

UNAM

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA



ALUMNA: LILIANA CITLALI AMBROSIO ARAGÓN

MATERIA: LABORATORIO DE DISEÑO DE SISTEMAS DIGITALES

PRÁCTICA 5:

DISEÑO DE UN MULTIPLICADOR

PROFESORA: M.I. NORMA ELVA CHÁVEZ RODRÍGUEZ

SEMESTRE: 2014-1

Fecha de entrega: 26/09/2013

DISEÑO DE UN MULTIPLICADOR BINARIO

OBJETIVO:

El alumno aprenderá a diseñar y construir multiplicadores de $N \times M$ utilizando la plataforma esquemática del software ISE, basado en la creación de macros (símbolos), utilizando como ejemplo de diseño un multiplicador de dos números ambos de dos bits, con visualización del resultado en un display de siete segmentos.

ANÁLISIS DE LA FUNCIÓN MULTIPLICACIÓN:

$$\begin{array}{r}
 \\
 X \\
 \hline
 A1 \\
 B1 \\
 \hline
 A1B0 \\
 A1B1 \\
 \hline
 M2 \\
 M1 \\
 M0
 \end{array}$$

análisis del peor de los casos ,
para saber el número de líneas requerido:

$$\begin{array}{r}
 1 \\
 X \\
 \hline
 1 \\
 1 \\
 \hline
 1 \\
 M3M2M1M0
 \end{array}$$

DIAGRAMA DE BLOQUES:

