CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

INGENIERIA EN COMUNICACIONES Y ELECTRONICA

Circuitos Digitales 1

Tarea 04
Multiplicador Estructural en VHDL

Nombre: Aguilar Rodríguez Carlos Adolfo

Código: 215860049

Fecha: 6 de marzo del 2018

Profesor: Chávez Martínez Ehecatl Joel

Contenido

- **Marco Teórico**

El multiplicador estructural es común que contenga el código más largo que el multiplicador comportamental aun así, la distribución de su codificación está más organizada, puesto que reparte todo el código en pequeños bloques de código en donde se puede trabajar por partes el código sin necesidad de alterar todo el código.

Además como ventaja es que se puede trabajar en equipo repartiendo partes del código a cada programador logrando un trabajo más rápido y eficiente.

Puede parecer enredoso la manera en cómo se distribuye la descripción del hardware de este tipo de código pero con solo darle un breve análisis visual podemos observar que es muy repetitivo, al hacer llamado de componentes (otros códigos) y simplemente aplicarlos al programa es cómo armar un rompecabezas de códigos y crear un código único con las funciones de los demás códigos.

Esto no quiere decir que sea la mejor manera de trabajar tiene sus ventajas y desventajas en la síntesis comportamental podemos detectar más fácil el funcionamiento del código en caso de que no se encuentre de manera explícita con comentarios para el programador la función del programa.

También puede ser cuestión de gustos unos prefieren la manera clásica y otros trabajar más en equipo de cualquier manera, se aplique la síntesis que se aplique mientras el circuito cumpla la función que se desea es recomendable usar la que mejor nos agrade.

En el caso de este código se hace llamado de un componente multiplicador de un bit que se va a repetir tantas veces como tantos bits se requieran operar

Para el caso de la simulación mientras cualquiera de las entradas sea 0 la salida será cero igual que una multiplicación común los demás se operan en sistema Hexadecimal

- **Código VHDL Multiplicador Jerárquico**

```
--Librerias
library ieee;
use ieee.std_logic_1164.all;
entity mul4bs is
	port(
	X,Y:in  std_logic_vector (3 downto 0);
	Z:	out std_logic_vector (7 downto 0));
end mul4bs;

architecture arq1 of mul4bs is
--componentes-----------------------------
component mul1bs
	port(
	si, x, y, ci:	in std_logic;
	co,so:	out std_logic);
end component;
--SeÃ±ales---------------------------------
	signal acarreo,suma: std_logic_vector(15 downto 0);
begin
--conexiones de componentes---------------
U1: mul1bs
	port map(
			si	=> '0',
			x	=> x(0),
			y	=> y(0),
			ci	=> '0',
			so	=> suma(0),
			co	=> acarreo(0));
U2: mul1bs
	port map(
			si	=> '0',
			x	=> x(1),
			y	=> y(0),
			ci	=> acarreo(0),
			so	=> suma(1),
			co	=> acarreo(1));
U3: mul1bs
	port map(
			si	=> '0',
			x	=> x(2),
			y	=> y(0),
			ci	=> acarreo(1),
			so	=> suma(2),
			co	=> acarreo(2));
```

Continuacion de codigo

```
U4: mullbs
        port map(
                        si      => '0',
                        x       => x(3),
                        y       => y(0),
                        ci      => acarreo(2),
                        so      => suma(3),
                        co      => acarreo(3));
U5: mullbs
        port map(
                        si      => suma(1),
                        x       => x(0),
                        y       => y(1),
                        ci      => '0',
                        so      => suma(4),
                        co      => acarreo(4));
U6: mullbs
        port map(
                        si      => suma(2),
                        x       => x(1),
                        y       => y(1),
                        ci      => acarreo(4),
                        so      => suma(5),
                        co      => acarreo(5));
U7: mullbs
        port map(
                        si      => suma(4),
                        x       => x(2),
                        y       => y(1),
                        ci      => acarreo(5),
                        so      => suma(6),
                        co      => acarreo(6));
U8: mullbs
        port map(
                        si      => acarreo(3),
                        x       => x(3),
                        y       => y(1),
                        ci      => acarreo(6),
                        so      => suma(7),
                        co      => acarreo(7));
U9: mullbs
        port map(
                        si      => suma(5),
                        x       => x(0),
                        y       => y(2),
                        ci      => '0',
                        so      => suma(8),
                        co      => acarreo(8));
```

