**Laboratorio de Arquitectura de computadores GITT Entrega 2**

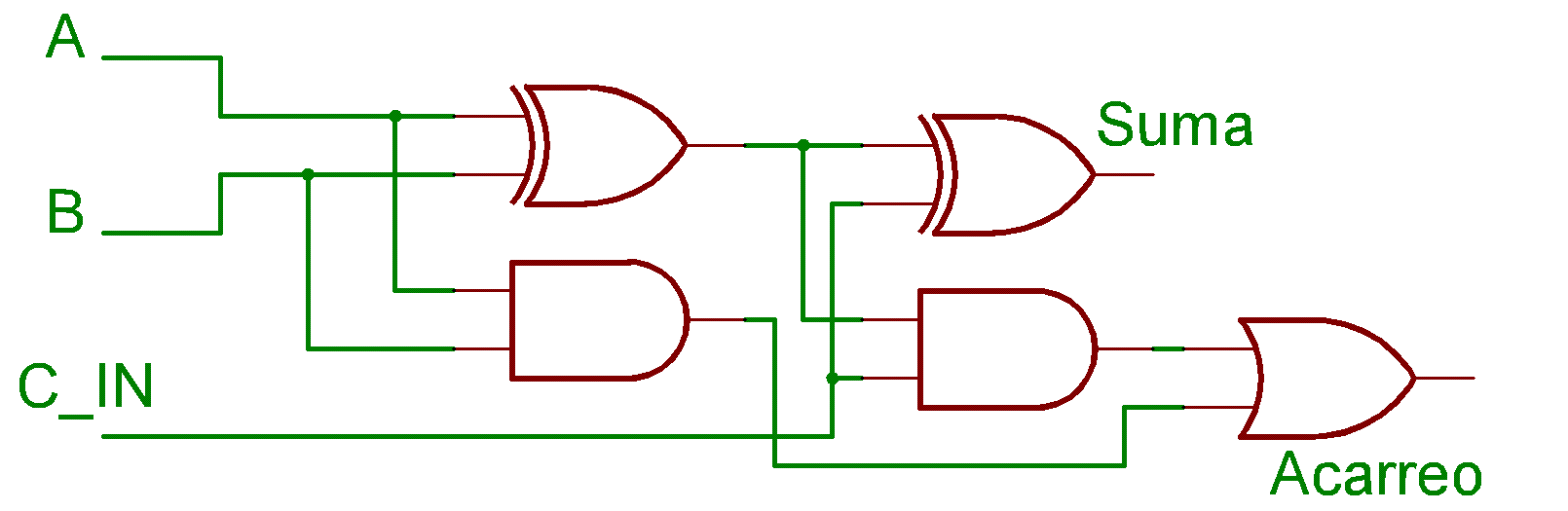
**Fecha de entrega: Viernes 19/10/2012 Grupo: Viernes**

**Autores: Manuel Montoya Catalá**

**Cuestión 1 y 2**

- Para calcular el tiempo que tarda en producirse la suma y acarreo tenemos que ver cuánto tardan en estabilizarse las salidas (suma y acarreo) en el peor de los casos.

Siendo la estructura del sumador elemental :

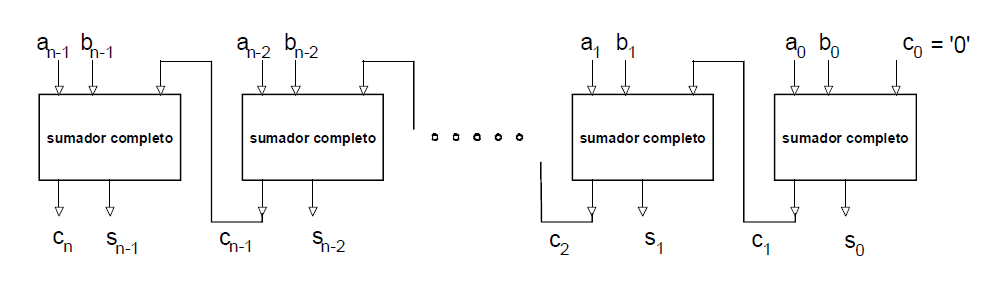


- Suponiendo que el retardo de propagación "T" sea el mismo para cada puerta,

en el peor de los casos:

· La suma tarda 2T

· El acarreo tarda 3T

- Cuando interconectamos varios sumadores elementales: 

- Las salidas finales de cada sumador no podrán empezar a generarse hasta que no se les propague el acarreo que produjo el sumador anterior

- En el peor de los casos, al cambiar el valor de las entradas, el acarreo de salida de un sumador puede cambiar, al hacerlo cambia el valor del acarreo de entrada del siguiente sumador, que a su vez, debido a esto puede también cambiar el valor de su acarreo de salida.

