

## Examen final IC (Part 1)

- Durada de l'examen: 2:30 hores.
- Les preguntes s'han de resoldre al **FULL DE RESPOSTES**.
- No podeu usar calculadora, mòbil, apunts, etc.
- La solució de l'examen es publicarà a Atenea demà al matí.

Per algunes de les preguntes són útils les figures i taules que es mostren en el full apart.

La puntuació de les preguntes d'aquesta part de l'examen corresponen a 6 punts de la nota final de l'examen.

## Pregunta 1) (0,2 punts)

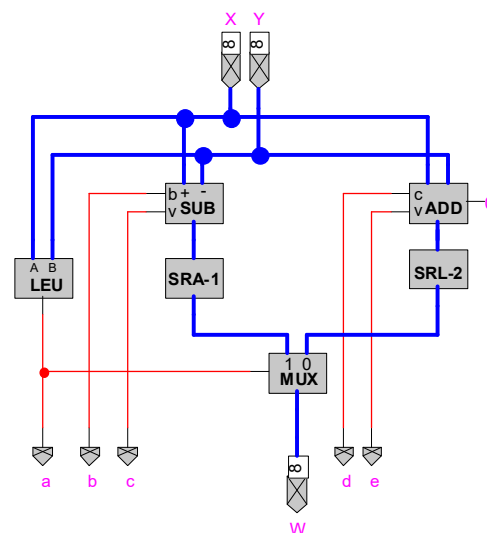
Cada fila de la taula té tres columnes amb: el vector de 8 bits  $X$ , el valor que representa  $X$  interpretat com un nombre natural codificat en binari,  $X_u$ , i el valor que representa  $X$  interpretat com un nombre enter codificat en complement a dos,  $X_s$ . Completeu totes les caselles buides.

$X$ (hexa)	$X_u$	$X_s$
	200	
BA		

## Pregunta 2) (0,4 punts)

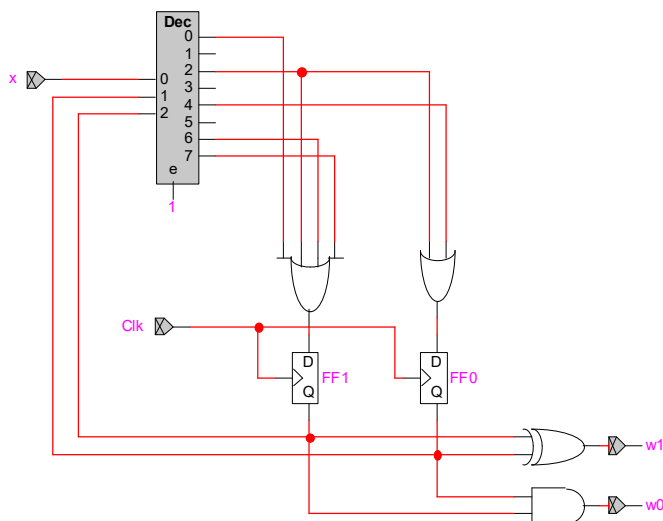
Donat l'esquema amb blocs del CLC de la dreta, completeu la següent taula que indica el valor de les sortides del circuit per cada un dels tres casos concrets de valors de les entrades (un cas per fila). El bloc LEU calcula  $A \leq B$ .

$X$	$Y$	$W$	$a$	$b$	$c$	$d$	$e$
11111111	10011111						
00111111	10011111						



## Pregunta 3) (0,3 punts)

Dibuixeu el graf d'estats que representa el comportament del següent circuit. No oblideu la llegenda i no cal que indiqueu l'estat inicial.



