Universitat Politècnica de Catalunya

Arquitectura de Computadores de Altas Prestaciones

**Práctica 2**

**Sumador de 4 bits**

Alumnos:

Manuel Velastegui

Carlos Andres Rodríguez Torres

**Grupo 5**

Cuatrimestre Primavera 2024-2025



**ÍNDICE**

[**1. Pregunta 1 3**](#_ucl316v6kc09)

[1.1 Justificación 3](#_j6nlsel2454t)

[1.2 Código 3](#_a1etykgl3r3b)

[1.3 Diagrama Sumador 4bits 4](#_mc5jpfegyduz)

[**2. Pregunta 2 4**](#_tun4wn8w50y1)

[2.1 Justificación. 4](#_kuvdgnd7jpt2)

[2.2 Código 5](#_qjnc7q9z3izz)

[2.3 Ventana Temporal 6](#_kpumbxkdv93r)

[**3. Pregunta 3 6**](#_ow5f5fi4v9lz)

[3.1 Retardo mínimo 7](#_kc3hracu5ik6)

[3.2 Retardo máximo 7](#_n4peie5aqf3d)

[3.3 Código 8](#_7s79ws4zht7b)

# 

# 1. Pregunta 1

Considere el sumador de 4 bits especificado en VHDL mediante sentencias Generate. Entregue la elaboración RTL que efectúa Quartus del diseño. Debe visualizarse la especificación RTL de uno de los sumadores.

## 1.1 Justificación

Para realizar el diagrama, se hizo la modificación del archivo snbits, en donde se ajusta el parámetro de los nbits que se quieren modelar.

## 1.2 Código

**entity snbits is**

**generic** (n: positive:= 4);

**port** (a: in std\_logic\_vector(n-1 downto 0);

b: in std\_logic\_vector(n-1 downto 0);

cen: in std\_logic;

s: out std\_logic\_vector(n-1 downto 0);

csal: out std\_logic );

**end** snbits;

**architecture estructural of** snbits **is**

**component** s1bit

generic(retardoxor: time ; retardoand: time ; retardoor: time );

port ( x: in std\_logic; y: in std\_logic; cen: in std\_logic;

s: out std\_logic; csal: out std\_logic);

**end component;**

**signal** c : std\_logic\_vector (n **downto** 0);

**begin**

c(0) <= cen;

sumador: for i in 0 to n-1 **generate**

sumi: s1bit **generic** map (retardoxor => 15 ns, retardoand => 10 ns, retardoor => 15 ns)

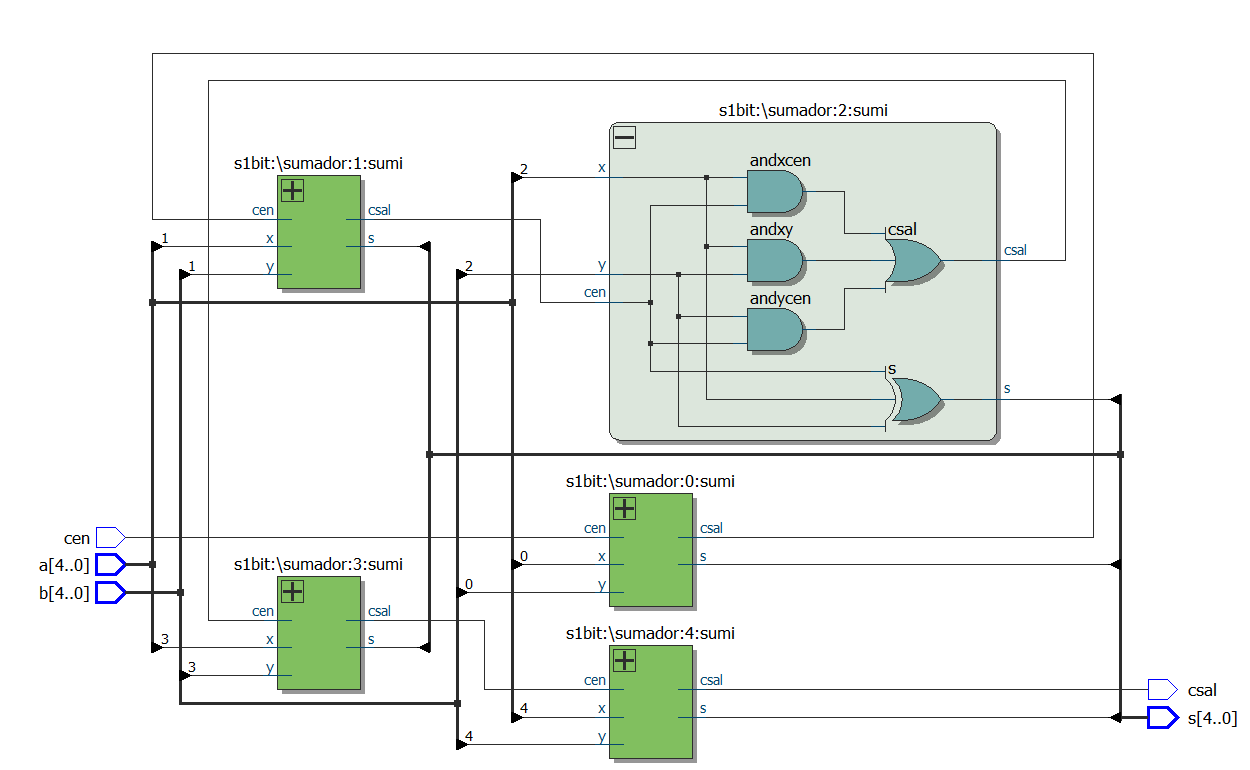
port map (x => a(i), y => b(i), cen => c(i), s => s(i), csal => c(i+1));