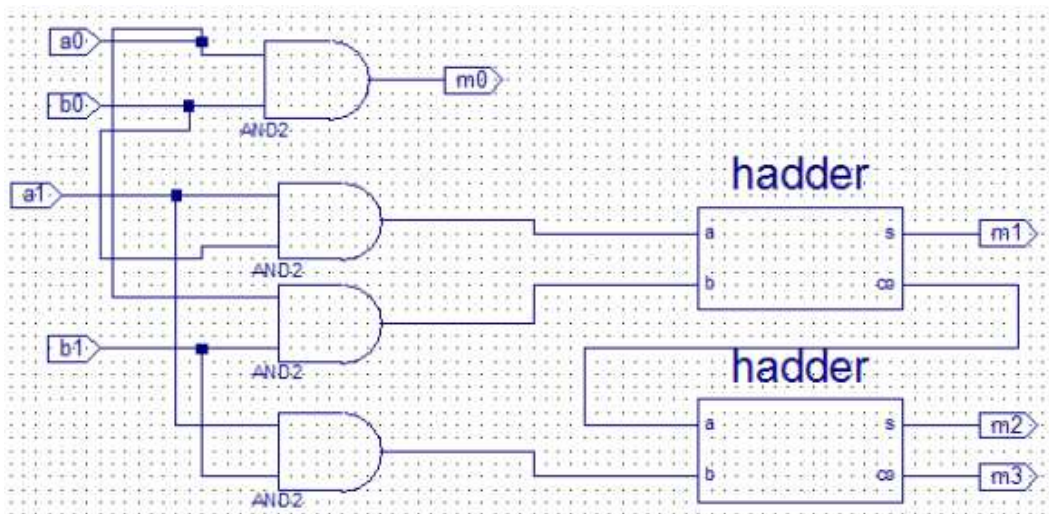
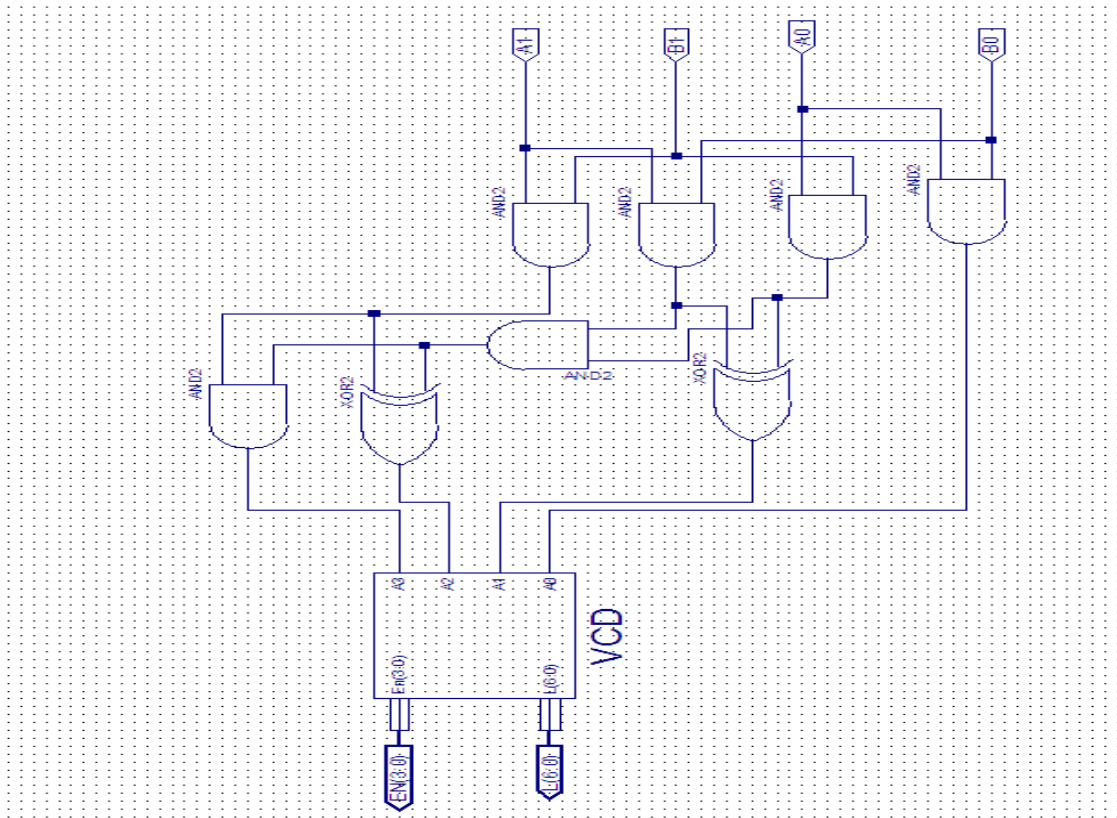
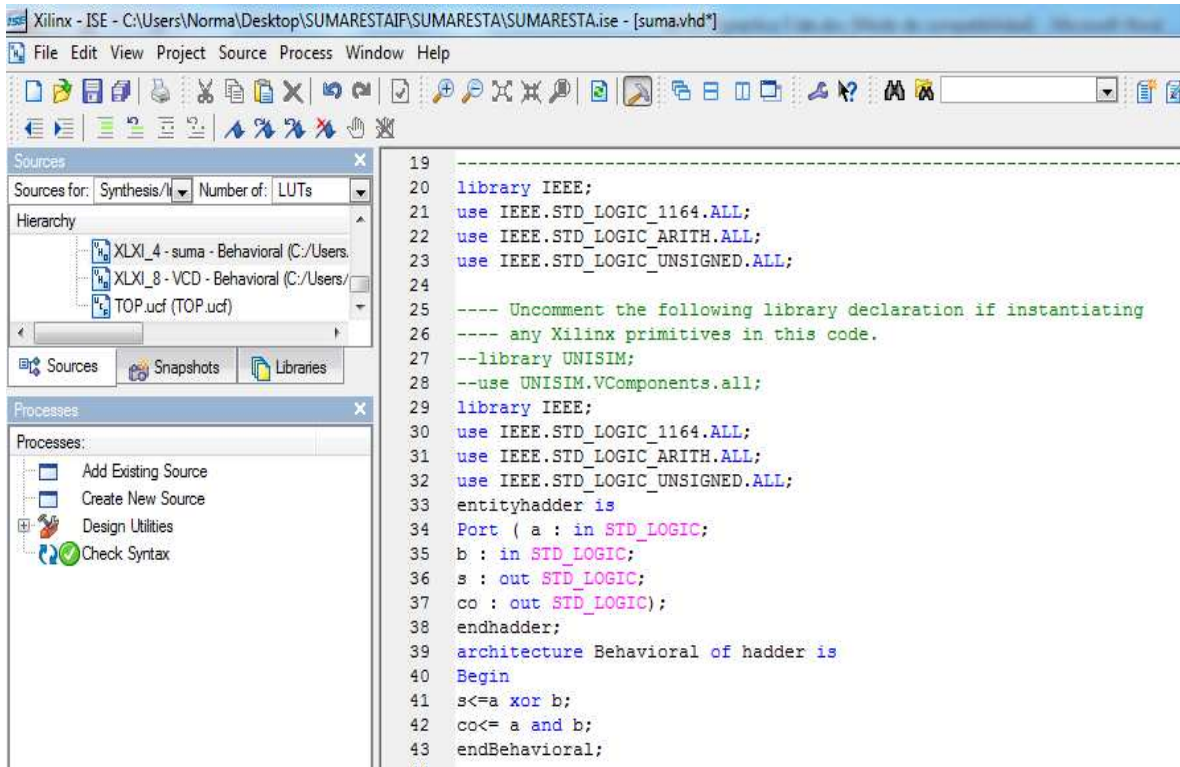


