



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



Práctica 1: Suma, Resta, Multiplicación, División

Alumnos: Alain Martínez
Antoni Llop
Carrera: Ingeniería Informática
Profesor: Carles Aliagas
Fecha: Febrero/Marzo 2025

FASE 1	4
TAREA 1	4
Decisiones de diseño	4
Implementación	4
Juego de pruebas	5
Retardos y área	5
TAREA 2	6
Decisiones de diseño	6
Implementación	6
Juego de pruebas	6
Retardos y área	7
TAREA 3	8
Decisiones de diseño	8
Implementación	8
Juego de pruebas	8
Retardos y área	9
FASE 2	10
TAREA 4	10
Decisiones de diseño	10
Implementación	10
Juego de pruebas	10
Retardos y área	11
TAREA 5	12
Decisiones de diseño	12
Implementación	12
Juego de pruebas	12
Retardos y área	13
TAREA 6	14
TAREA 7	16
Decisiones de diseño	16
Implementación	16
Juego de pruebas	16
Retardo y área	17
TAREA 8	18
Decisiones de diseño	18
Implementación	18
Juego de pruebas	19
Retardo y área	19
FASE 3	20
TAREA 9	20
Decisiones de diseño	20
Implementación	20
Juego de pruebas	21

Retardo y área	21
TAREA 10	22
Decisiones de diseño	22
Implementación	22
Juego de pruebas	22
Retardo y área	23
FASE 4	24
TAREA 11	24
Decisiones de diseño	24
Implementación	24
Juego de pruebas	24
Retardo y área	25
TAREA 12	26
Decisiones de diseño	26
Implementación	26
Juego de pruebas	27
Retardo y área	1
TAREA 13	1
Decisiones de diseño	1
Implementación	1
Juego de pruebas	1
Retardo y área	1
TAREA 14	1
Retardo y área	1
TAREA 15	1
FASE 5	1
TAREA 16	1
Decisiones de diseño	1
Implementación	1
Juego de pruebas	1
Retardo y área	1
TAREA 17	1
Decisiones de diseño	1
Implementación	1
Juego de pruebas	1
Retardo y área	1
TAREA 18	1
TAREA 19	1

FASE 1

TAREA 1

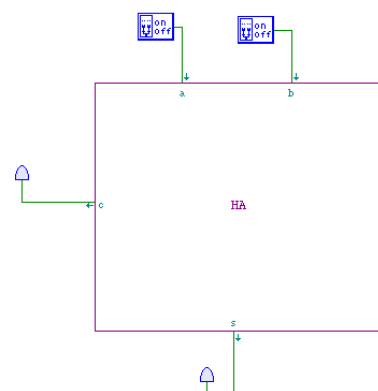
Realizad el circuito digital Half Adder (HA) de 1 bit que se muestra en la siguiente figura. Suponed que los retardos de las puertas lógicas utilizadas son $AND=3T$ y $XOR=4T$.

Decisiones de diseño

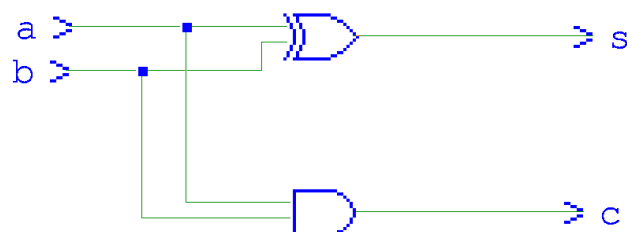
Pese a ser un circuito muy corto y poderlo realizar en el módulo main directamente, lo hemos hecho creando un módulo y así lo podíamos aprovechar para las siguientes tareas.

Implementación

Visto desde el módulo principal quedaría así.



Entrando en detalle.



Juego de pruebas

Como es un circuito muy pequeño, hemos probado que se cumpliera toda la tabla de la verdad entera.

Entradas		Salidas esperadas		Salidas obtenidas	
A	B	C	S	C	S
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	0	1	0

Retardos y área

Retardo

TS = 4T (hay 2 caminos, el de la XOR y el de la AND, al tener la XOR más retardo que la AND se convierte en el camino crítico.

TC = 3T.

Área HA = 14T (8 XOR + 6 AND). Este resultado lo usaremos más adelante, para todos los circuitos que requieran de un HA.

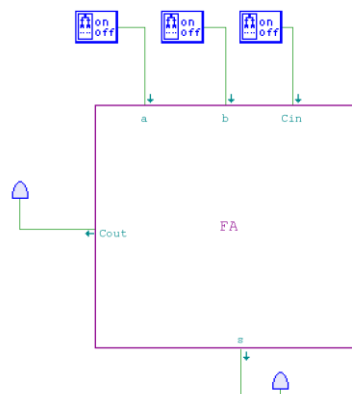
TAREA 2

Realizad el circuito digital Full Adder (FA) de 1 bit con acarreo de entrada utilizando sumadores Half Adders (HA) de 1 bit e indicad los tiempos de retardo y el área utilizada. Suponed que los retardos de las puertas lógicas utilizadas son de AND=3T, OR=3T y XOR=4T.

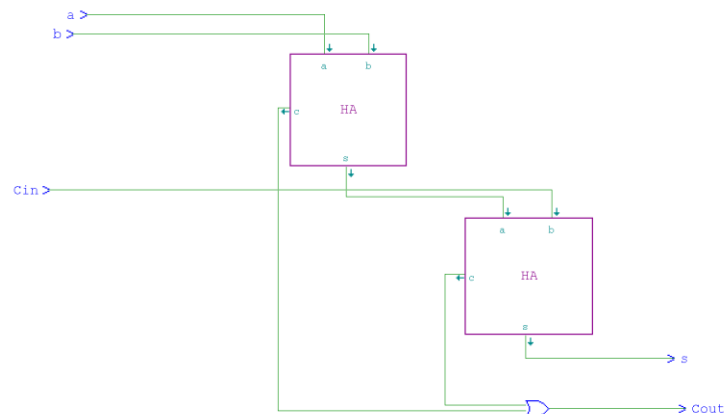
Decisiones de diseño

Hemos realizado un módulo FA y hemos reutilizado los módulos HA de la tarea anterior.

Implementación



Y dentro del FA:



Juego de pruebas

Igual que antes, al ser un circuito pequeño, hemos comprobado la tabla de la verdad entera:

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout
0	0	0	0	0	0	0

0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	1	1	1	1

Retardos y área

Retardo

TC: XOR + AND + OR = **10T**

TS: XOR + XOR = **8T**

* Área HA = 14 (tasca 1)

Área FA = $2 \cdot \text{HA} + \text{OR} = 2 \cdot 14 + 6 = \mathbf{34}$ (este resultado lo necesitaremos para todos los circuitos que tengan un FA).

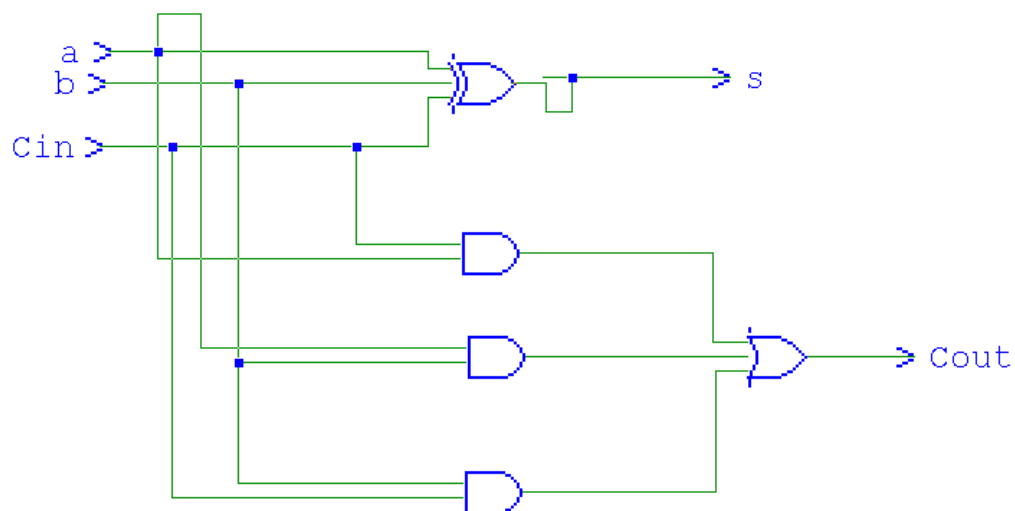
TAREA 3

Realizad una implementación alternativa al mismo circuito Full Adder (FA) de 1 bit con acarreo de entrada y comparad los tiempos de retardo y área con la solución anterior. Suponed los mismos retardos de las puertas lógicas utilizadas en la tarea anterior.

Decisiones de diseño

Hemos creado un módulo FA alternativo en el main. La implementación del main es exactamente igual que antes, un módulo con 3 entradas (A, B y Cin) y 2 salidas (Cout y S). Cambia la implementación del FA.

Implementación



Juego de pruebas

La tabla de la verdad es exactamente la misma que en el ejercicio anterior.

