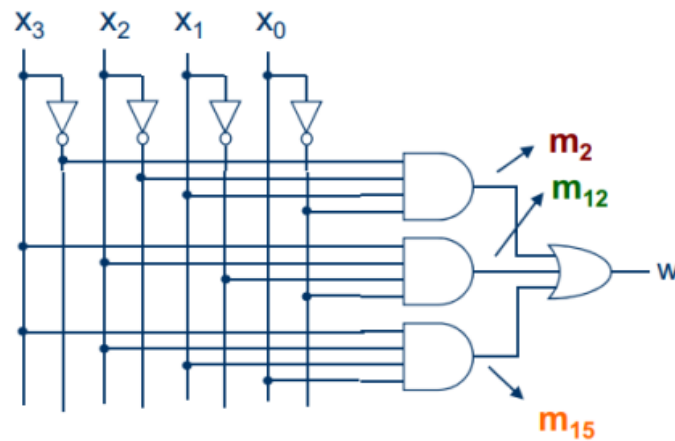


Exemple: 4 entrades i 1 sortida

x_3	x_2	x_1	x_0	w
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

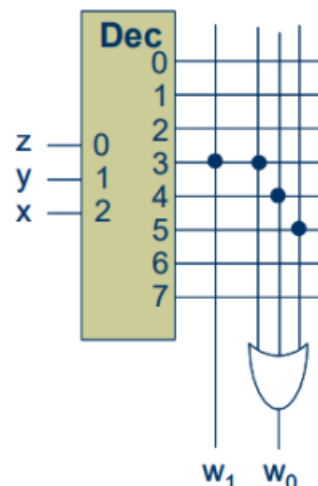


- $w = m_2 + m_{12} + m_{15} = \neg x_3 \cdot \neg x_2 \cdot x_1 \cdot \neg x_0 + x_3 \cdot x_2 \cdot \neg x_1 \cdot \neg x_0 + x_3 \cdot x_2 \cdot x_1 \cdot x_0$
- El CLC utilitza 4 portes NOT, 3 portes AND-4 i 1 porta OR-3
- Què faríeu si a la TV hi hagués alguna sortida a X (*Don't care*)?

Exemple: Síntesi CLC amb dues sortides

- L'ordre de les variables d'entrada a la TV (xyz) ha de correspondre's amb el pes de les entrades al bloc Dec-3-8
 - x amb pes 2, y amb pes 1, z amb pes 0
- Sortides:
 - Com la sortida w_1 només utilitza un *minterm*, no cal porta OR per sintetitzar-lo
 - Per sintetitzar w_0 cal una porta OR-3

x	y	z	w_1	w_0
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	0	0

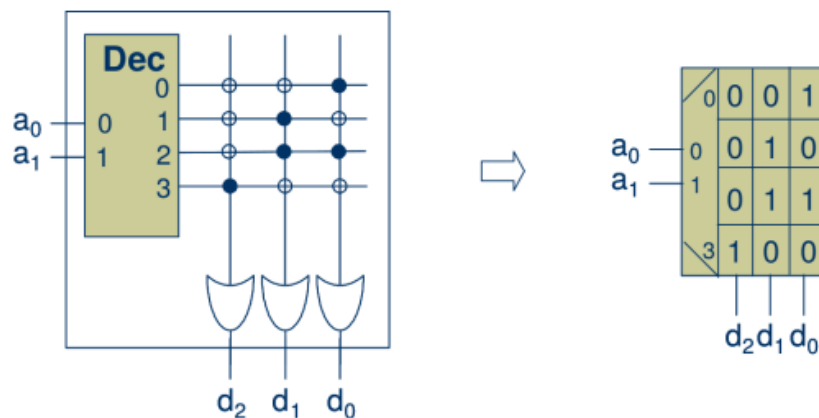


Exemple 1 (n=2, m=3)

- Sintetitzar un CLC amb 2 entrades i 3 sortides amb la següent TV

a_1	a_0	d_2	d_1	d_0
0	0	0	0	1
0	1	0	1	0
1	0	0	1	1
1	1	1	0	0

- Mostrem el connexionat així com la representació compacta
 - Vigileu amb els pesos dels senyals i l'ordre de les files!
 - A la representació compacta incloem la taula de veritat del CLC



Navigation icons: back, forward, search, etc.

Exercici 1617Q1-E1

a) ¿Cuántas puertas And y Or y de cuántas entradas cada una hacen falta para implementar directamente la expresión en suma de minterms de la función w de la siguiente tabla de verdad

a	b	c	w
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Número puertas AND = de entradas.