Continuación Codigo

```
U10: nullbs
        port map(
                        si      => suma(6),
                        x       => x(1),
                        y       => y(2),
                        ci      => acarreo(8),
                        so      => suma(9),
                        co      => acarreo(9));
U11: nullbs
        port map(
                        si      => suma(7),
                        x       => x(2),
                        y       => y(2),
                        ci      => acarreo(9),
                        so      => suma(10),
                        co      => acarreo(10));
U12: nullbs
        port map(
                        si      => acarreo(7),
                        x       => x(3),
                        y       => y(2),
                        ci      => acarreo(10),
                        so      => suma(11),
                        co      => acarreo(11));
U13: nullbs
        port map(
                        si      => suma(9),
                        x       => x(0),
                        y       => y(3),
                        ci      => '0',
                        so      => suma(12),
                        co      => acarreo(12));
U14: nullbs
        port map(
                        si      => suma(10),
                        x       => x(1),
                        y       => y(3),
                        ci      => acarreo(12),
                        so      => suma(13),
                        co      => acarreo(13));
U15: nullbs
        port map(
                        si      => suma(11),
                        x       => x(2),
                        y       => y(3),
                        ci      => acarreo(13),
                        so      => suma(14),
                        co      => acarreo(14));
```

Continuacion de Codigo

```
U16: nullbs
        port map(
                        si      => acarreo(11),
                        x       => x(3),
                        y       => y(3),
                        ci      => acarreo(14),
                        so      => suma(15),
                        co      => acarreo(15));

--asignaciones de seÃ±ales----------
Z(0) <= suma(0);
Z(1) <= suma(4);
Z(2) <= suma(8);
Z(3) <= suma(12);
Z(4) <= suma(13);
Z(5) <= suma(14);
Z(6) <= suma(15);
Z(7) <= acarreo(15);

end arq1;
```

- **Codigo ioc**

```
TOP(
(IOPIN x(0).0);
(IOPIN x(1).0);
(IOPIN x(2).0);
(IOPIN x(3).0);
)

RIGHT(
(IOPIN z(0).0);
(IOPIN z(1).0);
(IOPIN z(2).0);
(IOPIN z(3).0);
(IOPIN z(4).0);
(IOPIN z(5).0);
(IOPIN z(7).0);
)

BOTTOM(
(IOPIN y(0).0);
(IOPIN y(1).0);
(IOPIN y(2).0);
(IOPIN y(3).0);
)
```

- **Código pat**

```
--Puertos de datos-----------------------------
in X     (3 downto 0)     X;
in Y     (3 downto 0)     X;
out Z    (7 downto 0)     X;
--Puertos de Alimentacion----------------------
in VDD  B;
in VSS  B;
-----------------------------------------------
begin
--          X       Y       Z       VDD     VSS
<0ns>:  0       0       ?**     1       0;
<+50ns>:0       1       ?**     1       0;
<+50ns>:0       1       ?**     1       0;
<+50ns>:0       2       ?**     1       0;
<+50ns>:0       3       ?**     1       0;
<+50ns>:0       0       ?**     1       0;
<+50ns>:1       1       ?**     1       0;
<+50ns>:1       1       ?**     1       0;
<+50ns>:1       2       ?**     1       0;
<+50ns>:1       3       ?**     1       0;
<+50ns>:2       1       ?**     1       0;
<+50ns>:2       1       ?**     1       0;
<+50ns>:2       2       ?**     1       0;
<+50ns>:2       3       ?**     1       0;
<+50ns>:3       1       ?**     1       0;
<+50ns>:3       1       ?**     1       0;
<+50ns>:3       2       ?**     1       0;
<+50ns>:3       3       ?**     1       0;
-----------------------------------------------
end;
```

