

Laboratori DCISE

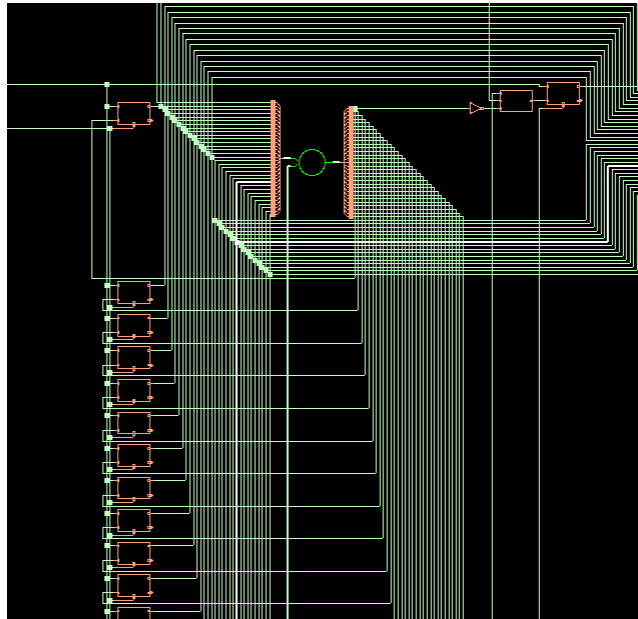
Sessió 3

Xavier Garcia Pich
David Guillen Fandos

Sessió 3

Pregunta 1

La síntesi genera 64 registres d'un bit per a implementar els dos registres de 32 bits i, enlloc de les portes AND i OR que calculen l'overflow, afegeix un circuit combinacional anomenat MAJ31 que calcula el bit de carry donades tres entrades. El sumador i el divisor queden integrats en un únic bloc combinacional.



Pregunta 2

El sumador que s'utilitza és un add_38_14 de tipus slow. Donat que no tenim cap restricció temporal el compilador opta per fer servir les versions més lentes i que tenen menor nombre de transistors.

Pregunta 3

El gate report ens dóna els següents resultats.

	Àrea (um ²)
DFC3	19800
ADD31	8190
Total	28465

Pregunta 4

El màxim retard del circuit és de 15,39ns i el camí crític és el que surt de data_reg[0] i acaba a filt_reg[31] passant per dins del sumador i per tots els blocs de càlcul de carry d'aquest.

Pregunta 5

Un cop afegides les restriccions de clock obtenim un retard màxim de 12.49ns. El camí crític és el mateix que abans.

Pregunta 6

El tipus d'adder es manté en slow donat que la restricció és de 100ns, que és molt de temps per a un sumador.

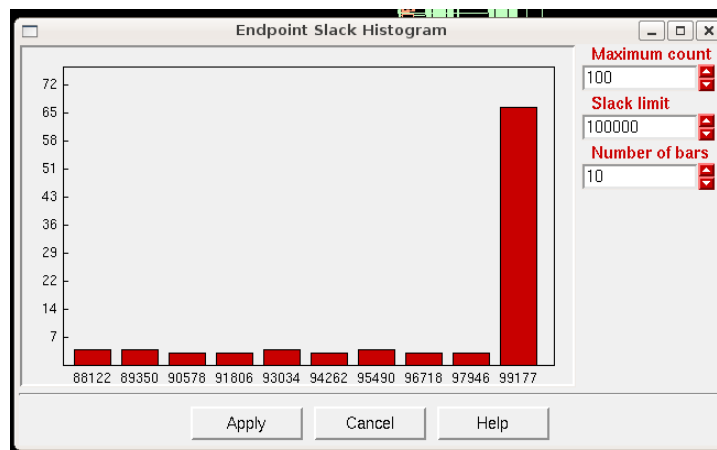
Pregunta 7

El slack obtingut és de 87.50ns, que és el resultat de restar el període del clock al retard màxim. Representa doncs els temps que “sobra” en un període de rellotge.

Pregunta 8

L'histograma mostra, per a cada conjunt de valors de slack, el nombre de components que tenen un slack contingut en l'interval. Així doncs veiem que tenim una columna molt gran per a slacks propers al període de rellotge. Això indica que hi ha molts components que estan llestos (acaben de calcular el seu resultat) al principi del període. En aquest cas són 65, que corresponen als registres d'un bit.

La resta de l'histograma és pla i de valors petits, cosa que indicia que existeix un camí molt llarg que passa per molts components, de manera que van calculant el seu valor en cascada. Un histograma ideal seria aquell que totes les columnes són de la mateixa grandària.