**end generate;**

csal <= c(n);

end **estructural**;

## 1.3 Diagrama Sumador 4bits

*Figura 1.Diagrama RTL Sumador de 4bits*

# 2. Pregunta 2

Suponga que antes del instante de tiempo t0 todas las señales del sumador son estables y que en este instante solo cambia una las 9 señales de entrada. Suponga también que la última señal de salida que se establiliza es s2 en el instante t1. Indique el retardo del circuito (caso peor), los valores de las entradas antes del instante t0 y en el instante t0. Modifique el programa de prueba (prueba\_S4bits.vhd) añadiendo un proceso que genere las señales de entrada correspondientes. Considere t0=200 ns. Compruebe el resultado con el simulador.Tenga en cuenta que los parámetros asociados a los retardos de las puertas se establecen al instanciar el componente S4bits en el programa de prueba.

## 2.1 Justificación.

El retardo es de 30 ns debido a que únicamente se actualiza el valor de uno de los sumadores de 1 bit que en este caso es del sum(2), los demás ya están estabilizados y no se modifica la señal de entrada.

| Entradas (t<t0) | | | Entradas (t=t0) | | | Retardo(t1 – t0) |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | Cen | A | B | Cen | Ret1(ns) |
| 0000 | 0000 | 0 | 0000 | 0100 | 0 | 30ns |

*Tabla 1. Retardos aplicados*

## 2.2 Código

**signal** A, B, SUM: std\_logic\_vector(3 **downto** 0);

**signal** cen, csal: **std\_logic**;

**begin**

sumador: S4bits **port** **map** (A => A, B => B, cen => cen, csal => csal, SUM => SUM)

prueba: **process**

**variable** t0: time := 200 ns;

**begin**

-- Antes del instante t0 todas las señales son estables

A <= x"0";

B <= x"0";

cen <= '0';

**wait** for t0;

-- Despues del instante t0 cambia 1 de las nueve señales

A <= x"0"; --- 0000

B <= x"4"; -- 0100

cen <= '0'; -- 0

**wait** **for** 400 ns;

A <= x"0";

B <= x"4";

cen <= '1';