Número puertas OR = de entradas.

b) Especificar el tamaño mínimo de la ROM para sintetizar un circuito de 4 entradas y 3 salidas.

Número de palabras =

Bits por palabra =

Dado el esquema del siguiente circuito (incluida la tabla de verdad del bloque C1) completad la tabla de verdad de la salida W y escribid la expresión lógica en suma de minterms.

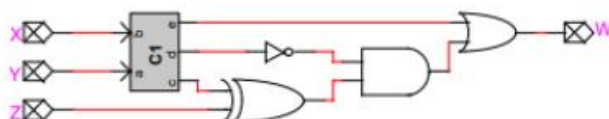


Tabla de verdad de W:

X	Y	Z	W
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Tabla de verdad de C1

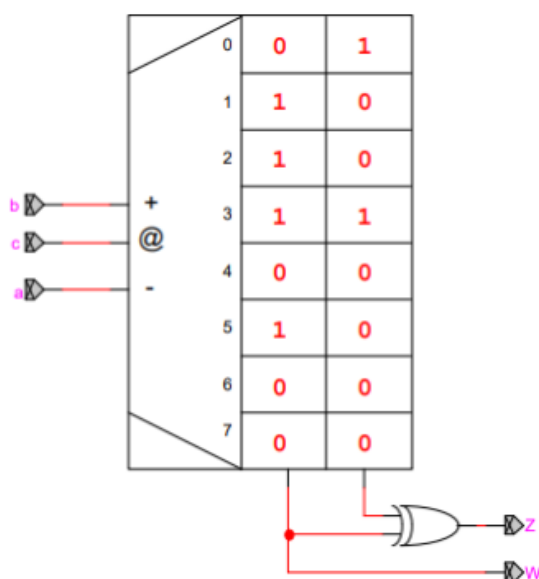
a	b	c	d	e
0	0	1	1	0
0	1	0	0	0
1	0	1	0	1
1	1	1	1	0

Expresión en suma de minterms de W: $!X \cdot Y \cdot !Z + !X \cdot Y \cdot Z + X \cdot !Y \cdot Z$

- Queremos implementar un circuito combinacional cuya función/comportamiento está descrito en la siguiente tabla de verdad. Para implementarlo ya tenemos el circuito diseñado. Rellena el contenido de la ROM del circuito para que haga la función descrita en la tabla de verdad

Tabla de verdad

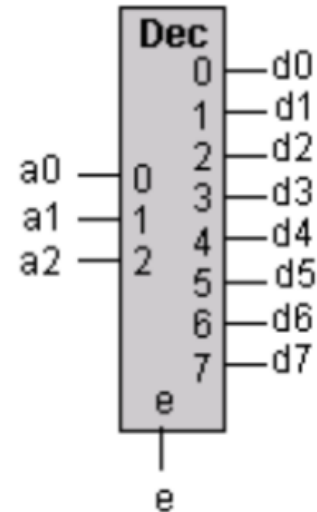
a	b	c	Z	W
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	0



- Afegeix una entrada extra al Dec- $n-2^n$, la senyal *enable*
 - Si val 0, totes les sortides del Decoder valdran 0
 - Si val 1, el Decoder operarà amb normalitat
- Exemple: Dec-3-8 amb senyal *enable*

Truth Table (compressed):

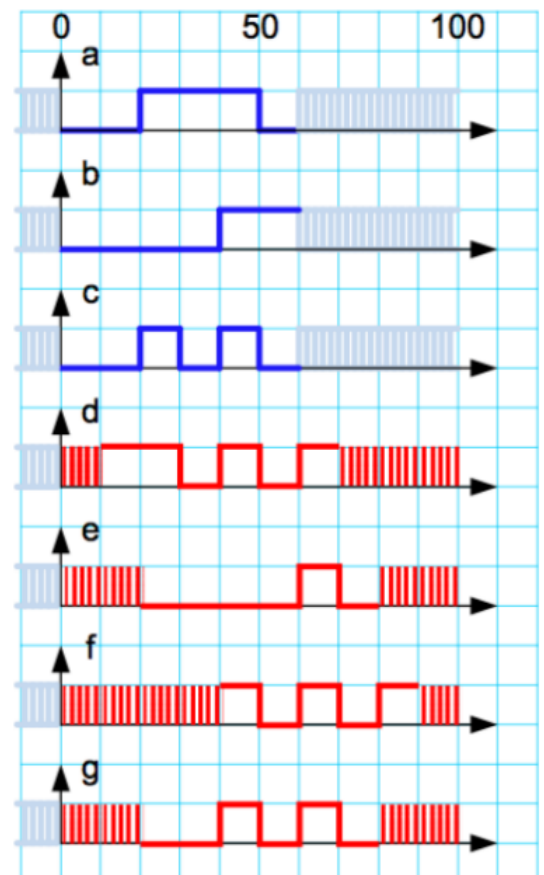
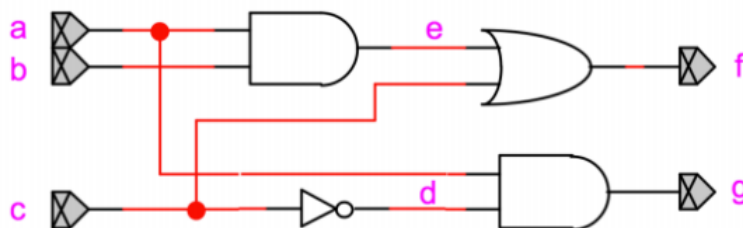
e	a2	a1	a0	d7	d6	d5	d4	d3	d2	d1	d0
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	1	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0