DIAGRAMA LÓGICO:



CÓDIGO EN VHDL PARA EL MEDIO SUMADOR

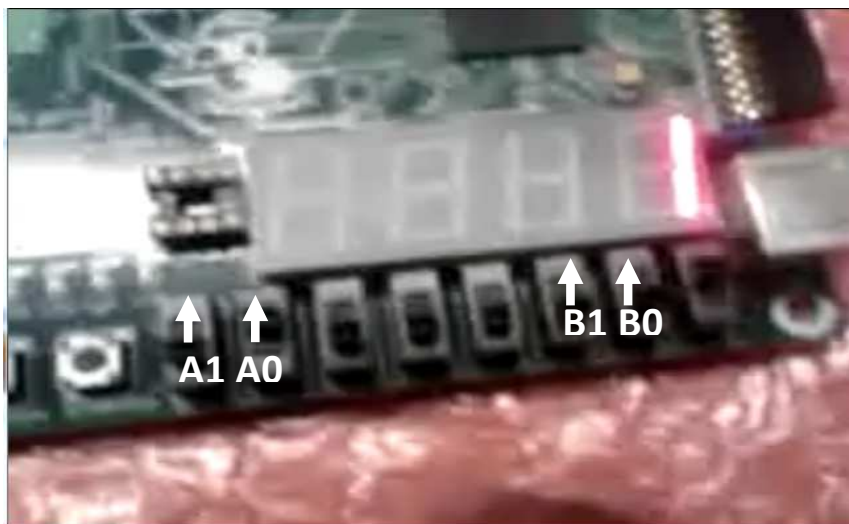


The screenshot shows the Xilinx ISE IDE with a VHDL file named [suma.vhd]. The code defines a half adder entity with two input ports (a and b) and two output ports (s and co). The architecture is Behavioral and uses the IEEE standard logic libraries. The logic is implemented as follows:

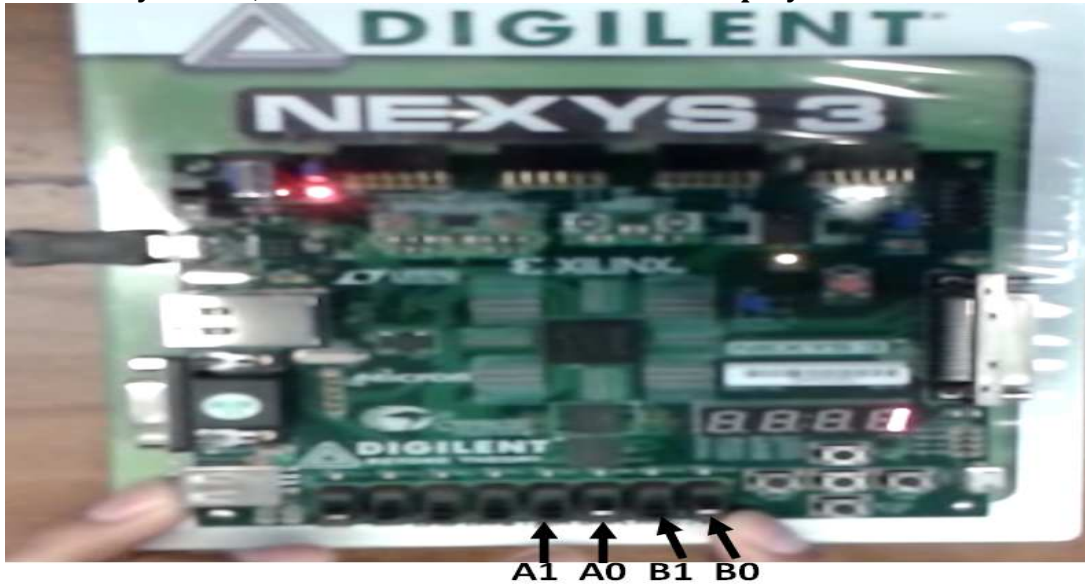
```
19 -----  
20 library IEEE;  
21 use IEEE.STD_LOGIC_1164.ALL;  
22 use IEEE.STD_LOGIC_ARITH.ALL;  
23 use IEEE.STD_LOGIC_UNSIGNED.ALL;  
24  
25 ---- Uncomment the following library declaration if instantiating  
26 ---- any Xilinx primitives in this code.  
27 --library UNISIM;  
28 --use UNISIM.VComponents.all;  
29 library IEEE;  
30 use IEEE.STD_LOGIC_1164.ALL;  
31 use IEEE.STD_LOGIC_ARITH.ALL;  
32 use IEEE.STD_LOGIC_UNSIGNED.ALL;  
33 entityhadder is  
34 Port ( a : in STD_LOGIC;  
35       b : in STD_LOGIC;  
36       s : out STD_LOGIC;  
37       co : out STD_LOGIC);  
38 endhadder;  
39 architecture Behavioral of hadder is  
40 Begin  
41   s<=a xor b;  
42   co<= a and b;  
43 endBehavioral;  
44
```

PRUEBAS AL DISEÑO:

Si A = 01 y B = 01, el resultado se muestra en el display



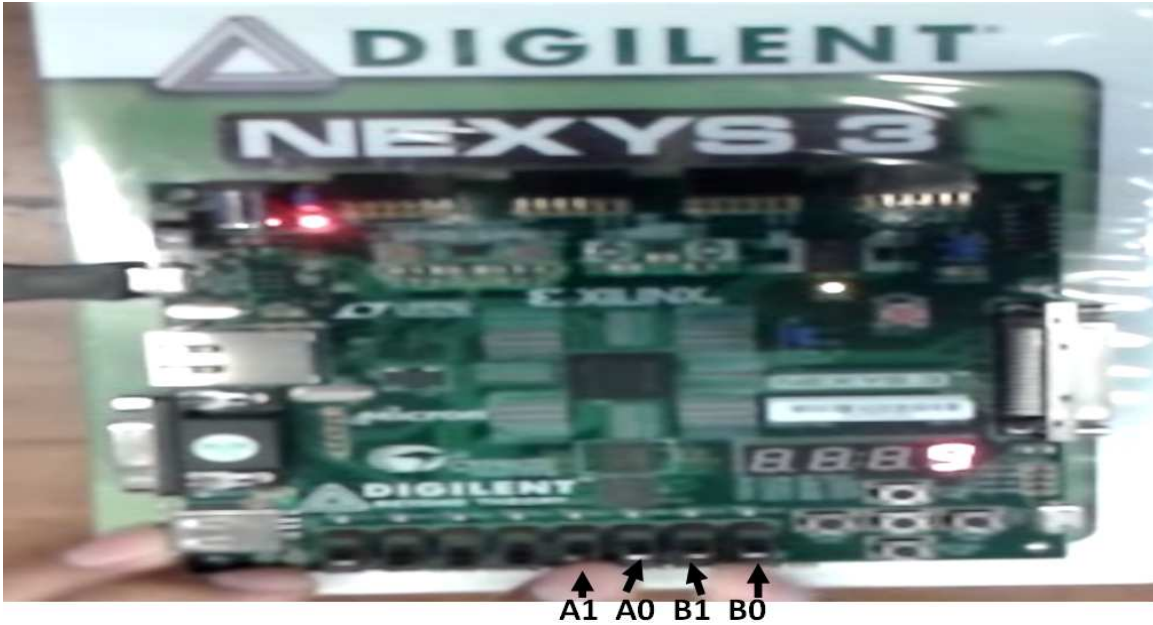
Si $A = 01$ y $B = 01$, el resultado se muestra en el display



