

PROVA DEL BLOC 4

EXPLICA COM FAS CADASCUN DELS PASSOS DEL PROBLEMA

Dissenya un sistema digital que generi la seqüència de números parells 0, 2, 4, 6, 8, i torni a començar a 0.

*Utilitza **NOMÉS registres de 4 bits** (el Clear és active low), **ALUs de 4 bits** i **comparadors de 4 bits** (no facis servir cap porta lògica) i **MUXs**.*

Indica quina ALU (quines funcions té definides) utilitzaries i en quins modes de funcionament la faries servir (no cal que la dissenyis).

Fes l'esquemàtic corresponent i explica el funcionament.

Per poder fer aquest problema ens calen, clarament, 3 elements mínims: 1 ALU que pugui fer la funció de sumar 2 a un número, un registre per poder emmagatzemar la informació i un comparador per decidir quan s'ha arribat a un número determinat i cal recomençar la sèrie. L'ús d'un multiplexor, tal com veurem, és opcional.

- 1) ALU: cal que indiquem que té una funció que és la suma. L'ALU sempre té, com a mínim, una variable de control, que ens determina el funcionament. Per tant aquí la **variable de control** és **F**. Podem assignar-li el valor que vulguem, per exemple, **1**. També és important recordar que **les sortides de l'ALU són una paraula de 4 bits** (que serà el resultat que sigui en funció de la variable de selecció) i **el Carry de sortida**. L'ALU, tot i que està basada en un sumador modificat, que dona com a sortides la suma aritmètica, la funció AND i la funció OR, només té 1 única sortida. Amb la variable de control triem quina de les funcions farem servir com a sortida. També cal recordar que, ha de tenir un Carry d'entrada per permetre connectar 2 o més ALUs per treballar amb paraules de més de bits, més de 4 en aquest exemple.
- 2) El registre genèric de 4 bits permet les operacions càrrega en paral·lel (entrades **D₃..D₀**), manteniment, desplaçar cap a l'esquerra (l'entrada és **D_R** i s'ha de posar sempre, encara que després no li donem cap valor) i desplaçar cap a la dreta (l'entrada és **D_L** també s'ha de posar sempre), per a la qual cosa té 2 variables de control: **S₁** i **S₀**. També cal indicar les sortides **Q₃..Q₀**. En el mode de càrrega en

paral·lel, posem $S_1=S_0=1$. El **Clear** ha d'estar a **1** per poder fer les operacions, i podem aprofitar el **Clear=0 per poder esborrar el registre**. També cal indicar sempre el Clock.

- 3) El comparador cal escriure sempre quina és la paraula A i quina la paraula B i dir si comparem A amb B o B amb A. En general, hem fet servir que comparem A amb B i, per tant, **G** vol dir $A > B$, **E** és $A = B$ i **L** és $A < B$. Però cal indicar-ho sempre.
- 4) Si volem fer servir un multiplexor, per exemple, de 2 entrades de dades i 1 entrada de selecció, hem de recordar que cada canal d'entrada de dades té només 1 bit i que si volem seleccionar entre dues paraules de 4 bits, cal que fem servir 1 MUX per cada bit i, per tant, 4 MUXs en aquest problema.
- 5) **Cal indicar SEMPRE quina és la sortida.**
- 6) Cal identificar sempre els diferents terminals (entrades i sortides) i, normalment, els circuits els representem amb les entrades a l'esquerra i les sortides a la dreta (o les entrades a dalt i les sortides a baix). Si hem de

Hi ha diverses opcions per dissenyar el circuit. Aquí en farem 3, però n'hi ha d'altres:

Opció A: la sortida $A < B$ del comparador dona 1 fins que el valor que surt de l'ALU és 10, moment en el que passa de 1 a 0 i actua sobre el Clear del Registre, el qual posa a 0 totes les sortides $Q_3..Q_0$.

Inconvenient: el registre té a la sortida el valor 8 durant un instant de temps curt, el que cal per que l'ALU sumi 2 i el comparador canviï la seva sortida. Això vol dir que la seqüència no té tota ella la mateixa durada.