Navigation icons: back, forward, search, etc.

Exercici 1415Q1 E1

Completad el siguiente cronograma de las señales del esquema lógico sabiendo que los tiempos de propagación de las puertas son: $T_p(\text{Not}) = 10$ u.t., $T_p(\text{And}) = T_p(\text{Or}) = 20$ u.t. Debéis operar adecuadamente con las zonas sombreadas (no se sabe el valor que tienen) y dibujar la señal sombreada cuando no se pueda saber si vale 0 o 1.



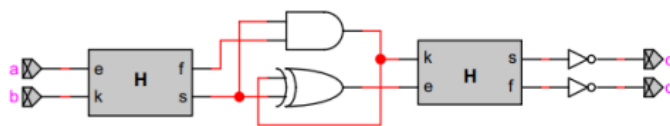
Navigation icons: back, forward, search, etc.

Dado el esquema del siguiente circuito (incluida la tabla de verdad del bloque H),

- Completad la tabla de verdad de las salidas c y d y escribid la expresión lógica en suma de minterms de c. (1 punto)
- Escribid el camino crítico (o uno de ellos si hay varios) y el tiempo de propagación desde la entrada a hasta la salida c. Se dan los tiempos de propagación de H (en la tabla) y de las puertas: $T_p(\text{Not}) = 10$, $T_p(\text{And}) = 20$, $T_p(\text{Or}) = 30$ y $T_p(\text{Xor}) = 50$ u.t. Por ejemplo, uno de los caminos de b a d se especificaría como: b - k - s - Xor - e - f - Not - d. (1 punto)

T.V. bloque H

e	k	f	s
0	0	1	1
0	1	1	0
1	0	0	1
1	1	1	1



a	b	c	d
0	0	1	0
0	1	0	0
1	0	0	1
1	1	1	0

T_p bloque H

T_p	f	s
e	60	50
k	90	80

Expresión en suma de minterms de c: $\neg a \cdot \neg b + ab$

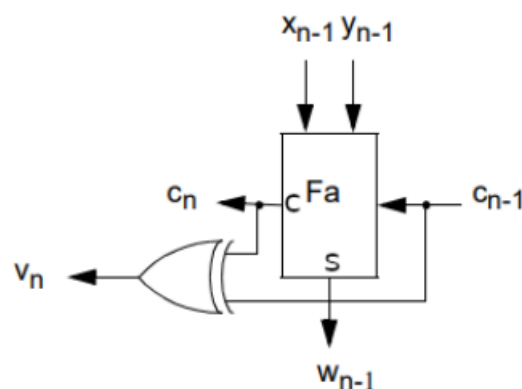
Camino crítico de a a c: a - e - f - And - Xor - e - s - Not - c

$T_{p_{a-c}}: 190$ u.t.

No representativitat a la suma d'enters

- Afegirem una nova sortida, v_n , al bloc ADD
 - $v_n = 1$ indicarà que el resultat de la suma d'enters amb aritmètica convencional no és representable en Ca2 de n bits
 - Cal detectar si els operands són del mateix signe però el resultat és de signe diferent
 - Els operands són positius però el resultat és negatiu: $\neg x_{n-1} \cdot \neg y_{n-1} \cdot w_{n-1}$
 - Els operands són negatius però el resultat és positiu: $x_{n-1} \cdot y_{n-1} \cdot \neg w_{n-1}$
 - $v_n = x_{n-1} \cdot y_{n-1} \cdot \neg w_{n-1} + \neg x_{n-1} \cdot \neg y_{n-1} \cdot w_{n-1}$
 - Amb la TV es pot demostrar que v_n és equivalent a $c_n \text{ XOR } c_{n-1}$

x_{n-1}	y_{n-1}	c_{n-1}	c_n	w_{n-1}	v_n
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

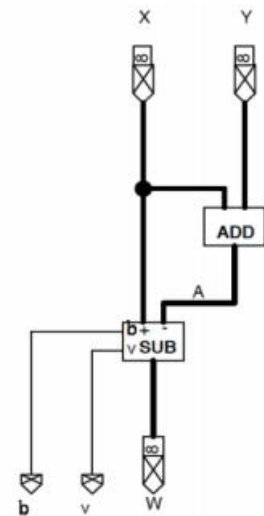


Exercici 1516Q1E2



Dado el siguiente circuito CLC y los vectores de 8 bits de entrada X y Y, completad la siguiente tabla.