Así pues el acarreo se va propagando de sumador en sumador, por lo que para que una salida sea correcta, se tiene que esperar a que su acarreo de entrada este estable.

# Asi pues, en el peor de los casos posibles cambia:

1-El acarreo de salida del primer sumador, debido a un cambio en sus entradas ( a o b)

2-El acarreo de salida de todos los demás sumadores , debido al cambio de acarreo del

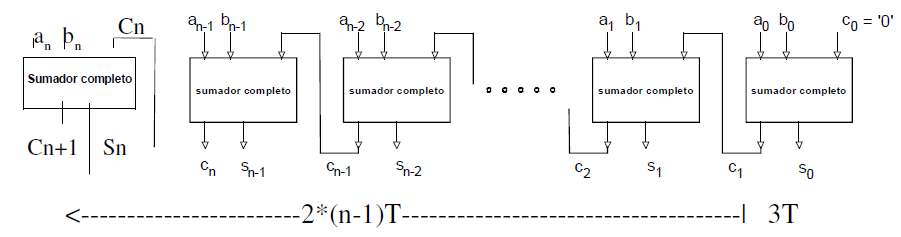
sumador anterior, propagándose asi el cambio entre los sumadores

**# Cálculo de retardos:**

- El primer cambio en el acarreo de salida tarda "3T".

- Como se puede ver en la figura 1, el sumador elemental tarda "2T" en dar un acarreo de salida estable cuando la entrada que varia es el acarreo de entrada.

Teniendo un sumador de "N" bits, el cambio de acarreo se propaga "N-1" veces en el peor de los casos, por lo que el tiempo necesario para que el acarreo de salida del último sumador sea estable es "2T\*(N-1)"



Asi pues, el tiempo que tarda en estabilizarse el acarreo de salida del ultimo sumador es:

**Retardo Acarreo = 2T\*(N-1) + 3T = 2T\*N + T**

- Para el caso del retardo máximo de la salida de la suma, tenemos en cuenta que también tenemos que esperar a que se propaguen el cambio en los acarreos en el peor de los casos

hasta el ultimo sumador.

El acarreo de entrada del último sumador tarda en estabilizarse "2T\*(N-2) + T"

Un cambio en la salida debido a un cambio en el acarreo tarda "T" segundos en propagarse:

**Retardo Suma = = 2T\*(N-2) + 3T + T = 2T\*N**

**Retardo de nuestro propagador de 8 bits**

# Para ver el tiempo que tarda en producirse la suma y acarreo de nuestro sumador de 8 bits, establecemos el peor de los casos, aquel en el que un cambio en la entrada del sumador (a o b)