DIAGRAMA ESQUEMATICO

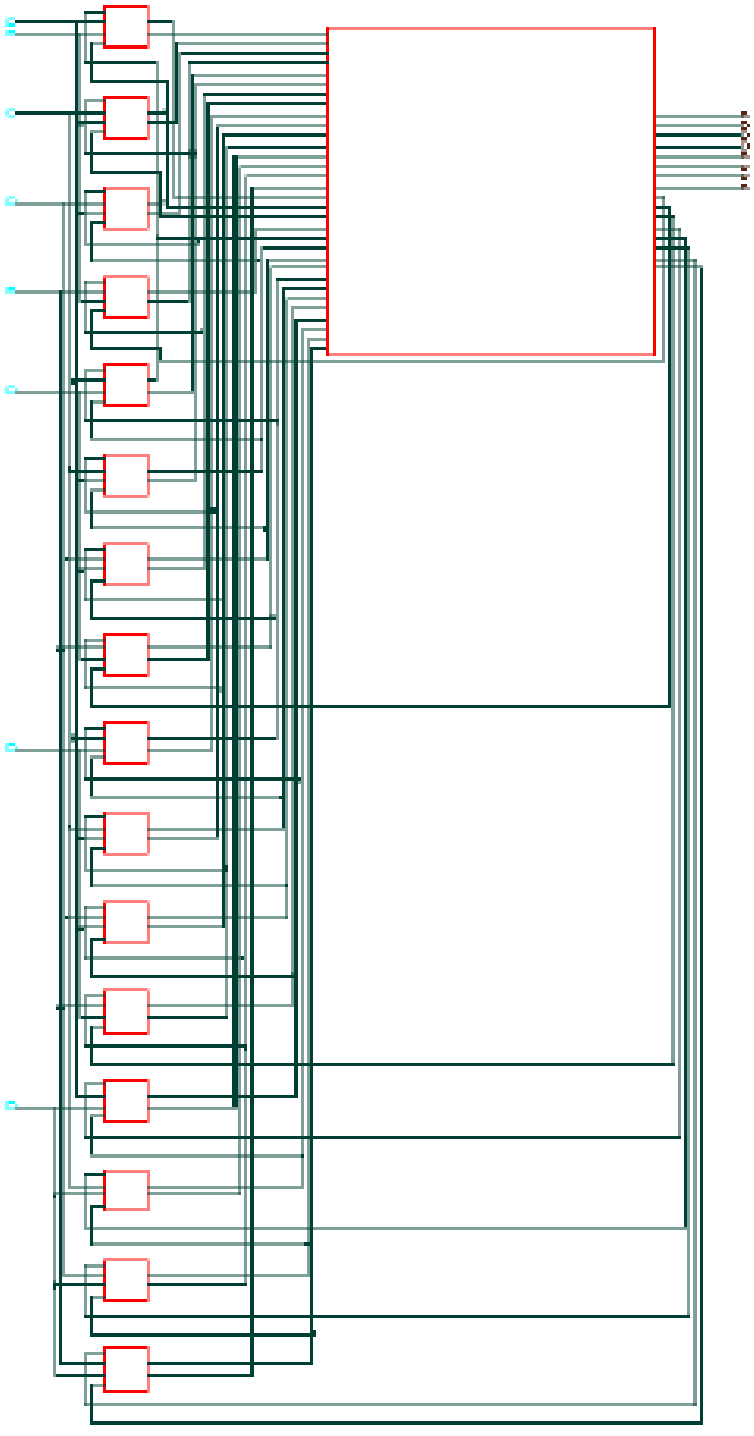