## Pregunta 4) (0,3 punts)

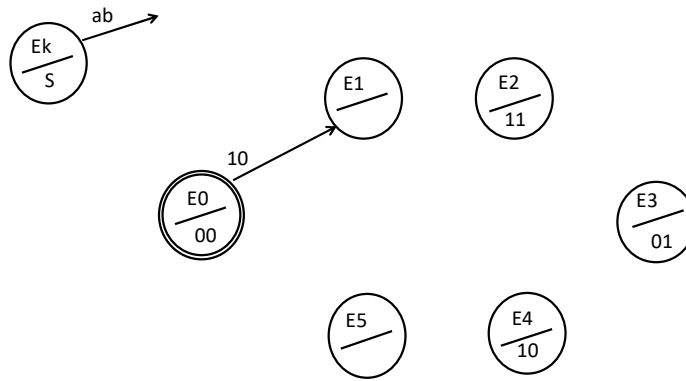
Completeu el graf d'estats d'un circuit seqüencial amb dos entrades d'1 bit  $a$  i  $b$  i una sortida  $S$  de 2 bits amb el següent funcionament:

La sortida  $S$ , de dos bits, ha de valdre 11 durant 1 cicle si per l'entrada  $a$  arriba la seqüència 10, ha de valdre 10 si per l'entrada  $a$  arriba la seqüència 11 i al mateix temps que arriba la seqüència 11 per l'entrada  $b$ . I ha de valdre 01 si per l'entrada  $a$  arriba la seqüència 11 i al mateix temps arriba la seqüència 10 per l'entrada  $b$ . En qualsevol altre cas la sortida  $S$  ha de valdre 00.

El reconeixement de les seqüències s'ha de fer amb encavalcament.

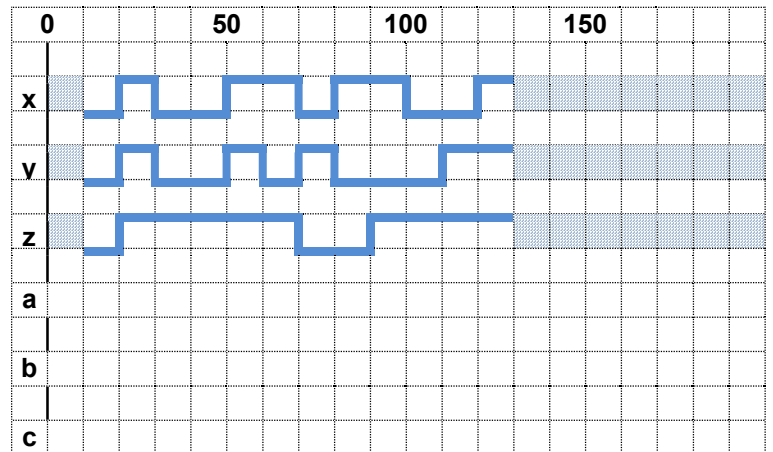
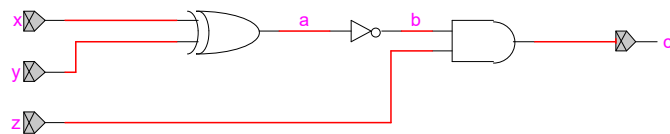
A continuació es mostra un exemple de funcionament del circuit, en que cada columna correspon a un cicle de rellotge:

$a$	0	0	1	0	0	1	1	1	0	1	0	0	0	1	1	1
$b$	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1
$S$	00	00	00	00	11	00	00	10	01	11	00	11	00	00	00	10



### Pregunta 5) (0,3 punts)

Completeu el següent cronograma dels senyals de l'esquema lògic considerant que els temps de propagació de les portes son:  $T_{p(Not)} = 10$ ,  $T_{p(And-2)} = 20$ ,  $T_{p(xor-2)} = 40$  u.t.. Heu d'operar adequadament amb les zones ombrejades (no se sap el valor que tenen) i heu de posar un senyal ombrejat quan no sabeu si val 0 o 1.



