

UNIVERSITAT POLITÈCNICA DE CATALUNYA
ARQUITECTURA DE COMPUTADORS D'ALTES PRESTACIONS

Práctica 1

Sumador de 1 bit

Carlota Catot
Miguel Antunez

Grupo 7

Quatrimestre primavera 2020-2021

Índice

1. Pregunta 1	2
2. Pregunta 2	3
2.1. Ventana textual	3
2.2. Ventana temporal	4
2.3. Tabla retardos	4
2.4. Justificación	4
3. Pregunta 3	5
3.1. Tabla de la verdad con retardos	5
3.2. Fila A de la tabla	6
3.3. Fila B de la tabla	6
3.4. Explicación ejercicio	7

1. Pregunta 1

Entregue la elaboración RTL que efectúa Quartus del diseño efectuado.

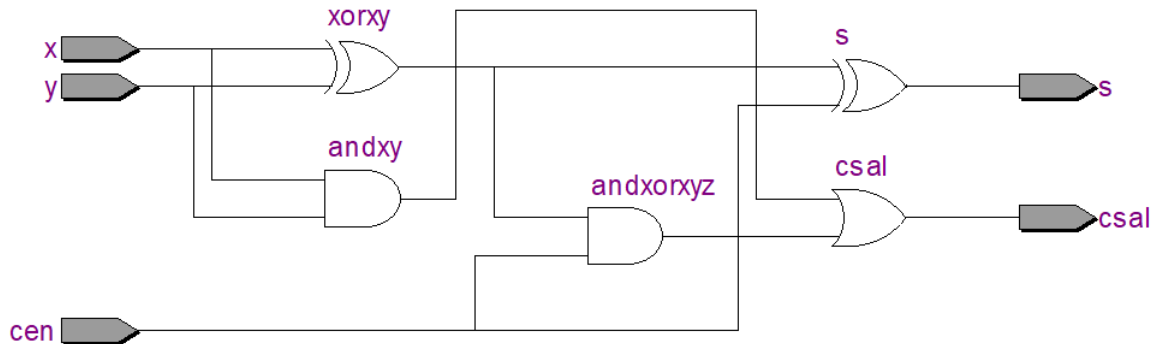


Figura 1: Diagrama RTL Quartus

En la figura 1 podemos ver el RTL que nos muestra quartus, que como podemos ver, es muy parecido al esquema del enunciado, las únicas diferencias que percibimos son las del orden de los cables.

Para poder generar el esquema hemos usado el código VHL presentado a continuación:

```
library ieee;
use ieee.std_logic_1164.all;
entity s1bit is
port(
    x : in std_logic;
    y : in std_logic;
    cen: in std_logic;
    s: out std_logic;
    csal: out std_logic
);
end s1bit;

architecture flujodatos of s1bit is
    constant retardoxor: time := 15 ns;
    constant retardoand: time := 10 ns;
    constant retardoor: time := 15 ns;

    signal xorxy, andxy, andxorxyz : std_logic;

begin
    xorxy <= x xor y after retardoxor;
    andxy <= x and y after retardoand;
    s <= xorxy xor cen after retardoor;
    andxorxyz <= xorxy and cen after retardoand;
    csal <= andxorxyz or andxy after retardoor;
end flujodatos;
```

2. Pregunta 2

Cree un frente de onda para medir el retardo de las 8 posibles combinaciones de los valores de las señales de entrada de un sumador de 1 bit. Utilice la orden force para estimular los puertos de entrada en el simulador Modelsim. Adjunte la parte de la ventana textual (list.lst) y la parte de la ventana temporal (wave.bmp) cuando se estimula el circuito con el vector de entrada ($x = 0$, $y = 1$, $cen = 1$). Muestre en la ventana textual el cálculo del retardo. Así mismo, identifique en ventana temporal la evolución de las señales para este vector (utilice los cursores). Para cada vector de entrada indique el retardo observado. Justifique la respuesta.

