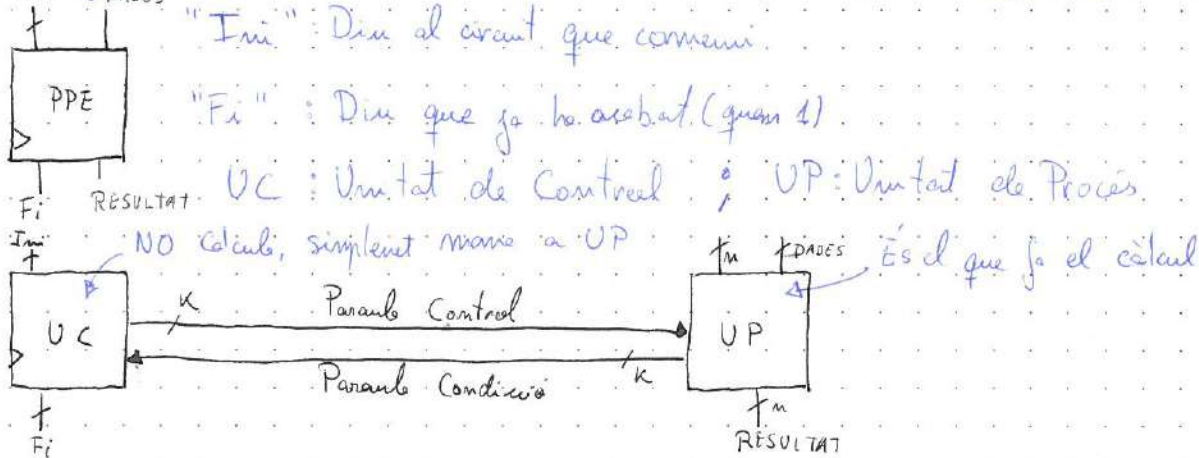


# Processadors de Propòsit Específic (PPE)

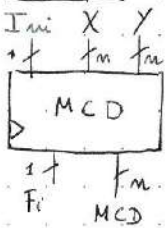
Dissej d'un processador que realitza un càlcul, però no és programable !!



Paraula Control: Codifiquem ordres a fer b UP.

Paraula de Condició: Codifica events rellevants per b evolució del càlcul.

## Exemple



$A = x, B = y, Ini = 1$

While ( $A \neq B$ ) {  
 if ( $A > B$ )  $A := A - B$ ;  
 else  $B := B - A$ ;  
}

$MCD := A$ ;  $Fi = 1$ ;

## Llista Etapes

- Inicial: Esperem a començar [ $A := x$ ;  $B := y$ ].
- $A \neq B$ .
- $A > B$ .
- $A := A - B$ .
- $B := B - A$ .
- Final

## Llista Events Rellevants

- $A \neq B$
- $A > B$

## UP (Unitat de Procés)

a) Elements de Memòria. Necessitem 2 registres amb càrrega (1 per  $A-B$  i 1 per  $B-A$ ).

b) Elements de Càlcul. 1x EQ, 2x SUB, 1x LT. [La NOT no jo feta aquí perq no calc]

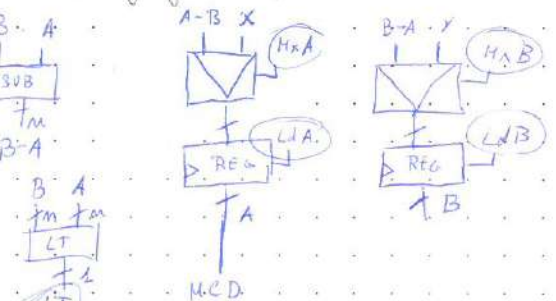
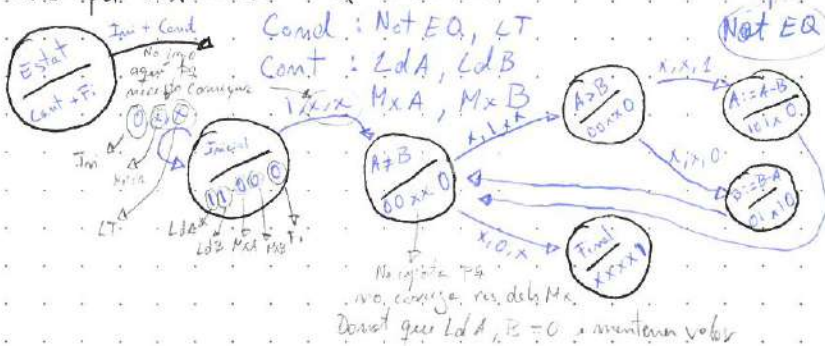
c) Camins de dades entre a) i b)