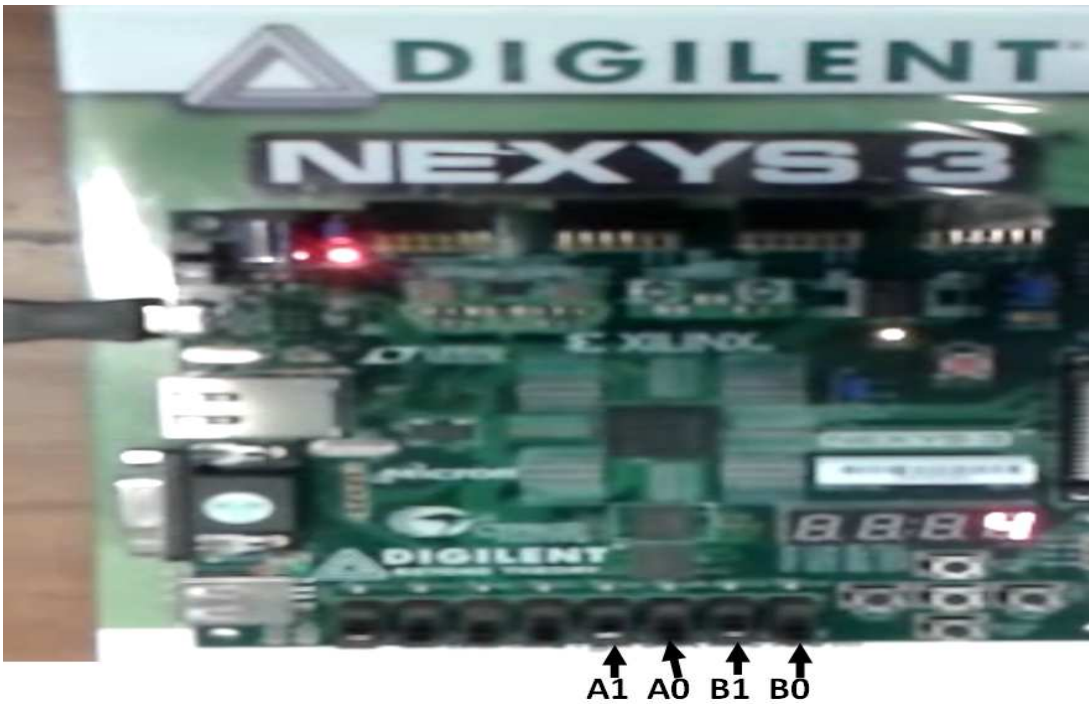
Si $A = 01$ y $B = 11$, el resultado se muestra en el display



