



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



ESTRUCTURA DE COMPUTADORS

Curs 2024-25

Estudiants: Oupman Miralles, Raquel Cornadó

Professor/a: Carles Aliagas

ÍNDEx

ÍNDEx	2
Fase 1: Sumadores HA, FA	1
Tarea 1.....	1
Tarea 2.....	2
Tarea 3.....	4
Fase 2: Sumadores CPA	7
Tarea 4.....	7
Tarea 5.....	8
Tarea 6.....	10
Tarea 7.....	11
Tarea 8.....	13
Fase 3: Sumadores CSA	15
Tarea 9.....	15
Tarea 10.....	16
Fase 4: Sumadores CLA	18
Tarea 11.....	18
Tarea 12.....	20
Tarea 13.....	22
Tarea 14.....	24

Tarea 15.....	26
Fase 5: Multiplicador RCA.....	27
Tarea 16.....	27
Tarea 17.....	29
Tarea 18.....	32
Tarea 19.....	34

Fase 1: Sumadores HA, FA

Tarea 1

Especificación: 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 $XOR=4T$.

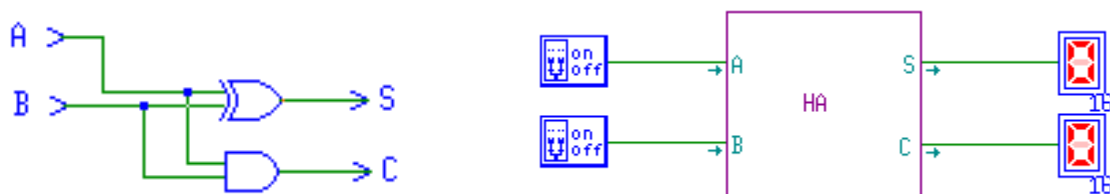
Diseño: En aquesta tasca implementem el circuit Half Adder de la manera en la que es mostra en l'enunciat.

- 2 Inputs (A & B) de 1 bit.
 - Connectats a un switch de 1 bit.
- 2 output (Sum & C_{out}) de 1 bit.
 - Connectats a un led de 1 bit en HEXA.

Les sortides del Half Adder es poden calcular amb la seva formula teòrica:

- $S = \bar{A} * B + A * \bar{B} = A \oplus B$
- $C = A * B$

Implementación:

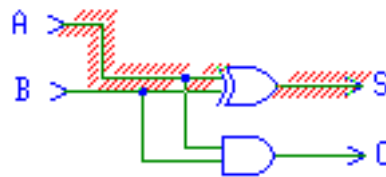


Juego de pruebas: per comprovar el funcinament del Half Adder fem us dels switchs i els leds i comprovem les diferents sortides possibles del resultat i el Carry.

INPUTS		OUTPUS	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Análisis de Resultados:

- Temps de propagació de S:
 - $T_S = T_{XOR} = 4T$
- Temps de propagació de C:
 - $T_C = T_{AND} = 3T$
- Camí crític:
 - $T_{RETARD} = \max(T_S, T_C) = T_S = 4T$

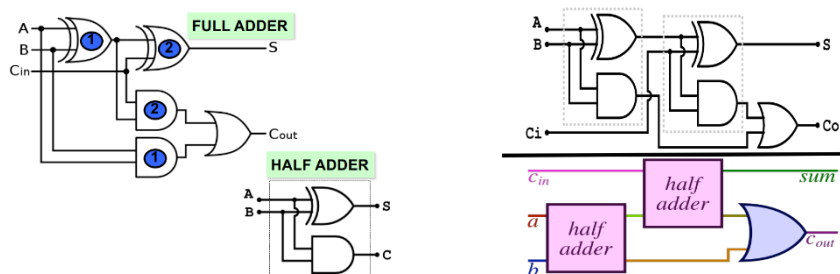


- Càlcul de l'àrea:
 - $A_{AND} = 6u$
 - $A_{XOR} = 8u$
 - $A_{TOTAL} = A_{AND} + A_{XOR} = 6 + 8 = 14u$

Tarea 2

Especificación: 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.

Diseño: En aquesta tasca implementem el circuit Full Adder amb els Half Adders de la tasca anterior. La idea es que un Half Adder conté una porta XOR i una AND, pel que podem transformar el circuit de l'exemple utilitzant 2 HA i afegint una OR.

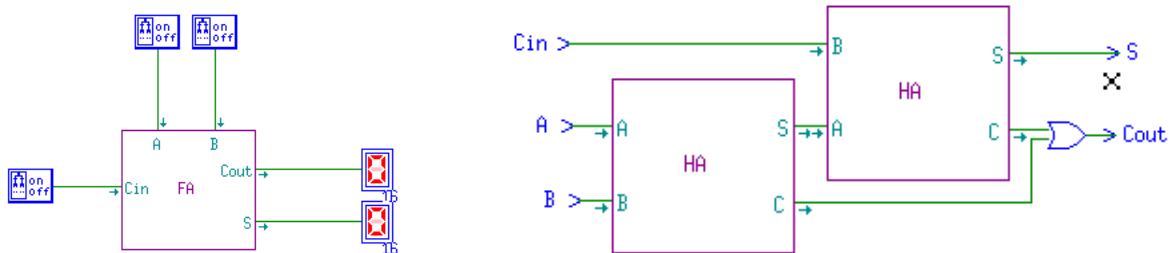


- 3 Inputs (A, B, C_{in}) de 1 bit.
 - Conectats a un switch de 1 bit.
- 2 outputs (Sum & C_{out}) de 1 bit.
 - Conectats a un led de 1 bit en HEXA.

El càlcul de resultats utilitzant 2 Half Adders seria:

- $S = A \oplus B \oplus C_{in}$
- $C = (A * B) + C_{in} * (A \oplus B)$

Implementación:



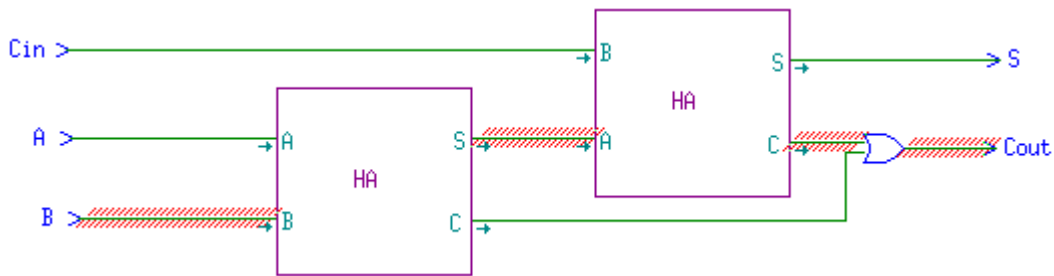
Juego de pruebas: per comprovar el funcionament del Full Adder fem ús dels switches i els leds i comprovem les diferents sortides possibles de Sum i Carry.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1
1	1	1	1	1

Análisis de Resultados:

- Temps de propagació de S:
 - $T_S = 2T_{XOR} = 2 * 4 = 8T$
- Temps de propagació de C:
 - $T_C = T_{XOR} + T_{AND} + T_{OR} = 4 + 2 * 3 = 10T$

- Camí crític:
 - $T_{\text{RETARD}} = \max(T_S, T_C) = T_C = 10T$



- Càlcul de l'àrea:
 - $A_{\text{AND}} = 6u$
 - $A_{\text{XOR}} = 8u$
 - $A_{\text{OR}} = 6u$
 - $A_{\text{TOTAL}} = 2A_{\text{AND}} + 2A_{\text{XOR}} + A_{\text{OR}} = 3 * 6 + 2 * 8 = 34u$

Tarea 3

Especificación: 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.

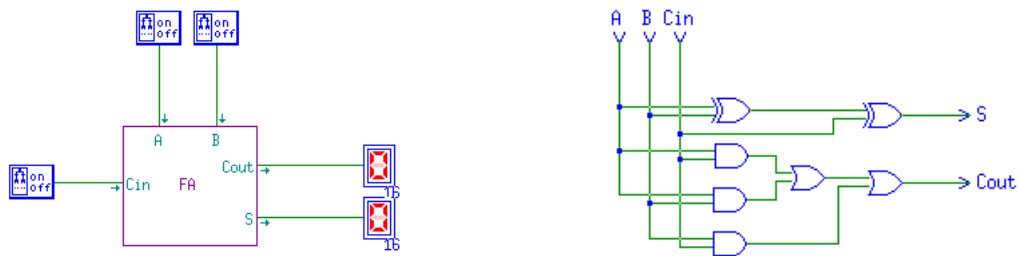
Diseño: En aquesta tasca hem realitzat una implementació alternativa a la de la tasca 2. En aquest cas utilitzarem més portes lògiques tal i com hem vist a teoria.

- 3 Inputs (A, B, C_{in}) de 1 bit.
 - Conectats a un switch de 1 bit.
- 2 outpus (Sum & C_{out}) de 1 bit.
 - Conectats a un led de 1 bit en HEXA.

Les sortides del Full Adder alternatiu es poden calcular amb la seva formula teòrica:

- $S = A \oplus B \oplus C$
- $C = (A * B) + (A * C_{in}) + (B * C_{in})$

Implementación:

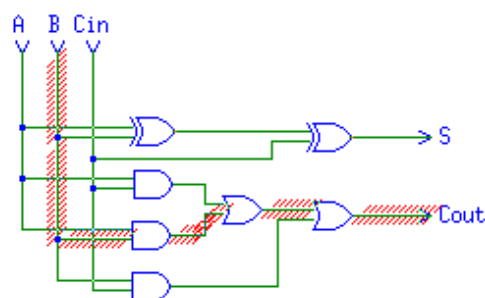


Juego de pruebas: per comprovar el funcionament del Full Adder fem us dels switches i els leds i comprovem les diferents sortides possibles de Sum i Carry.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1
1	1	1	1	1

Análisis de Resultados:

- Temps de propagació de S:
 - $T_S = 2T_{XOR} = 2 * 4 = 8T$
- Temps de propagació de C:
 - $T_C = T_{AND} + 2T_{OR} = 3 * 3 = 9T$
- Camí crític:
 - $T_{RETARD} = \max(T_S, T_C) = T_C = 9T$



- Càlcul de l'àrea:
 - $A_{AND} = 6u$
 - $A_{XOR} = 8u$
 - $A_{OR} = 6u$
 - $A_{TOTAL} = 3A_{AND} + 2A_{XOR} + 2A_{OR} = 5 * 6 + 2 * 8 = 46u$

Diferències entre les dues implementacions:

	FULL ADDER 1	FULL ADDER 2
RETARD Sum	8 T	8 T
RETARD Cout	10 T	9 T
AREA	34	46

Observem que la sortida de la Suma tarda el mateix en les dues implementacions, però en canvi, la sortida del Carry tarda menys en el FA2. Això és deu a utilitzar més portes lògiques per poder fer operacions de forma paral·lela amb les AND i reduir el temps de càlcul que no pot fer el FA1 amb les portes en sèrie. Però això bé amb la desavantatge d'haver d'utilitzar més àrea, la qual incrementa significativament en un 26%.