X	Y	A	W	Wu	Ws	b	v
00000000	11111111	11111111	00000001	1	1	1	0
11111111	00000001	00000000	11111111	255	-1	0	0
10000000	10101010	00101010	01010110	86	86	0	1

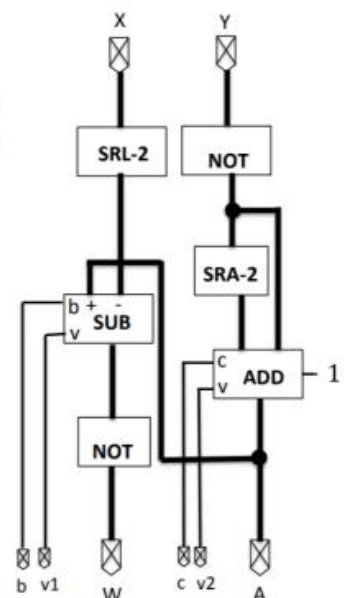


Exercici 1819Q1E2



Dado el esquema del CLC a bloques de la derecha, completad la siguiente tabla que indica el valor de las salidas del circuito para cada uno de los cuatro casos concretos de valores de las entradas (un caso por fila).

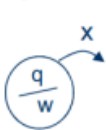
X	Y	b	v1	c	v2	W	A
0000 0000	1111 1111	0	0	0	0	11111110	00000001
1000 0001	1111 1100	1	0	0	0	00011011	00000100
1000 0000	0111 1111	0	0	1	1	10111110	01100001
1111 1111	1000 0000	0	1	0	1	10011111	10011111



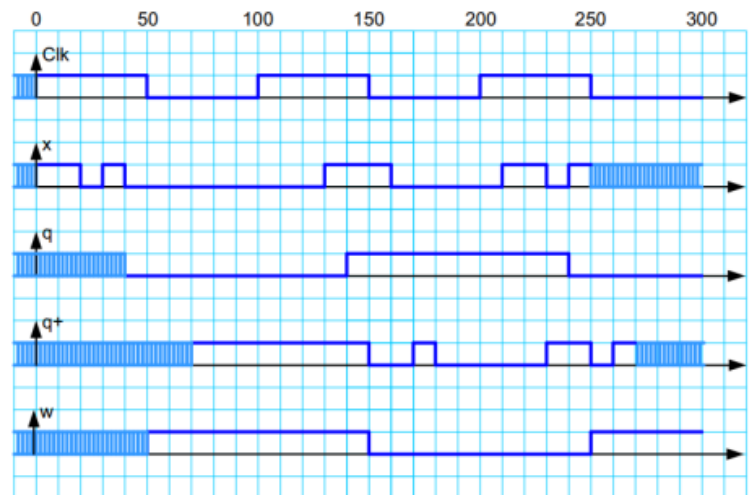
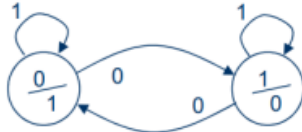
- Per fer el graf d'estats caldria construir les taules de veritat del CLC que calcula les sortides i del CLC que actualitza l'estat

Solució:

Leyenda:

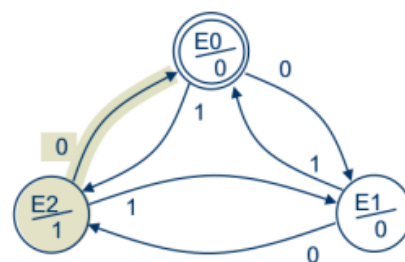
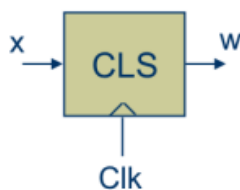


Grafo:



Seguiment del graf d'estats: exercici 1

- A partir del graf d'estats, podem completar el cronograma simplificat que mostra l'evolució de l'estat i de la sortida a mesura que l'entrada canvia de valor.
 - Exemple on $n = 1, m = 1, k = 2$