### Pregunta 6) (0,4 punts)

Dibuixeu el mapa de Karnaugh amb les agrupacions adequades per obtenir l'expressió mínima en suma de productes de la funció  $w$  d'un circuit al que li correspon la següent taula de veritat:

a	b	c	d	w
0	0	0	0	0
0	0	0	1	0
0	0	1	0	x
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	x
1	0	1	1	x
1	1	0	0	x
1	1	0	1	1
1	1	1	0	x
1	1	1	1	0

- Dibuixeu el Mapa de Karnaugh on es vegi clarament els grups que heu escollit. (0,2 punts)
- Indiqueu l'expressió mínima en suma de productes de  $w$  (0,1 punts)
- Si implementeu la funció  $w$  amb una ROM. Quantes paraules i de Quants bits té la ROM ? (0,1 punts)

### Pregunta 7) (0,6 punts)

Completeu la següent taula assemblant les instruccions en ensamblador SISA o desassemblant les instruccions en llenguatge màquina segons sigui necessari. Poseu NA a la casella en els casos en el que la instrucció no sigui de llenguatge SISA. Indiqueu quins canvis es produeixen en l'estat del computador SISC Von Neumann després d'executar cada una de les instruccions de la taula suposant que abans d'executar-se cada una d'elles el PC val 0xCAFE, el contingut de tots el registres és 0xF0FA, el contingut del byte de totes les adreces parells de memòria és 0x24 i el de totes les imparells és 0x91, i el contingut de tots els ports d'entrada és 1 i el dels ports de sortida 2. Useu la notació  $MEMw[0x..]=0x..$  per indicar qualsevol canvi en la memòria,  $PORTIN[0x..]=0x$  per canvis als ports d'entrada i  $PORTOUT[0x..]=0x..$  per canvis als ports de sortida.

Llenguatge màquina SISA	Llenguatge ensamblador SISA	Canvis a l'estat del computador
0x0456		
	LDB R1, -3 (R6)	
0x87FA		

**Pregunta 8) (1,2 punts)**

Es vol dissenyar un PPE que controli el funcionament d'un semàfor d'un pas de vianants, on els vianants han de prémer un botó per poder passar. El circuit té una entrada **B** que es posa a 1 durant un cicle quan un vianant prem el botó per demanar pas, i una sortida **R** que ha de valer 1 quan el semàfor ha d'estar Roig i 0 quan ha d'estar Verd.

El semàfor, habitualment, està en Roig i només canvia de color quan arriba un primer 1 per l'entrada **B** des de que el semàfor està a Roig.

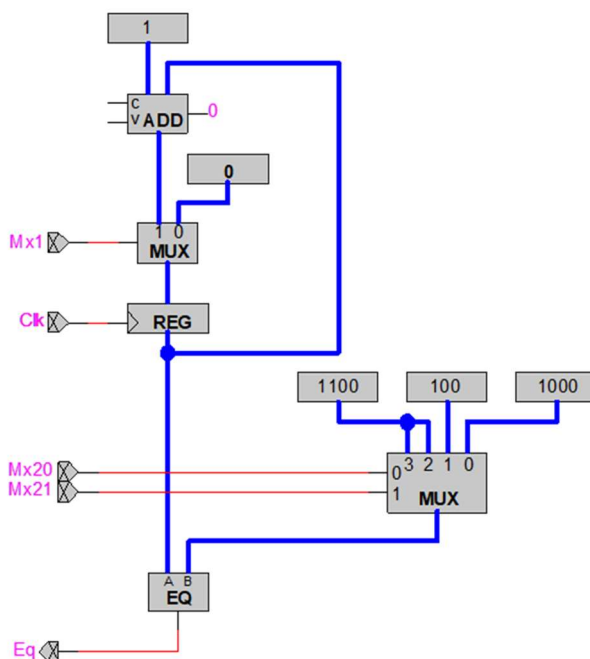
Quan arriba aquest primer 1 per l'entrada **B**, si el semàfor porta 1000 o més cicles a Roig, continua a Roig durant 100 cicles més i, a continuació, es posa a Verd durant 1000 cicles, passant després a la situació habitual, Roig. Si el semàfor encara no porta 1000 cicles a Roig, continua a Roig fins arribar als 1100 cicles, passant després 1000 cicles a Verd.