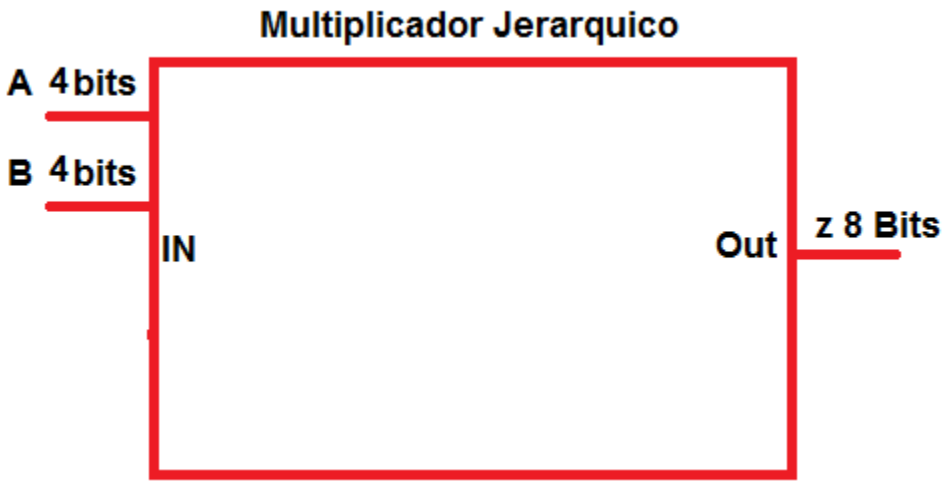
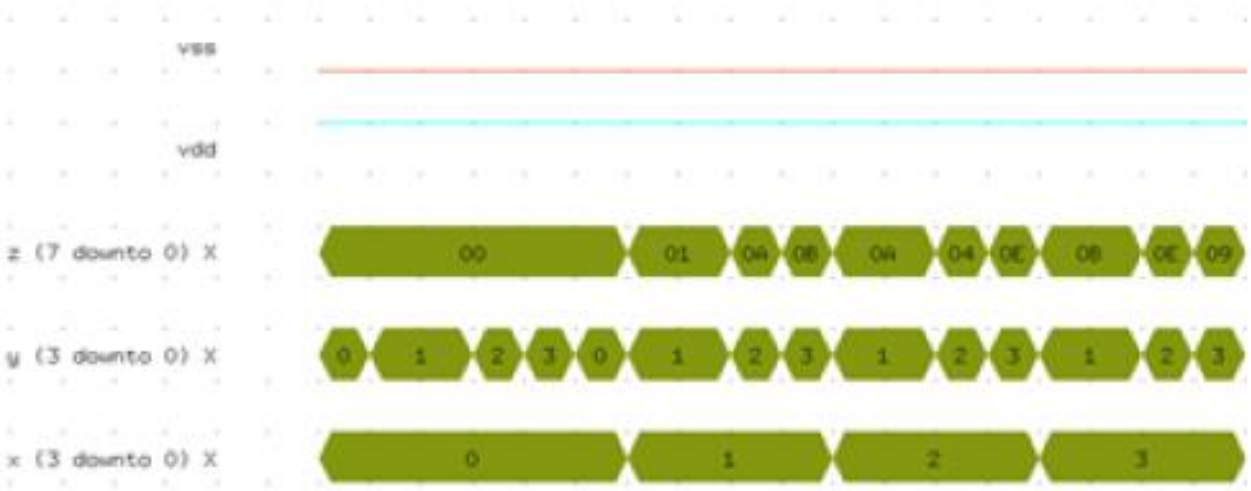