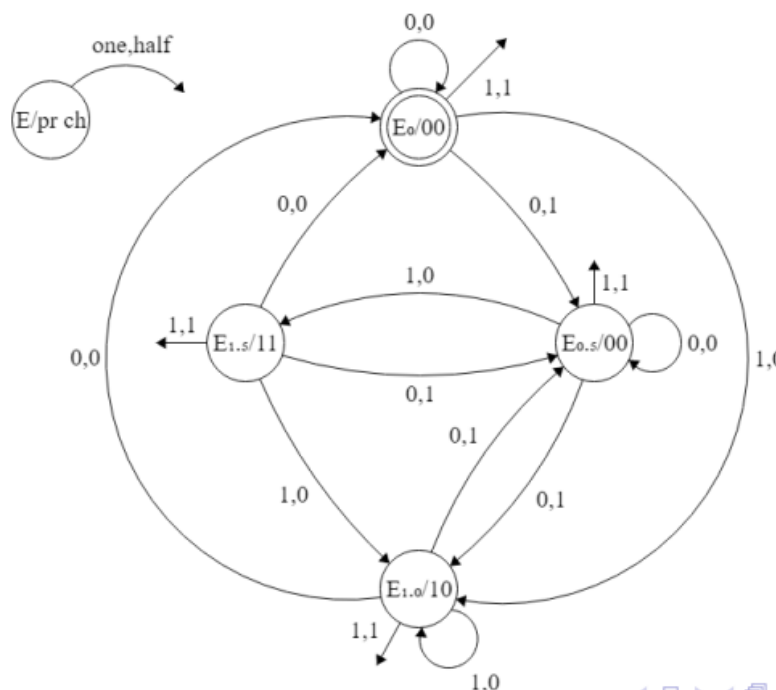


Núm. Ciclo	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
Estado	E0	E2	E0	E1	E2	E0	E1	E0	E2	E1	E2	E0	E2	E0	E2
Entrada x	1	0	0	0	0	0	1	1	1	0	0	1	0	1	0
Salida w	0	1	0	0	1	0	0	0	1	0	1	0	1	0	1

- Dibuixeu el graf d'estats d'un CLS corresponent al mecanisme de control d'una màquina de venda de productes de 1€ que admeti monedes de 1€ i de 0,5€ tal que:
 - Entrades:
 - *half*: val 1 si en aquest cicle s'ha introduït una moneda de 0,50€
 - *one*: val 1 si en aquest cicle s'ha introduït una moneda de 1€
 - Podeu assumir que *one* i *half* no poden valer 1 simultàniament
 - Sortides:
 - *product* (pr): valdrà 1 durant un cicle si s'ha rebut un mínim de 1€
 - *change* (ch): valdrà 1 durant un cicle si cal retornar canvi (0,50€)
 - Un cop que el CLS notifica que cal servir un producte (i, si cal, retornar canvi), el CLS es disposarà a rebre monedes per a un nou producte.
- Orientacions:
 - Determineu quins estats necessitaríeu i les transicions entre ells en funció de les entrades rebudes
 - Amb n senyals d'entrada, de cada node han de sortir 2^n arestes
 - No oblideu dibuixar la llegenda i determinar l'estat inicial

Exemple: Màquina de venda

- Nombre d'estats: 4
 - Un per cada quantitat acumulada possible abans de servir el producte
 - 0€, 0.5€, 1.0€ i 1.5€
- Com $one = 1$ i $half = 1$ és impossible, les arestes no porten enlloc



Construir graf d'estats (E2 Q2 18/19)



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:

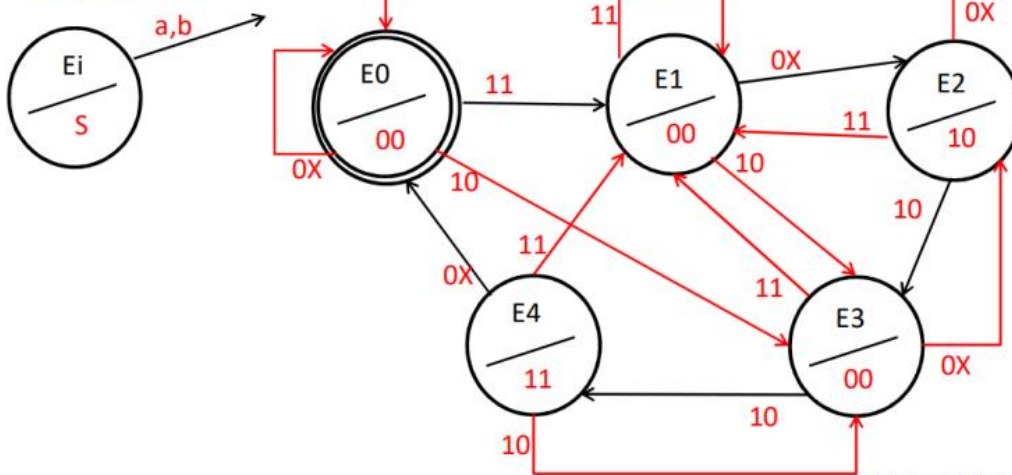
La sortida **S**, de dos bits, ha de valdre 10 durant 1 cicle si per l'entrada **a** arriba la seqüència 10, i ha de valdre 11 si per l'entrada **a** arriba la seqüència 11 al mateix temps que arriba la seqüència 00 per l'entrada **b**. En qualsevol altre cas la sortida **S** ha de valdre 00.