La millor implementació doncs és FA1, ja que aquesta escala molt millor a major numero de bits. En el cas d'un CPA, el camí crític es la propagació del carry:

- $RETARD_C = T_C + (n - 1) * (T_C - P_P)$
- $RETARD_{C1} = 10T + (n - 1) * (10T - 4T) = 10T + (n - 1) * 6T$
- $RETARD_{C2} = 9T + (n - 1) * (9T - 0T) = 9T + (n - 1) * 9T$

Tot i que FA2 calcula 3 AND en paral·lel, 2 d'elles depenen de Cin, pel que igualment hauran de esperar el temps de 1 AND i 2 OR seqüencials. A més a més, l'ús de l'àrea també seria molt superior. En conclusió, el FA2 té un 33% més de velocitat de propagació del carry en cascada, i un 26% menys de hardware utilitzat.

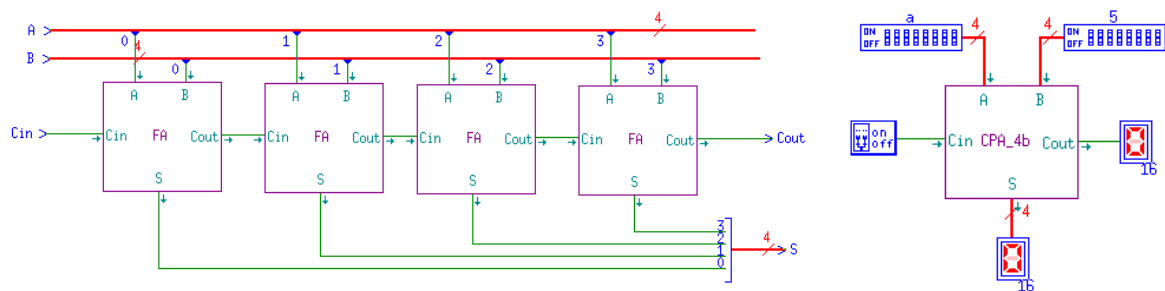
Fase 2: Sumadores CPA

Tarea 4

Especificación: 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.

Diseño: Per implementar un CPA hem connectat 4 FA en cascada, de manera que el primer rep el carry extern i després la resta rep el Cin de l'anterior full adder, propagant el carry fins al final. Al final es concatena el resultat de cada suma bit a bit per donar el resultat S i un carry de sortida Cout.

Implementación:



Juego de pruebas: hem comprovat diferents casos amb i sense carry in per comprovar l'increment en 1 sobre el resultat. Podem veure també el overflow quan superem els 4 bits.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
3	5	0	8	0
3	5	1	9	0
F	0	0	F	0
F	0	1	0	1
F	F	0	E	1
F	F	1	F	1

Análisis de Resultados:

- Temps de propagació de S:

$$\text{RETARD}_S = T_C + (n - 2) * (T_C - P_P) + (T_S - P_P) = 10 + (4 - 2) * (10 - 4) + (8 - 4) = 26T$$

- Temps de propagació de C:

$$\text{RETARD}_C = T_C + (n - 1) * (T_C - P_P) = 10 + (4 - 1) * (10 - 4) = 28T$$

- Camí crític:

$$\circ \quad T_{\text{RETARD}} = \max(T_S, T_C) = T_C = 28T$$

- Càlcul de l'àrea:

$$\circ \quad A_{\text{FA}} = 34u$$

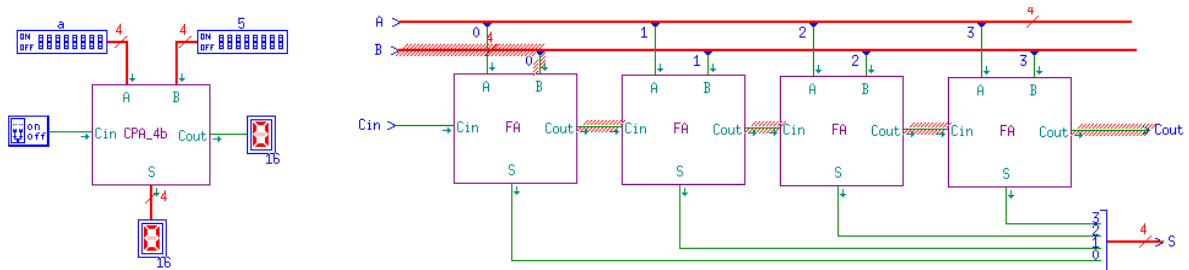
$$\circ \quad A_{\text{TOTAL}} = 4A_{\text{FA}} = 4 * 34 = 136u$$

Tarea 5

Especificación: 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.

Diseño: tenim el mateix disseny que la tasca anterior però utilitzant el FA de la tasca 3.

Implementación:



Juego de pruebas: realitzem el mateix joc de proves que el CPA anterior.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
3	5	0	8	0
3	5	1	9	0
F	0	0	F	0
F	0	1	0	1
F	F	0	E	1
F	F	1	F	1

Análisis de Resultados:

- Temps de propagació de S:

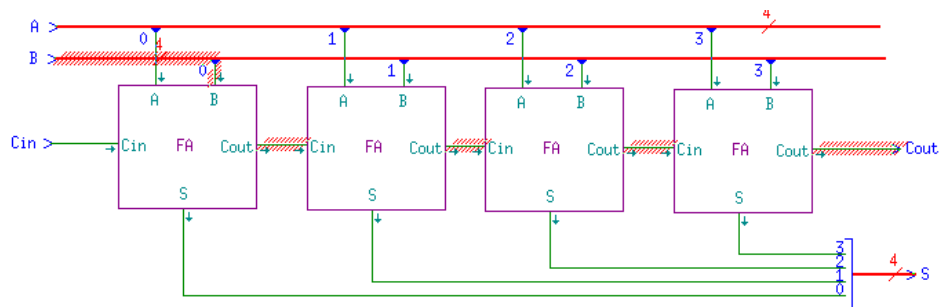
$$\text{RETARD}_S = T_C + (n - 2) * (T_C - P_P) + (T_S - P_P) = 9 + (4 - 2) * (9 - 0) + (8 - 0) = 35T$$

- Temps de propagació de C:

$$\text{RETARD}_C = T_C + (n - 1) * (T_C - P_P) = 9 + (4 - 1) * (9 - 0) = 36T$$

- Camí crític:

$$\text{T}_{\text{RETARD}} = \max(T_S, T_C) = T_C = 36T$$



- Càlcul de l'àrea:

- $A_{FA} = 46u$
- $A_{TOTAL} = 4A_{FA} = 4 * 46 = 184u$

Com bé havíem comentat en la Tasca 3, es veu clarament com la implementació del FA1 és molt millor tant en velocitat com consum que el FA2.

Tarea 6

Especificación: 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

Análisis de Resultados: les formules utilitzades en els temps de retard del CPA han sigut les següents:

- Temps de propagació de S:
 - $RETARD_S = T_C + (n - 2) * (T_C - P_P) + (T_S - P_P)$
- Temps de propagació de C:
 - $RETARD_C = T_C + (n - 1) * (T_C - P_P)$

En les quals tenim:

- $T_C \rightarrow$ retard del Carry
- $T_S \rightarrow$ retard del Sum
- $n \rightarrow$ numero de bits del circuit
- $P_P \rightarrow$ fracció de temps que es realitza en paral·lel (com per exemple $A \wedge B$)

Utilitzant aquestes formules calcularem els retards de un CPA de 8, 16, 32, 64 i 128 bits amb les diferents implementacions dels FA tenint en compte els diferents temps que hem tret anteriorment. Simplifiquem la fórmula de RS i RC:

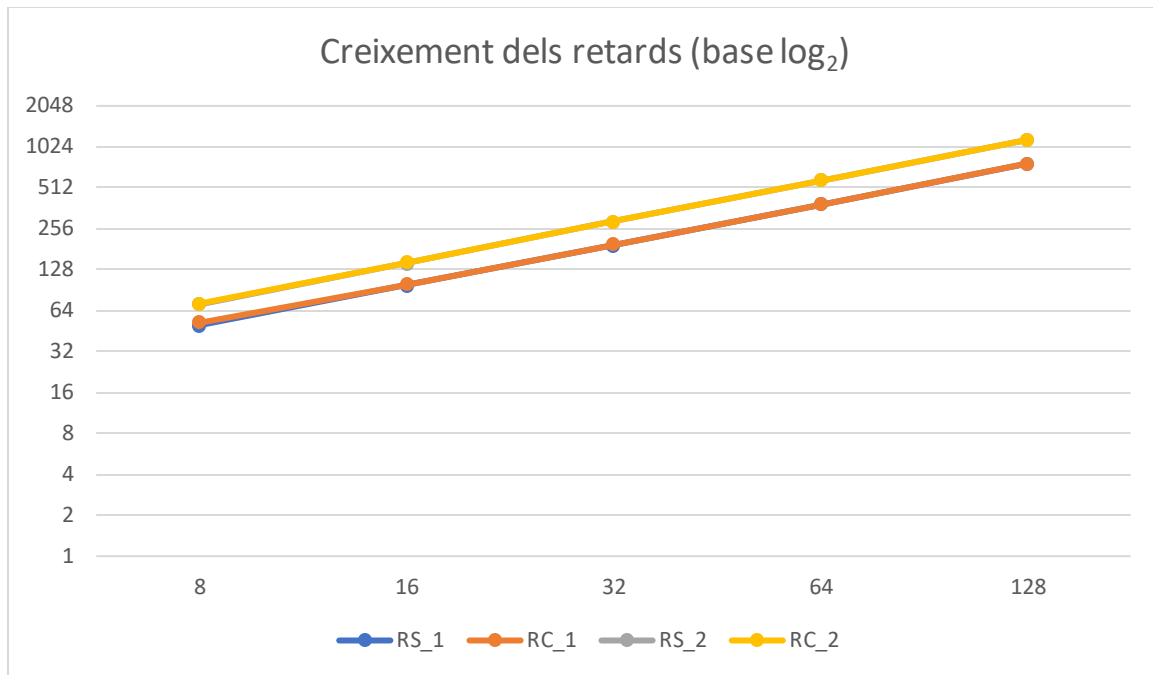
CPA (Tasca 4) $\rightarrow T_S = 10, T_C = 8, P_P = 4$

