Universitat de GIRONA
Estudis d'ETIG/ETIS. **Curs: 1er** **2005/2006** **10 de Febrer de 2006**
Estuctura i Tecnologia de Computadors **Examen parcial** **Grup** Matí / Tarda

Nom i cognoms: ....

PROVA DE PROBLEMES

Temps: 2h 15m. Obtenint 3 o més punts sobre 10, la prova puntuarà el 50% del total de l'examen. La puntuació està posada al costat de cada exercici. Cal fer cada problema en fulls a part. Les notes es publicaran a partir del dia 20. La revisió de l'examen es farà dimarts dia 21 a l'aula PI-01 a les 12.

1. <u>ÀLGEBRA DE BOOLE</u>. (2 Punts).

a. Simplifiqueu, tant com es pugui, la funció: **(1 Punt)**

$$f(a,b,c,d)=\overline{((a\overline{(b+c)}+\bar{b})(a\overline{(b+c)}+\overline{(c+d)})+\bar{a}\bar{b}\bar{d})}$$

$$f(a,b,c,d)=\overline{((a\overline{(\boldsymbol{b+c})}+\bar{b})(a\overline{(\boldsymbol{b+c})}+\overline{(\boldsymbol{c+d})})+\bar{a}\bar{b}\bar{d})} \quad \text{[de Morgan]}$$

$$f(a,b,c,d)=\overline{((\boldsymbol{a\bar{b}\bar{c}+\bar{b}})(a\bar{b}\bar{c}+\bar{c}\bar{d})+\bar{a}\bar{b}\bar{d})} \quad \text{[Simplificació Teoremes]}$$

$$f(a,b,c,d)=\overline{((\boldsymbol{\bar{b}(a\bar{b}\bar{c}+\bar{c}\bar{d})})+\bar{a}\bar{b}\bar{d})} \quad \text{[Distributiva]}$$

$$f(a,b,c,d)=\overline{(\boldsymbol{a\bar{b}\bar{c}+\bar{b}\bar{c}\bar{d}+\bar{a}\bar{b}\bar{d}})} \quad \text{[de Morgan]}$$

$$f(a,b,c,d)=(\bar{a}+b+c)(b+c+d)(a+b+d)$$

Per acabar de simplificar al màxim es pot utilitzar *Karnaugh,* però cal acabar de modificar l'expressió per arribar a, per exemple, un producte de màxterms.

$$f(a,b,c,d)=(\boldsymbol{\bar{a}+b+c})(b+c+d)(a+b+d)$$

$$f(a,b,c,d)=(\bar{a}+b+c+d)(\bar{a}+b+c+\bar{d})(\boldsymbol{b+c+d})(a+b+d)$$

$$f(a,b,c,d)=(\bar{a}+b+c+d)(\bar{a}+b+c+\bar{d})(a+b+c+d)(\bar{a}+b+c+d)(\boldsymbol{a+b+d})$$

$$f(a,b,c,d)=(\bar{a}+b+c+d)(\bar{a}+b+c+\bar{d})(a+b+c+d)(\bar{a}+b+c+d)(a+b+c+d)(a+b+\bar{c}+d)$$

Ara la funció ja es troba en producte de màxterms:

$$f(a,b,c,d)=\prod\nolimits_4(0,2,8,9)$$

Es pot simplificar fàcilment per *Karnaugh* agafant els zeros o els uns:

c d \ a b	***00***	***01***	***11***	***10***
00	**0** 0	**1** 4	**1** 12	**0** 8
01	**1** 1	**1** 5	**1** 13	**0** 9
11	**1** 3	**1** 7	**1** 15	**1** 11
10	**0** 2	**1** 6	**1** 14	**1** 10

Agafant els zeros (màxterms):

$$\boxed{\boldsymbol{f(a,b,c,d)=(\bar{a}+b+c)(a+b+d)}}$$

Agafant els uns (mínterms):

$$\boxed{\boldsymbol{f(a,b,c,d)=b+(a\,c)+(\bar{a}\,d)}}$$

b. Escriviu la següent funció amb el mínim número de portes NOR de <u>dues entrades</u>. Quantes són necessàries? **(1 Punt)**

$$f(a,b,c,d)=\sum\nolimits_{4}(3,5,6,7,9,11)+d(1,4,12,13)$$

Per obtenir portes NOR de la forma més directa a partir de l'expressió en suma de mínterms, es pot simplificar per *Karnaugh* agafant els zeros per obtenir l'expressió simplificada en producte de sumes.

