

# Práctica 3

## Núcleo del camino de datos de un procesador y buffer circular

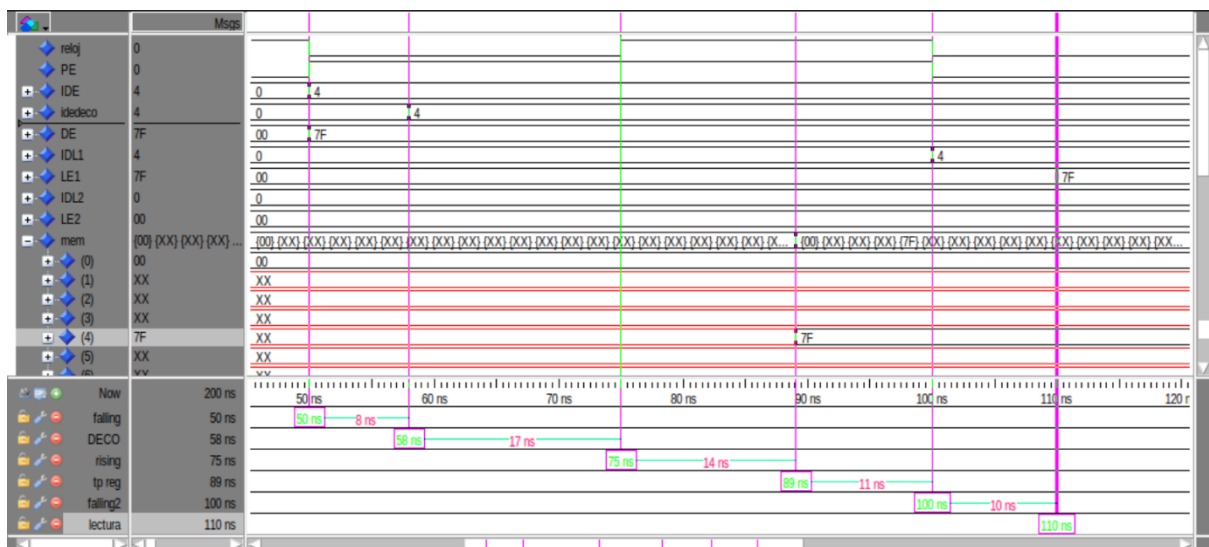
.....

Nom i Cognoms	Jafeth Rafael Concepción Castillo
Nom i Cognoms	Eloi Merino Serrat

Número de grup de laboratori	
------------------------------	--

### Preguntes

#### Exercici 1



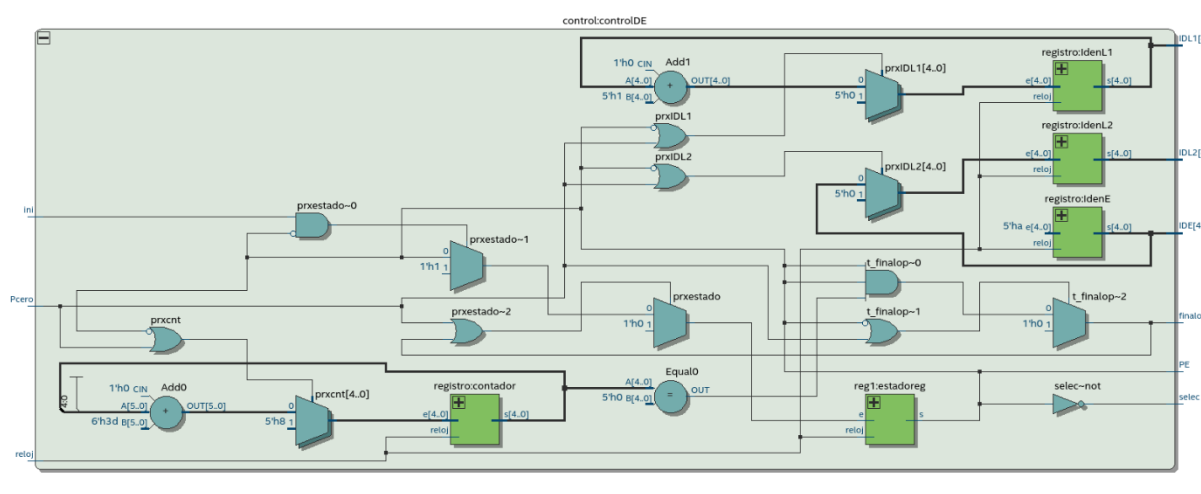
#### Esriptura

Podem observar com transcorren **39 ns** des del primer flanc descendent (que inicia la operació d'escriptura, amb tots els senyals ja estables), fins que el valor escrit apareix a la memòria del banc de registres.