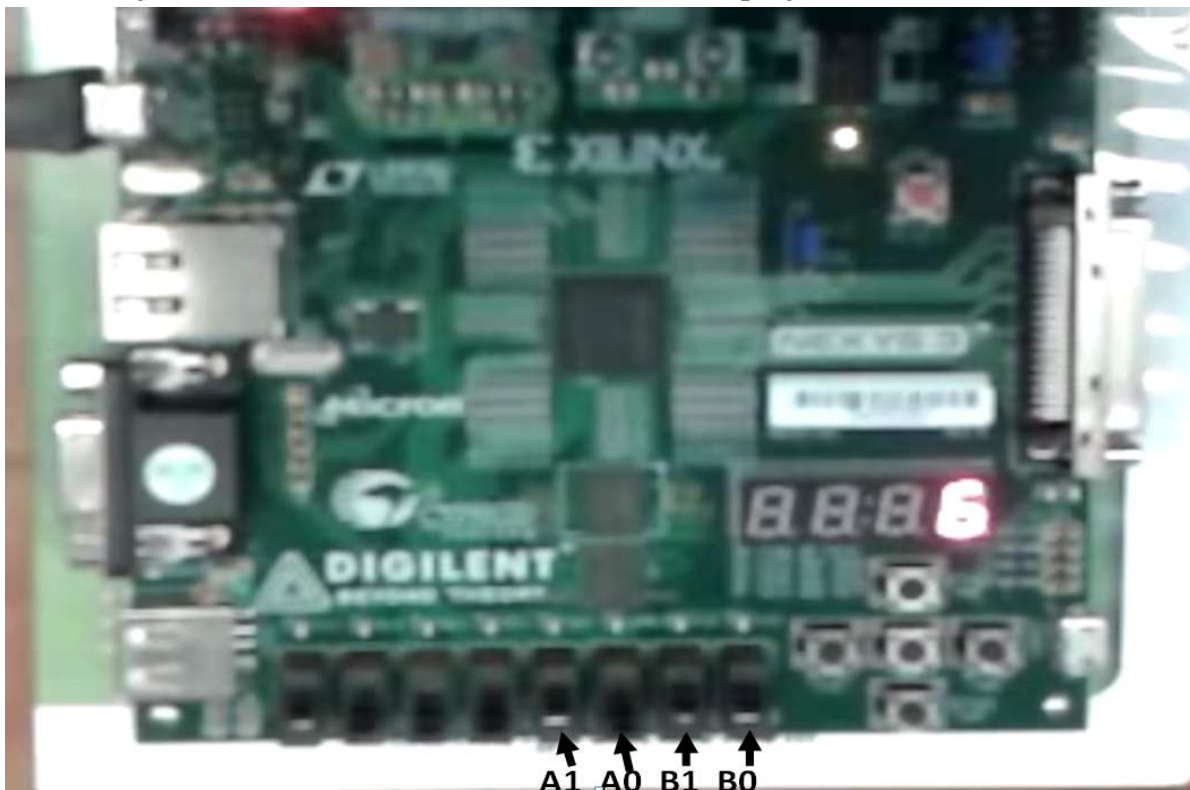
Si $A = 11$ y $B = 11$, el resultado se muestra en el display



Si $A = 11$ y $B = 11$, el resultado se muestra en el display



Si $A = 10$ y $B = 11$, el resultado se muestra en el display



CONCLUSIÓN

Ésta práctica es adecuada para comprender la operación de la multiplicación con números binarios, pero no solamente eso sino que la implementación de ello a la vida real, ya que podemos utilizar el ingenio y no solamente utilizarlo en una tarjeta sino en alguna combinación de algún proyecto, es impresionante poder manipular los dispositivos en este caso las tarjetas de la compañía Digilent, ya que al realizar la práctica yo me vuelvo la diseñadora, por lo que decido donde y cuantas entradas y salidas tiene mi sistema y es así como de alguna manera hago que mi dispositivo funcione y obtenga en forma sencilla el diseño digital que yo quiera.