- $RETARD_S = 10 + (n - 2) * (10 - 4) + (8 - 4) = 14 + 6 * (n - 2)$
- $RETARD_C = 10 + (n - 1) * (10 - 4) = 10 + 6 * (n - 1)$

CPA (Tasca 5) $\rightarrow T_S = 9, T_C = 8, P_P = 0$

- $RETARD_S = 9 + (n - 2) * (9 - 0) + (8 - 0) = 17 + 9 * (n - 2)$
- $RETARD_C = 9 + (n - 1) * (9 - 0) = 9 + 9 * (n - 1)$

n	CPA (Tasca 4)		CPA (Tasca 5)	
	RS	RC	RS	RC
8	50	52	71	72
16	98	100	143	144
32	194	196	287	288
64	386	388	575	576
128	770	772	1151	1152



Com ja havíem demostrat abans, el CPA que incorpora el Full Adder de la tasca 2 és molt millor a major numero de bits, tant en velocitat com en consum. Tenim un creixement lineal dels valors de RC i RS per als dos CPA, ja que cada vegada que dupliquem n , RS i RC també es dupliquen. Podem veure doncs, que les formules del temps realment son formules lineals:

$$RS = 14 + 6 * (n - 2) \rightarrow y = n + mx$$

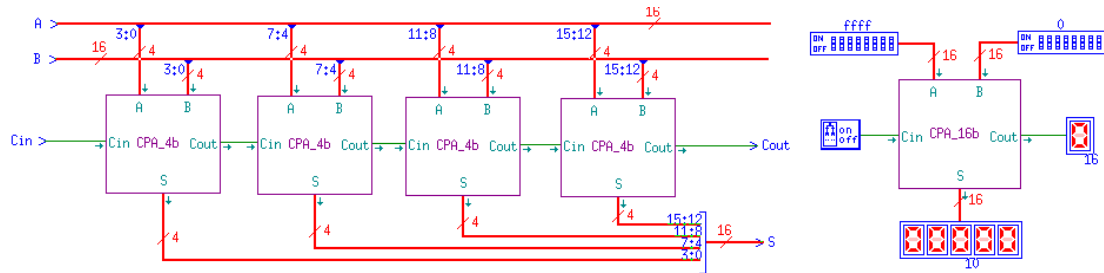
Deduïm que el pendent depèn de $(T_c - P_p)$ pel que el Full Adder de la tasca 3 el qual no té part paral·lela fa que el retard sigui d'un 30% major. Finalment l'àrea depèn de n , significant que en 128 bits les seves àrees fossin 4352 i 5888 respectivament, una diferència del 26%, la qual és bastant gran.

Tarea 7

Especificación: 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.

Diseño: connectem 4 CPAs de 4 bits en cascada, assignant cada un rang de 4 bits del bus d'entrada (A i B) de 16 bits.

Implementación:



Juego de pruebas: hem comprovat diferents casos amb i sense carry in per comprovar l'increment en 1 sobre el resultat. Podem veure també el overflow quan superem els 16 bits.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
FF	ABC	0	BBB	0
FFFF	0	0	FFFF	0
FFFF	0	1	0	1
FFFF	FFFF	0	FFFE	1
FFFF	FFFF	1	FFFF	1

Análisis de Resultados:

- Temps de propagació de S:
 - $\text{RETARD}_S = 14 + 6 * (n - 2) = 98T$
- Temps de propagació de C:
 - $\text{RETARD}_C = 10 + 6 * (n - 1) = 100T$
- Camí crític:
 - $T_{\text{RETARD}} = \max(T_S, T_C) = T_C = 100T$
- Càlcul de l'àrea:
 - $A_{\text{CPA}_{4b}} = 136u$
 - $A_{\text{TOTAL}} = 4A_{\text{CPA}_{4b}} = 4 * 136 = 544u$

Tarea 8

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

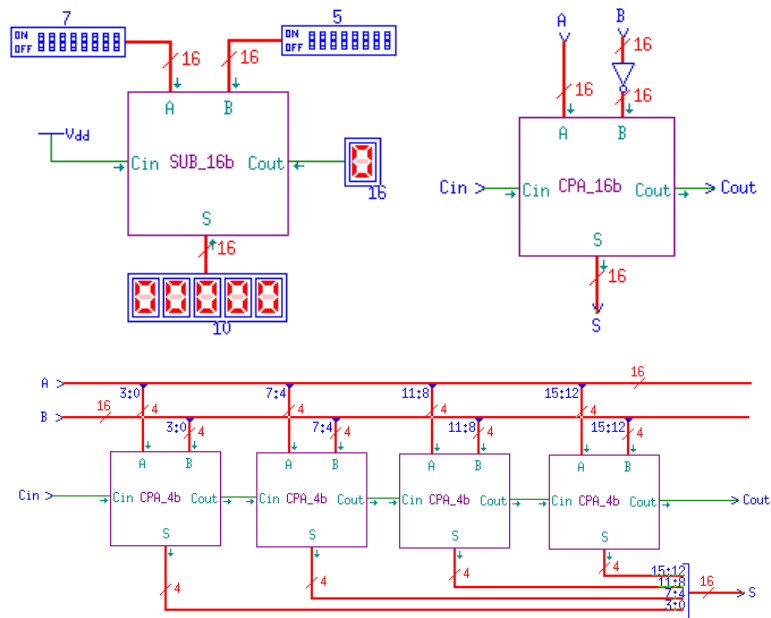
Diseño: per a realitzar aquest circuit podem construir-lo des de zero, o aprofitar el CPA de 16 bits que ja tenim. Evidentment, nosaltres reutilitzarem el CPA de 16 bits. Per a poder realitzar una resta necessitem fer el Complement a 2 de un dels dos valors d'entrada, i després sumar 1, així que el que realment estem fent és una suma, ja que com a tal no podem fer restes.

$$\bullet \quad A - B = A + \text{Ca2}(B) = A + (\bar{B} + 1)$$

En un principi vam utilitzar 2 CPAs, el primer per fer el Ca2 de B i després el segon per sumar A i -B. Però això resultava en un retard i una àrea massa grans. Com que estem realitzant un restador dedicat (que només faci restes, no farà sumes) vam veure que podíem estalviar-nos un CPA i fer-ho només amb un.

Per això utilitzem una NOT de 16 bits per negar el valor B per obtenir el complement a 1, llavors per obtenir el complement a 2 necessitem sumar-li, pel que el C_{in} sempre serà 1, així que el connectem a un Voltage at the Drain (VDD). Deixem el retard de la NOT per defecte, i a dins del Restador trobarem un CPA de 16 bits normal.

Implementación:



Juego de pruebas: hem comprovat diferents casos, i aquí sempre tindrem overflow quan $A \geq B$, mentre que quan $A < B$ no en tindrem, és a dir si el resultat de la resta és positiu tindrem overflow, si el resultat és negatiu no tindrem overflow.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	1	0	1
1	0	1	1	1
0	1	1	FFFF	0
FFFF	0	1	FFFF	1
FFFF	F	1	FFF0	1
FFFF	FFFE	1	1	1
FFFE	FFFF	1	FFFF	0

Análisis de Resultados:

- Temps de propagació de S:
 - $\text{RETARD}_S = 14 + 6 * (n - 2) + R_{\text{NOT}} = 98 + 2 = 100T$
- Temps de propagació de C:
 - $\text{RETARD}_C = 10 + 6 * (n - 1) + R_{\text{NOT}} = 100 + 2 = 102T$
- Camí crític:
 - $T_{\text{RETARD}} = \max(T_S, T_C) = T_C = 102T$
- Càlcul de l'àrea:
 - $A_{\text{CPA}_{16b}} = 544u$
 - $A_{\text{TOTAL}} = A_{\text{CPA}_{16b}} + 16A_{\text{NOT}_{1b}} = 544 + 16 * 2 = 576u$

Hem suposat el retard per defecte de la NOT (2T), i com que la seva àrea en 1 bit es de 2u, en 16 bits serà de 32u.

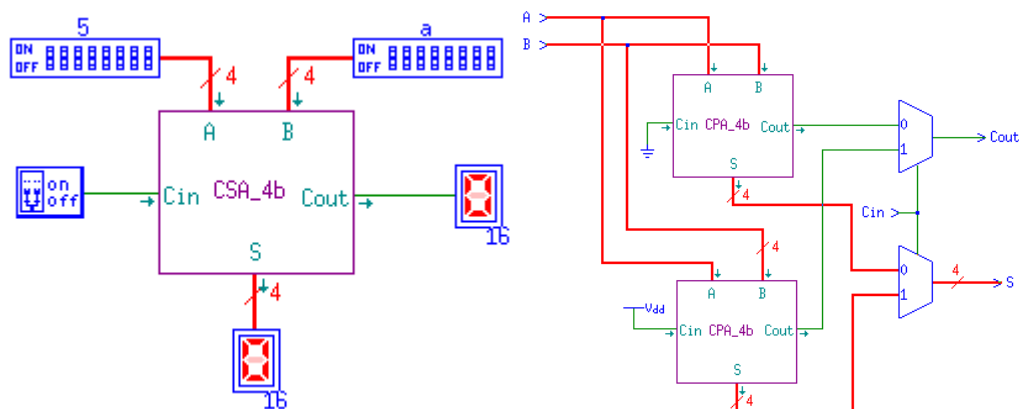
Fase 3: Sumadores CSA

Tarea 9

Especificación: 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.

Diseño: se'ns demana utilitzar els CPAs de 4 bits, cosa que ens permet només haver d'utilitzar 2 multiplexors, a diferència dels 5 que veiem en la figura de l'enunciat. Així que ens queden 2 CPAs de 4 bits connectats a 2 multiplexors, els quals s'encarreguen de seleccionar la sortida Sum i Cout.