c d \ a b	*00*	*01*	*11*	*10*
00	**0** 0	**X** 4	**X** 12	**0** 8
01	**X** 1	**1** 5	**X** 13	**1** 9
11	**1** 3	**1** 7	**0** 15	**1** 11
10	**0** 2	**1** 6	**0** 14	**0** 10

$$\boxed{f(a,b,c,d)=(b+d)(\bar{a}+\bar{b})}$$

Ara cal eliminar la AND entre els termes aplicant una doble negació a tota la funció:

$$f(a,b,c,d)=\overline{\overline{((b+d)(\bar{a}+\bar{b}))}}$$

$$\boxed{f(a,b,c,d)=\overline{(\overline{(b+d)}+\overline{(\bar{a}+\bar{b})})}}$$

El total de NORs que es necessiten és 5: dues NOR per implementar cada un dels inversors, les dues NORs de cada terme i la NOR total. Totes són de dues entrades.

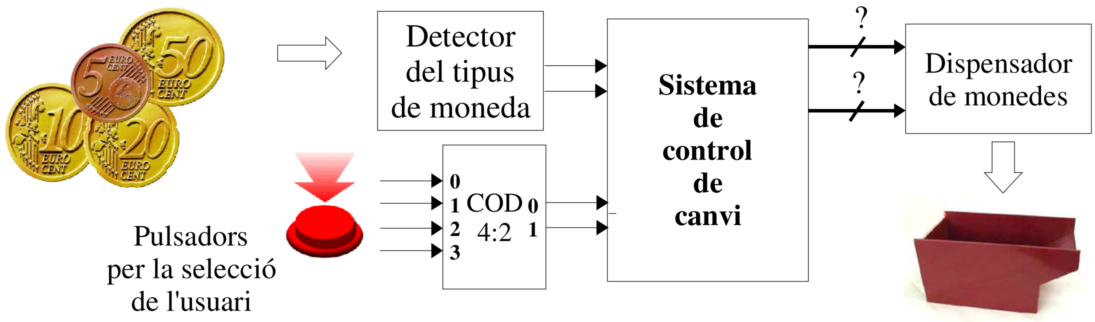
2. <u>SISTEMES COMBINACIONALS</u>. La màquina del canvi. (3 Punts).

Ens han encarregat que dissenyem i implementem el mecanisme de control d'una màquina de canvi de monedes. Aquesta màquina permet, per part de l'usuari, l'entrada d'<u>una sola moneda</u> i escollir amb quin tipus de moneda es vol el canvi. Si l'elecció és correcta retorna el canvi solicitat. En cas contrari retorna <u>una moneda del mateix tipus</u>.

La màquina només treballa amb monedes de 5, 10, 20 i 50 cèntims d'Euro. El sistema disposa d'un detector que proporciona un codi per identificar cada un dels tipus de moneda. A més a més, també disposa d'un codificador que genera el codi de moneda escollida per part de l'usuari a partir de quatre pulsadors.

El sistema de control de canvi que cal implementar ha de controlar un dispensador de monedes que necessita saber <u>el tipus de moneda</u> i la <u>quantitat d'aquest tipus</u> que cal retornar al calaix de la màquina.

<u>Dissenyeu i implementeu el sistema combinacional</u> que controla el mecanisme dispensador de monedes (Sistema de control de canvi) perquè la màquina operi tal i com s'ha especificat.



a. Identifiqueu i codifiqueu les entrades i sortides del sistema i feu la taula de <u>veritat de totes les funcions de sortida</u> **(1 Punt).**

Com que existeixen 4 tipus diferents de moneda, l'usuari només pot entrar una sola moneda d'un dels tipus i demanar canvi només d'un sol tipus de moneda, el *Sistema de control de canvi* serà un sistema combinacional amb 4 entrades:

- Dues línies codificaran el tipus de moneda entrada.
- Dues línies més per codificar el tipus de moneda que cal retornar.