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

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

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

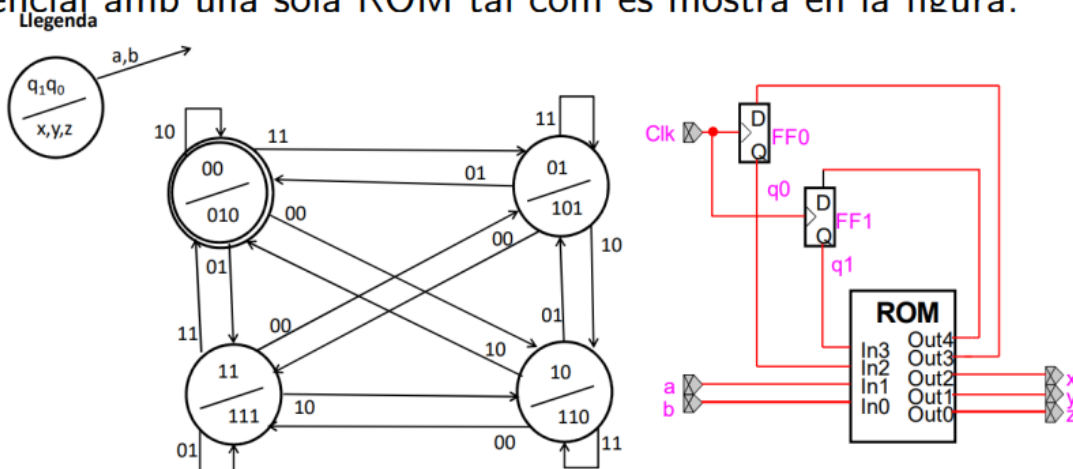
Llegenda



Síntesi CLS's (E2 Q2 18/19)

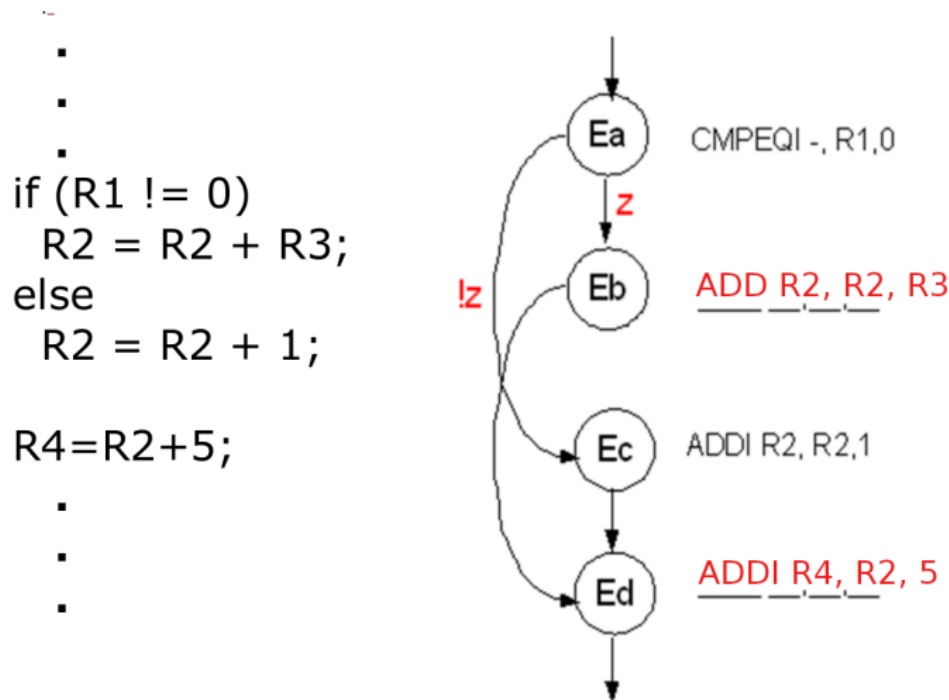


- Si es vol implementar el graf d'estats següent amb un circuit seqüencial amb una sola ROM tal com es mostra en la figura.