#### Lectura

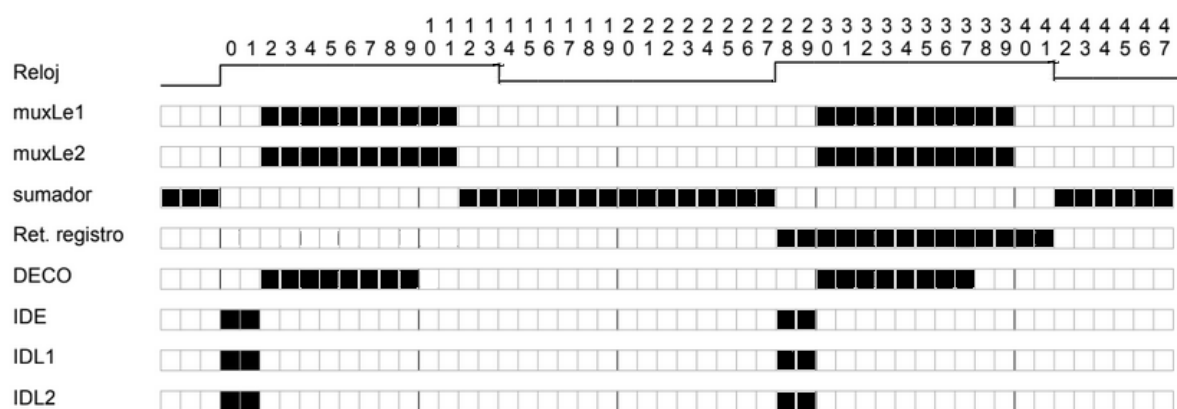
Des del flanc descendent que inicia la operació fins que podem observar la senyal ja estable a LE1 passen **10 ns**, que equival al retard de muxL.

## Exercici 2



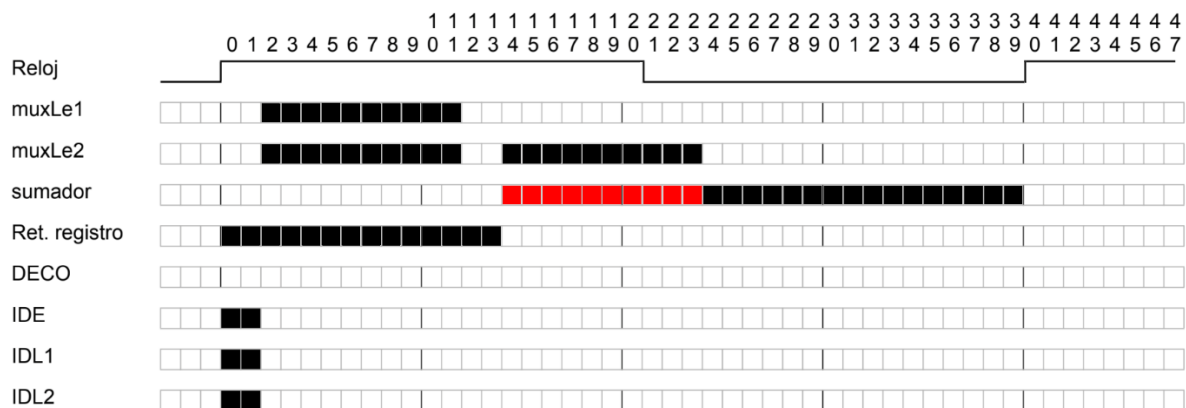
El que alimenta la entrada de l'estat del registre, depèn del comptador i de l'estat actual. D'aquesta manera, recalculem el seu estat cada cop que el comptador augmenta. La sortida *IDL2* només és 0 quan *Pcero* = '1', en qualsevol altre cas, és 10. La senyal *IDE* és sempre 10 (hardcoded).

### Exercici 3



Podríem usar un semi període de **14 ns. 28 ns** de període total quadrat. No podem fer-ho en menys perquè la suma necessita el flanc ascendent després de computar el resultat per a poder guardar-lo al registre IDE, i la suma triga 28 ns exactes des del flanc ascendent.

#### Exercici 4



L'identificador de registre que el controlador posa a la línia IDE és el mateix que el que s'està llegint per a fer la suma (IDL2). Amb això ens trobem que mentre s'està acabant de computar aquesta suma, el valor de la entrada del sumador canvia (10 -> 0, que és l'estat escrit al cicle anterior). Per tant, necessitem esperar a que aquest valor estigui disponible per a que la suma correcta es comenci a computar. En vermell, podem veure aquells cicles on el valor que la suma està resolent és incorrecte, pel que el resultat serà ignorat. Quan ja tenim el valor correcte a la línia LE2, esperem el retard del sumador (tsum) i fem flanc ascendent alhora que aquest apareix per DE. Reduïm així el temps de cicle total i acabem amb **20 ns** per semi període i **40 ns** per període complet.