Codi	Tipus de Moneda
00	5 Cèntims
01	10 Cèntims
10	20 Cèntims
11	50 Cèntims

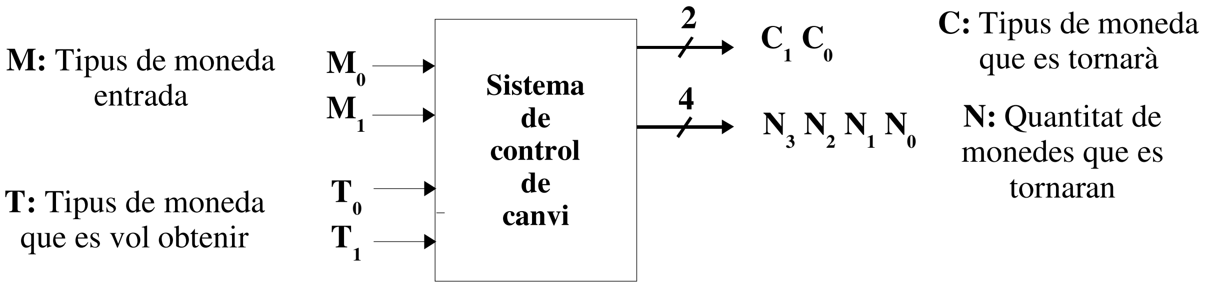
El S*istema de control de canvi* que s'ha d'implementar cal que indiqui al *Dispensador de monedes* de quin tipus i quantes monedes d'aquest tipus s'han de retornar. El tipus de moneda es pot codificar com a la taula anterior, per tant, també caldran 2 bits.
Quant al número de monedes, en el cas pitjor, cal retornar:

50 Cèntims / 5 Cèntims = 10 Monedes

Per codificar un 10 es necessiten:

$$n = \lfloor \ln(10) / \ln(2) \rfloor$$
$$n = \lfloor 2.3025 / 0.6931 \rfloor$$
$$\mathbf{n = 4\ bits}$$

Per tant el sistema es pot formalitzar en el següent bloc lògic combinacional:



Cal dissenyar i implementar un sistema format per 2 + 4 = 6 funcions de sortida de 4 variables cada una. La taula de veritat d'aquest sistema és la següent:

	M	$M_1 M_0$	*T*	$T_1 T_0$	*C*	C_1	C_0	*N*	N_3	N_2	N_1	N_0
0	*5*	**0 0**	*5*	**0 0**	*5*	**0**	**0**	*1*	**0**	**0**	**0**	**1**
1		**0 0**	*10*	**0 1**	*5*	**0**	**0**	*1*	**0**	**0**	**0**	**1**
2		**0 0**	*20*	**1 0**	*5*	**0**	**0**	*1*	**0**	**0**	**0**	**1**
3		**0 0**	*50*	**1 1**	*5*	**0**	**0**	*1*	**0**	**0**	**0**	**1**
4	*10*	**0 1**	*5*	**0 0**	*5*	**0**	**0**	*2*	**0**	**0**	**1**	**0**
5		**0 1**	*10*	**0 1**	*10*	**0**	**1**	*1*	**0**	**0**	**0**	**1**
6		**0 1**	*20*	**1 0**	*10*	**0**	**1**	*1*	**0**	**0**	**0**	**1**
7		**0 1**	*50*	**1 1**	*10*	**0**	**1**	*1*	**0**	**0**	**0**	**1**
8	*20*	**1 0**	*5*	**0 0**	*5*	**0**	**0**	*4*	**0**	**1**	**0**	**0**
9		**1 0**	*10*	**0 1**	*10*	**0**	**1**	*2*	**0**	**0**	**1**	**0**
10		**1 0**	*20*	**1 0**	*20*	**1**	**0**	*1*	**0**	**0**	**0**	**1**
11		**1 0**	*50*	**1 1**	*20*	**1**	**0**	*1*	**0**	**0**	**0**	**1**
12	*50*	**1 1**	*5*	**0 0**	*5*	**0**	**0**	*10*	**1**	**0**	**1**	**0**
13		**1 1**	*10*	**0 1**	*10*	**0**	**1**	*5*	**0**	**1**	**0**	**1**
14		**1 1**	*20*	**1 0**	*50*	**1**	**1**	*1*	**0**	**0**	**0**	**1**
15		**1 1**	*50*	**1 1**	*50*	**1**	**1**	*1*	**0**	**0**	**0**	**1**