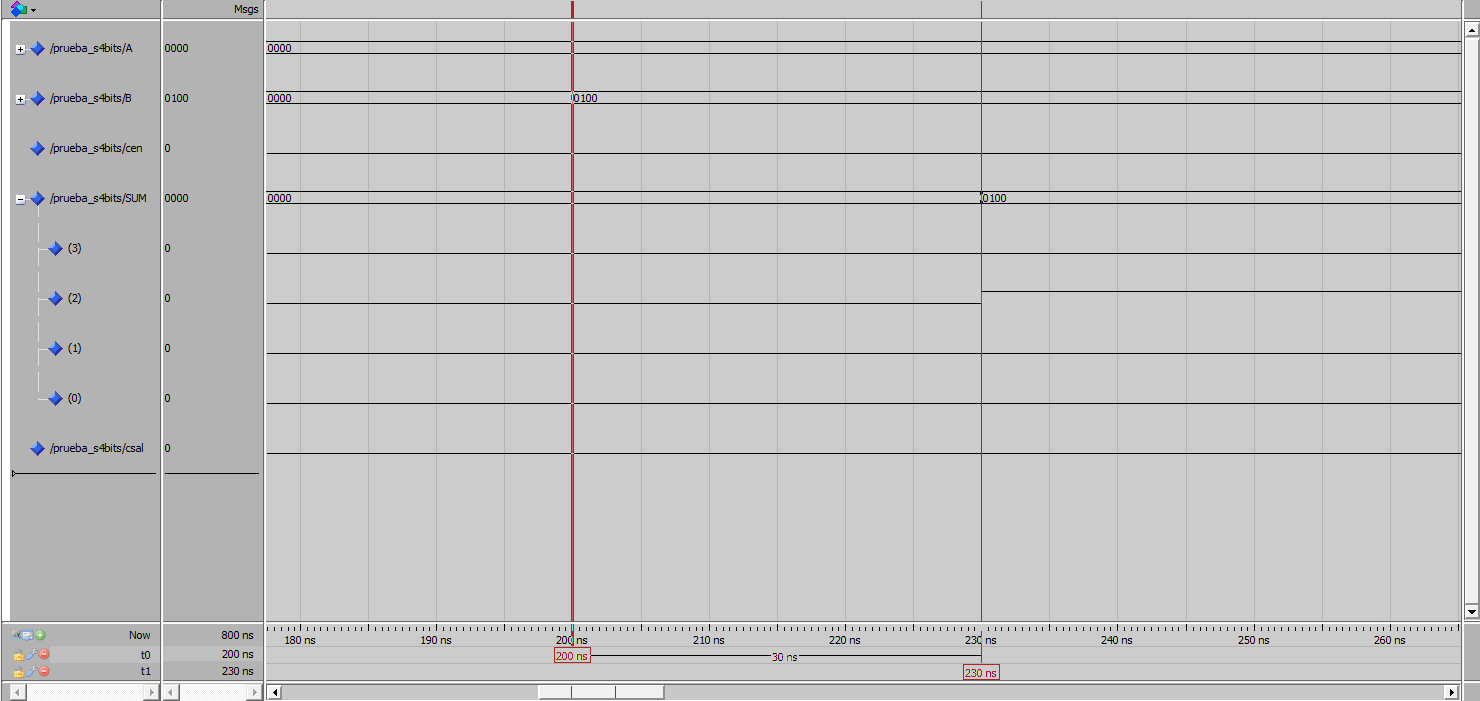
**wait** **for** 200 ns;

**wait**; -- Esto evita que el proceso termine

**end process;**

**end prueba;**

## 2.3 Ventana Temporal

*Figura 2. Ventana temporal generado con Modelsim*

# 3. Pregunta 3

Considere el sumador de 4 bits especificado en VHDL mediante sentencias generate (página 83). Tenga en cuenta que, en este diseño, los parámetros asociados a los retardos de las puertas se establecen al instanciar los componentes s1bits en el fichero snbits.vhd. Modifique el programa de pruebas (prueba\_snbits\_reloj.vhd) para que imprima el primer valor de las entradas en el que se observa el retardo máximo y el mínimo. Utilice los ficheros wave\_reloj.do y formato\_ventana.do, ubicados en el directorio PRUEBAS, para formatear las señales en la ventana temporal y eliminar algunos mensajes de advertencia (warnings) en la ventana textual.

Para esta pregunta hemos modificado el fichero prueba\_snbits\_reloj, siguiendo las instrucciones las instrucciones brindadas en la guía, además de utilizar los ficheros para la modificación de señales en la venta temporal.

## 3.1 Retardo mínimo

| vectores de bits Retardo Mínimo Justificación | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| A | 0 | 0 | 0 | 0 | 40 ns | Hemos detectado que el retardo mínimo es de 40 ns. Esto se puede explicar de una manera muy simple con el acarreo, ya que las tres posiciones más altas son 0 y no existe dependencia. Por lo tanto, el retardo mínimo será la suma de las puertas lógicas AND (10 ns), OR (15 ns) y XOR (15 ns)." |
| B | 0 | 0 | 0 | 1 |
| cen | 0 | | | |

## 3.2 Retardo máximo

| vectores de bits | | | | Retardo máximo Justificación | | |
| --- | --- | --- | --- | --- | --- | --- |
| A | 0 | 0 | 1 | 0 | 90 ns | Hemos identificado que el retardo máximo es de 90ns. Debido a que las 3 posiciones más altas tiene dependencia del valor de acarreo para tener su resultado. |
| B | 0 | 1 | 0 | 0 |
| Cen | 0 | | |  |

## 

## 

## 3.3 Código

t\_retardo\_ns := integer(t\_retardo / 1 ns); -- Convert time to nanoseconds

-- Update max and min delay values

if t\_retardo\_ns <300 and t\_retardo\_ns > retardo\_max then

retardo\_max := t\_retardo\_ns; -- Update max

A\_max := A; -- Store corresponding A value

B\_max := B; -- Store corresponding B value

cen\_max := cen; -- Store corresponding cen value

end if;

if (not min\_initialized) or (t\_retardo\_ns < retardo\_min) then

retardo\_min := t\_retardo\_ns; -- Update min

A\_min := A; -- Store corresponding A value

B\_min := B; -- Store corresponding B value

cen\_min := cen; -- Store corresponding cen value

min\_initialized := true; -- Set the flag to true

end if;