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0

1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	1	1	1	1

Retardos y área

Retardos

TC = AND+OR=**6T**

TS = XOR = **4T**

Área = XOR(3 entradas) + OR(3 entradas)+ 3*AND= 11+8+3*6=**37**

FASE 2

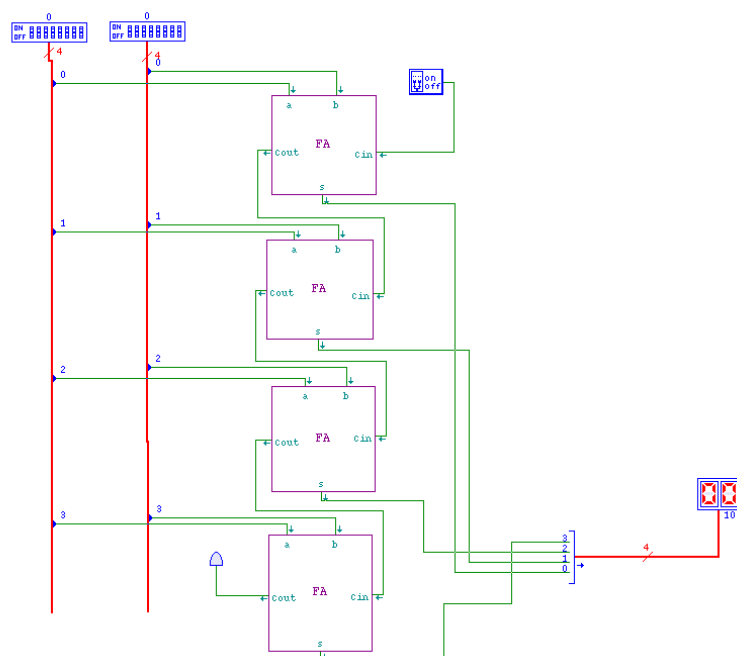
TAREA 4

Realizad el circuito digital Carry Propagate Adder (CPA) de 4 bits que se muestra en la siguiente figura e indicad y formulad los tiempos de retardo y el área utilizada. Asumid el Full Adder (FA) de 1 bit considerado en la Tarea 2.

Decisiones de diseño

Esta tarea la hicimos demasiado pronto, antes siquiera de ver la teoría y los próximos ejercicios y la implementamos del CPA ha sido sin módulo, directamente en el main. Es un error ya que en el CPA de 16 bits hemos tenido que repetir el mismo circuito. Pero gracias a este pequeño fallo intrascendente hemos aprendido la lección de que es muy importante hacer módulos, exactamente igual que cuando haces funciones cuando programas para no repetir código.

Implementación



Juego de pruebas

A y B son números hexadecimales. Cin y Cout están en binario. S es decimal.

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout

F	1	1	1	1	1	1
F	F	0	14	1	14	1
3	8	1	12	0	12	0
2	F	0	1	1	1	1
E	6	1	5	1	5	1

Retardos y área

Retardo → TC = 28, TS = 26T

TC de un FA = 10 (ejercicio 2)

ParteParalela → En nuestro diseño de FA implementado con 2 HA, la parte paralela es el primer HA, cuyo camino crítico es TS = 4. El segundo HA tiene que esperar esos 4T y que llegue el Carry, por ende esa es la parte paralela que se puede ir haciendo solo con A y B.

TC CPA 4 bits = TC(bit 0) + (n - 1 bits) * (TC - ParteParalela)

TC CPA 4 bits = 10T + 3 bits * (10T - 4T) = 10 + 3*6 = **28T**

La ParteParalelaS es el primer HA, es decir, 4T.

TS CPA 4 bits = TC(bit 0) + (n - 2 bits) * (TC - ParteParalela) + (TS - ParteParalelaS)

TS CPA 4 bits = 10T + 2 bits * (10T - 4T) + (8T - 4T) = 10 + 2*6 + 4 = **26T**

* Área FA = 34

Área = 4FA* = 4 * 34 = **136**

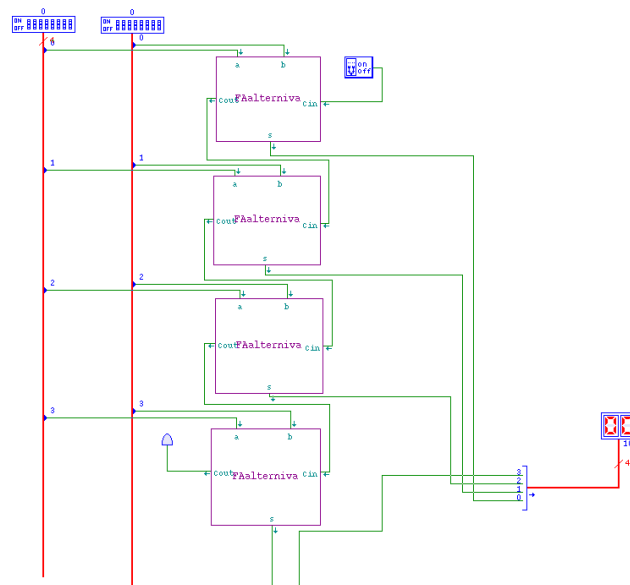
TAREA 5

Realizad el mismo circuito digital Carry Propagate Adder (CPA) de 4 bits que se muestra en la tarea anterior y formulad los tiempos de retardo y el área utilizada. Asumid el Full Adder (FA) de 1 bit considerado en la Tarea 3.

Decisiones de diseño

Hemos creado el CPA en el main directamente, sin usar módulos. En este ejercicio sí que tiene sentido hacer esto ya que el CPA que íbamos a usar en los siguientes ejercicios era el otro. Ahora bien, sería más conveniente hacerlos en módulos ya que no se sabe nunca cuándo los puedes requerir de nuevo.

Implementación



Juego de pruebas

Hemos utilizado la misma tabla de antes para demostrar que los resultados son iguales y que ambos circuitos hacen lo mismo.

A y B son números hexadecimales. Cin y Cout están en binario. S es decimal.

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout
F	1	1	1	1	1	1
F	F	0	14	1	14	1
3	8	1	12	0	12	0

2	F	0	1	1	1	1
E	6	1	5	1	5	1

Retardos y área

Retardo \rightarrow TC = 24, TS = 22

TC 1 FA = 6T (tasca 3)

TC CPA 4 bits = TC(bit 0) + (n - 1 bits) * (TC - ParteParalela)

TC CPA 4 bits = 6T + 3 bits * 6T = 6 + 18 = **24T**

La ParteParalelaS es 0.

TS 1 FA = 4T

TS CPA 4 bits = TC(bit 0) + (n - 2 bits) * (TC - ParteParalela) + (TS - ParteParalelaS)

TS CPA 4 bits = 6T + 2 bits * 6T + 4T = 6 + 2*6 + 4 = **22T**

FA alternativo = 37 (tasca 3)

Área = 4FA = 148

TAREA 6

Indicad las fórmulas que describen los tiempos de retardo del circuito digital Carry Propagate Adder (CPA) de 4 bits implementado en las tareas anteriores. Aplicando esa fórmula, mostrad los tiempos de retardo que introduciría un CPA de 8 bits, 16 bits, 32 bits, 64 bits y 128 bits para cada una de las dos posibles implementaciones de Full Adder (FA) de 1 bit consideradas en las tareas anteriores.

Fórmulas generales

$$TC \text{ CPA } n \text{ bits} = TC(\text{bit } 0) + (n - 1 \text{ bits}) * (TC - \text{ParteParalela})$$

$$TS \text{ CPA } n \text{ bits} = TC(\text{bit } 0) + (n - 2 \text{ bits}) * (TC - \text{ParteParalela}) + (TS - \text{ParteParalelaS})$$

Implementación 1 del FA (ejercicios 2 y 4):

$$TC = 10T, \text{ con ParteParalela} = 4T$$

$$TS = 8T, \text{ con ParteParalela} = 4T$$

$$TC \text{ CPA } 8 \text{ bits} = 10 + 7 * (10 - 4) = 52T$$

$$TS \text{ CPA } 8 \text{ bits} = 10 + 6 * 6 + 4 = 50T$$

$$TC \text{ CPA } 16 \text{ bits} = 10 + 15 * (10 - 4) = 100T$$

$$TS \text{ CPA } 16 \text{ bits} = 10 + 14 * 6 + 4 = 98T$$

$$TC \text{ CPA } 32 \text{ bits} = 10 + 31 * (10 - 4) = 196T$$

$$TS \text{ CPA } 32 \text{ bits} = 10 + 30 * 6 + 4 = 194T$$

$$TC \text{ CPA } 64 \text{ bits} = 10 + 63 * (10 - 4) = 388T$$

$$TS \text{ CPA } 64 \text{ bits} = 10 + 62 * 6 + 4 = 386T$$

$$TC \text{ CPA } 128 \text{ bits} = 10 + 127 * (10 - 4) = 772T$$

$$TS \text{ CPA } 128 \text{ bits} = 10 + 126 * 6 + 4 = 770T$$

Implementación 2 del FA (ejercicios 3 y 5):