Una alternativa per estalviar sortides pot ser codificar el número de monedes ja que, amb 4 bits, es poden obtenir 16 combinacions diferents i d'aquestes només se n'utilitzen 5: 1, 2, 4, 5 i 10 Monedes.
Per codificar això només seran necessaris 3 bits K_2, K_1 i K_0:

Codi K_2 K_1 K_0	Número de Monedes
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	5
1 0 0	10
1 0 1	*X*
1 1 0	*X*
1 1 1	*X*

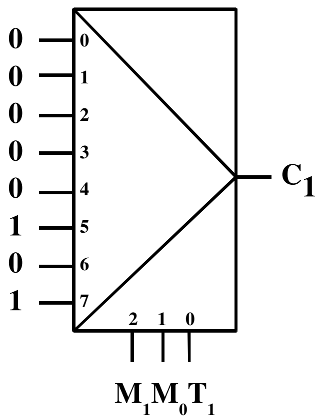
Utilitzant aquesta nova codificació per les monedes, la taula de la veritat varia de la següent forma:

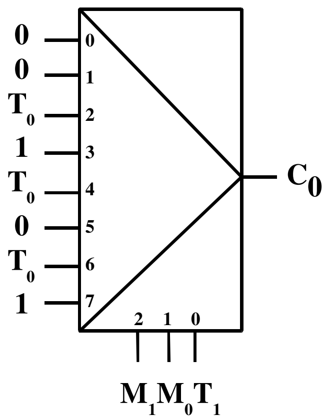
	M	M_1 M_0	*T*	T_1 T_0	*C*	C_1	C_0	*K*	K_2	K_1	K_0
0	*5*	0 0	*5*	0 0	*5*	0	0	*1*	0	0	0
1		0 0	*10*	0 1	*5*	0	0	*1*	0	0	0
2		0 0	*20*	1 0	*5*	0	0	*1*	0	0	0
3		0 0	*50*	1 1	*5*	0	0	*1*	0	0	0
4	*10*	0 1	*5*	0 0	*5*	0	0	*2*	0	0	1
5		0 1	*10*	0 1	*10*	0	1	*1*	0	0	0
6		0 1	*20*	1 0	*10*	0	1	*1*	0	0	0
7		0 1	*50*	1 1	*10*	0	1	*1*	0	0	0
8	*20*	1 0	*5*	0 0	*5*	0	0	*4*	0	1	0
9		1 0	*10*	0 1	*10*	0	1	*2*	0	0	1
10		1 0	*20*	1 0	*20*	1	0	*1*	0	0	0
11		1 0	*50*	1 1	*20*	1	0	*1*	0	0	0
12	*50*	1 1	*5*	0 0	*5*	0	0	*10*	1	0	0
13		1 1	*10*	0 1	*10*	0	1	*5*	0	1	1
14		1 1	*20*	1 0	*50*	1	1	*1*	0	0	0
15		1 1	*50*	1 1	*50*	1	1	*1*	0	0	0

b. Implementeu les funcions <u>dedicades al tipus de moneda</u> amb multiplexors de 3 entrades de control **(1 Punt).**

Les funcions dedicades al tipus de moneda són C_1 i C_0. Utilitzant les varibles M_1, M_0, i T_1, a les línies de control, es pot decidir cada una de les funcions a partir del valor de la variable d'entrada de menys pes T_0:

M_1 M_0 T_1	T_0	C_1	C_1	C_0	C_0
0 0 0	**0**	0	**0**	0	**0**
0 0 0	**1**	0		0	
0 0 1	**0**	0	**0**	0	**0**
0 0 1	**1**	0		0	
0 1 0	**0**	0	**0**	0	**T_0**
0 1 0	**1**	0		1	
0 1 1	**0**	0	**0**	1	**1**
0 1 1	**1**	0		1	
1 0 0	**0**	0	**0**	0	**T_0**
1 0 0	**1**	0		1	
1 0 1	**0**	1	**1**	0	**0**
1 0 1	**1**	1		0	
1 1 0	**0**	0	**0**	0	**T_0**
1 1 0	**1**	0		1	
1 1 1	**0**	1	**1**	1	**1**
1 1 1	**1**	1		1	

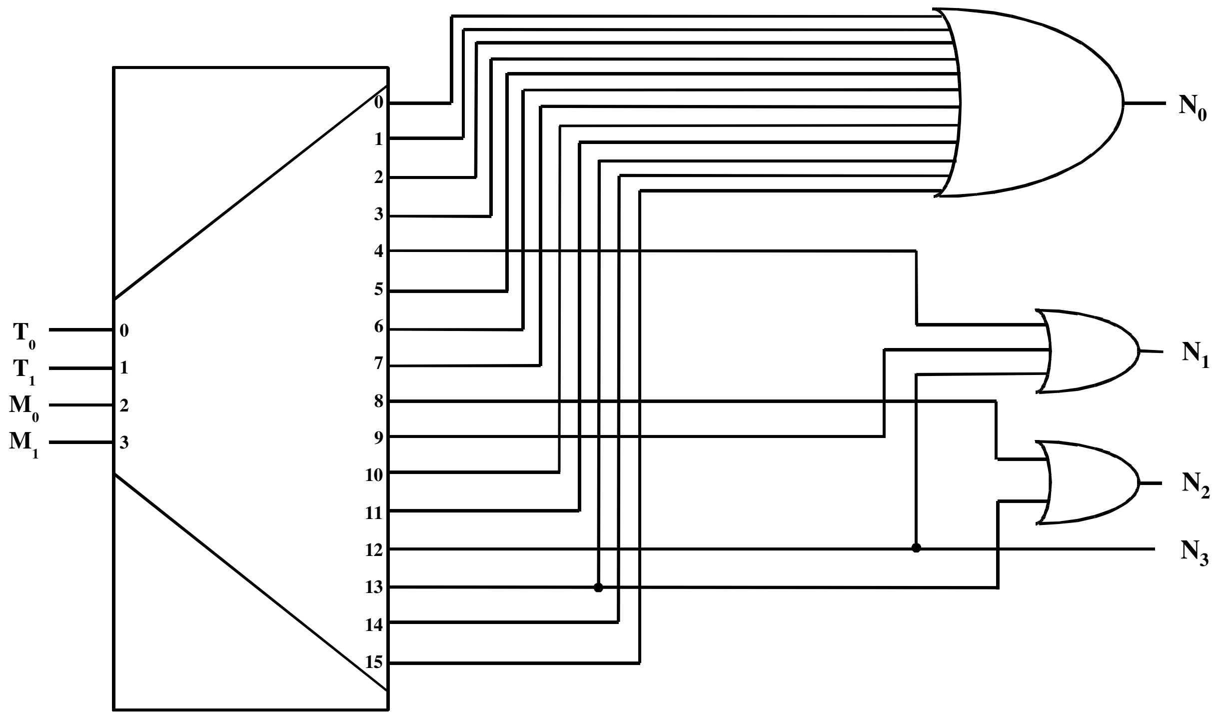




c. Implementeu les fucions <u>dedicades a la quantitat de monedes</u> amb un descodificador 4:16 i portes OR (**1 Punt).**