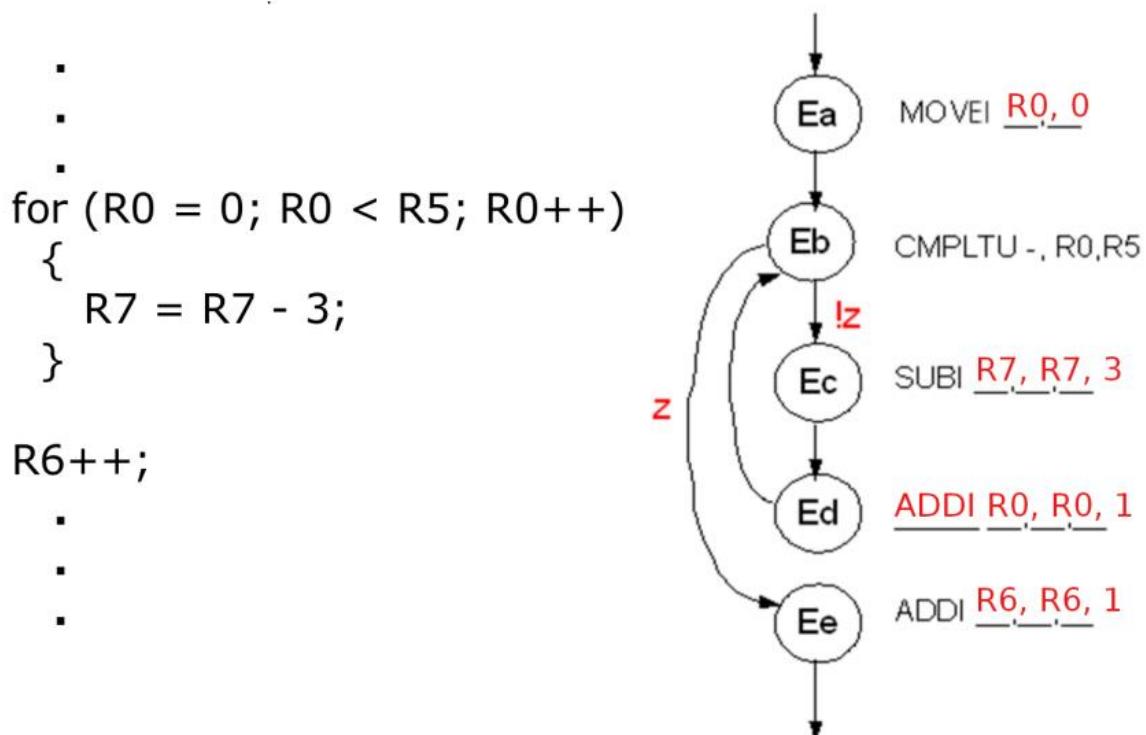


- Indiqueu en hexa el contingut de les següents adreces de la ROM:
 - ROM[0x4]=0x1D
 - ROM[0xA]=0x06
 - ROM[0xE]=0x17

- Completeu els mnemotècnics i les etiquetes z/!z al graf d'estats:



- Completeu els mnemotècnics i les etiquetes z/!z al graf d'estats:



Exercici 8.5d

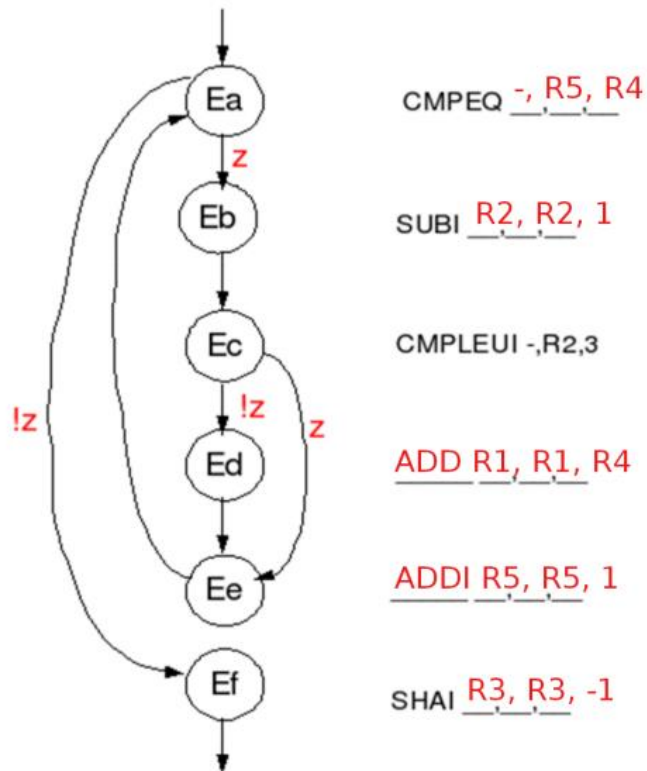


- Completeu els mnemotècnics i les etiquetes z/!z al graf d'estats:

```

while (R5 != R4)
{
    R2 = R2 - 1;
    if (R2 <= 3) R1 = R1 + R4;
    R5++;
}
R3 = R3 / 2;

```



Parcial E3 Q1 16-17

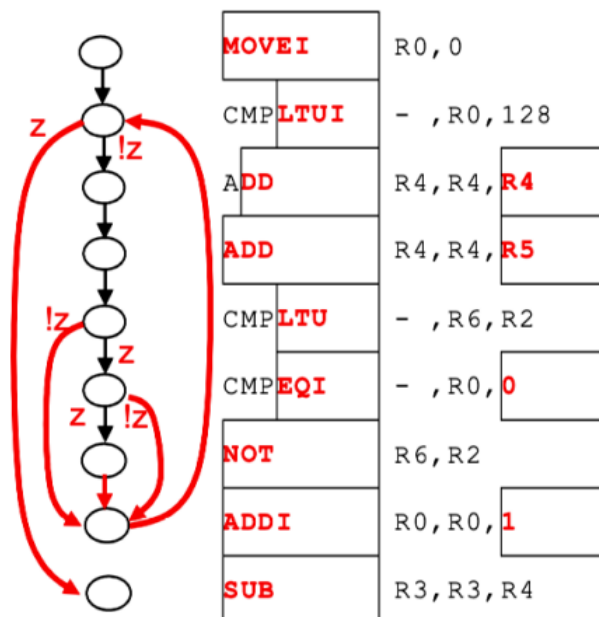


- Completeu (arcs, etiquetes i mnemotècnics) el fragment graf d'estats de la UC perquè, juntament amb la UPG, implementi el fragment de codi indicat.
- Les dades són de tipus natural.

```

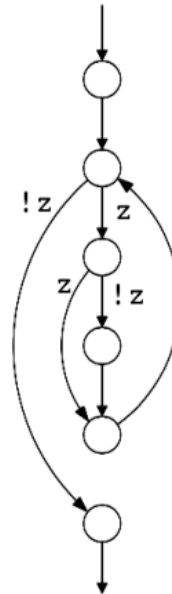
for (R0=0; R0<128; R0++) {
    R4=R4*2+R5;
    if ((R6>=R2) && (R0!=0))
        R6=not(R2)
}
R3=R3-R4;

```



- Completeu (arcs, etiquetes i mnemotècnics) el fragment graf d'estats de la UC perquè, juntament amb la UPG, implementi el fragment de codi indicat.
- Les dades són de tipus enter.

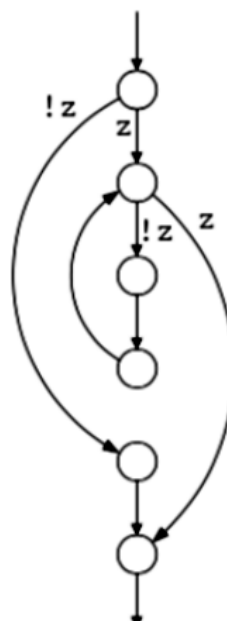
```
for(R1=133; R1>-133; R1--) {
    // R1<0> es el bit
    // de menys pes de R1
    if (R1<0>==1)
        R2=R2+R1
}
R2=R2/4;
```



MOVEI	R1,	133
CMPLEI	-, R1,	-133
ANDI	-, R1,	1
ADD	R2,	R2, R1
SUBI	R1,	R1, 1
SHAI	R2,	R2, -2

- Completeu (arcs, etiquetes i mnemotècnics) el fragment graf d'estats de la UC perquè, juntament amb la UPG, implementi el fragment de codi indicat.
- Les dades són de tipus natural.

```
if (R3 > 50) {
    while (R0 >= R2) {
        R0 = R0 / 2;
        R3 = R3 - 1;
    }
} else {
    // | es OR bit a bit
    R3 = R2 | R3;
}
R4 = R0 + R3;
```



CMPLEUI	-, R3,	50
CMPLEU	-, R2,	R0
SHLI	R0,	R0, -1
SUBI	R3,	R3, 1
OR	R3,	R2, R3
ADD	R4,	R0, R3