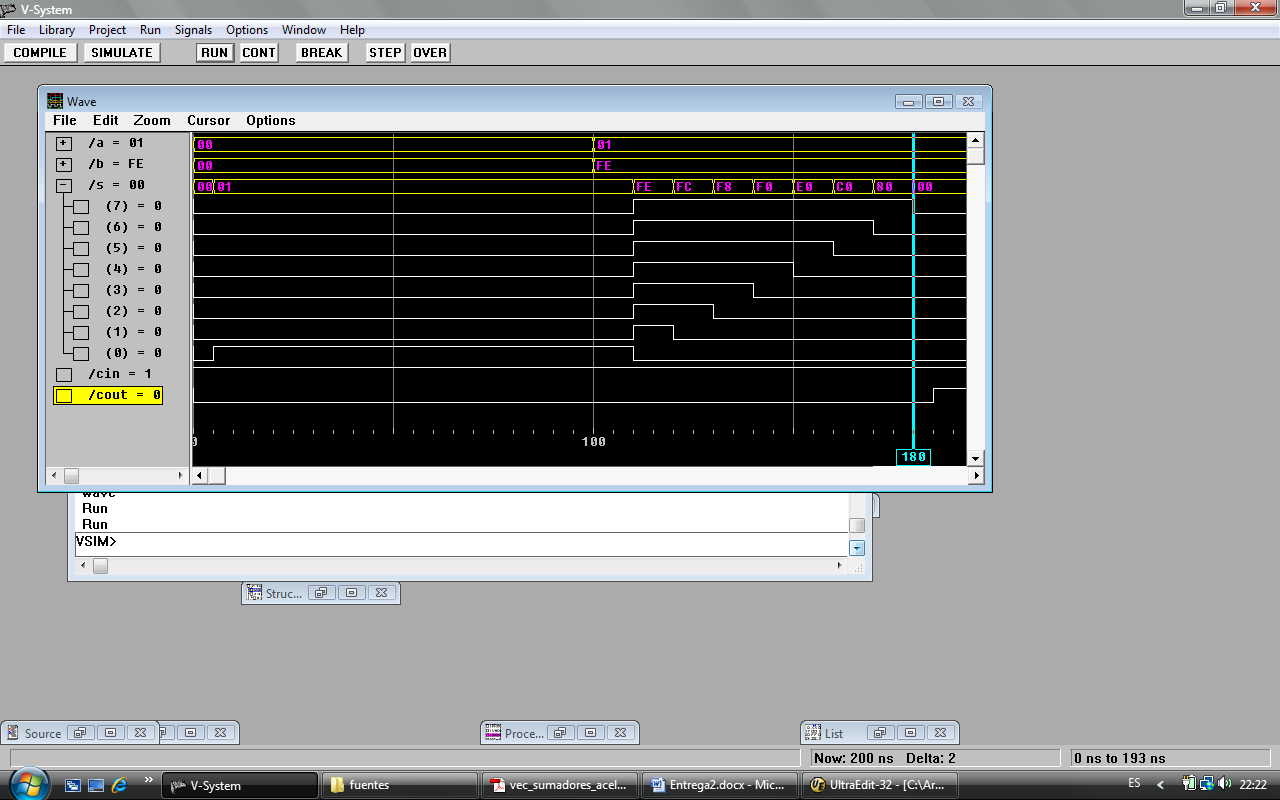
produce un cambio en el acarreo de salida, dicho cambio se propaga por todos los sumadores hasta el ultimo, donde además tiene que cambiar también el valor de la salida. Para ello:

- Ponemos un Byte de entrada con todo 1´s menos el más bajo b = "1111 1110"

y el otro a 0´s a= "0000 0000" con el acarro de entrada a 1.

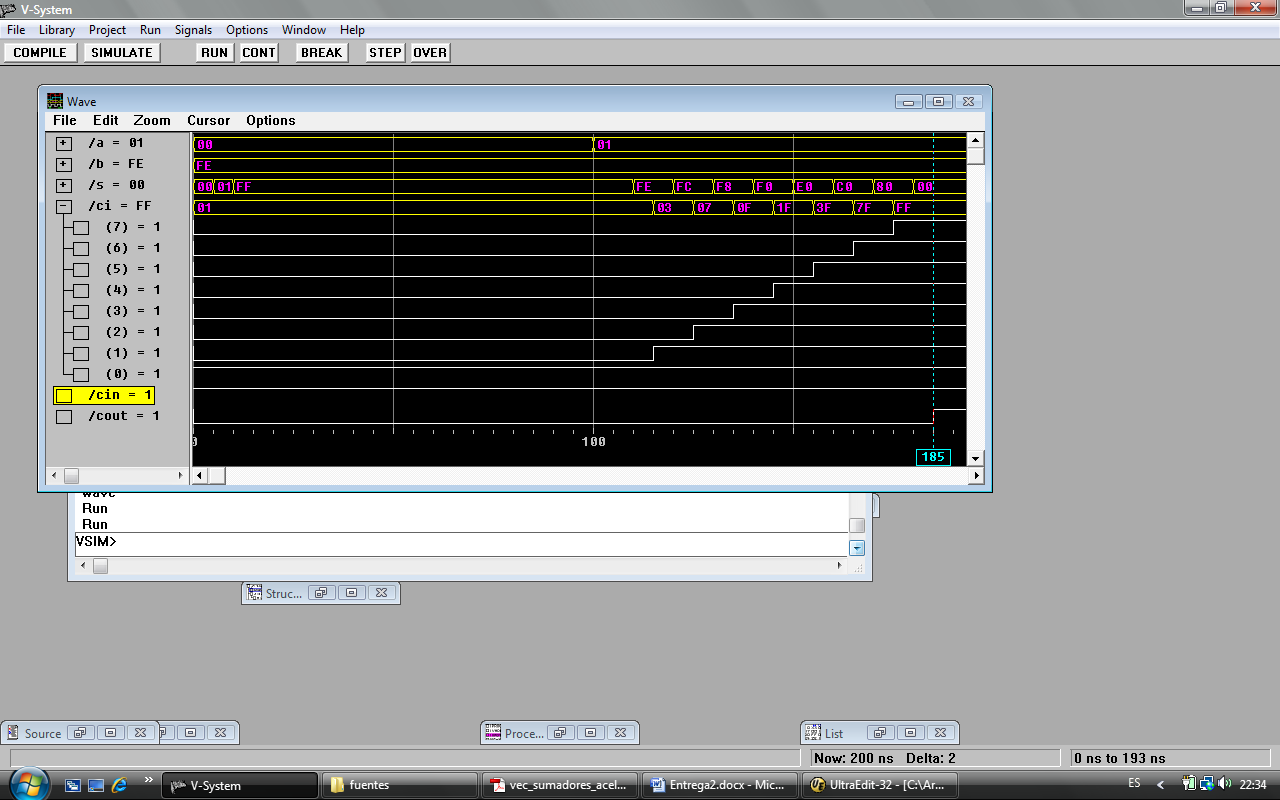
- Si ahora cambiamos a : a= "0000 0001" tenemos el peor de los casos

