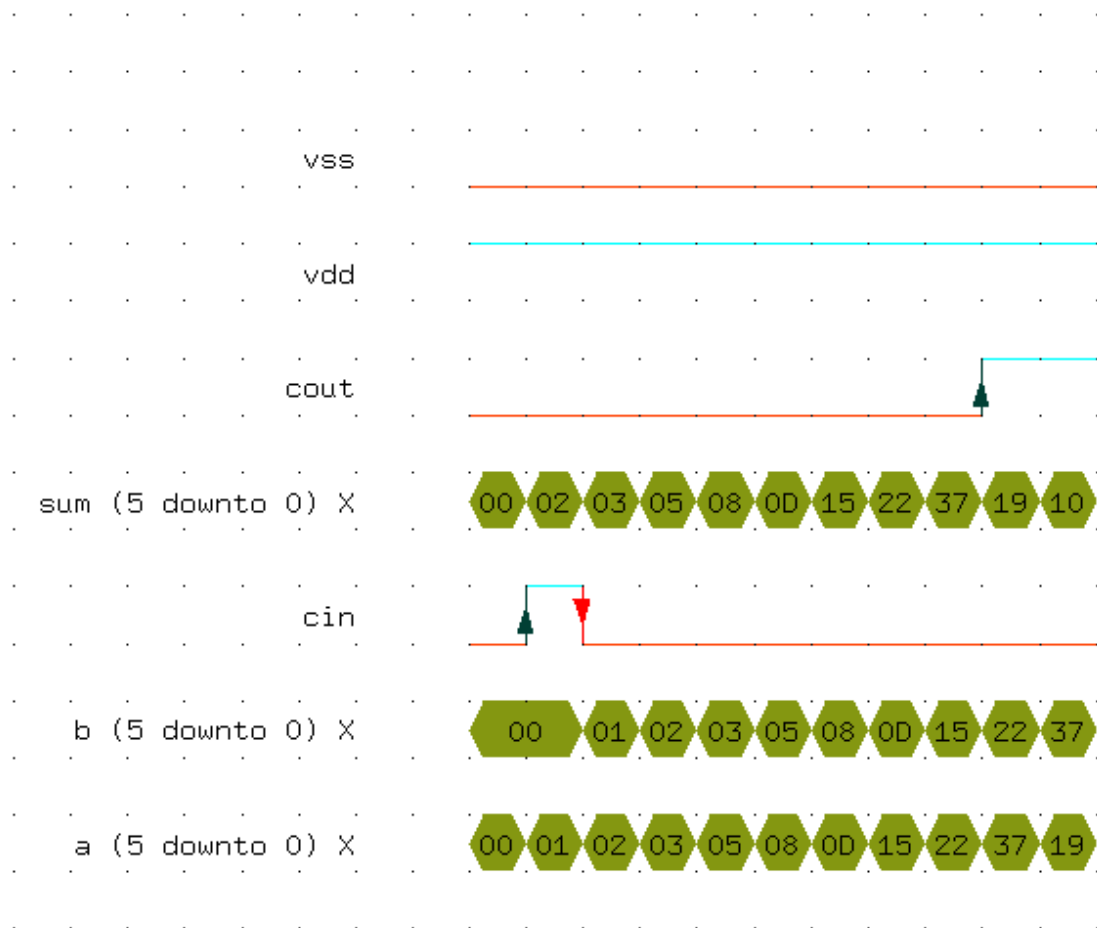
## DIAGRAMA ESQUEMATICO



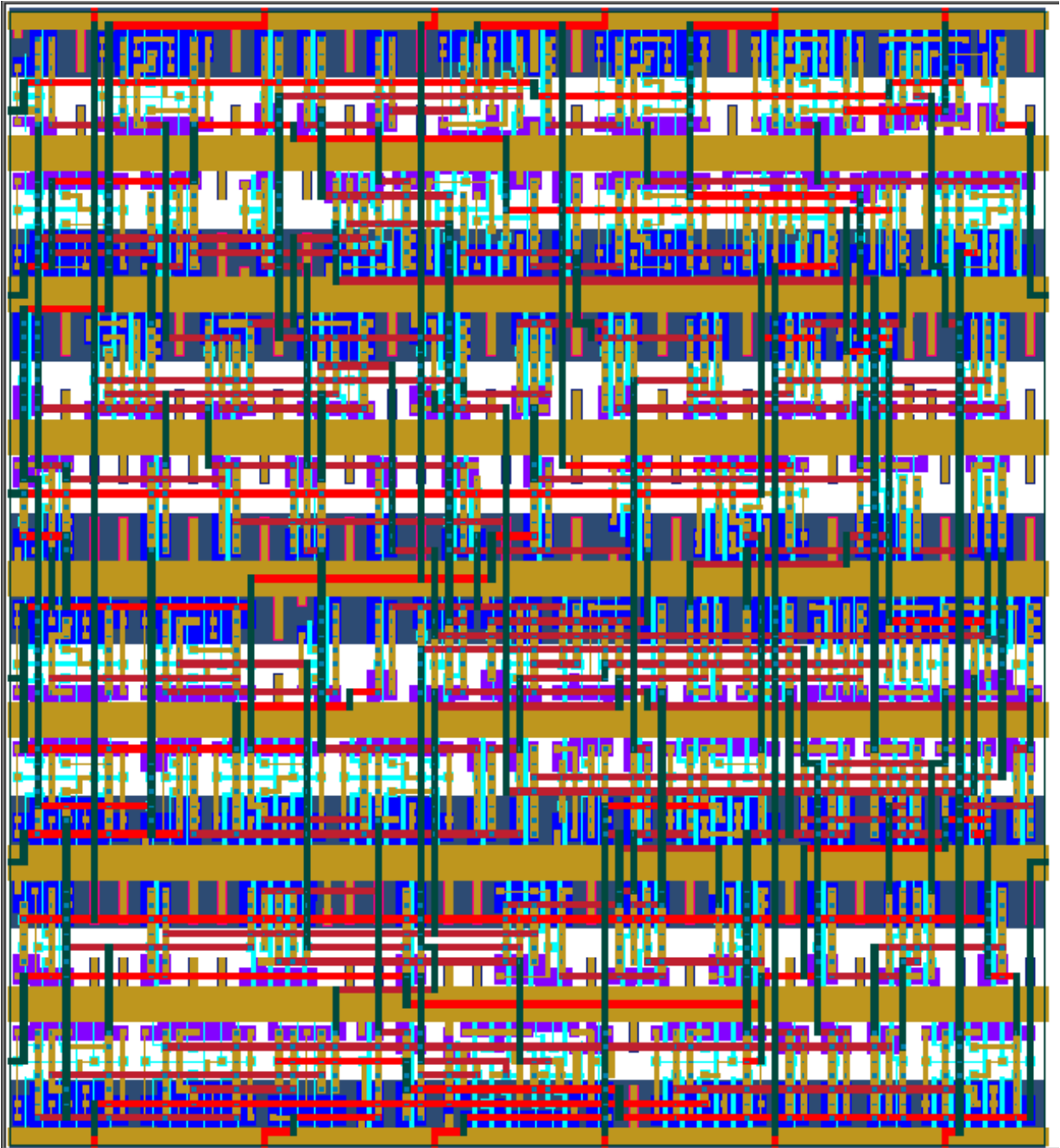
# Diagrama caja negra



## • Simulación



- Plano real





- **Tabla de resultados**

```
--LOON-----
Area on file 'sum6bc_boog.vst'...118000 lamda² (with over-cell routing)
Area on file 'sum6bc.vst'...122000 lamda² (with over-cell routing)
Critical path (no warranty)...3720 ps from 'b 1' to 'sum 4'
-----

--COUGAR-----
---> Figure size   : (   -100,   -100 )
                   (  36600,  40100 )
---> Build transistors
<--- 574
-----
```

Critical path 3720 pico segundos  
Area  $A = 366000 \text{ micrometros}^2$   
Transistores =574  
Figura= 1,467,660,000