Implementación:



Juego de pruebas: hem comprovat diferents casos amb i sense carry in per comprovar l'increment en 1 sobre el resultat. Podem veure també el overflow quan superem els 4 bits.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
3	5	0	8	0
3	5	1	9	0
F	0	0	F	0
F	0	1	0	1
F	F	0	E	1
F	F	1	F	1

Análisis de Resultados:

- Temps de propagació de S:
 - $\text{RETARD}_S = T_{C_CPA} + T_M = 26 + 2 = 28T$
- Temps de propagació de C:
 - $\text{RETARD}_C = T_{S_CPA} + T_M = 28 + 2 = 30T$
- Camí crític:
 - $T_{\text{RETARD}} = \max(T_S, T_C) = T_C = 30T$
- Càlcul de l'àrea:
 - $A_{CPA_{4b}} = 136u$
 - $A_M = 8u$
 - $A_{\text{TOTAL}} = 2A_{CPA_{4b}} + 5A_M = 2 * 136 + 5 * 8 = 312u$

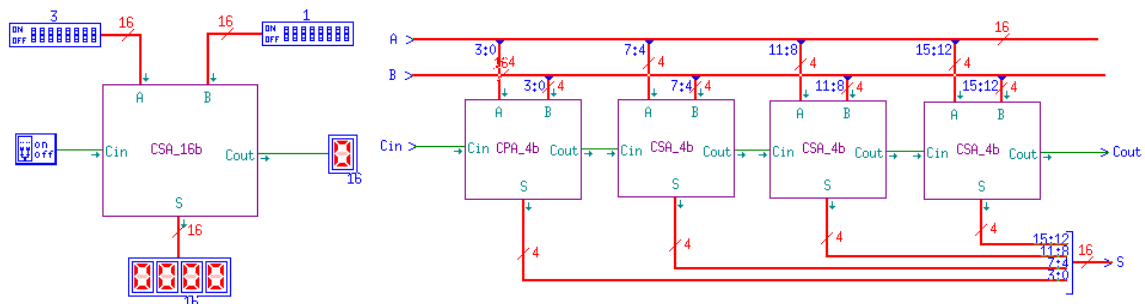
Un multiplexor de 1 bit té un àrea de 8u, pel que un multiplexor de 4 bits tindrà 32u.

Tarea 10

Especificación: 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.

Diseño: per a realitzar aquest circuit, connectem un CPA de 4 bits procedit per tres CSA de 4 bits en cascada.

Implementación:

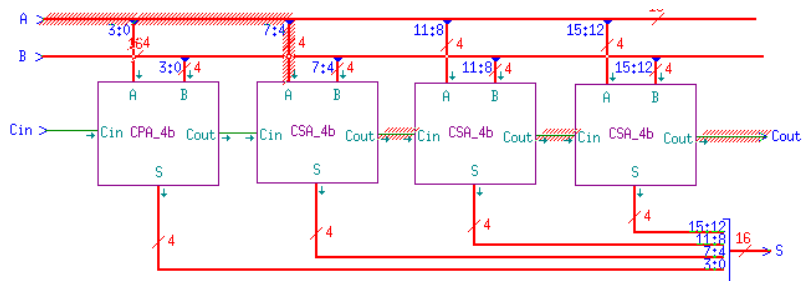


Juego de pruebas: hem comprovat diferents casos amb i sense carry in per comprovar l'increment en 1 sobre el resultat. Podem veure també el overflow quan superem els 16 bits.

INPUTS			OUPUTS	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
FF	ABC	0	BBB	0
FFFF	0	0	FFFF	0
FFFF	0	1	0	1
FFFF	FFFF	0	FFFE	1
FFFF	FFFF	1	FFFF	1

Análisis de Resultados:

- Temps de propagació de S:
 - $\text{RETARD}_S = T_{C_CPA0} + (M - 1) * T_M = 26 + (4 - 1) * 2 = 32T$
- Temps de propagació de C:
 - $\text{RETARD}_C = T_{S_CPA0} + (M - 1) * T_M = 28 + (4 - 1) * 2 = 34T$
- Camí crític:
 - $T_{\text{RETARD}} = \max(T_S, T_C) = T_C = 34T$



- Càlcul de l'àrea:
 - $A_{CPA4b} = 136u$
 - $A_{CSA4b} = 312u$
 - $A_{\text{TOTAL}} = A_{CPA4b} + 3 * A_{CSA4b} = 136 + 3 * 312 = 1072u$

La raó per la qual s'utilitza un CPA i tres CSA en lloc de quatre CSA en cascada és perquè el CPA té un retard i àrea inferior a un CSA. (28T, 136u / 30T, 312u).

- $312u * 4 > 312u * 3 + 132u$
- $34T < 36T$

Podem utilitzar el primer mòdul com a CPA perquè no tenim necessitat de calcular dues sumes en paral·lel, ja que el valor del carry el sabem des del moment 0.

Fase 4: Sumadores CLA

Tarea 11

Especificación: 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

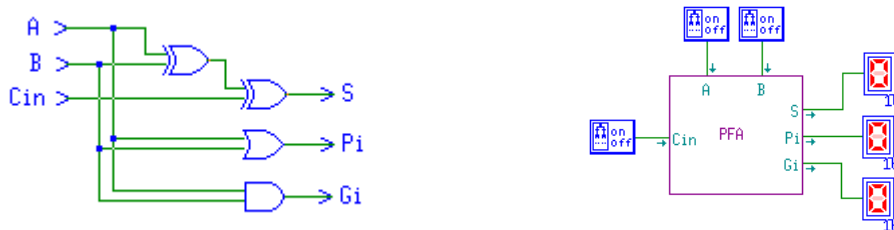
Diseño: En aquesta tasca implementem el Partial Full Adder de la manera en la que es mostra en l'enunciat.

- 3 Inputs (A, B & C_{in}) de 1 bit.
 - Connectats a un switch de 1 bit.
- 3 outpus (S, P_i , G_i) de 1 bit.
 - Connectats a un LED de 1 bit en HEXA.

Les sortides es poden calcular amb la seva formula teòrica:

- $S = A \oplus B \oplus C_{in}$
- $P_i = A + B$
- $G_i = A \cdot B$

Implementación:

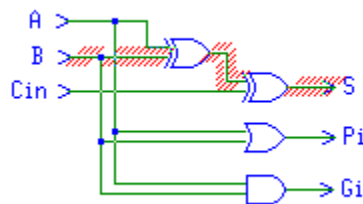


Juego de pruebas: Hem comprovat diferents casos amb i sense *carry in* per comprovar que es propaga quan almenys una de les entrades és 1 i C_{in} també és 1, fent que el carry es transmeti a la següent posició i que només es genera quan ambdues entrades són 1, independentment de C_{in} , ja que en aquest cas el carry sempre serà 1.

INPUTS			OUTPUTS		
A	B	Cin	S	Pi	Gi
0	0	0	0	0	0
1	0	0	1	1	0
1	1	0	0	1	1
0	0	1	1	0	0
1	0	1	0	1	0
0	1	1	0	1	0
1	1	1	1	1	1

Anàlisi de Resultados:

- Temps de propagació de S:
 - $RETARD_S = 2XOR = 2 \cdot 4 = 8T$
- Temps de propagació de G:
 - $RETARD_{G_i} = 3T$
- Temps de propagació de P:
 - $RETARD_{P_i} = 3T$
- Camí crític:
 - $T_{RETARD} = \max(T_S, T_{G_i}, T_{P_i}) = T_S = 8T$



- Càlcul de l'àrea:
 - $A_{XOR} = 8u$
 - $A_{OR} = 6u$
 - $A_{AND} = 6u$
 - $A_{TOTAL} = 2 \cdot A_{XOR} + A_{OR} + A_{AND} = 2 \cdot 8 + 2 \cdot 6 = 16 + 12 = 28u$

Tarea 12

Especificación: 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

Diseño: Els 4 PFAs reben:

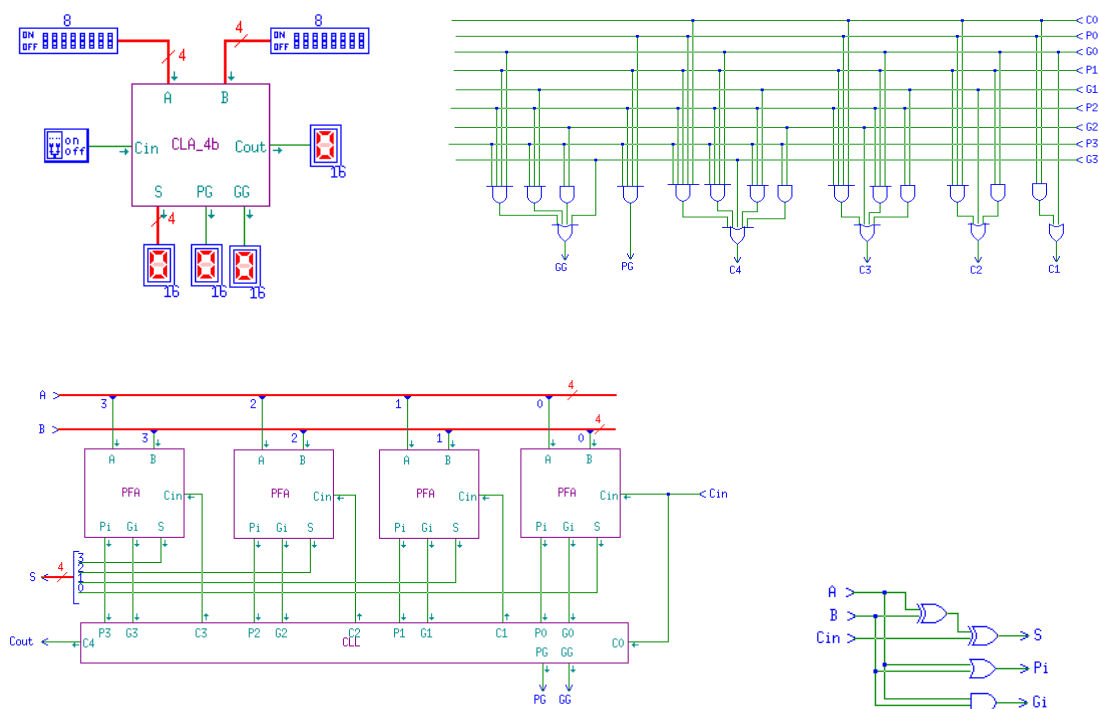
- 2 inputs A_i B_i del bus de 4 bits
- 1 input C_{in} , però que no es propaga en cascada, sinó que es calcula a la CLL per a la resta de PFA

La Carry Look-Ahead Logic:

- Calcula els carrys intermedis C_1 , C_2 , C_3 i C_4 utilitzant la fórmula:

$$C_{i+1} = G_i + (P_i \cdot C_i)$$

Implementación:

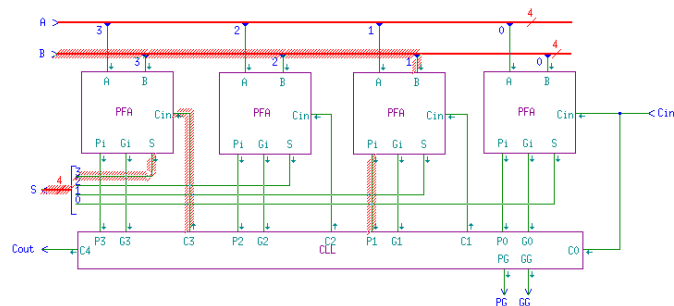


Juego de pruebas: hem comprovat diferents casos amb i sense carry in i sumant valors crítics per comprovar la propagació i generació del carry. Podem veure també l'overflow quan superem els 4 bits.