$$TC = 6T, \text{ sin parte paralela}$$

$$TS = 4T, \text{ sin parte paralela}$$

$$TC \text{ CPA } 8 \text{ bits} = 6 + 7 * 6 = 48T$$

$$TS \text{ CPA } 8 \text{ bits} = 6 + 6 * 6 + 4 = 46T$$

$$TC \text{ CPA } 16 \text{ bits} = 6 + 15 * 6 = 96T$$

$$TS \text{ CPA } 16 \text{ bits} = 6 + 14 * 6 + 4 = 94T$$

$$TC \text{ CPA } 32 \text{ bits} = 6 + 31 * 6 = 192T$$

$$TS \text{ CPA } 32 \text{ bits} = 6 + 30 * 6 + 4 = 190T$$

$$\text{TC CPA 64 bits} = 6 + 63 * 6 = 384\text{T}$$

$$\text{TS CPA 64 bits} = 6 + 62 * 6 + 4 = 382\text{T}$$

$$\text{TC CPA 128 bits} = 6 + 127 * 6 = 768\text{T}$$

$$\text{TS CPA 128 bits} = 6 + 126 * 6 + 4 = 766\text{T}$$

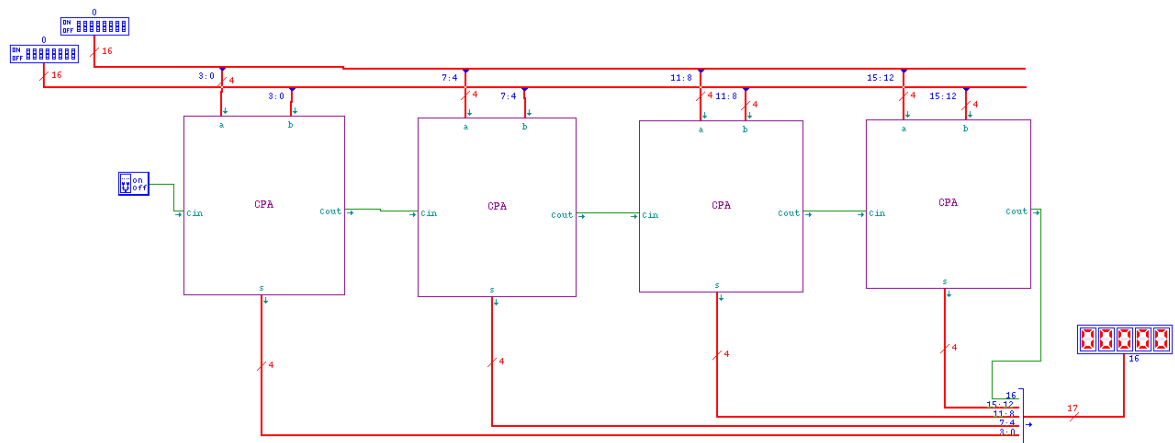
TAREA 7

Realizad un circuito digital Carry Propagate Adder (CPA) de 16 bits e indicad y formulad los tiempos de retardo y el área utilizada. Utilizad para esta implementación el circuito Carry Propagate Adder (CPA) de 4 bits implementado en la Tarea 4.

Decisiones de diseño

El CPA de 16 bits tiene una salida conectada a un display de 16 bits y además un Carry de salida. Nosotros hemos conectado el Carry a las salidas y hemos hecho que el display sea de 17 bits, en lugar de hacerlo de 16 bits y conectar un LED al Carry de salida como hemos hecho en ejercicios anteriores.

Implementación



Juego de pruebas

A, B y S en hexadecimal. Cin en binario.

Entradas			Salidas esperadas	Salidas obtenidas
A	B	Cin	S (Cout incluido)	S (Cout incluido)
F	F0	0	FF	FF
F	F0	1	100	100
2AE	EF33	0	F1E1	F1E1
FFFF	FFFF	1	1FFFF	1FFFF
3333	EEEE	1	12222	12222

Retardo y área

Los tiempos de retardo ya los hemos calculado en la tarea anterior:

$$\mathbf{TC\ CPA\ 16\ bits = 10 + 15 * (10 - 4) = 100T}$$

$$\mathbf{TS\ CPA\ 16\ bits = 10 + 14 * 6 + 4 = 98T}$$

Usamos el área del CPA que hemos calculado en la tasca 4

$$\mathbf{A_CPA = n*(A_CPA) = 4*(136) = 544}$$

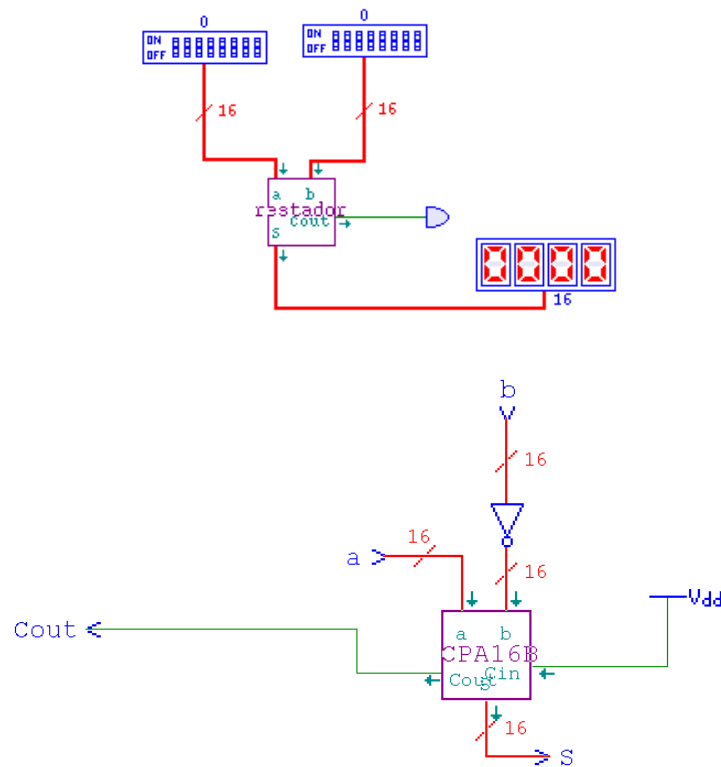
TAREA 8

Realizad un circuito digital restador de 16 bits e indicad los tiempos de retardo y el área utilizada.

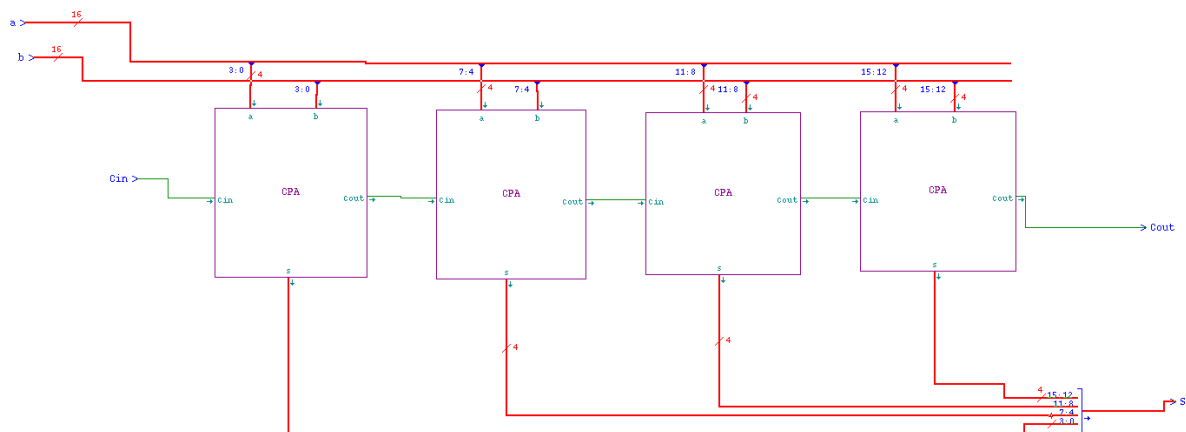
Decisiones de diseño

Hemos usado un CPA de 16 bits, exactamente igual que para la suma, pero negando la entrada B antes de llegar al CPA y sumando 1 haciendo que el Carry de entrada sea siempre 1. En este caso hemos dejado la salida de 16 bits y el Carry de salida lo hemos conectado a un LED (en la tarea anterior habíamos conectado el Carry a la salida)

Implementación



La implementación del CPA es la misma que en el ejercicio anterior:



Juego de pruebas

A, B y S en hexadecimal. Cout en binario. Cin siempre 1.

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout
FFFF	FFFF	1	0	1	0	1
FFFF	A321	1	5CDE	1	5CDE	1
0	FFFF	1	1	0	1	0
3333	2222	1	1111	1	1111	1
8765	4321	1	4444	1	4444	1

Retardo y área

Retardo → El retardo es el mismo que el de un CPA porque el circuito es exactamente el mismo. TC CPA 16 bits = 100T. En este caso habría que sumar 2T para invertir la entrada B, por lo que sería TC = 102T.

Lo mismo con TS CPA 16 bits = 98T + 2T para invertir B = 100T.

Área CPA = 136.

Área NOT = 2.