2.1. Ventana textual

ps	/s1bit/x	/s1bit/xorxy	/s1bit/y	/s1bit/andxy	/s1bit/cen	/s1bit/andxorxyz
0	0	0	0	0	0	0
10000	0	0	0	0	0	0
15000	0	0	0	0	0	0
25000	0	0	1	0	0	0
40000	0	0	1	1	0	0
50000	0	1	0	1	0	0
65000	0	1	0	0	0	0
75000	0	1	1	0	0	0
85000	0	1	1	0	0	1
100000	1	0	0	0	1	1
110000	1	0	0	0	1	0
115000	1	0	0	1	1	0
125000	1	0	1	1	0	0
135000	1	0	1	1	0	1
140000	1	0	1	0	0	1
150000	1	1	0	0	1	1
160000	1	1	0	0	1	0
165000	1	1	0	1	1	0
175000	1	1	1	1	1	0

Figura 2: Fichero list.lst para la ejecución de las 8 combinaciones

2.2. Ventana temporal

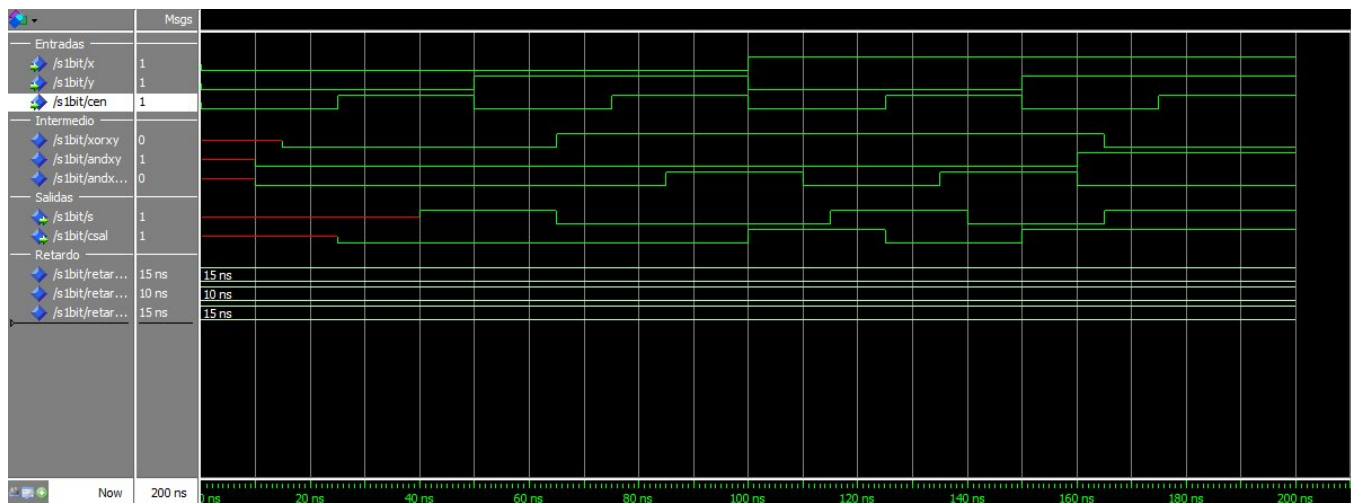


Figura 3: Fichero list.lst para la ejecución de las 8 combinaciones

2.3. Tabla retardos

El retardo presentado en las 8 combinaciones se puede observar en la siguiente tabla (siendo los valores del vector de entrada (x,y,cen) en el orden establecido):

vector de entrada	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
retardo observado (ns)	30	40	30	40	30	40	30	30

2.4. Justificación

El retardo mínimo es de 30ns el cual corresponde al tiempo de retardo de las dos puertas XOR, en los casos que el retardo es de 40ns es debido a la combinación de los retardos de las tres puertas (xorxy, andxorxyz y el Csal), este caso ocurre cuando cen tiene valor 1, exceptuando el último caso (111). Las puertas XOR, siempre tienen que esperar las dos entradas para operar, mientras que las puertas AND y OR, se pueden ejecutar cuando reciben una entrada de 0 y 1 respectivamente.

3. Pregunta 3

Indique, para cada valor del vector entrada E1, un vector de entrada previo E0 que provoque el retardo (ret1) observado en la pregunta anterior. Para cada caso, minimice el número de cambios en los bits de los vectores de entrada.

Adjunte los frentes de onda utilizados para los casos a y b indicados en la tabla y la ventana temporal. Utilce los cursores para medir el retardo.