INPUTS						
A	B	Cin	S	PG	GG	Cout
0	0	1	1	0	0	0
1	0	1	2	0	0	0
0	F	1	0	1	0	1
C	C	0	8	0	1	1
C	C	1	9	0	1	1
F	F	0	E	1	1	1
F	F	1	F	1	1	1

Análisis de resultados:

- Temps de propagació de P_i :
 - $RETARD_{P_i} = T_{OR} = 3T$
- Temps de propagació de G_i :
 - $RETARD_{G_i} = T_{AND} = 3T$
- Temps de propagació de S :
 - $RETARD_S = T_{XOR} + T_{Cin} = 4T + 9T = 13T$
- Temps de propagació de C :
 - $C_{i+1} = G_i + (P_i \cdot C_i)$
 - $RETARD_C = T_{P \circ G} + T_{AND} + T_{OR} = 3T + 3T + 3T = 9T$
- Temps de propagació de GG :
 - $RETARD_{GG} = RETARD_{G_i} + RETARD_{C_{LL}} = 3T + 3T + 3T = 9T$
- Temps de propagació de PG :
 - $RETARD_{PG} = RETARD_{P_i} + RETARD_{C_{LL}} = 3T + 3T = 6T$
- Camí crític:
 - $T_{RETARD} = \max(T_S, T_C, T_{PG}, T_{GG}) = 13T$



El camí crític serà sempre la sortida S. L'enfocament del CLA és poder computar totes les entrades de carry de forma paral·lela, que és l'únic factor del qual es depen per calcular la sortida. Així que el primer mòdul calcula la sortda S molt ràpid perquè té el carry in en temps 0, mentre que els altres la seva sortida es càlculara quan arribi el carry in, que es calcula en tots al mateix temps amb els resultats de Pi i Gi.

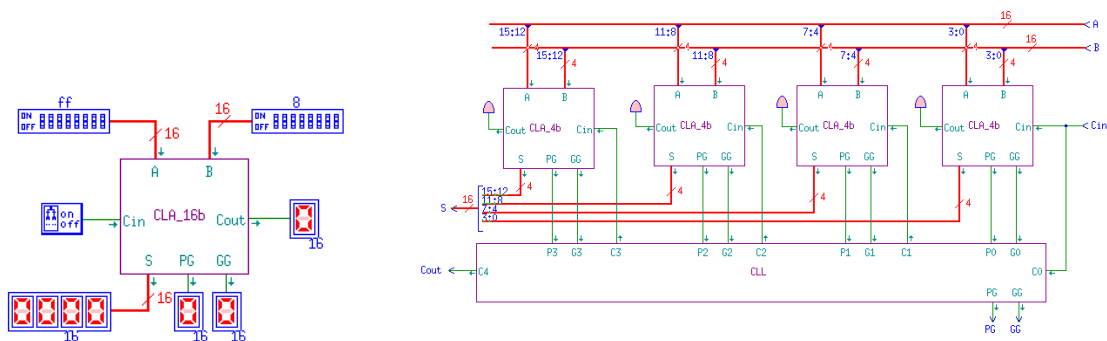
- Càlcul de l'àrea:
 - $A_{PFA} = 28u$
 - $A_{CLL} = 5 \cdot A_{AND_2} + 4 \cdot A_{AND_3} + 4 \cdot A_{AND_4} + A_{AND_5} + A_{XOR_2} + A_{XOR_3} + 2 \cdot A_{XOR_4} + A_{XOR_5} = 6 \cdot 6 + 5 \cdot 8 + 6 \cdot 10 + 2 \cdot 12 = 160u$
 - $A_{OR}, A_{AND}(2 \text{ inputs}) = 6u$
 - $A_{OR}, A_{AND}(3 \text{ inputs}) = 8u$
 - $A_{OR}, A_{AND}(4 \text{ inputs}) = 10u$
 - $A_{OR}, A_{AND}(5 \text{ inputs}) = 12u$
 - $A_{CLA-4b} = 4 \cdot A_{PFA} + A_{CLL} = 4 \cdot 28 + 160 = 272u$

Tarea 13

Especificación: 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

Diseño: connectem 4 Partial Full Adders de manera que el primer Cin entra al primer PFA i també a la Carry Look-Ahead Logic, així com el PG i GG de cada PFA. El Cout dels sumadors individuals està connectat a un LED i la suma individual de cadascun es concatena, donant el resultat S. Cout l'obtenim del CLL.

Implementación:

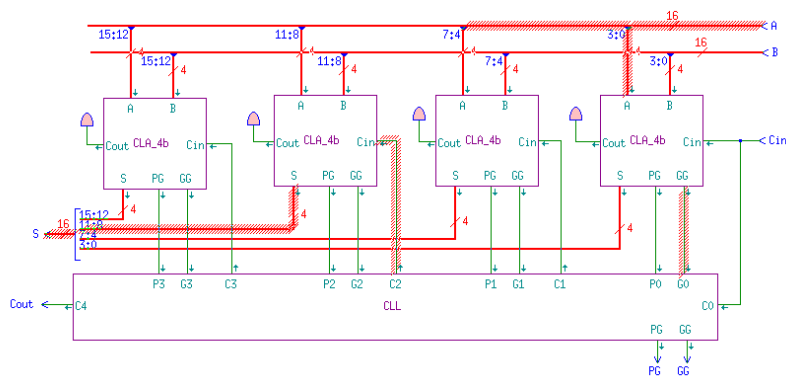


Juego de pruebas: hem comprovat diferents casos amb i sense carry, sumant valors crítics per comprovar la propagació i generació del carry.

INPUTS			OUTPUT			
A	B	Cin	S	PG	GG	Cout
0	0	1	1	0	0	0
ABC	777	1	1243	0	0	0
FFFF	1	1	1	1	0	1
FFFF	1	0	0	1	1	1
FFFF	0	1	0	1	0	1
FFFF	FFFF	0	FFFE	1	1	1
FFFF	FFFF	1	FFFF	1	1	1

Análisis de resultados:

- Temps de propagació de Gj:
 - $RETARD_{Gj} = T_{Gi o Pi} + T_{AND} + T_{OR} = 3T + 3T + 3T = 9T$
- Temps de propagació de Pj:
 - $RETARD_{Pj} = T_{Pi o Gi} + T_{AND} + T_{OR} = 3T + 3T + 3T = 9T$
- Temps de propagació de C:
 - $RETARD_C = T_{PG o GG} + T_{AND} + T_{OR} = 9T + 3T + 3T = 15T$
- Temps de propagació de S:
 - $RETARD_{C_CLA1} = T_C + T_{AND} + T_{OR} = 15T + 3T + 3T = 21T$
 - $RETARD_S = T_{C_CLA1} + T_{XOR} = 21T + 4T = 25T$
- Temps de propagació de GG i PG:
 - $RETARD_{GG,PG} = T_C = 15T$
- Camí crític:
 - $T_{RETARD} = \max(T_C, T_S) = T_S = 25T$



- Càlcul de l'àrea:
 - $A_{CLA-4B} = 272u$
 - $A_{CLL} = 160u$
 - $A_{CLA-16B} = 4 \cdot A_{CLA-4B} + A_{CLL} = 4 \cdot 272 + 160 = 1248u$

Tarea 14

Especificación: 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

Análisis de resultados:

Considerem P_i i G_i com les sortides internes de P_i i G_i de cada CLA de 4 bits, i PG i GG com les P_i i G_i del CLA de 16 bits. La PG i GG del CLA de 16 bits tenen un retard de 15T.

	S	Cin	Pi	Gi	PG	GG
CLA0	13	0	3	3	6	9
CLA1	22	12	3	3	6	9
CLA2..3	25	15	3	3	6	9

CLA0 calcula:

- P_i, G_i : 3T
- S_0 : $4 + 4 = 8T$
- $C_{1..4}$: $3 + 3 + 3 = 9T$
- $S_{1..3}$: $9 + 4 = 13T$
- PG : $3 + 3 = 6T$
- GG : $3 + 3 + 3 = 9T$

CLL_16b calcula:

- C_1 : $9 + 3 = 12T$
- PG : $6 + 3 = 9T$
- GG : $9 + 3 + 3 = 15T$

CLA1 calcula:

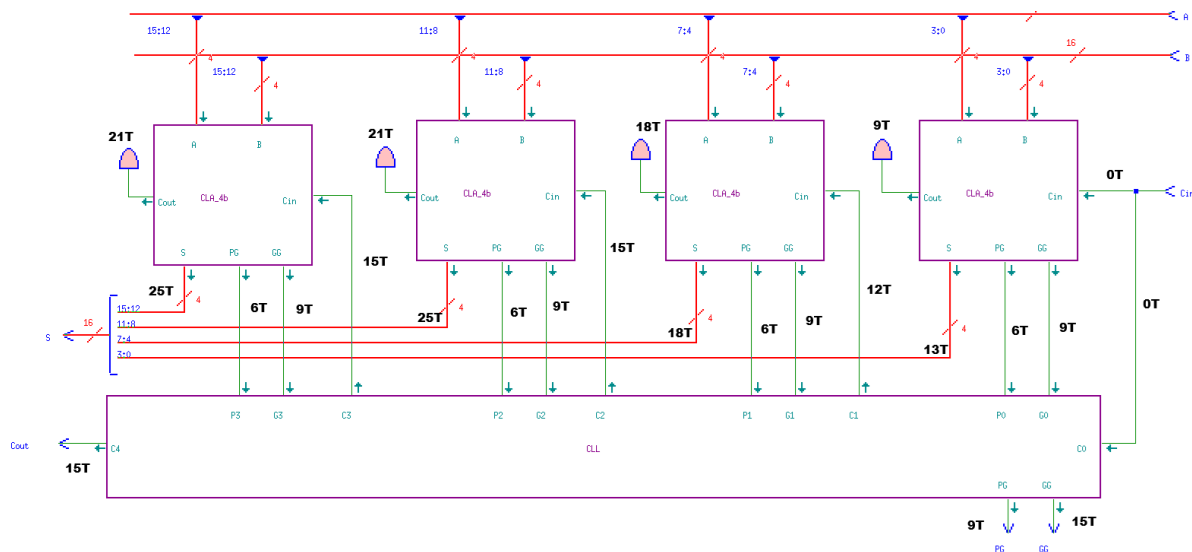
- P_i, G_i : 3T
- S_0 : $12 + 4 = 16T$
- $C_{1..4}$: $12 + 3 + 3 = 18T$
- $S_{1..3}$: $18 + 4 = 22T$