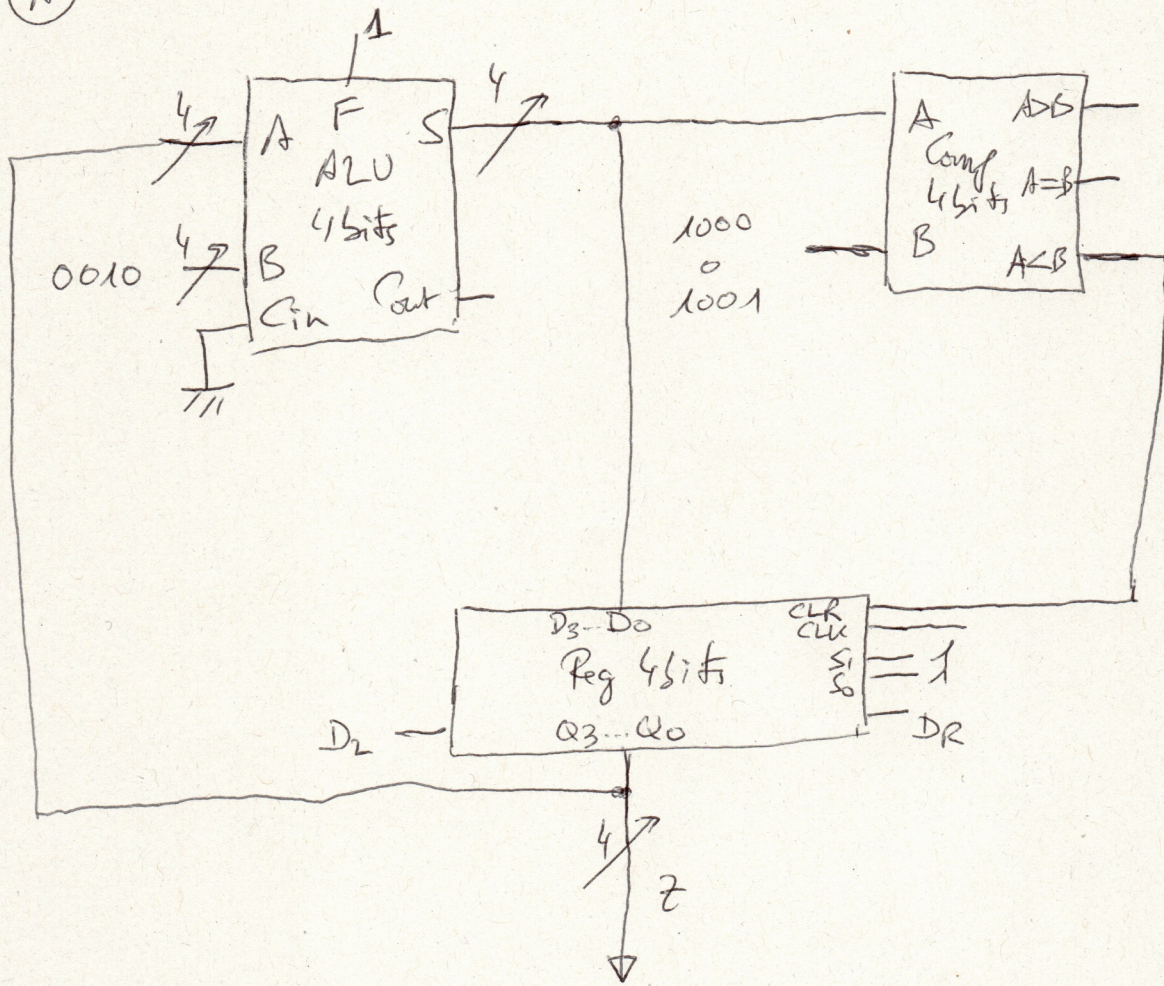
Opció B: La sortida del comparador actua de variable de selecció de 4 MUXs (1 per cada bit de la paraula), de manera que o deixen passar els bits de la paraula ($sel=1$) o deixen passar tot 0s ($sel=0$).

Avantatge: és un circuit síncron i cada valor de la sortida està durant un cicle de rellotge complet.