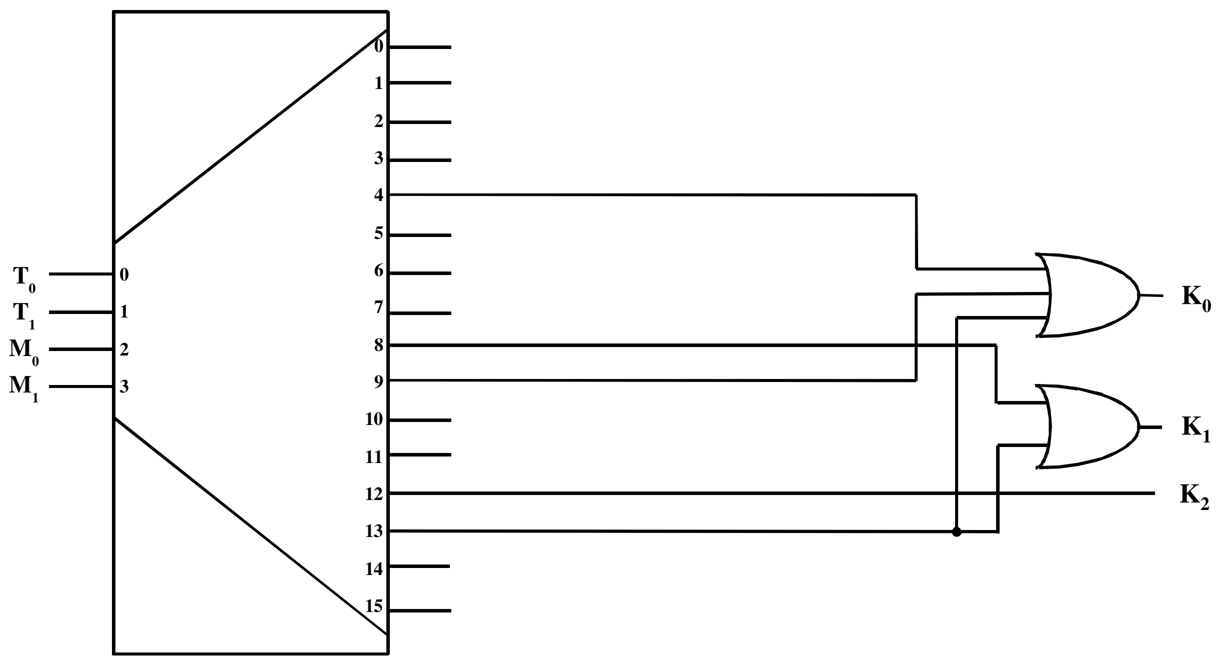
Utilitzant la primera aproximació, les funcions dedicades al tipus de moneda són N_3, N_2, N_1, N_0 que són funció de M_1, M_0, T_1 i T_0:

Mínterm	M_1 M_0 T_1 T_0	*N*	N_3	N_2	N_1	N_0
0	**0 0 0 0**	*1*	**0**	**0**	**0**	**1**
1	**0 0 0 1**	*1*	**0**	**0**	**0**	**1**
2	**0 0 1 0**	*1*	**0**	**0**	**0**	**1**
3	**0 0 1 1**	*1*	**0**	**0**	**0**	**1**
4	**0 1 0 0**	*2*	**0**	**0**	**1**	**0**
5	**0 1 0 1**	*1*	**0**	**0**	**0**	**1**
6	**0 1 1 0**	*1*	**0**	**0**	**0**	**1**
7	**0 1 1 1**	*1*	**0**	**0**	**0**	**1**
8	**1 0 0 0**	*4*	**0**	**1**	**0**	**0**
9	**1 0 0 1**	*2*	**0**	**0**	**1**	**0**
10	**1 0 1 0**	*1*	**0**	**0**	**0**	**1**
11	**1 0 1 1**	*1*	**0**	**0**	**0**	**1**
12	**1 1 0 0**	*10*	**1**	**0**	**1**	**0**
13	**1 1 0 1**	*5*	**0**	**1**	**0**	**1**
14	**1 1 1 0**	*1*	**0**	**0**	**0**	**1**
15	**1 1 1 1**	*1*	**0**	**0**	**0**	**1**