Només es considera el senyal **B** la primera vegada que val 1 cada vegada que el semàfor es posa a Roig. Quan està a Verd no es considera.

Pot ser que cap vianant premi el boto **B** durant milions de cicles.

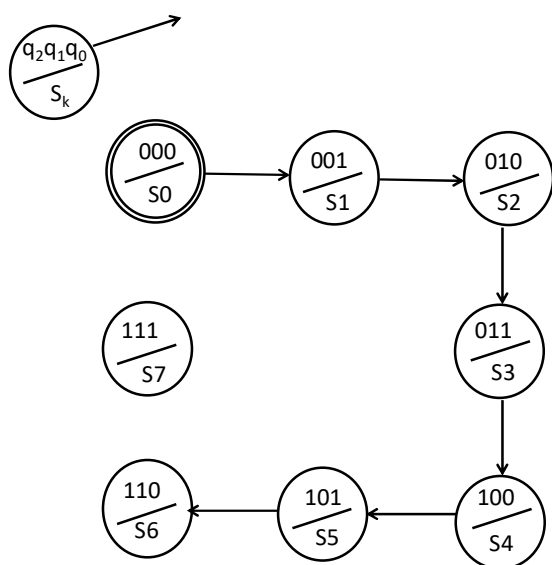
Les constants 100, 1000 i 1100 son nombres decimals.

A continuació teniu l'esquema complet de la Unitat de Procés del PPE i el graf d'estats incomplet de la Unitat de Control del PPE.



Graf incomplet de la Unitat de Control:

Taula de sortides



$S_k$	Mx1	Mx21	Mx20	R
S0				
S1				
S2				
S3				
S4				
S5				
S6				
S7				

a) Indiqueu la mida de la ROM que implementa, conjuntament amb un multiplexor de busos, l' UC (0,2 punts).