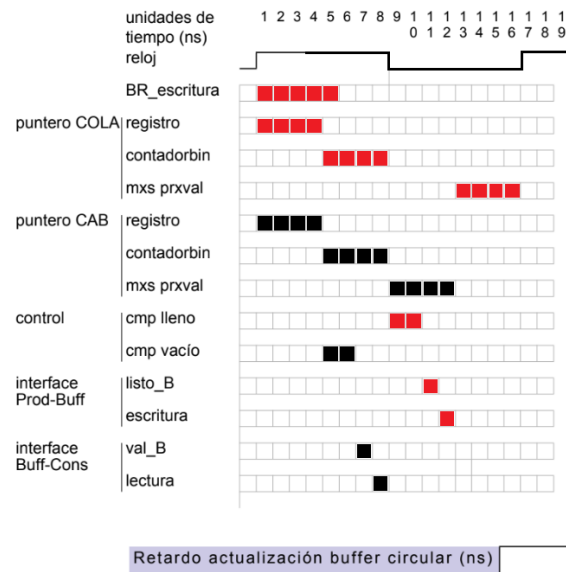
#### Exercici 5

**\*\*Fotografia del cronograma a la última fulla.**

#### Exercici 6

Tant escriptura com PE valen 1 al flanc ascendent següent de la quarta escriptura, fent que s'escrigui una altra dada addicional (s'inicia una altra operació).

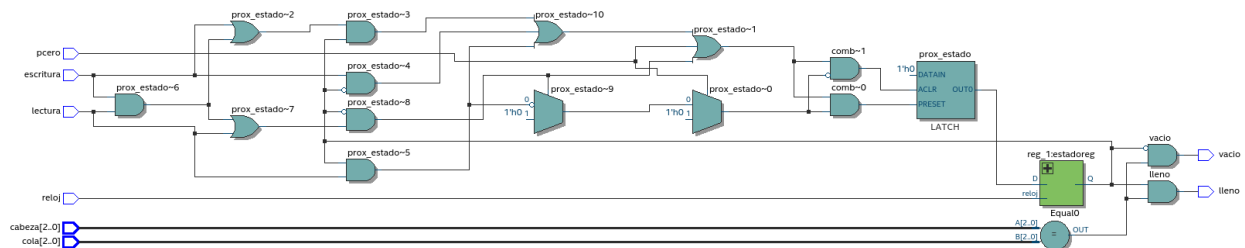
## Exercici 7



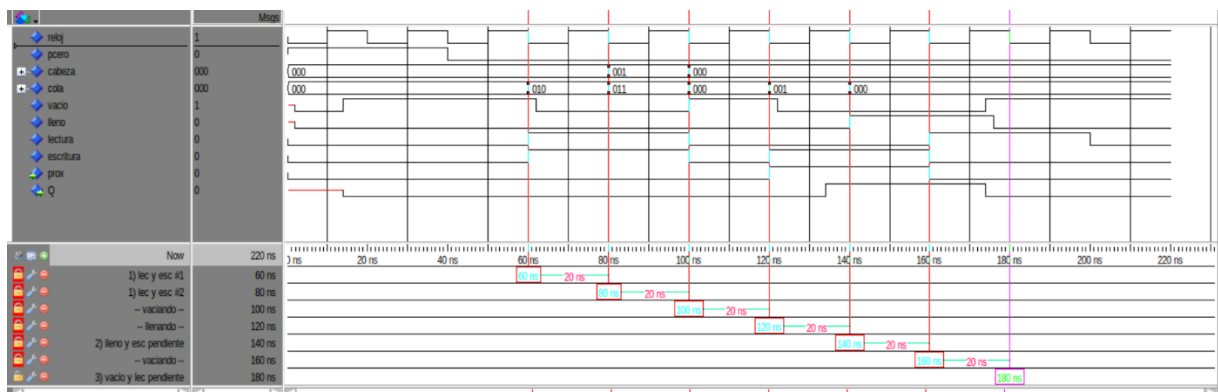
Com que el registre *reg* deixa anar el valor a comparar del mòdul de *control* 4 ns després del flanc ascendent, en el cas de “vacío”, tant el comparador com el comptador poden començar els seus càlculs (i per tant els seus retards) alhora.

En el cas de “lleno”, però, la lògica de control ha d’esperar que el comptador binari tregui el valor per a realitzar aquesta mateixa operació (+4 ns). Aquest és precisament el camí crític. Triguem 16 ns en completar totes les operacions.

## Exercici 8



## Exercici 9



### Exercici 5