Área = 4 * Área CPA + 16 bits entrada B * Área NOT = 4 * 136 + 16 * 2 = 576

FASE 3

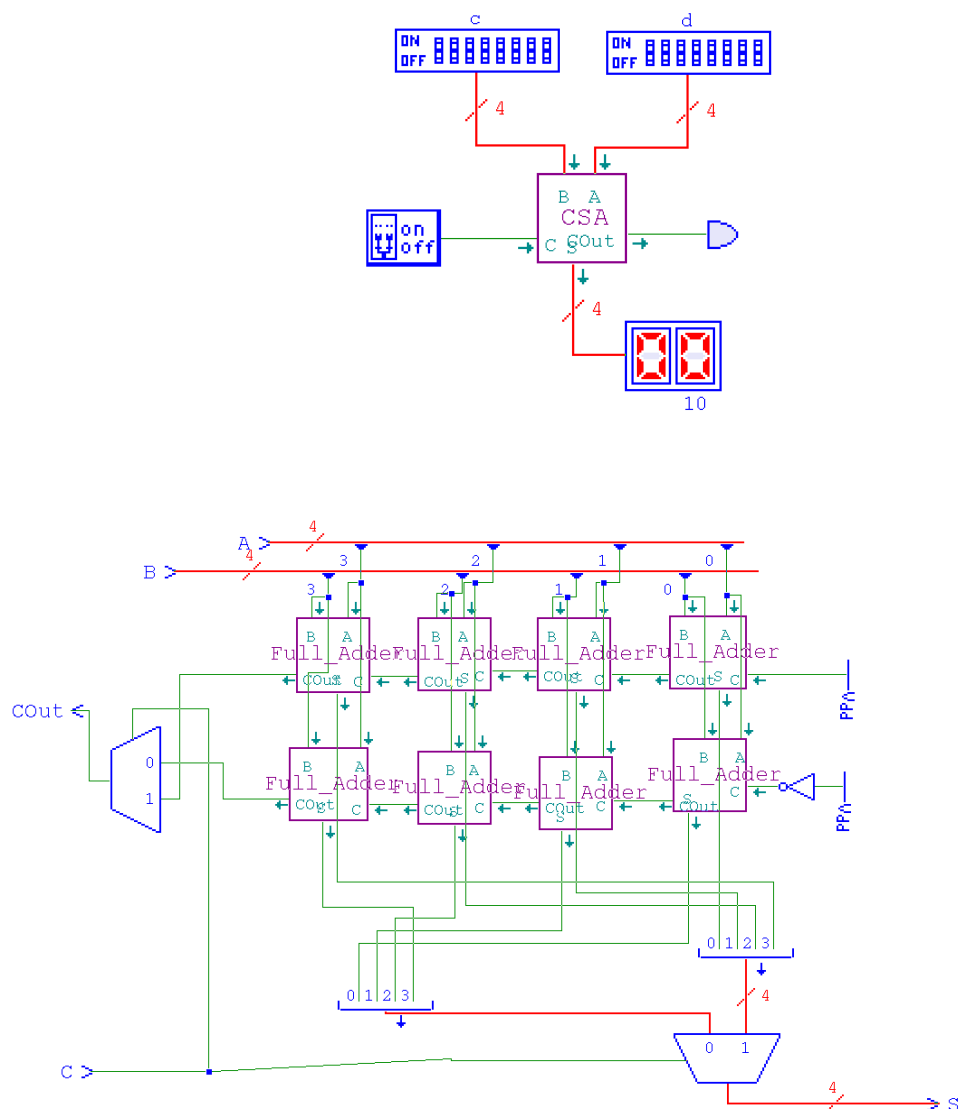
TAREA 9

Realizad el circuito digital sumador Carry Select Adder (CSA) de 4 bits que se muestra en la siguiente figura e indicad y formulad los tiempos de retardo y el área utilizada. Asumid el diseño de Carry Propagate Adder (CPA) de 4 bits implementado en la Tarea 4 y un retardo para el multiplexor de 2T.

Decisiones de diseño

Esta vez hemos utilizado un módulo CSA y dentro hemos implementado todos los FA y multiplexores.

Implementación



Juego de pruebas

A y B están en HEX. Cin y Cout están en binario. S en decimal.

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout
C	D	1	10	1	10	1
F	F	0	14	1	14	1
1	F	0	0	1	0	1
D	2	1	0	1	0	1
D	2	0	15	0	15	0

Retardo y área

Retardo \rightarrow TC = 30. TS = 28T

RetardoCPA TC = 28T (tarea 4)

RetardoCPA TS = 26T.

TC \rightarrow RetardoCSA = RetardoCPA + RetardoMUX = 28T + 2T = 30T

TS \rightarrow RetardoCSA = RetardoCPA + RetardoMUX = 26T + 2T = 28T

Área CPA = 136.

Área MUX = 8.

Área NOT = 8.

Área = 314 \rightarrow 2 CPA + 5 MUX + 1 NOT = 2 * 136 + 5 * 8 + 8 = 314.

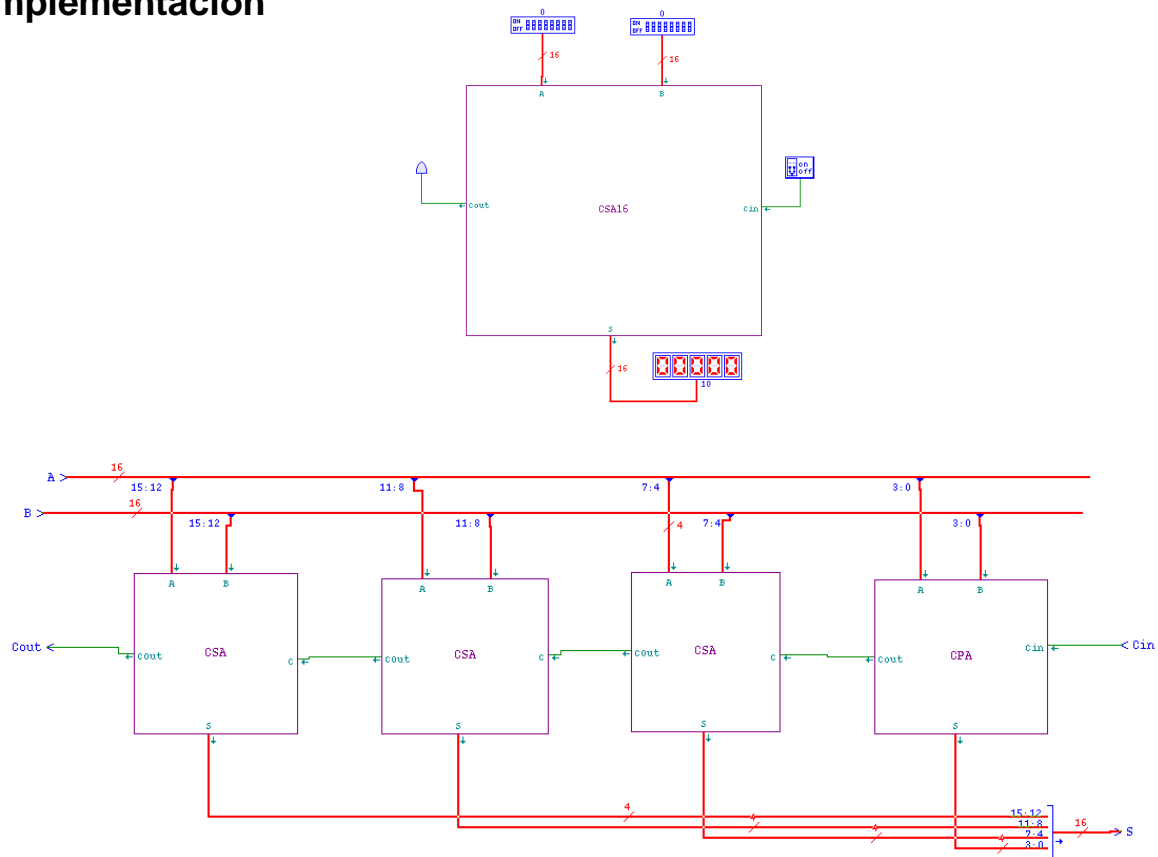
TAREA 10

Realizad un circuito digital Carry Select Adder (CSA) de 16 bits e indicad y formulad los tiempos de retardo y el área utilizada. Utilizad para esta implementación el circuito Carry Select Adder (CSA) de 4 bits implementado en una tarea anterior.

Decisiones de diseño

Hemos creado un módulo CSA 16 bits formado por 4 módulos CSA de 4 bits.

Implementación



Juego de pruebas

A y B están en HEX. Cin y Cout están en binario. S en decimal.

Entradas			Salidas esperadas		Salidas obtenidas	
A	B	Cin	S	Cout	S	Cout
FFFF	EEEE	1	61166	1	61166	1
1234	1234	1	9321	0	9321	0
FFFF	1111	0	4368	1	4368	1

2828	8282	0	43690	0	43690	0
4444	EEEE	1	13107	1	13107	1

Retardo y área

Retardo \rightarrow TC = 34. TS = 32.

RetardoCPA TC = 28T.

RetardoCPA TS = 26T.

RetardoMUX = 2T

Retardo CSA = RetardoCPA + (n módulos - 1) * Retardo MUX.

TC \rightarrow Retardo CSA = 28T + 3 * 2T = 34.

TS \rightarrow Retardo CSA = 26T + 3 * 2T = 32.

Área CSA = 314

Área CPA = 136

Área = 3 * Área CSA + Área CPA = 3 * 314 + 136 = 1078

FASE 4

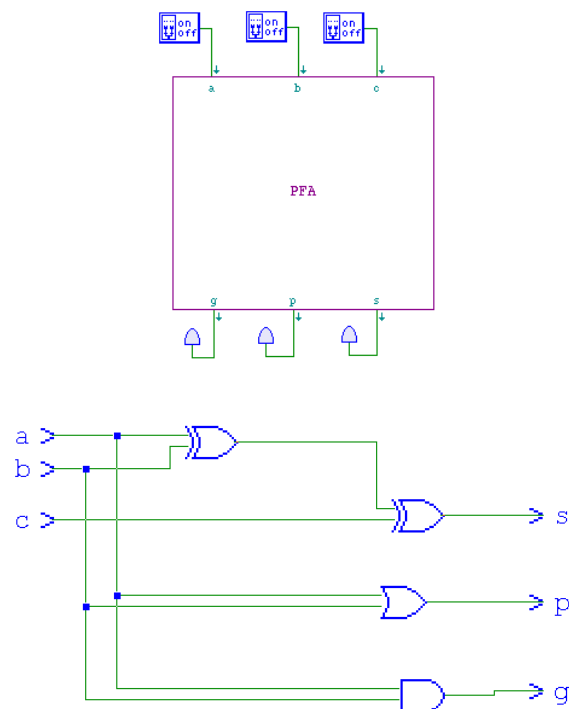
TAREA 11

Realizad el circuito digital Partial Full Adder (PFA) de 1 bit con acarreo de entrada que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Suponed que los retardos de las puertas lógicas utilizadas son de AND=3T, OR=3T y XOR=4T.