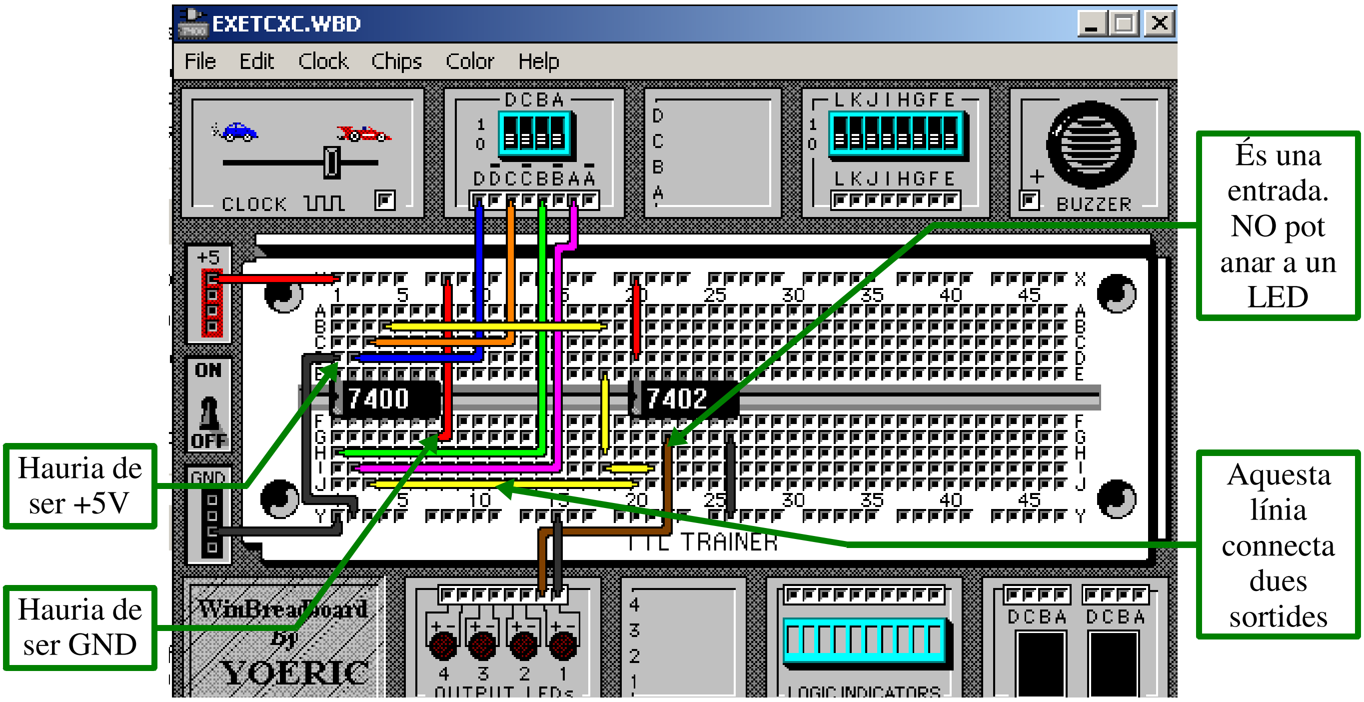
Utilitzant la segona aproximació, les funcions dedicades al tipus de moneda són K_2, K_1, K_0 que són funció de M_1, M_0, T_1 i T_0:

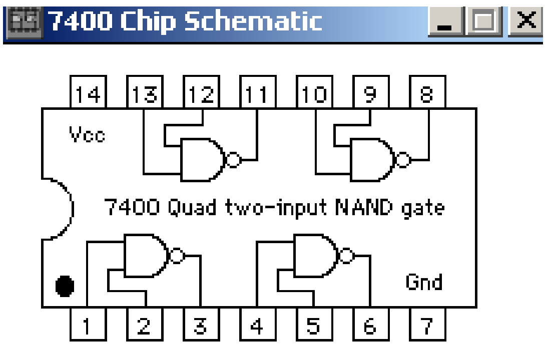
Mínterm	M_1 M_0 T_1 T_0	*K*	K_2	K_1	K_0
0	**0 0 0 0**	*1*	**0**	**0**	**0**
1	**0 0 0 1**	*1*	**0**	**0**	**0**
2	**0 0 1 0**	*1*	**0**	**0**	**0**
3	**0 0 1 1**	*1*	**0**	**0**	**0**
4	**0 1 0 0**	*2*	**0**	**0**	**1**
5	**0 1 0 1**	*1*	**0**	**0**	**0**
6	**0 1 1 0**	*1*	**0**	**0**	**0**
7	**0 1 1 1**	*1*	**0**	**0**	**0**
8	**1 0 0 0**	*4*	**0**	**1**	**0**
9	**1 0 0 1**	*2*	**0**	**0**	**1**
10	**1 0 1 0**	*1*	**0**	**0**	**0**
11	**1 0 1 1**	*1*	**0**	**0**	**0**
12	**1 1 0 0**	*10*	**1**	**0**	**0**
13	**1 1 0 1**	*5*	**0**	**1**	**1**
14	**1 1 1 0**	*1*	**0**	**0**	**0**
15	**1 1 1 1**	*1*	**0**	**0**	**0**