Bits d'entrada=

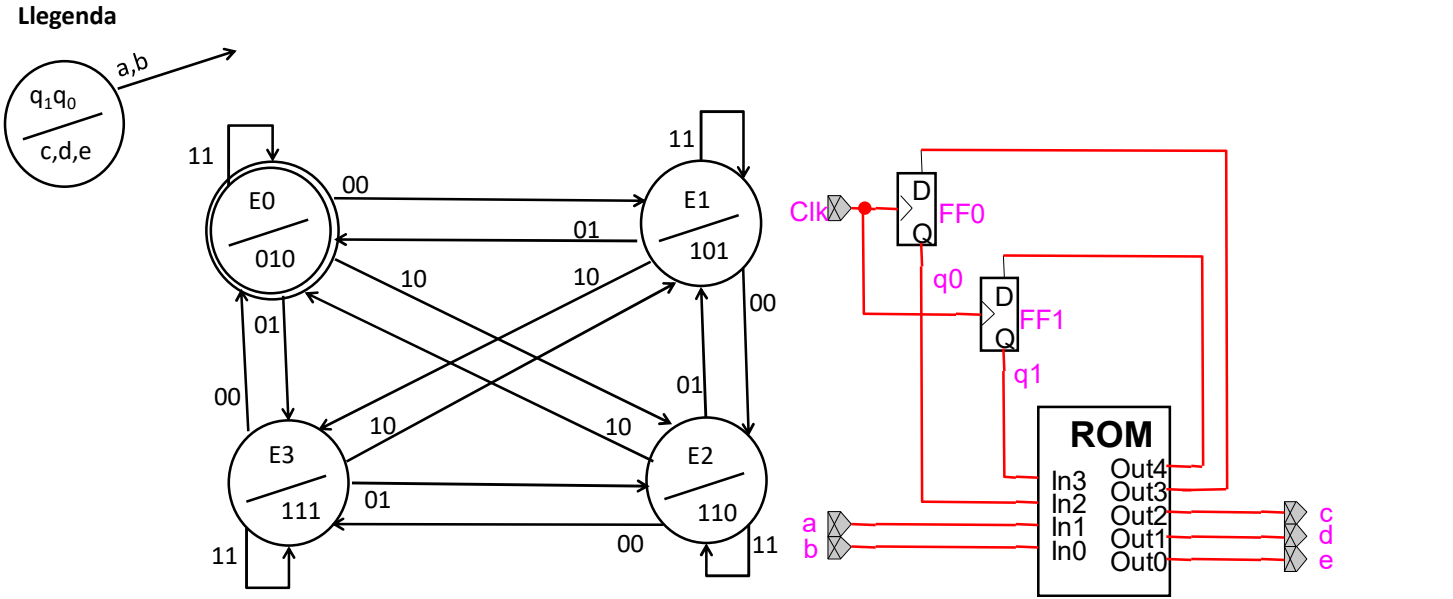
Bits de sortida=

Mida de la ROM (en bits)=

b) Completeu el graf d'estats d l'UC, en el que li falten arcs i etiquetes dels arcs, així com el valor de les sortides de cada node, que heu d'indicar a la taula. No oblideu la llegenda. (1 punt)

**Pregunta 9) (0,2 punts)**

Si es vol implementar el graf d'estats següent amb un circuit seqüencial amb una sola ROM tal com es mostra en la figura. L'estat "Ei" es codifica amb el natural "i".

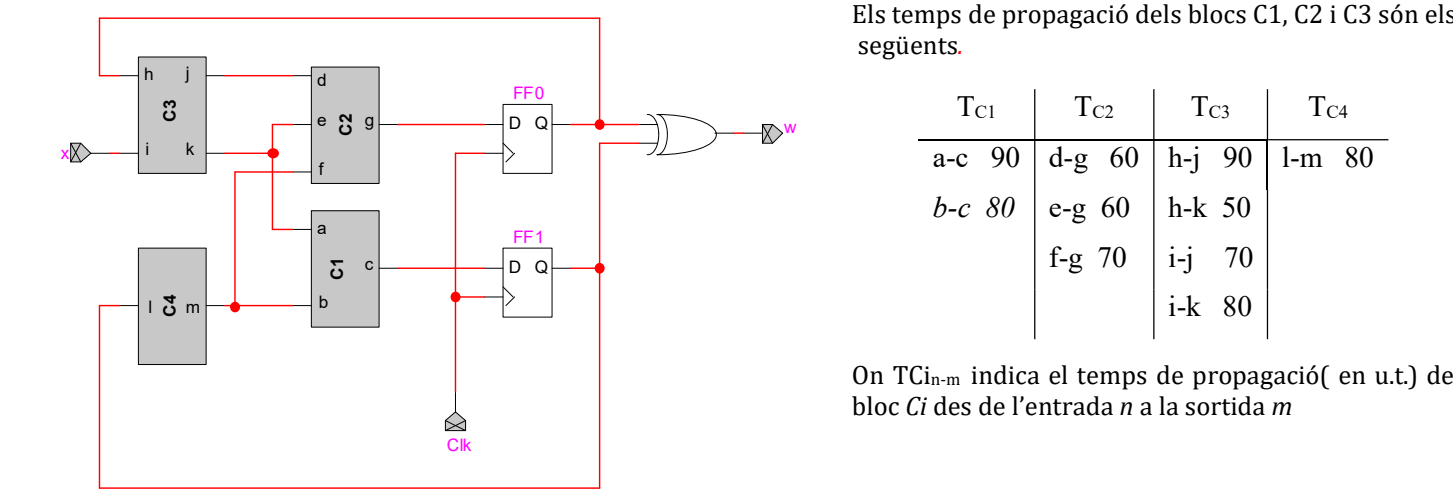


Indiqueu en hexadecimal el contingut de les següents adreces de la ROM:

ROM[0x3]=0x	ROM[0xB]=0x	ROM[0xF]=0x
-------------	-------------	-------------

**Pregunta 10) (0,2 punts)**

El següent circuit seqüencial forma part d'un sistema més complex. Se sap que l'entrada *x* s'estabilitza passades 80 u.t. des de l'arribada del flanc ascendent de rellotge i que la sortida *w* ha d'estar estable 60 u.t. abans de l'arribada del flanc ascendent del rellotge. Se sap que el camí crític passa per aquest circuit i que els blocs C1, C2,C3 i C4 són blocs combinacionals dels que no coneixem la seva implementació interna. El temps de propagació de cada biestable és 100 u.t. i el Tp de la porta XOR és 50 u.t.