3.1. Tabla de la verdad con retardos

	E0			E1			retardo observado
	x	y	cen	x	y	cen	ret1 (ns)
A	1	0	0	0	0	0	30
	0	1	1	0	0	1	40
	1	1	0	0	1	0	30
B	0	0	1	0	1	1	40
	1	1	0	1	0	0	30
	0	0	1	1	0	1	40
	1	1	1	1	1	0	30
	0	1	0	1	1	1	30

Para generar la tabla completa hemos usado el un fichero con un frente de onda. El fichero *estimulacion.do* con el codigo presentado a continuación generando la onda de la figura 6

```

force s1bit/x 1 0 ns, 0 50 ns, 0 100 ns, 0 150 ns, 1 200 ns, 0 250 ns, 0 300 ns, 0 350 ns,
              1 400 ns, 1 450 ns, 0 500 ns, 1 550 ns, 1 600 ns, 1 650 ns, 0 700 ns, 1 750 ns

force s1bit/y 0 0 ns, 0 50 ns, 1 100 ns, 0 150 ns, 1 200 ns, 1 250 ns, 0 300 ns, 1 350 ns,
              1 400 ns, 0 450 ns, 0 500 ns, 0 550 ns, 1 600 ns, 1 650 ns, 1 700 ns, 1 750 ns

force s1bit/cen 0 0 ns, 0 50 ns, 1 100 ns, 1 150 ns, 0 200 ns, 0 250 ns, 1 300 ns, 1 350 ns,
                0 400 ns, 0 450 ns, 1 500 ns, 1 550 ns, 1 600 ns, 0 650 ns, 0 700 ns, 1 750 ns

run -all

```

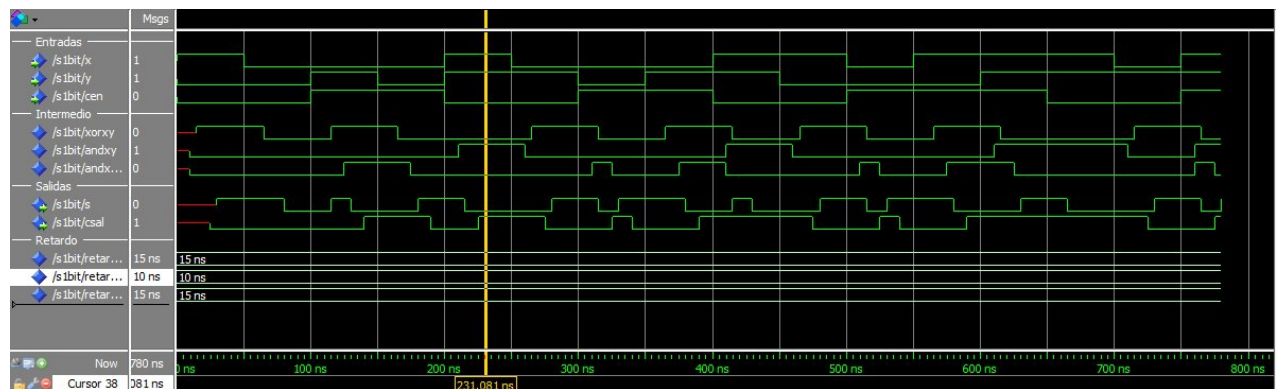


Figura 4: Ventana temporal con las ondas generadas a partir de la tabla anterior

Para las filas de la tabla marcadas con una A i una B hemos generado unas ondas específicas, para ello se han creado archivos concretos de frente de onda con únicamente las dos entradas correspondientes E0 y E1, para poder ver correctamente la ventana temporal, ajustar los cursores y apreciar los retardos.