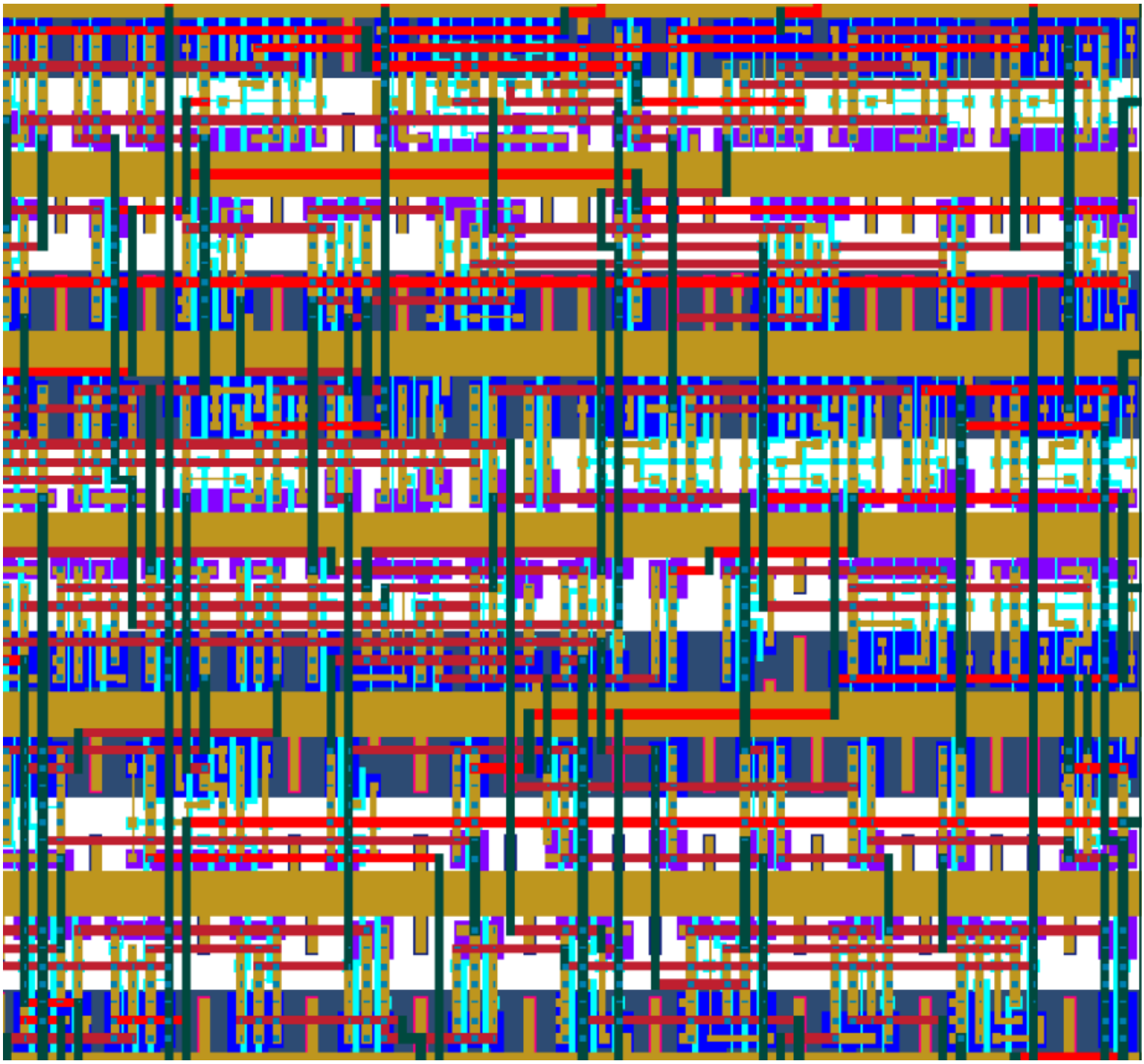
Diagrama caja negra



- **Simulación**



- **Plano real**



- **Tabla de resultados**

```
--COUGAR----------------------------------------------------------

---> Figure size :(  -100,  -100 )

                   (  44100,  40100 )

---> Buildtransistors

                   <--- 712

----------------------------------------------------------------
```

Critical path 2472 pico segundos
Area $A = 41700 micrometros^2$
Transistores =712
Figura= 1,768,410,000