Pregunta 9

L'anàlisi de potència és:

Leakage (nW)	Internal (mW)	Net (mW)	Switching (mW)
21.24	1.6	0.67	2.27

El *area report* obtingut és:

Component	cell	net
filter	28465	1764
adder	8481	1458

Pregunta 10

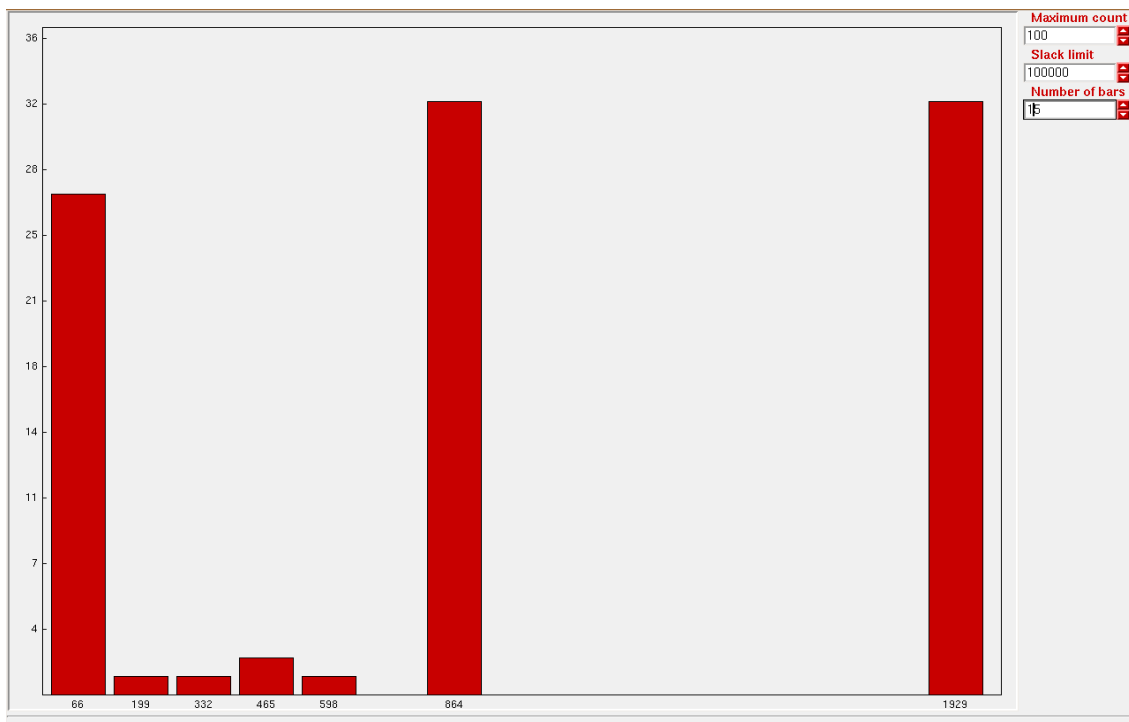
Trobem que amb un clock de 2202ps obtenim un slack de 0ps. Tot i això no sempre s'obté el mateix resultat amb una restricció petita donat que el programa no genera sempre el mateix esquema per als mateixos valors d'entrada (té una certa aleatorietat per a restriccions de rellotge petites).

Pregunta 11

L'adder que s'ha generat és del tipus “very fast”. La seva implementació no és la clàssica en cascada sinó una molt més complexa i amb moltes portes, suposadament farà ús d'algun adder esmentat a l'estudi previ, tot i que és difícil d'identificar amb un anàlisi simple.

Pregunta 12

L'histograma que s'obté és bastant pla. La majoria dels components treballen en paral·lel, pel que s'obté un circuit molt més ràpid.



Pregunta 13

El *power report* és:

Leakage (nW)	Internal (mW)	Net (mW)	Switching (mW)
37.89	13.17	6.58	19.76

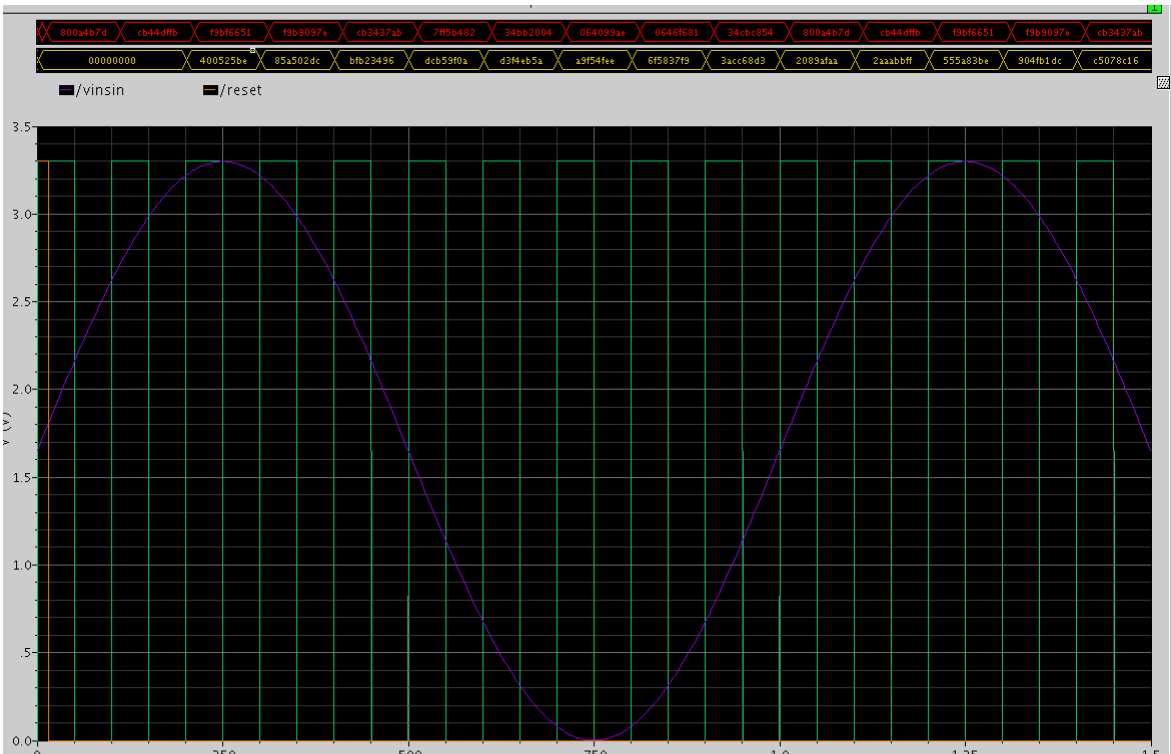
El *area report* obtingut és:

Component	cell	net
filter	53672	8577
adder	33688	8253

Com es pot veure augmenta molt la potència i l'àrea, especialment la del component “adder”.

Pregunta 14

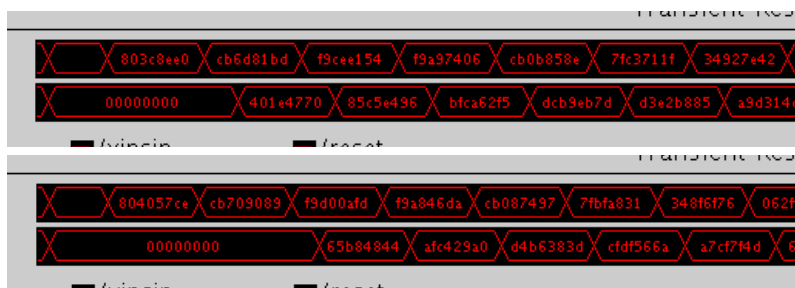
Simulem el circuit per a uns quants valors de període de rellotge i verifiquem que funciona correctament. L'operació per a comprovar-ho és sumar la sortida del filtre en un cicle amb l'entrada en el mateix cicle i veure que, dividida per dos, es correspon amb la sortida del següent cicle.



Pregunta 15

Reduint el període de rellotge trobem que per a un valor de 17ns funciona correctament (comprovant els 7 o 8 primers valors) mentre que per a 16ns ja trobem operacions amb un resultat erroni. Això ens porta a dir que el període de rellotge mínim és 17ns, tot i que per seguretat hauria de ser una mica més gran, ja que el model emprat és en alguns aspectes ideal (anàlisi digital enlloc d'analògic).

La diferència dels 2,2ns dels apartats anteriors seran degudes a diferències entre els models de simulació dels dos programes. Molt possiblement la simulació realitzada anteriorment no tingués en compte algunes característiques del circuit, mentre que la nova simulació fa servir un model molt més proper a la realitat.



Com es pot veure en el primer cas es compleix que $(0x803C8EE0 + 0x0) = 0x401E4770$, mentre que $(0x804057CE + 0x0) \neq 0x65B84844$.