3.2. Fila A de la tabla

```
force s1bit/x 1 0 ns, 0 100 ns
force s1bit/y 1 0 ns, 1 100 ns
force s1bit/cen 0 0 ns, 0 100 ns
run -all
```

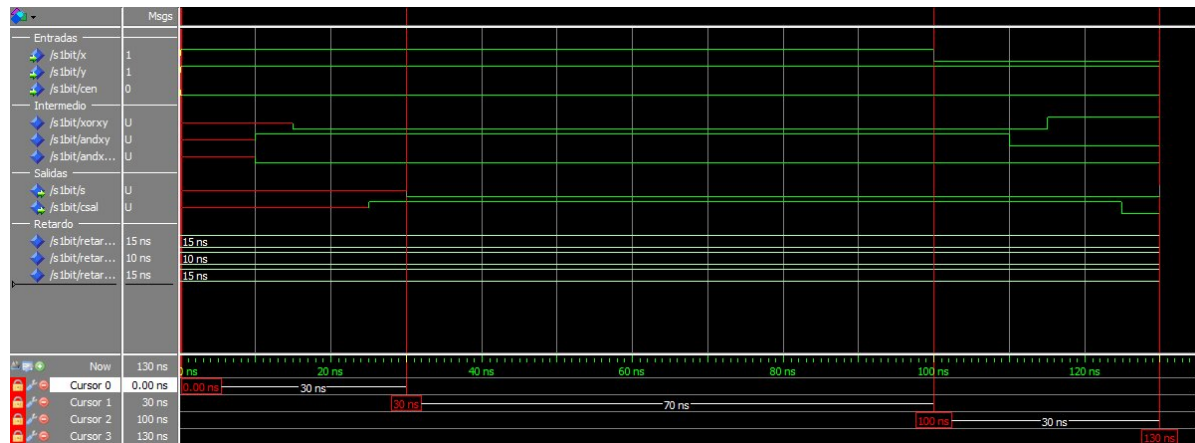


Figura 5: Ventana temporal para la fila A de la tabla

3.3. Fila B de la tabla

```
force s1bit/x 0 0 ns, 0 100 ns
force s1bit/y 0 0 ns, 1 100 ns
force s1bit/cen 1 0 ns, 1 100 ns
run -all
```

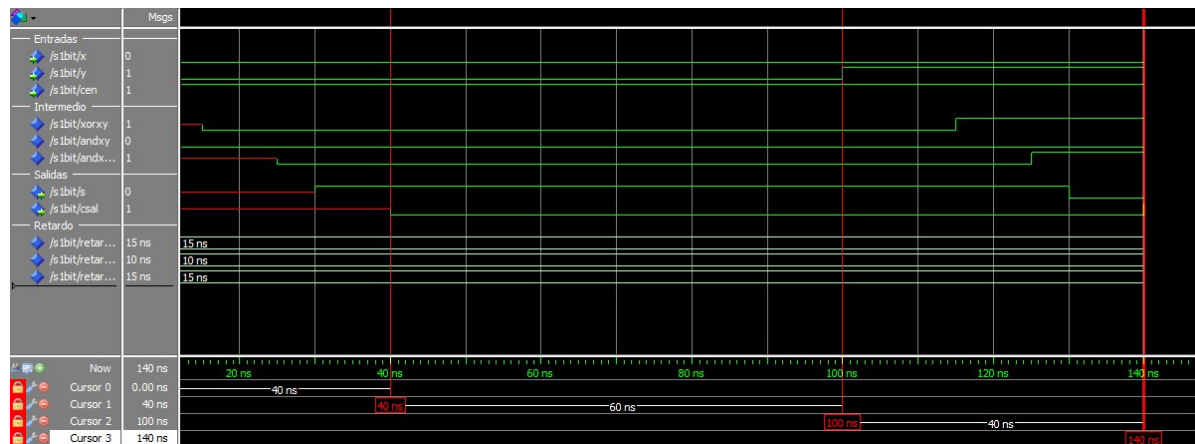


Figura 6: Ventana temporal para la fila B de la tabla

3.4. Explicación ejercicio

Una vez calculado los retardos de todas las posibles combinaciones en la pregunta 2, podemos obtener el retardo $ret1$.

Calculamos la entrada $E0$ en función del retardo de la entrada $E1$ y además que $E0$ provoque un cambio en la salida respecto a $E1$. Una vez se estabiliza la señal se calcula el retardo. En la siguiente iteración se tendrá el retardo acumulado de la anterior pero no se sumará ningún retardo más porque la señal ya está estabilizada.

El periodo mínimo de cambio de estados ha de ser mayor que el retardo máximo, en otro caso el sumador no funcionaria correctamente.