CLL_16b calcula:

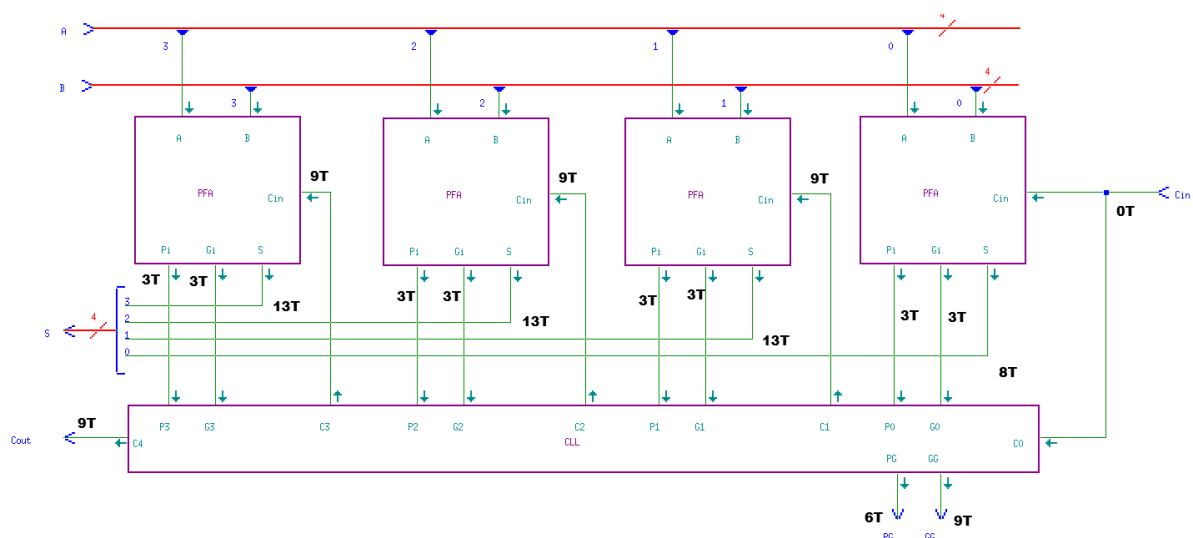
- $C_{2..4}$: $9 + 3 + 3 = 15T$

CLA2..3 calcula:

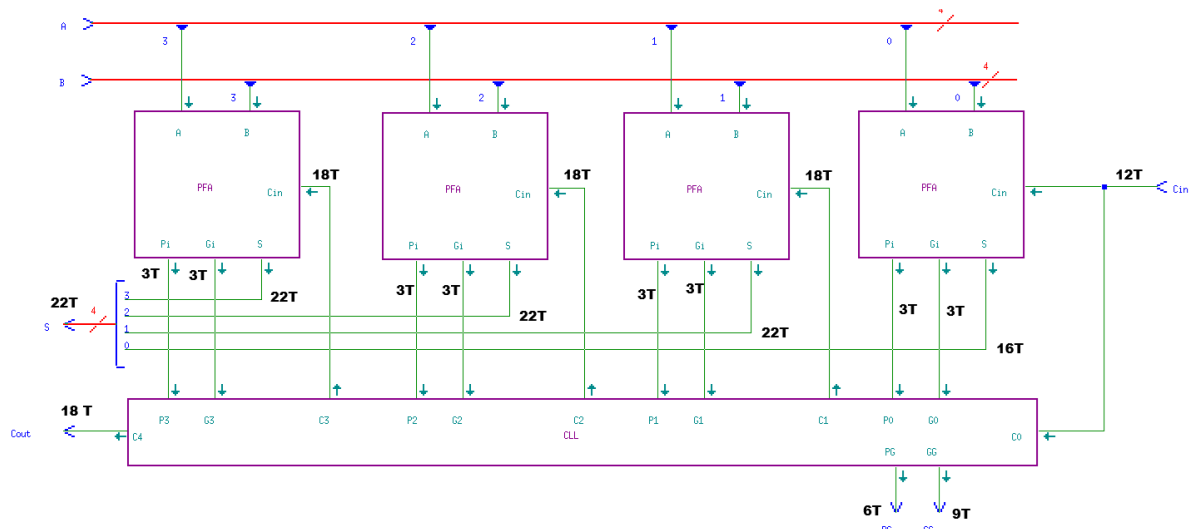
- P_i, G_i : 3T
- $S0$: $15 + 4 = 19T$
- $C1..4$: $15 + 3 + 3 = 21T$
- $S1..3$: $21 + 4 = 25T$
- PG : $6 + 3 = 9T$
- GG : $9 + 3 + 3 = 15T$



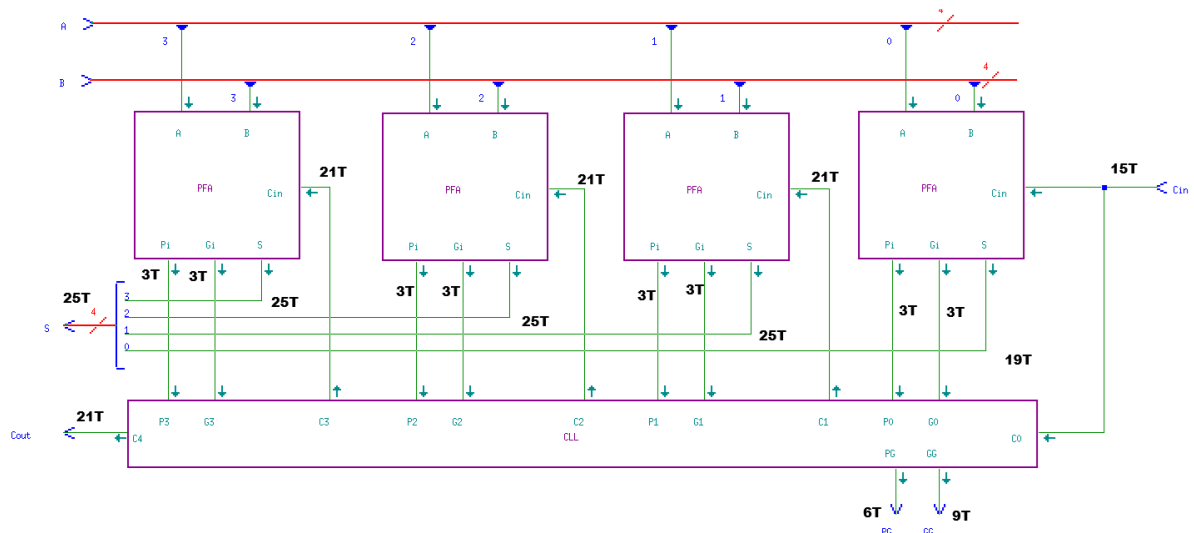
Retard de totes les conexions de CLA 16b



Retard de totes les conexions de CLA0



Retard de totes les connexions de CLA1



Retard de totes les connexions de CLA2 i CLA3

Tarea 15

Especificación: 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

Análisis de resultados:

	T_{RETARD}	Àrea
CPA	100T	544u
CSA	34T	1072u
CLA	25T	1248u

La principal diferència que veiem en els 3 sumadors és la inversa proporcionalitat entre el retard i l'àrea. El que fem és ampliar el hardware per tal de reduir el retard de la suma.

El CPA són Full Adders connectats en cascada, el que implica que s'arrossega el retard del carry al següent mòdul, generant un retard molt elevat en la sortida de la suma.

La primera solució plantejada a aquest problema és la qual precomputa una doble suma, una assumint carry 1 i l'altre carry 0. Això ens permet tenir tots els resultats de les sumes al mateix temps, i seleccionar la sortida correcta una vegada s'hagi calculat el carry amb un multiplexor. Malauradament, amb 16 bits ens trobem el mateix problema, que entre CSAs de 4 bits, al estar connectats en cascada, tot i tenir la suma precomputada al mateix temps en tots els mòduls, la sortida no la obtim fins que arribi el carry, pel que seguim arrossegant el retard del carry entre mòduls. Tal i com hem explicat anteriorment, l'ús d'un CPA com a primer mòdul ens ajuda a tenir una primera sortida més ràpid, evitant fer una doble suma ja que el carry el sabem al moment 0.

Finalment tenim una millora del CSA, el CLA. El CLA calcula tots els carries de forma paral·lela gràcies al Carry Look-Ahead Logic. Aquesta lògica es duu a terme utilitzant ANDs i ORs de forma paral·lela, però tot i generar els carries de forma molt ràpida, aquest circuit requereix una implementació de hardware molt extensa per tal d'assolir aquesta paral·lelització

En conclusió, depenent de si es vol la resposta més ràpida o un menor consum i complexitat escollirem un circuit o un altre, sent:

- CPA: òptim per un baix consum si la sortida de la suma no és imprescindible.
- CSA: òptim si es busca un equilibri entre el consum i temps de retard.
- CLA: òptim si la velocitat de càlcul és la major prioritat.

Fase 5: Multiplicador RCA

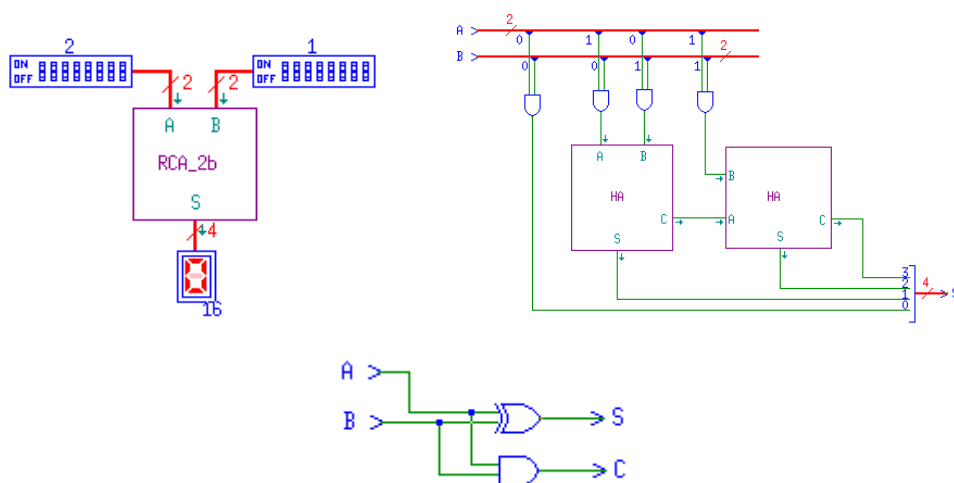
Tarea 16

Especificación: 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.