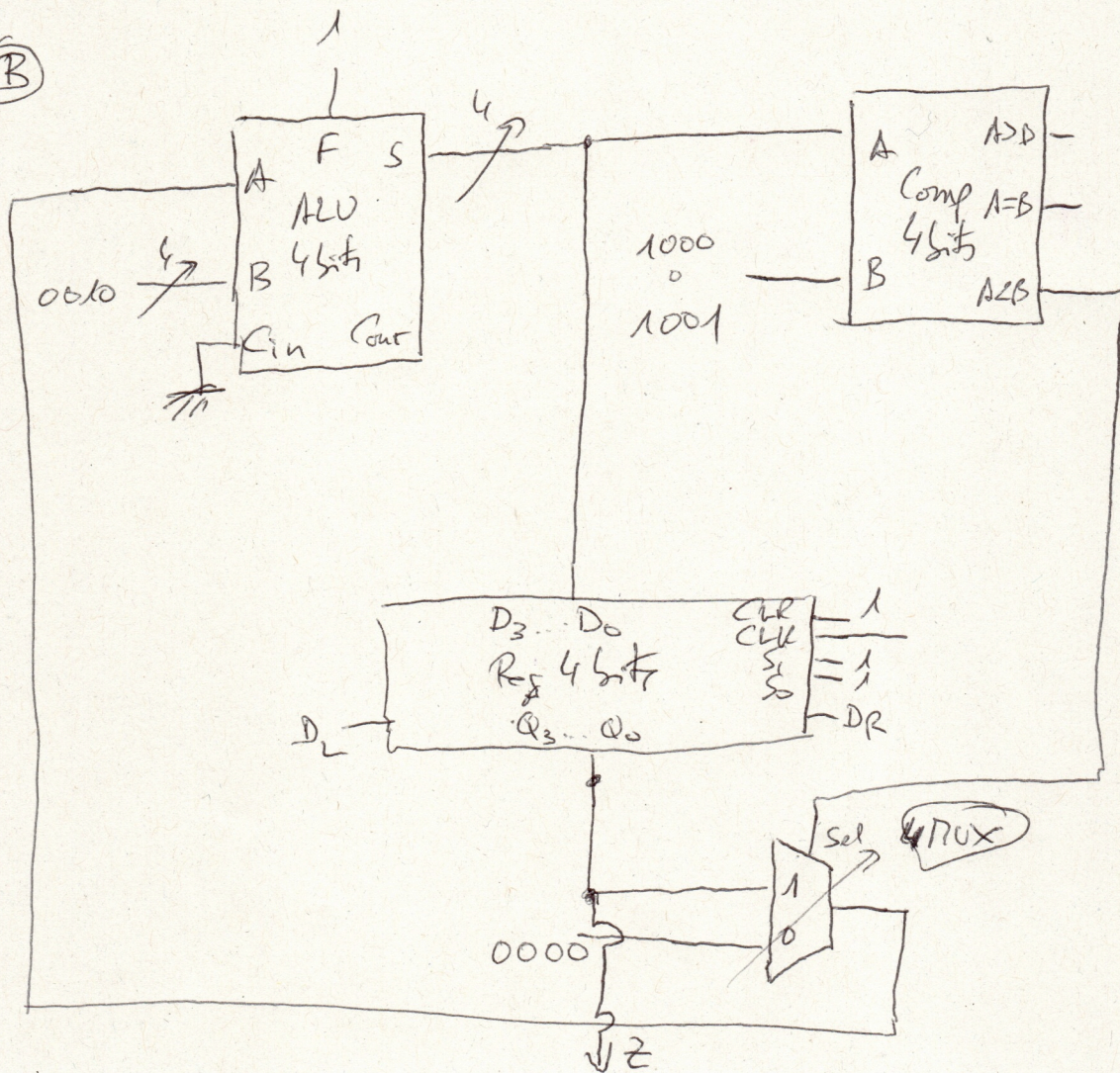
Opció C: La sortida del comparador actua sobre la variable de selecció S_0 . de mode del Registre, passant de càrrega en paral·lel ($S_1=S_0=1$) a desplaçar cap a l'esquerra ($S_1=1$, $S_0=0$).

Avantatge: Aquest circuit també és síncron, igual que la Opció B.

(A)



(B)



(C)