d) Paraules de Condició: NotEQ, LT

\*) Paraules de Control: LdA, LdB, MxA, MxB

## UC

✗ Etapes → ✗ Estats → ✗ Bietables



# Si surt Mx de UP,  
 la igual que el control  
 SEMPRE surt la paraula  
 de Control

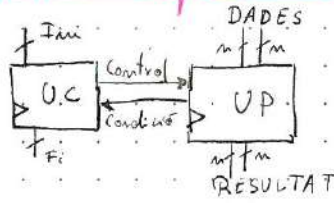
# Análisis Temporal PPE

<sup>UC</sup>Entrada → Bientable<sup>UC</sup><sub>UP</sub>

<sup>UC</sup>Bientable → Bientable<sup>UC</sup><sub>UP</sub>

<sup>UC</sup>Bientable → Sortida<sup>UC</sup><sub>UP</sub>

⚠ Bientable nomén en sumo si SUAT del Bientable.



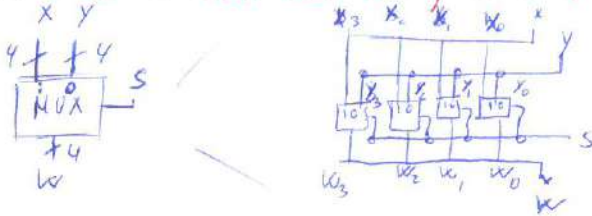
Es el mateix Circuit

aiñ que Bi → Bi pot ser

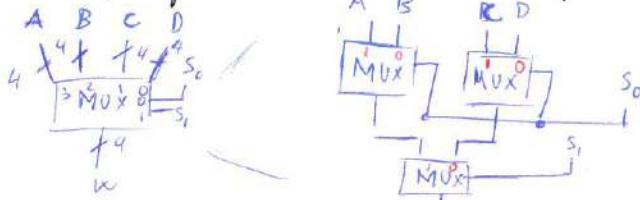
que siga  $\begin{matrix} UC \rightarrow UC \\ UC \rightarrow UP \\ UP \rightarrow UP \end{matrix}$  } Depèn.

7.1. Esquema lògic intern del MUX-2-1 a partir de Mx-2-1.

⚠ NO ES el mateix. Un permut Bus de bits i l'altre treballa bit-bit.

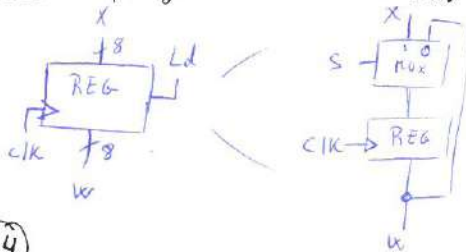


7.2. Esq. lògic intern MUX-4-1 a partir de MUX-2-1



⚠ NO OBLIDAR  
o ficar valor a les entrades '0,1,...' i a les de control

7.3. Esq. lògic REG (8bits) amb (Ld) a partir de REG (8bits) i MUX-2-1



Recorde el CLK ⚠

7.4

#Estava malament el punte algorithme.

a) Completa cromo) 0x00, 0x3A, 0x00, 0x3B, 0x3B, 0x00, 0xFA, 0xFA, 0x00, 0x00, 0x01, 0xFF, 0xFF

b) Completa cromo) 0xE, 0xF, 0x0, 0x4, 0x0, 0x1, 0x2, 0x3, 0x0, 0x1, 0x2, 0x3, 0x4

c) Cromo) 0x27, 0x8D, 0x00, 0x3B, 0x40, 0x8A, 0x00, 0x0C, 0xCC, 0x00, 0x00, 0xFF, 0xA9