Decisiones de diseño

Hemos creado el módulo PFA en lugar de hacerlo en el main, para así poder reutilizar el módulo y no tener que repetirlo.

Implementación



Juego de pruebas

Entradas			Salidas esperadas			Salidas obtenidas		
A	B	C	S	P	G	S	P	G
0	0	0	0	0	0	0	0	0

1	1	1	1	1	1	1	1	1
0	1	1	0	1	0	0	1	0
1	0	1	0	1	0	0	1	0
1	0	0	1	1	0	1	1	0

Retardo y área

Retardo \rightarrow TS = 8T (2 XOR).

TP = 3T (OR)

TG = 3T (AND)

Área XOR = 8

Área AND = 6

Área OR = 6

Área = $2 * 8 + 6 + 6 = 28$

TAREA 12

Realizad el circuito digital Carry Look-Ahead Adder (CLA) de 4 bits que se muestra en la siguiente figura e indicad los tiempos de retardo y el área utilizada. Asumid el diseño de Partial Full Adder (FA) de 1 bit implementado en la tarea anterior. Suponed que los retardos de las puertas lógicas utilizadas son de AND=3T, OR=3T y XOR=4T.

Decisiones de diseño

Este es el circuito realizado dentro del módulo CLALogic.

Para hacer este circuito básicamente hemos desarrollado de C1 a C4 de esta manera, incluyendo también la PGeneral y la GGeneral:

$$C1 = G0 + P0 \cdot C0$$

$$C2 = G1 + P1 \cdot G0 + P1 \cdot P0 \cdot C0$$

$$C3 = G2 + P2 \cdot G1 + P2 \cdot P1 \cdot G0 + P2 \cdot P1 \cdot P0 \cdot C0$$

$$C4 = G3 + P3 \cdot G2 + P3 \cdot P2 \cdot G1 + P3 \cdot P2 \cdot P1 \cdot G0 + P3 \cdot P2 \cdot P1 \cdot P0 \cdot C0$$

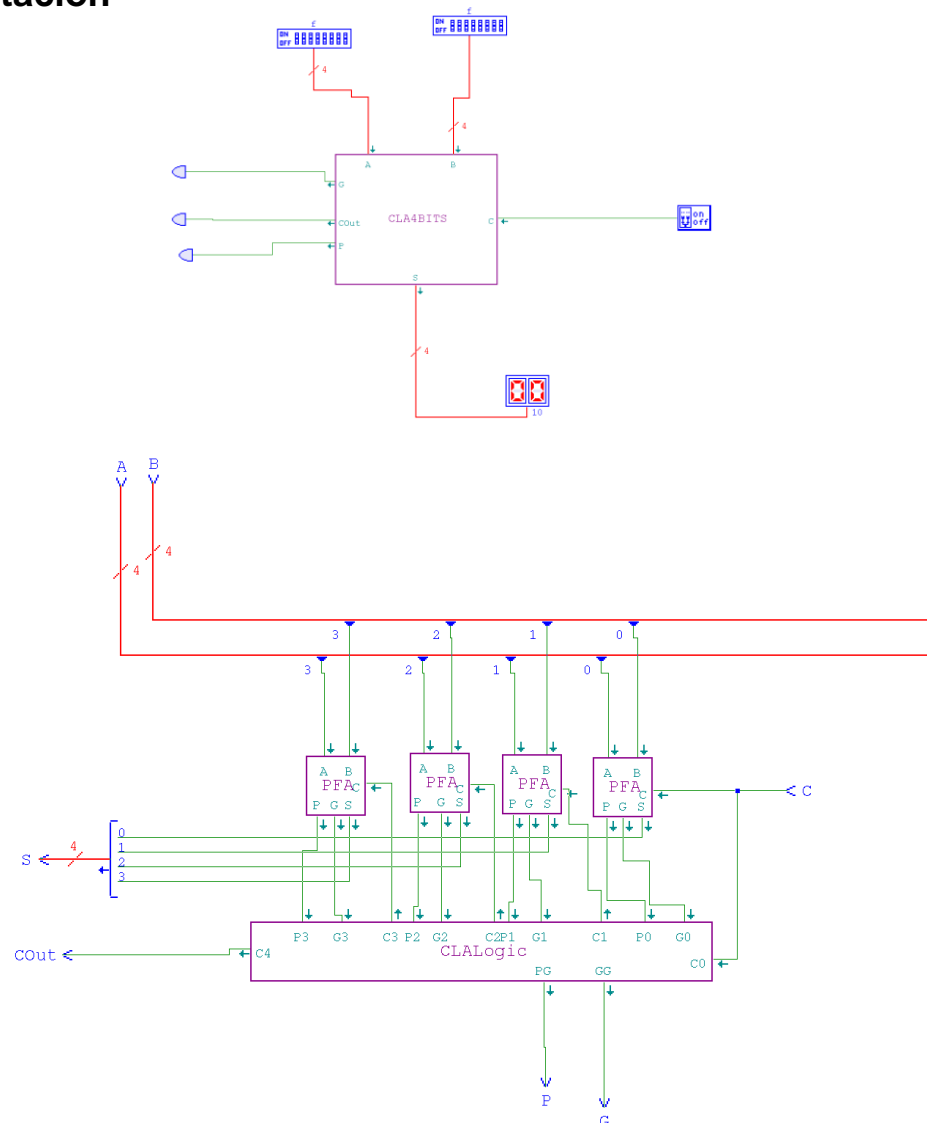
Y el cálculo de las dos generales:

$$PG = P3 \cdot P2 \cdot P1 \cdot P0$$

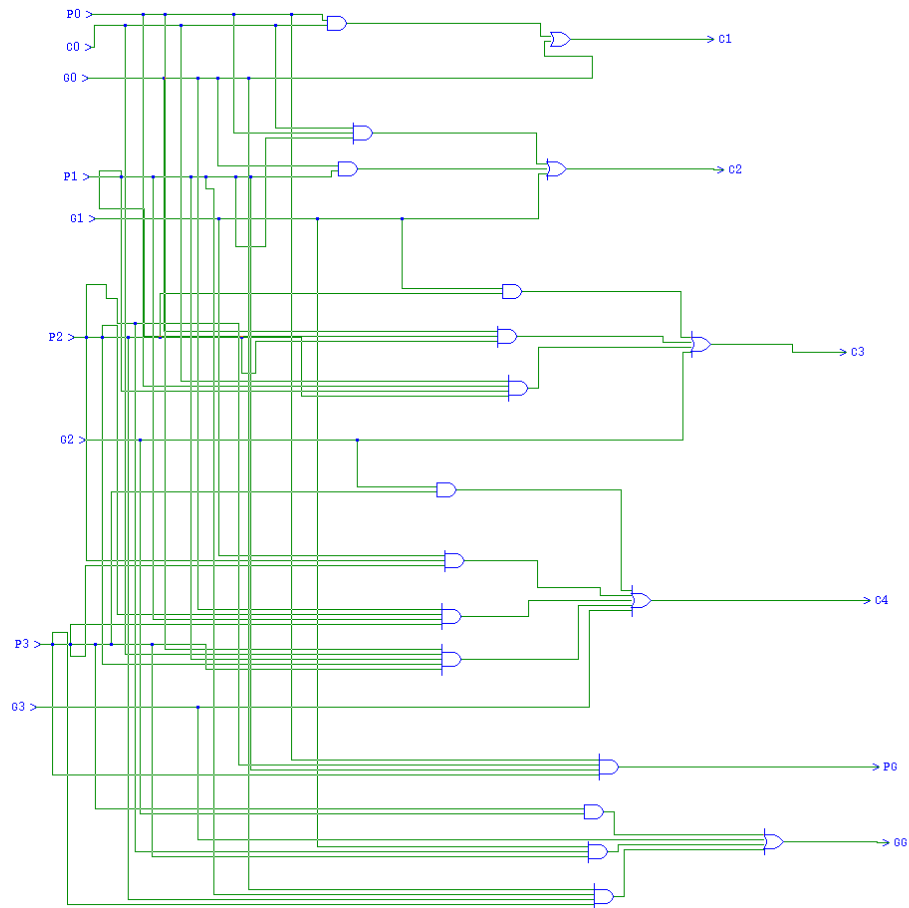
$$GG = G3 + P3 \cdot P2 \cdot P1 \cdot G0$$

Donde los + son AND y · son OR

Implementación



El CLA logic:



Juego de pruebas

A y B están en formato HEX. S en formato decimal. Los Carry, P y G en binario.