#Simulación del retardo de suma: T = 5 ns



Retardo de suma = 2T\*N = 2\*8\*5 ns = 80 ns **Coincide !!!!**

#Simulación del retardo de acarreo: T = 5 ns



Retardo de acarreo = 2T\*N + T = 2\*8\*5 + 5 ns = 85 ns **Coincide !!!!**

**Código de sumprop8:**

entity sumprop8 is

generic (retardo\_puerta:time := 3 ns);

port (a: IN bit\_vector (7 downto 0);

b: IN bit\_vector (7 downto 0);

cin: In bit; --Acarreo entrada

s: OUT bit\_vector (7 downto 0);

cout: OUT bit); --Acarreo salida

end;

architecture sumprop8\_rtl of sumprop8 is

signal ci:bit\_vector (7 downto 0) := "00000000"; --Acarreos internos

begin

ci(0) <= cin; -- cin es mas visual a la hora de hacer represencaciones que ci(0)

--cin es el acarreo de entrada al sumprop8

-- Instanciamos los 8 sumadores y hacemos las conexiones entre ellos

ss0 : entity work.sum\_elemental generic map (5 ns)

port map (a(0),b(0),ci(0),s(0),ci(1));

ss1 : entity work.sum\_elemental generic map (5 ns)

port map (a(1),b(1),ci(1),s(1),ci(2));

ss2 : entity work.sum\_elemental generic map (5 ns)

port map (a(2),b(2),ci(2),s(2),ci(3));

ss3 : entity work.sum\_elemental generic map (5 ns)

port map (a(3),b(3),ci(3),s(3),ci(4));

ss4 : entity work.sum\_elemental generic map (5 ns)

port map (a(4),b(4),ci(4),s(4),ci(5));

ss5 : entity work.sum\_elemental generic map (5 ns)

port map (a(5),b(5),ci(5),s(5),ci(6));

ss6 : entity work.sum\_elemental generic map (5 ns)

port map (a(6),b(6),ci(6),s(6),ci(7));

ss7 : entity work.sum\_elemental generic map (5 ns)

port map (a(7),b(7),ci(7),s(7),cout);

--cout es el acarreo de salida al sumprop8

end;

**Código de la entidad de pruebas:**

-- Entidad que usa el sumprop8 y le da valores a las entradas

-- para testearlo

entity sumprop8\_1\_test is end;

architecture test of sumprop8\_1\_test is

signal a,b,s :bit\_vector (7 downto 0) := "00000000"; --Entradas, salidas

signal cin, cout :bit := '0'; --Acarreo de entrada y salida del sumprop8

begin

ss1: entity work.sumprop8 generic map (5 ns) port map (a,b,cin,s,cout);

process

begin

a<= "00000000"; b <= "11111110"; cin <= '1';

wait for 100 ns;

a<= "00000001";

wait for 100 ns;

end process;

end;

**Código de sum\_elemental**

entity sum\_elemental is

generic (retardo\_puerta:time := 3 ns);

port (a: IN bit;

b: IN bit;

c: In bit; --Acarreo

s: OUT bit;

c2: OUT bit); --Acarreo salida

end;

architecture sum\_elem\_rtl of sum\_elemental is

signal Xs1,Xc1,Xc2,Xc3 : bit;

begin

Xs1 <= a xor b after retardo\_puerta;

s <= Xs1 xor c after retardo\_puerta;

Xc1 <= a and b after retardo\_puerta;

Xc2 <= a or b after retardo\_puerta;

Xc3 <= c and Xc2 after retardo\_puerta;

c2 <= Xc1 or Xc3 after retardo\_puerta;

end;