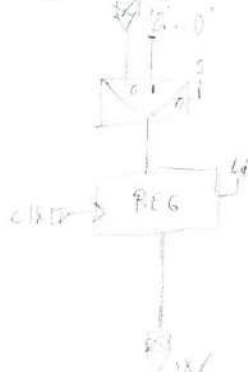
d) Cromo) 0x00, 0x33, 0x63, 0xA2, 0xD2, 0x8E, 0xA5, 0x83, 0x66, 0x7D, 0x1D, 0x80, 0xD4

7.5

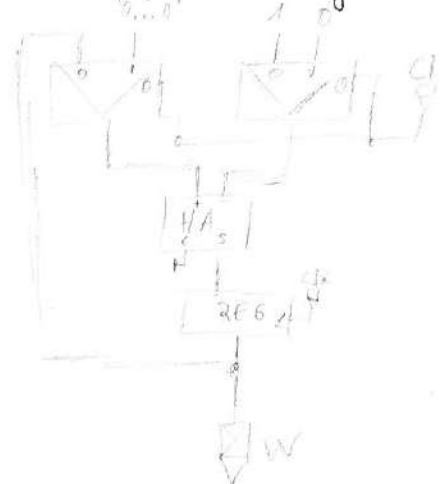
a) REG-CI  
Reg. 8b; senyal posada 0 (CI)  
IN: X OUT: W; Mx-2-1



b) REG-CILd  
Reg. 8b; senyal posada '0'  
senyal canvia Mx i Reg.



d) Seq. - 0-7-CI  
Gran. 4bits seq. nombre 0-15  
a nombre per cycle. Te senyal Clear. (CI)





e) Accumulator - C1

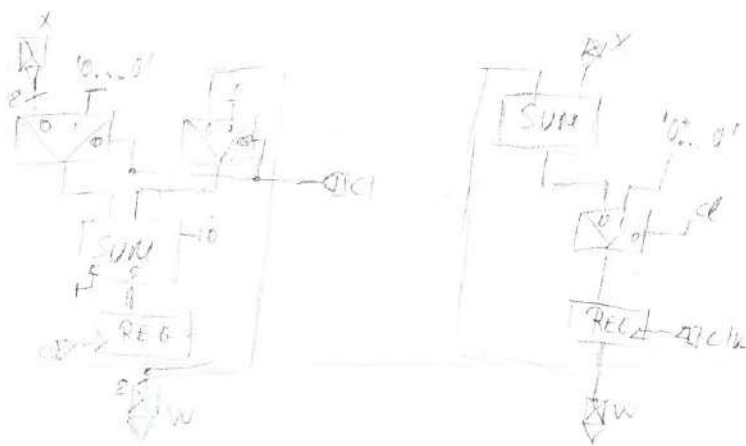
Suma tots valors entrades desde

Cicle després que C1 hagi valgut 1.

$X, W$  8bits. Es fa suma 2.

Final de cicle senyal C1 = 1

$W$  valdrà 0 al cicle següent.

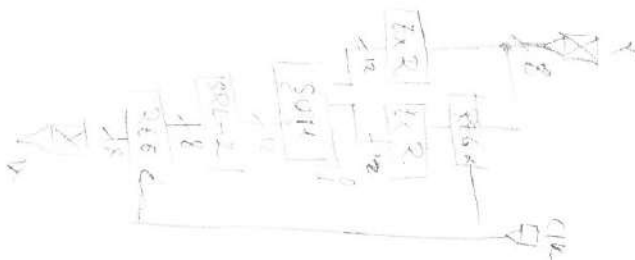


f) Mitjana - 2- Mostres

$$W_n(c+1) = [X_n(c) + X_n(c-1)] \div 2$$

No importa valor 2 primers ac.

Bussos de 8bits. Pot ser necessari fer extensió de Rang.



$N = 0.03$

7.6. Complete Disseny de Suma-N.

i) Completa cronograma