Es demana:

- Escriure tots esl camins crítics que hi hagi, llistant els blocs i portes per les que passa i indicant en cada bloc l'entrada i sortida per les que passa el camí. (0,1 punts)
- Quin és el temps de cicle mínim ?(0,1 punts)

**Pregunta 11) (0,6 punts)**

Indiqueu el valor que ha de tenir cada un dels bits de la paraula de control de l'UPG (sense subsistema d'I/O, ni memòria, veure "xuletari") perquè es realitzi, durant un cycle, l'acció concreta especificada amb el mnemotècnic. Poseu una x a les caselles que el valor **no importa** per l'execució de la instrucció. En el cas de que no es pugui realitzar l'acció heu de tatxar tota la línia de senyals.

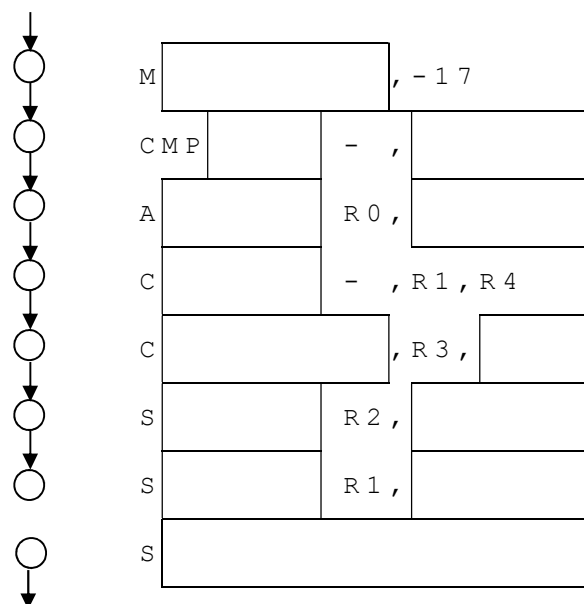
Mnemotècnic	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
BZ R1,-6									
OUT R1 // ADDI R2,R1,-3									
XOR -,R2,R3									

**Pregunta 12) (0,6 punts)**

Completeu el fragment del graf d'estats de l'UC de **propòsit específic** perquè junt amb l'UPG formin un processador que faci la funcionalitat descrita mitjançant el següent codi en C (el codi no té perquè fer res útil). Indiqueu els arcs que falten, les etiquetes dels arcs (z, !z, o res) i completeu les caselles de cada paraula de control que s'especifica amb mnemotècnics a la dreta de cada node del graf. A les comparacions, heu d'interpretar les dades com valors **enters**.

```
for (R0=-17; R0 > 128; R0++) {
    while ((R1>R4) && (R3<R2)) {
        R2=R2-R3;
        R1=R1-1;
    }
}
R2=R2/8;
```

Nota: el símbol && correspon a l'AND lògica (booleana).

**Pregunta 13) (0,7 punts)**

Completeu el fragment de programa en llenguatge ensamblador SISA, perquè el processador format per la unitat de control de propòsit general (UCG) junt amb l'UPG faci la funcionalitat descrita mitjançant el següent codi en C (el codi no té perquè fer res útil). El codi SISA només ha d'escriure en els registres R4, R5, R6 per implementar la funcionalitat descrita) i a R7 per valors temporals. A les comparacions heu d'interpretar les dades com valors **enters**. Ompliu la part que falta. No podeu usar etiquetes pels salts.

```
if (R6 >= -160) {
    while (R4 <= 50) {
        R5 = (R5 - 1) - R3;
        R4 = R5;
    }
} else {
    R6 = R4 + R4
}
R5 = R6 / 8;
```