Diseño: Per implementar aquest RCA hem utilitzat els Half Adders de la tasca 1 en cascada. Els productes parcials es generen a través de portes AND i després es sumen utilitzant Half Adders. Primer, es generen els productes parcials corresponents als bits de A i B. Aquests productes es combinen mitjançant dos half adders per propagar correctament les sumes i el carry a la següent etapa. Finalment, els bits de sortida es concatenen per formar el resultat de 4 bits. És bàsicament com el que es descriu a continuació:

$$\begin{array}{r}
 \begin{array}{cc}
 A1 & A0 \\
 \times & B1 & B0 \\
 \hline
 & A1 B0 & A0 B0 \\
 + & A1 B1 & A0 B1 \\
 \hline
 P3 & P2 & P1 & P0
 \end{array}
 \end{array}$$

Implementación:



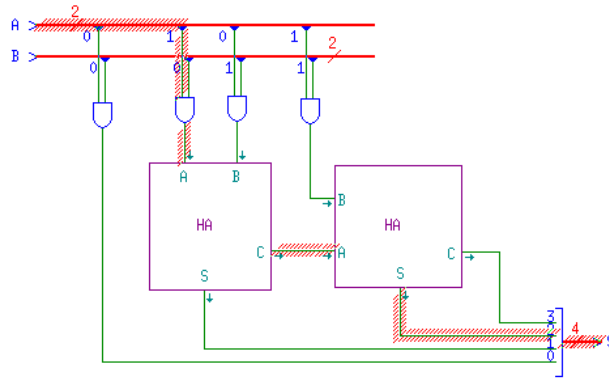
Juego de pruebas: hem comprovat totes les sortides

INPUTS		OUPUT
A	B	S
0	0	0
0	1	0
0	2	0
0	3	0
1	0	0
1	1	1
1	2	2
1	3	3

INPUTS		OUPUT
A	B	S
2	0	0
2	1	2
2	2	4
2	3	6
3	0	0
3	1	3
3	2	6
3	3	9

Análisis de resultados:

- Temps de propagació de S:
 - $T_S = T_{AND} + T_{C-HA} + T_{S-HA} = 3T + 3T + 4T = 10T$
- Camí crític:
 - $T_{RETARD} = T_S = 10T$



- Càlcul de l'àrea:
 - $A_{HA} = 14u$
 - $A_{AND} = 6u$
 - $A_{TOTAL} = 4 \cdot A_{AND} + 2 \cdot A_{HA} = 4 \cdot 6 + 2 \cdot 14 = 52u$

Tarea 17

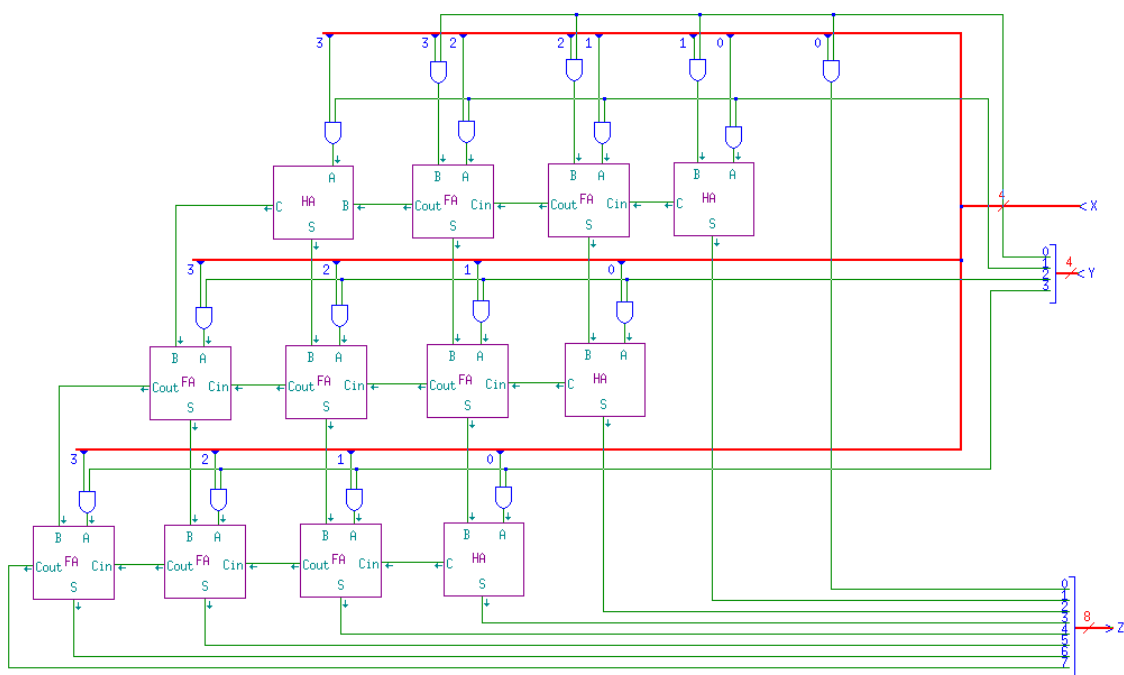
Especificación: Realizad el circuito digital multiplicador Riple Carry Array de 4 bits que se muestra en la siguiente figura e indicad el tiempo de retardo y el área utilizada. Asumid los diseños de Half Adder (HA) y Full Adder (FA) de 1 bit implementados en las tareas anteriores. Suponed también que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

Diseño: Aquest multiplicador funciona descomponent la multiplicació en productes parcials i sumant-los de manera seqüencial amb una xarxa de sumadors.

Primer, cada bit de XXXX es combina amb cada bit de YYY amb portes AND, generant els productes parcials. Els productes es distribueixen entre nivells de sumadors, on les primeres sumes es fan amb Half Adders, ja que no tenen carry d'entrada, mentre que les següents requereixen Full Adders per gestionar la propagació del carry.

					A3	A2	A1	A0
x					B3	B2	B1	B0
					A3 B0	A2 B0	A1 B0	A0 B0
+					A3 B1	A2 B1	A1 B1	A0 B1
+					A3 B2	A2 B2	A1 B2	A0 B2
+					A3 B3	A2 B3	A1 B3	A0 B3
	P7	P6	P5	P4	P3	P2	P1	P0

Implementación:



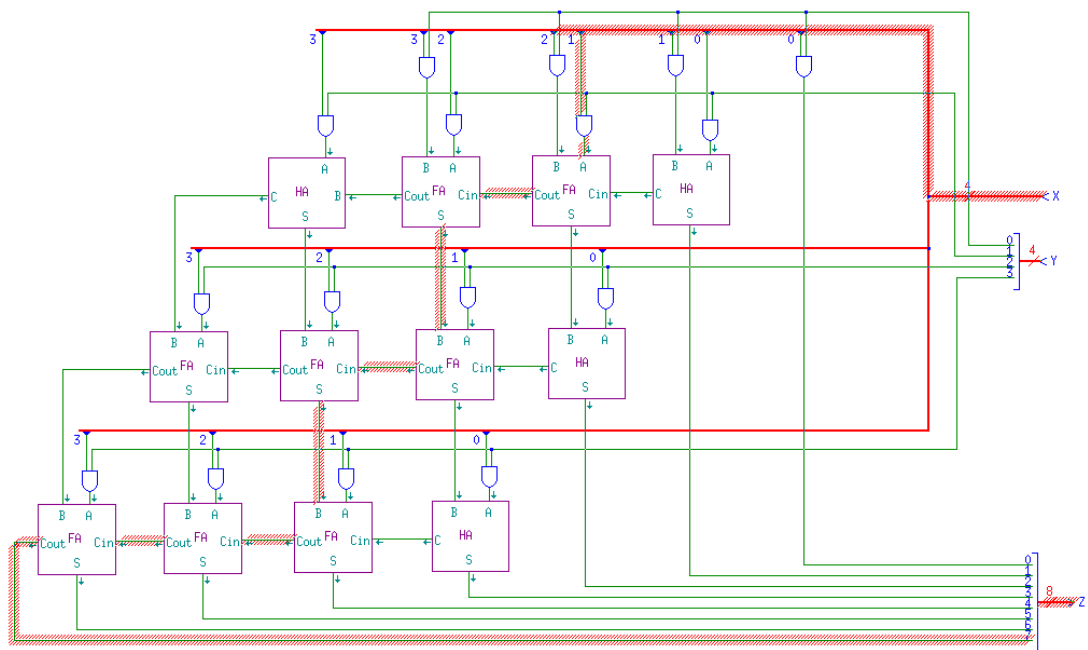
Juego de pruebas: hem desenvolupat el següent joc de proves amb valors (en hexadecimal) crítics i casos en què hi pothaver *overflow* o propagació de *carrys* .

INPUTS		OUPUTS
A	B	S
0	0	0
8	8	40
F	F	E1
A	5	32
7	6	2A
F	2	1E
3	3	9

Análisis de resultados:

- Temps de propagació de S:
 - Calulat a mà: 51T
- Camí crític:
 - Calulat a mà: 53T

Càlculs demostrats a la tasca 18.