Entradas			Salidas obtenidas				Salidas esperadas			
A	B	Cin	S	P	G	Cout	S	P	G	Cout
F	0	0	15	1	0	0	15	1	0	0
F	0	1	0	1	0	1	0	1	0	1
4	4	0	8	0	0	0	8	0	0	0
5	B	1	1	1	1	1	1	1	1	1
B	B	1	7	0	1	1	7	0	1	1

Retardo y área

$$\mathbf{S} = 2 * \text{XOR} = 2 * 4T = \mathbf{8T}$$

$$\mathbf{P} = \text{OR} = \mathbf{3T}$$

$$\mathbf{G} = \text{AND} = \mathbf{3T}$$

$$\mathbf{Cin} = \text{CalcPG}(\text{Si P o G es más grande, en este caso son iguales}) + \text{AND}(\text{muchas AND juntas}) + \text{OR}(1 \text{ gran OR}) = 3T + 3T + 3T = \mathbf{9T}$$

$$\mathbf{Sfinal} = \text{La Llegada de Cin} (9T) + \text{XOR} = 9T + 4T = \mathbf{13T}$$

$$\mathbf{Cout} = \text{CalcPG} + \text{AND} + \text{OR} = \mathbf{9T}$$

$$\mathbf{PG} = \text{CalcPG} + \text{AND} = 3T + 3T = \mathbf{6T}$$

$$\mathbf{GG} = \text{CalcPG} + \text{AND} + \text{OR} = \mathbf{9T}$$

El camino crítico del CLA es la señal de **Sfinal = 13T**

$$A_PFA = 28$$

$$A_CLA = 160$$

$$\text{Área} = 4 * A_PFA + CLA = 272$$

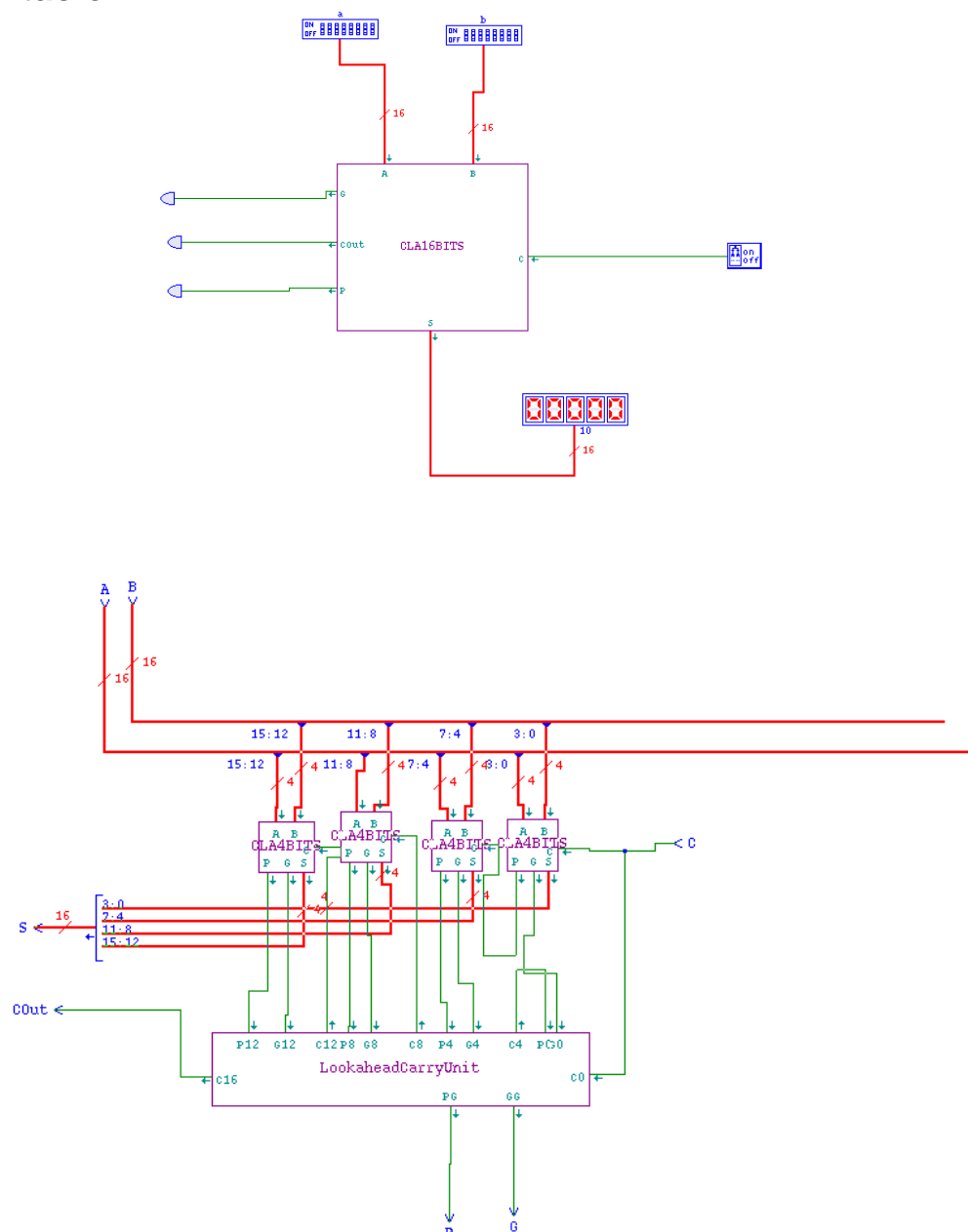
TAREA 13

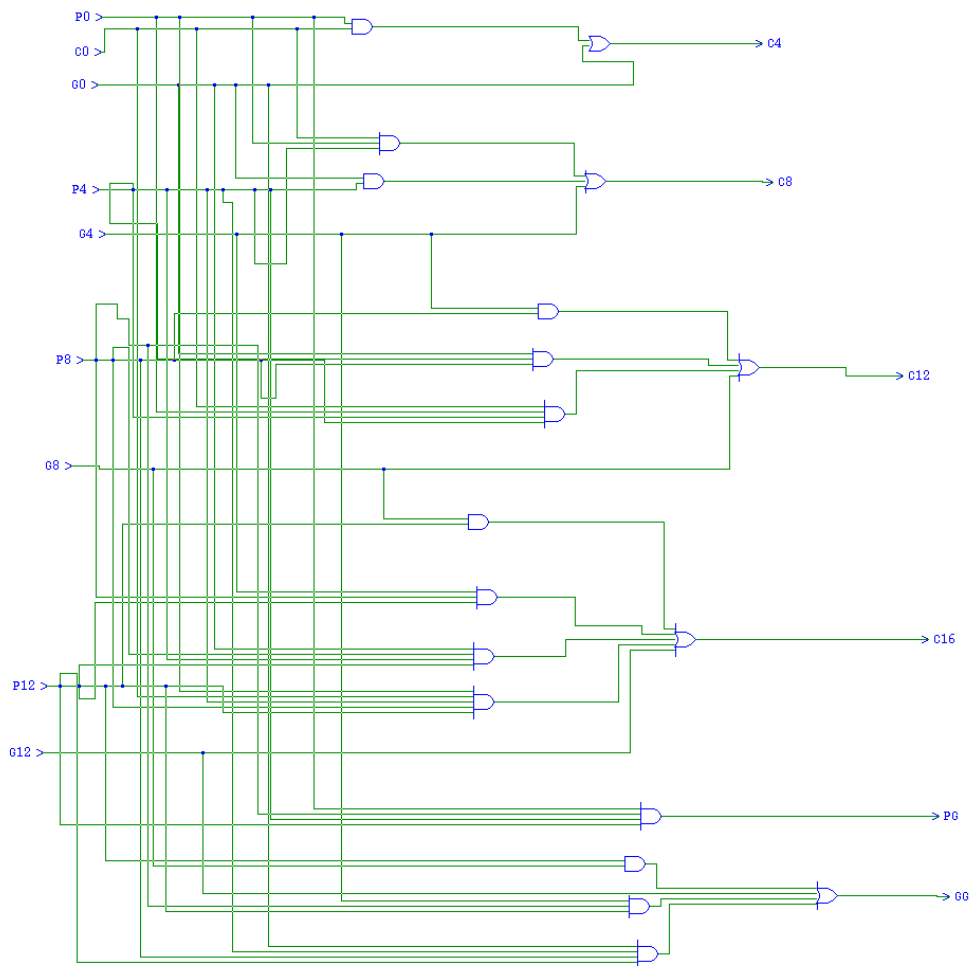
Realizad un circuito digital Carry Look-Ahead Adder (CLA) de 16 bits mediante los CLA de 4 bits implementados en la tarea anterior y conectadlos en cascada tal y como se muestra en la siguiente figura. Indicad los tiempos de retardo y el área utilizada.

Decisiones de diseño

Para hacer esta tarea básicamente hemos cogido el CLALogic y hemos cambiado la terminología para las P y G respectivamente.

Implementación





Juego de pruebas

Entradas			Salidas esperadas				Salidas obtenidas			
A	B	Cin	S	P	G	Cout	S	P	G	Cout
0	0	0	0	0	0	0	0	0	0	0
F	F	1	15	1	1	1	15	1	1	1
FFFF	FFFF	1	6553 5	1	1	1	6553 5	1	1	1

FFFF	FFFF	0	6553 4	1	1	1	6553 5	1	1	1
A	B	1	22	0	0	0	22	0	0	0

Retardo y área

LCU 16 BITS:

$$\mathbf{PG_{fin} = PG + AND = 6T + 3T = 9T}$$

$$\mathbf{GG_{fin} = GG + AND + OR = 9T + 3T + 3T = 15T}$$

$$\mathbf{C_4 = GG + OR = 9T + 3T = 12T}$$

$$\mathbf{C_8 = C_{12} = C_{16} = GG + AND + OR = 9T + 3T + 3T = 15T}$$

CLA0:

$$\mathbf{S_0 = 2 * XOR = 8T}$$

$$\mathbf{C_{1,2,3} = CalcPG + AND + OR = 3T + 3T + 3T = 9T}$$

$$\mathbf{S_{1,2,3} = C_{1,2,3} + XOR(la\ 2a) = 9T + 4T = 13T}$$

$$\mathbf{PG = 6T}$$

$$\mathbf{GG = 9T}$$

CLA1:

$$\mathbf{PG = 10T}$$

$$\mathbf{GG = 15T}$$

$$\mathbf{S_4 = C_4 + XOR = 12T + 4T = 16T}$$

$$\mathbf{C_{5,6,7} = C_4 + AND + OR = 12T + 3T + 3T = 18T}$$

$$\mathbf{S_{5,6,7} = C_{5,6,7} + XOR = 18T + 4T = 22T}$$

CLA2=CLA3:

$$\mathbf{PG = 10T}$$

$$\mathbf{GG = 15T}$$

$$\mathbf{S_8 = S_{12} = C_8 + XOR = 15T + 4T = 19T}$$

$$\mathbf{C_{9,10,11} = C_{13,14,15} = C_8 + AND + OR = 15T + 3T + 3T = 21T}$$

$$\mathbf{S_{9,10,11} = S_{13,14,15} = S_{12} = S_{16} = C_{9,10,11} + XOR = 21T + 4T = 25T}$$

El camino crítico sería el de la **S** = 25T

$$A_CLA4B = 272 \text{ (tarea anterior)}$$

$$A_LCU = 160$$

$$\text{Área} = 4 * A_CLA + A_LCU = 1248$$

TAREA 14

Calculad y justificad los tiempos de retardo de todas las señales del circuito digital Carry Look-Ahead Adder (CLA) de 16 bits implementado en la tarea anterior.