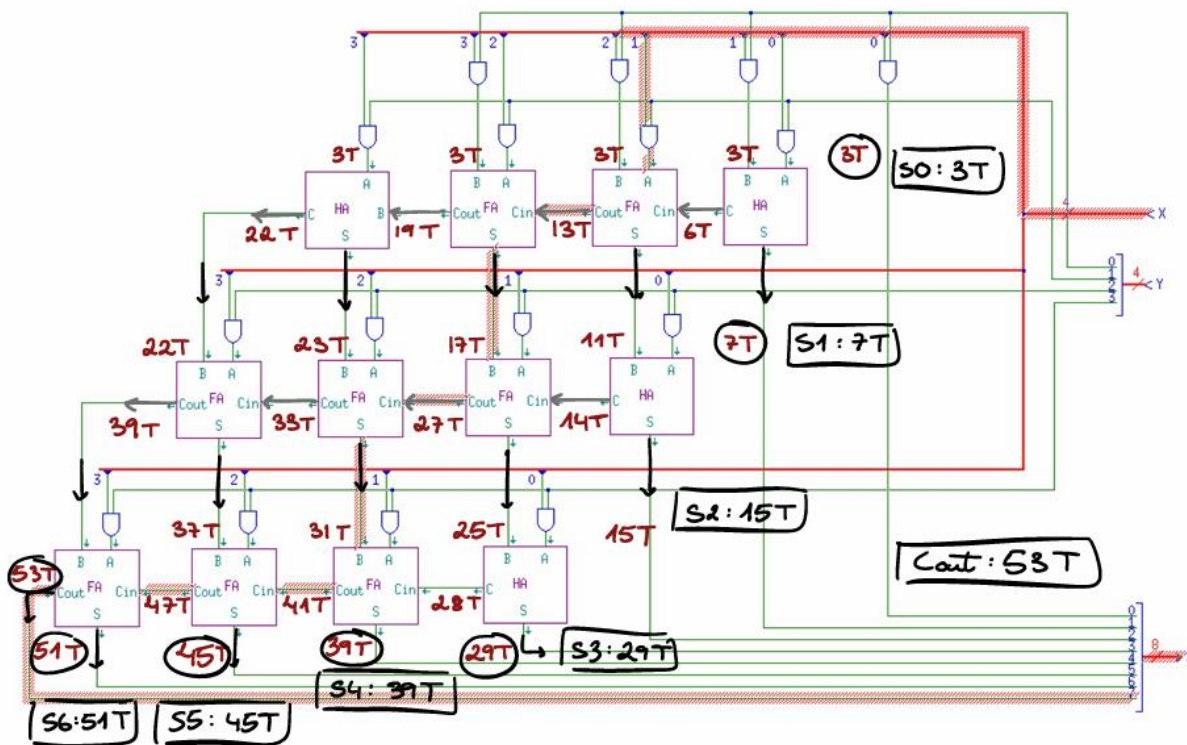


- Càlcul de l'àrea:
 - $A_{HA} = 14u$
 - $A_{FA} = 34u$
 - $A_{AND} = 6u$
 - $A_{RCA-4b} = 4 \cdot A_{HA} + 8 \cdot A_{FA} + 16 \cdot A_{AND} = 4 \cdot 14 + 8 \cdot 34 + 16 \cdot 6 = 424u$

Aquest tipus de multiplicador es coneix com a *fast multipliers*. Això és gràcies a que totes

Tarea 18

Especificación: 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.



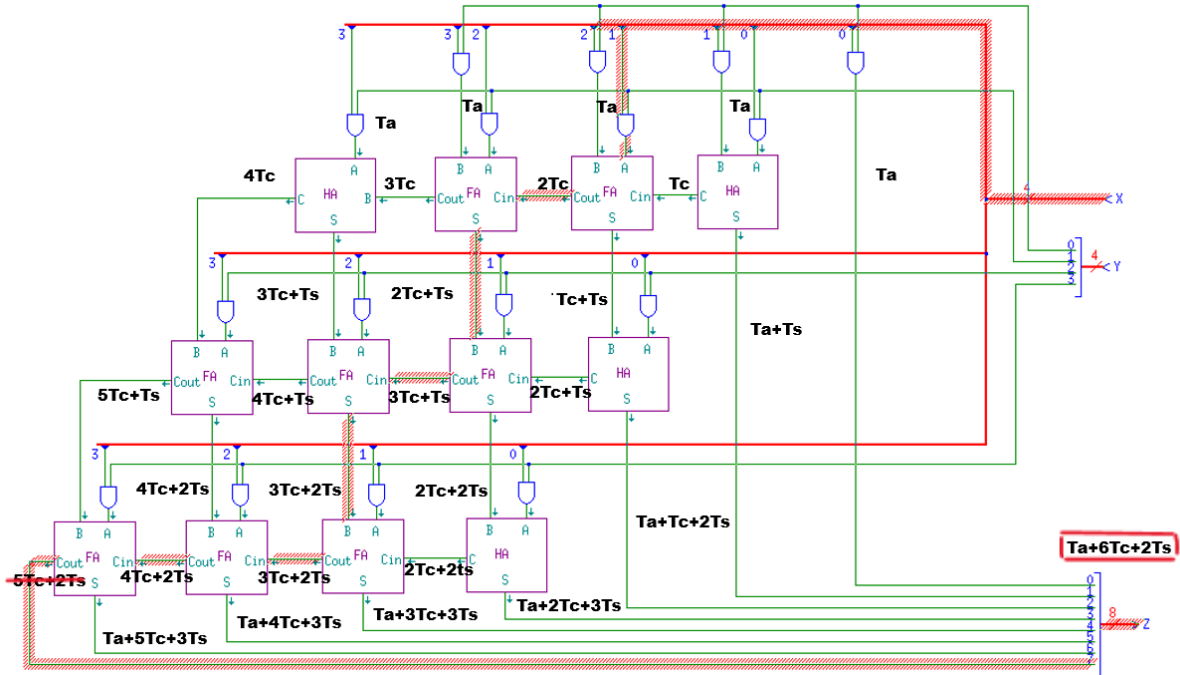
Com es pot veure, el procediment de càlcul és anar seguint el circuit tenint quins temps predominen en cada porta i anar sumant. Tenim fórmules a la teoria, però no són aplicables ja que només es poden utilitzar quan s'assumeixen aquests circuits amb FA que siguin paral·lels o no.

Tenim llavors, que una multiplicació de N-bits tindrà el següent número de components:

- $AND = N^2 = 4^2 = 16$
- $HA = N = 4$
- $FA = N * (N - 2) = 4 * (4 - 2) = 8$

Veiem que això coincideix amb el càlcul de l'àrea que em fet a la tasca anterior. Tantmateix, es compleix per al multiplicador de 2 bits, on:

- $AND = N^2 = 2^2 = 4$
- $HA = N = 2$
- $FA = N * (N - 2) = 2 * (2 - 2) = 0$



Finalment podem treure una formula per tal de treure el retard de la sortida Z.

- $T_A = RETARD_{AND}$
- $T_C = RETARD_{CARRY_HA1}$
- $T_A = RETARD_{SORTIDA_HA}$

En la primera fila, A i B entren en temps T_A , i aleshores es realitza la suma en el primer HA, i treu el carry en T_C . El següent FA espera el Carry del HA, realitza la suma, i treu el carry a $2T_C$. I així successivament.

D'igual manera, en la segona fila passa exactament el mateix. Cada entrada arriba amb el temps del carry acumulat més la sortida. I el carry es propaga en el temps de l'entrada més el carry. És a dir, si l'entrada es a $T_C + T_S$, això se li sumara el T_C que es genera, pel que el carry serà $2T_C + T_S$. Així propagant-se per tota la fila amb Carry i Suma.

Seguim aplicant aquesta lògica fins arribar a l'última sortida, la Z, la qual serà el camí crític ja que serà la última en ser calculada. Per tant si calculem el retard de la sortida Z amb l'expressió deduïda:

- $T_A + 6T_C + 2T_S = 3 + 6 * 6 + 2 * 7 = 53T$

Cosa que equival al retard total del resultat de la multiplicació

Tarea 19

Especificación: 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.

En els processadors actuals, les operacions aritmètiques com la suma i la multiplicació es realitzen dins de la ALU (arithmet-logic unit) i la FPU (Float-Point Unit). Les diferents empreses opten per als circuits que més s'adaptin al seu disseny / necessitats.

- Kogge-Stone

És un sumador paral·lel molt ràpid desenvolupat per Peter M. Kogge i Harold S. Stone al 1973, el qual es pot veure implementat en processadors de Intel, AMD i NVIDIA. Funciona de tal manera que genera i propaga, per cada bit es determina si es genera un desplaçament (G) o si es propaga un desplaçament existent (P). S'utilitza una xarxa de prefixos per calcular els desplaçaments de manera paral·lela, reduint el temps de propagació dels carrys. Un cop calculats els desplaçaments, es realitza una suma final dels bits. Aquest circuit doncs, podríem dir que funciona similarment a un CLA, tot i que el CLA connecta en cascada els mòduls, i comparat amb el Kogge-Stone no té tant bona escalabilitat.

<https://www.anandtech.com/show/16214/amd-zen-3-ryzen-deep-dive-review-5950x-5900x-5800x-and-5700x-tested/6#:~:text=In%20Zen%20%2C%20a%20single,with%20a%20throughput%20of%20%2Fclock>

- Dadda Multiplier

Creat per Luigi Dadda al 1965, aquest multiplicador binari es centra en optimitzar el procés de multiplicació, sent utilitzat sobretot en processadors digitals i en GPUs. El seu funcionament es basa en generar productes parcials multiplicant cada bit a bit dels dos nombres, i aplica una estratègia de reducció dels productes parcials amb etapes de full adders i half adders fins que només queden 2 bits del mateix pes. Al final es sumen aquets dos, obtenint així el resultat. És una optimització del Wallace, el qual redueix el màxim possible cada capa, mentre que el Dadda minimitza el màxim possible l'àrea i el retard del circuit, a costa de tenir nombres finals amb bits més llargs que no pas el de Wallace, cosa que implica tenir adders de més grans.

https://en.wikipedia.org/wiki/Dadda_multiplier

- FMA

El Floating-Point-Accumulate és una unitat funcioan que realitza operacions de multiplicació i acumulació en coma flotant bastant emprat en processadors actuals. Realitza l'operació $S = (A \times B) + C$, on A i B son valors en coma flotant que es multipliquen, i C es un valor en coma flotant que s'afageix al resultat de la multiplicació. Aquesta unitat esta disenyada per realitzar operacions en un sol cicle de rellotge, o en els menors possibles depenent de l'arquitectura del processador. Es sol veure molt en les unitats FPU de AMD. En aquest cas hem trobat que el Zen 3 te un nucli amb 2 unitats de FMAC, on cada un pot realitzar operacions multiplicació-acumulació per cicle de rellotge, permetent un rendiment de 2 operacions FMAC per cicle. També és molt util aplicat a processaments gràfics 3D, videojocs i intel·ligència artificial.

<https://www.anandtech.com/show/16214/amd-zen-3-ryzen-deep-dive-review-5950x-5900x-5800x-and-5700x-tested/6#:~:text=In%20Zen%20%2C%20a%20single,with%20a%20throughput%20of%20%2Fclock>