Retardo y área

LCU 16 BITS:

$$\mathbf{PG_{fin} = PG + AND = 6T + 3T = 9T}$$

$$\mathbf{GG_{fin} = GG + AND + OR = 9T + 3T + 3T = 15T}$$

$$\mathbf{C_4 = GG + OR = 9T + 3T = 12T}$$

$$\mathbf{C_8 = C_{12} = C_{16} = GG + AND + OR = 9T + 3T + 3T = 15T}$$

CLA0:

$$\mathbf{S_0 = 2 * XOR = 8T}$$

$$\mathbf{C_{1,2,3} = CalcPG + AND + OR = 3T + 3T + 3T = 9T}$$

$$\mathbf{S_{1,2,3} = C_{1,2,3} + XOR(la\ 2a) = 9T + 4T = 13T}$$

$$\mathbf{PG = 6T}$$

$$\mathbf{GG = 9T}$$

CLA1:

$$\mathbf{PG = 10T}$$

$$\mathbf{GG = 15T}$$

$$\mathbf{S_4 = C_4 + XOR = 12T + 4T = 16T}$$

$$\mathbf{C_{5,6,7} = C_4 + AND + OR = 12T + 3T + 3T = 18T}$$

$$\mathbf{S_{5,6,7} = C_{5,6,7} + XOR = 18T + 4T = 22T}$$

CLA2=CLA3:

$$\mathbf{PG = 10T}$$

$$\mathbf{GG = 15T}$$

$$\mathbf{S_8 = S_{12} = C_8 + XOR = 15T + 4T = 19T}$$

$$\mathbf{C_{9,10,11} = C_{13,14,15} = C_8 + AND + OR = 15T + 3T + 3T = 21T}$$

$$\mathbf{S_{9,10,11} = S_{13,14,15} = S_{12} = S_{16} = C_{9,10,11} + XOR = 21T + 4T = 25T}$$

El camino crítico sería el de la **S** = 25T

TAREA 15

Comparad los tiempos y áreas de los sumadores Carry Propagate Adder (CPA), Carry Select Adder (CSA) y Carry Look-Ahead Adder (CLA) de 16 bits realizados anteriormente.

Los resultados quedan de la siguiente manera:

CPA 16 BITS

TC = 100T (camino crítico)

TS = 98T

Área = 544

CSA 16 BITS

TC = 34T (camino crítico)

TS = 32T

Área = 1078

CLA 16 BITS

TC = 15T

TS = 25T (camino crítico)

A = 1248

Viendo los resultados obtenidos podemos afirmar que los CPA son más simples que los CSA y CLA en cuanto a componentes, ya que necesitan aproximadamente la mitad de recursos que los otros dos. Ahora bien, el problema está en el tiempo, que es 3 o 4 veces mayor, o sea que es mucho más lento. El CSA resuelve el problema del tiempo del CPA pero incrementando el área. El CLA mejora todavía más el tiempo del CSA incrementando ligeramente el área.

En resumen, a la hora de implementar dichos circuitos en la vida real, habría que preguntarse qué queremos. Si queremos algo simple y de bajo coste, podemos optar por CPA. Si por lo contrario queremos un alto rendimiento, sin importar nada más, solo buscando velocidad, la opción por la que decantarnos sería el CLA. En un nivel intermedio podría emplearse el CSA.

FASE 5

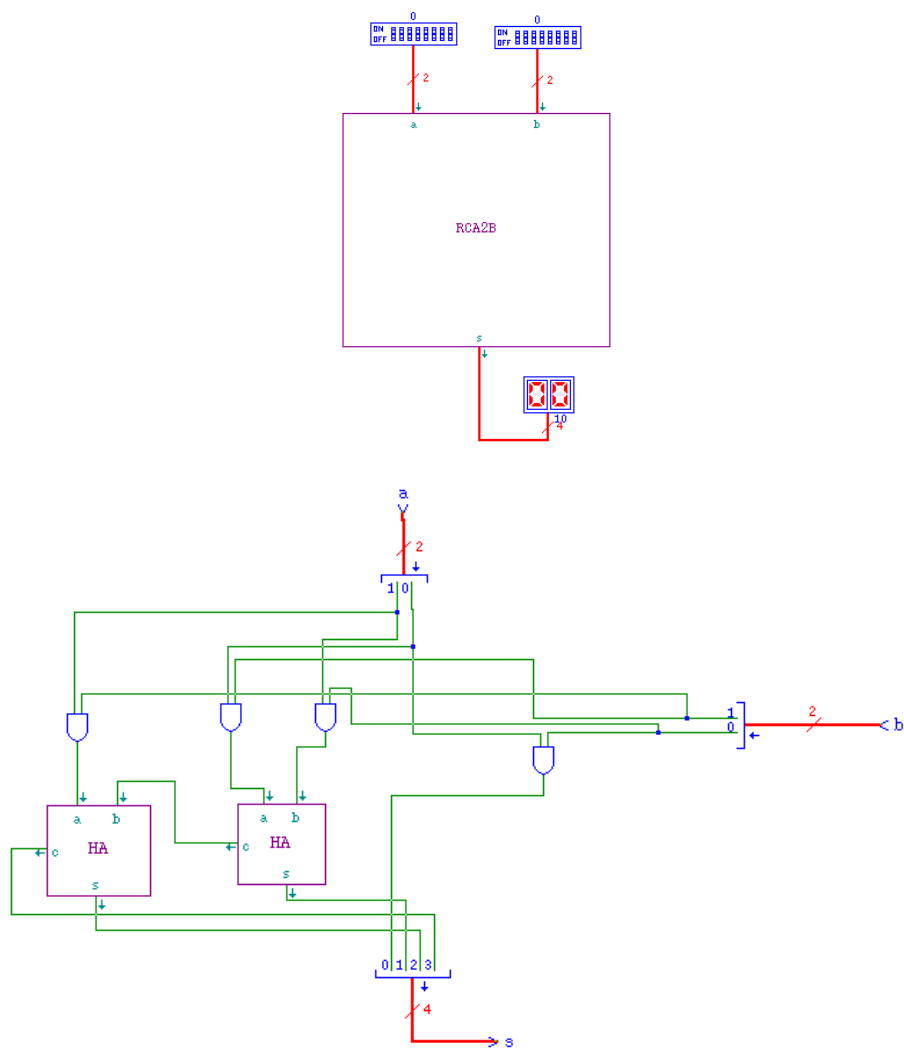
TAREA 16

Realizad un circuito digital multiplicador Riple Carry Array de 2 bits e indicad los tiempos de retardo y el área utilizada. Asumid el diseño de Half Adder (HA) implementado en una tarea anterior y un retardo para las puertas AND de 3T.

Decisiones de diseño

Hemos creado un módulo RCA de 2 bits.

Implementación



Juego de pruebas

Al tener solo 2 bits en las entradas, el número más grande que podemos poner es un 3 (1 1 en binario). El número más grande que podemos tener en la S es un 9. La salida es de 4 bits y en el display se puede mostrar hasta el número 15 en decimal, pero como las entradas son de 2 bits, el número más grande se puede conseguir multiplicando 3*3 que es 9.

Entradas		Salida esperada	Salida obtenida
A	B	S	S
3	2	6	6
0	2	0	0
2	1	2	2
2	2	4	4
1	1	1	1

Retardo y área

Retardo → $TS = 10T$

Aquí no hay Carry de salida. Lo que sería el Carry de salida del segundo HA es parte de la salida. Para calcular el tiempo sumamos el tiempo de todas las AND, el tiempo de Carry del primer HA y el tiempo de salida del segundo HA (consideramos el tiempo de salida del segundo HA y no el de Carry porque el de salida tarda más).

$$TS = T_AND + TC_FA + TS_FA = 3 + 3 + 4 = 10T$$

$$A_HA = 14$$

$$A_AND = 6$$

$$\text{Área} = 2 \cdot A_HA + 4 \cdot A_AND = 28 + 24 = 52$$

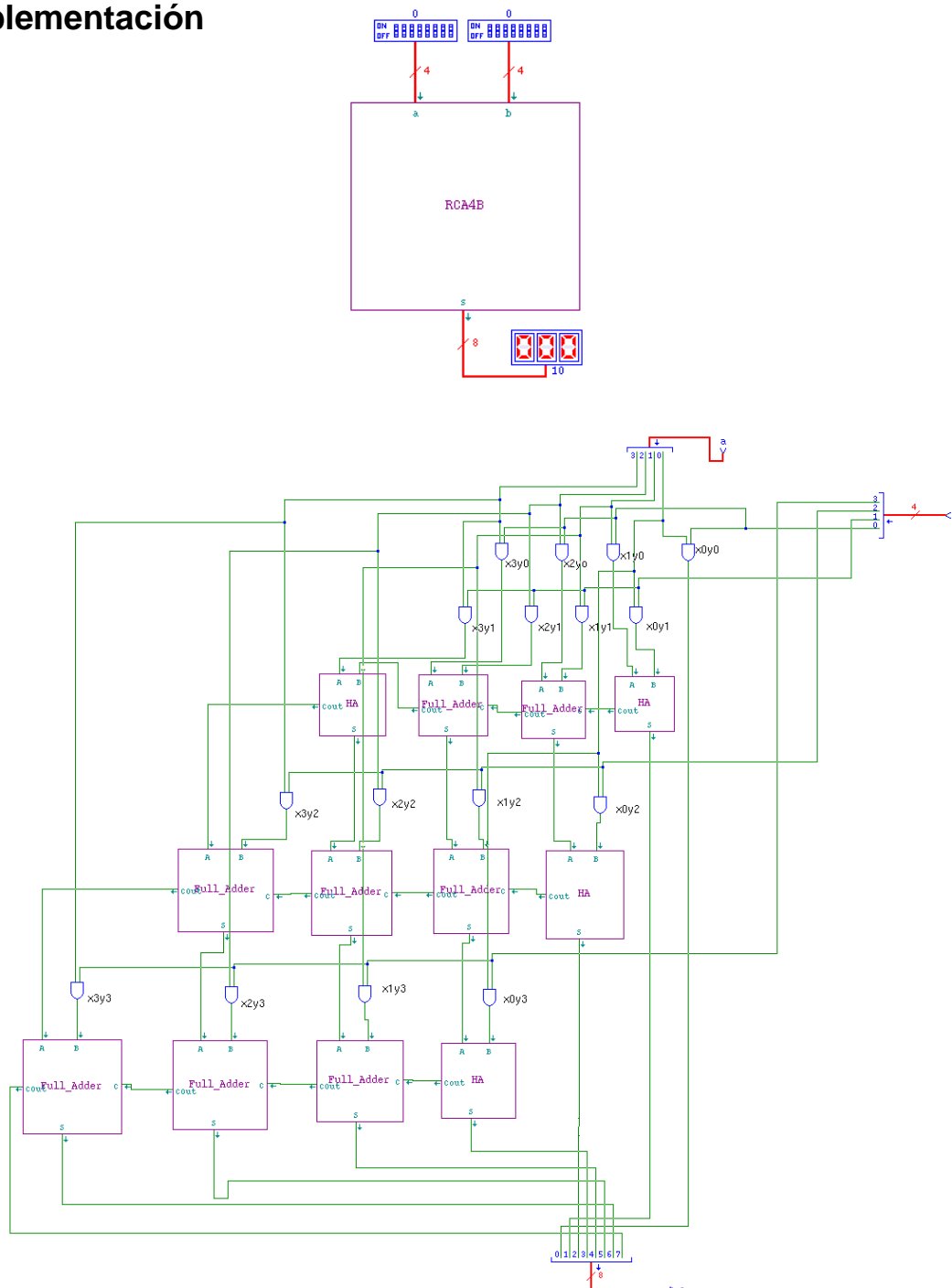
TAREA 17

Realizad el circuito digital multiplicador Ripple Carry Array de 4 bits que se muestra en la siguiente figura e indicad el tiempo de retardo y el área utilizada. Asumid los diseños de Half Adder (HA) y Full Adder (FA) de 1 bit implementados en las tareas anteriores. Suponed también que los retardos de las puertas lógicas utilizadas son de AND=3T, OR=3T y XOR=4T.

Decisiones de diseño

Pese a que no hacía falta crear un módulo para esta tarea, hemos creado uno y dentro hemos implementado todo el circuito. Nos gusta más ver una salida limpia y con pocos dibujos a la hora de simular para que se vea más claro y no se mezcle con toda la implementación.

Implementación



Juego de pruebas

Al tener 4 bits, el número más grande que podemos poner en las entradas es F (15). Por ende, el número más grande que puede aparecer en el display es 225 (15*15 o F*F).

En este caso las entradas están en HEX y las salidas en DEC.

Entradas		Salida esperada	Salida obtenida
A	B	S	S
F	F	225	225
A	A	100	100
2	8	16	16
C	3	36	36
2	7	14	14

Retardo y área

Camí crític = 53T. Los retardos están calculados en la tarea siguiente. Pese a que hemos puesto 53T como camino crítico, en la tarea siguiente hemos puesto 52, ya que nuestros cálculos a mano nos han dado otro resultado.

$$A_{FA} = 34$$

$$A_{HA} = 14$$

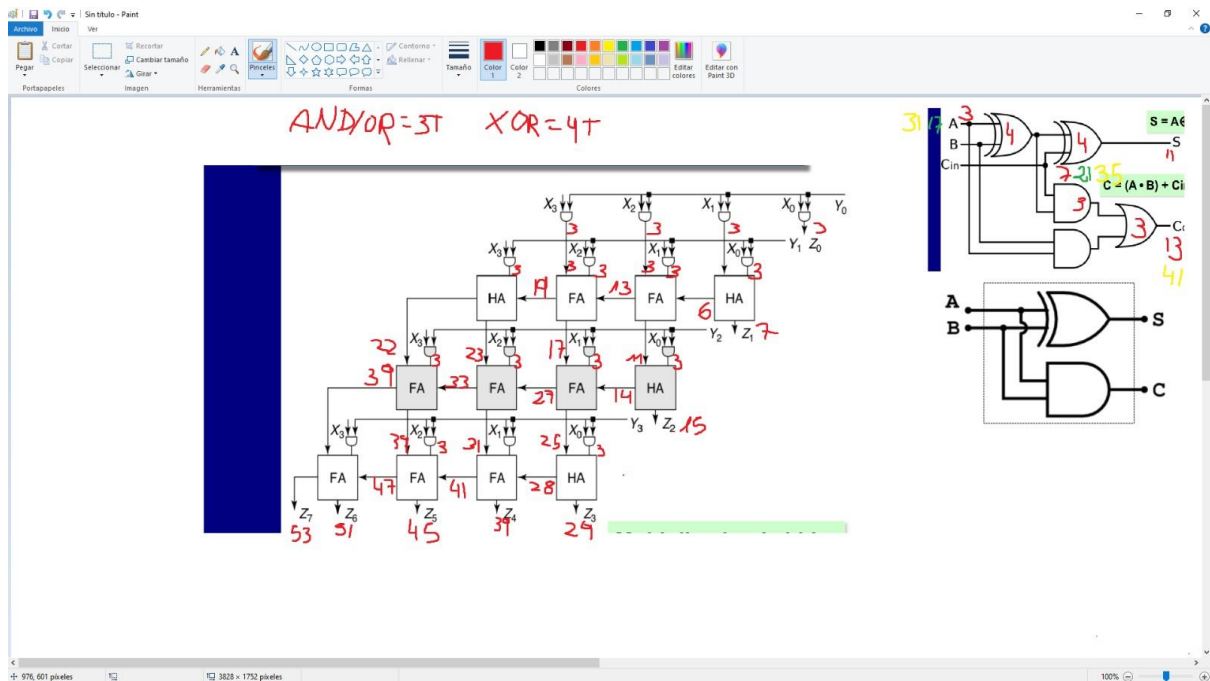
$$A_{AND} = 6$$

$$\text{Área} = 8 \cdot A_{FA} + 4 \cdot A_{HA} + 16 \cdot A_{AND} = 272 + 56 + 96 = 424$$

TAREA 18

Calculad los tiempos de retardo de todas las señales de todos los Half Adder (HA) y Full Adder (FA) de 1 bit del circuito digital multiplicador Riple Carry Array de 4 bits implementado en la tarea anterior.

Hemos adjuntado la imagen con los tiempos al lado de cada salida. De esta forma, se ve mucho más claro.



TAREA 19

Investiga qué tipo de sumadores/multiplicadores se emplean en procesadores comerciales actuales (Intel, AMD, ARM, Apple, NVIDIA, Qualcomm, etc). Para responder, debes consultar y citar fuentes fiables como documentación técnica de los fabricantes, artículos de revistas especializadas en arquitectura de procesadores, conferencias de arquitectura de computadores o análisis detallados de expertos en hardware.

Intel: Desde la microarquitectura Haswell (2013), Intel ha incorporado instrucciones FMA3 en sus procesadores, permitiendo operaciones FMA eficientes. en.wikipedia.org

AMD: Las arquitecturas de AMD, comenzando con Piledriver (2012), han soportado tanto FMA3 como FMA4, proporcionando flexibilidad en operaciones de punto flotante. en.wikipedia.org

ARM: Los procesadores ARM con VFPv4 y/o NEONv2, como el Cortex-A7 (2013), soportan operaciones FMA, mejorando el rendimiento en aplicaciones multimedia y científicas. en.wikipedia.org

Apple: Los procesadores de Apple, comenzando con el A6 (2012), han incorporado soporte para operaciones FMA, optimizando tareas de cálculo intensivo en sus dispositivos. en.wikipedia.org

NVIDIA: Las GPUs de NVIDIA, desde la arquitectura Fermi (2010), han implementado operaciones FMA, cruciales para cálculos gráficos y computación paralela. en.wikipedia.org

Qualcomm: Los procesadores Snapdragon, como el Krait (2012), integran soporte para FMA, mejorando el rendimiento en aplicaciones multimedia y de señal. en.wikipedia.org