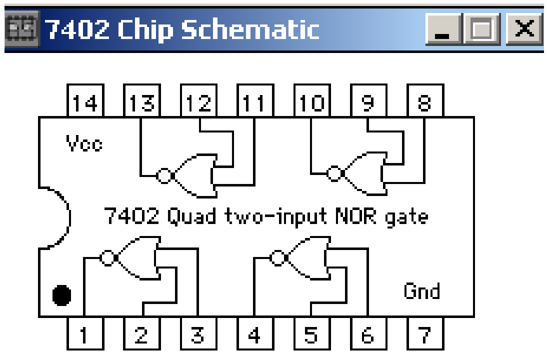


3. <u>SISTEMES COMBINACIONALS</u>. Problemes amb la fusta de tallar pa. (1 Punts).

Ens han encarregat que detectem els errors d'implementació del circuit format per portes de la següent figura **(1 Punt)**.







4. <u>SISTEMES SEQÜENCIALS</u>. Disseny del circuit de control d'un neteja parabrises. (4 Punts).

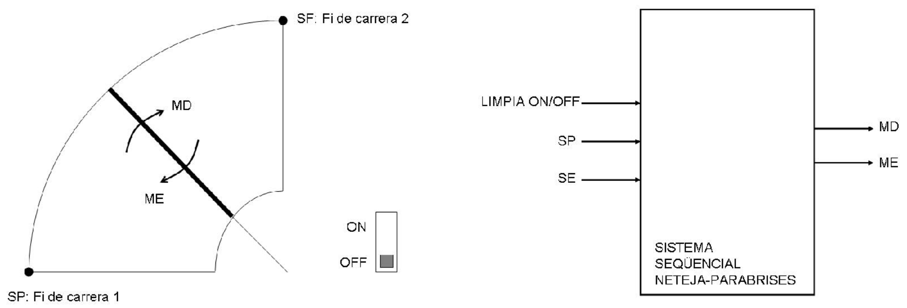
Cal dissenyar el circuit de control del neteja parabrises d'un cotxe de la gama baixa d'una coneguda marca comercial. Es tracta d'un circuit molt senzill que disposa dels següents elements:

- Un interruptor que connecta (ON) o desconnecta (OFF) el funcionament del neteja parabrises.
- Dos sensors (SP i SF) que indiquen, cada un d'ells, quan l'escombreta arriba a la posició horitzontal (de repòs) i a la posició vertical màxima en el vidre.

El mòdul que cal dissenyar ha de proporcionar dues senyals al motor del neteja parabrises:

- MD per fer anar el neteja parabrises cap a la dreta, és a dir, des de la posició horitzontal a la vertical.
- ME per fer anar el neteja parabrises cap a l'esquerra, és a dir, des de la posició vertical a l'horitzontal.

En la figura, hi ha l'esquema del neteja parabrises i el diagrama de blocs del circuit seqüencial que hem de dissenyar:



El funcionament que se'ns demana és el següent:

- Se suposa que quan s'activa el neteja parabrises (ON) l'escombreta es troba en posició horitzontal. Si no és així, s'ha de col·locar en aquesta posició abans d'operar normalment.
- Quan es troba funcionant amb normalitat (ON) i va fent el recorregut s'ha d'activar el senyal MD (i ME desactivat) per fer anar el neteja parabrises fins a la posició vertical aleshores, quan s'activa el sensor SF, s'ha d'activar el senyal ME (i MD desactivat) per fer anar el neteja parabrises fins a la posició horitzontal. Quan s'activa el sensor SP s'ha de tornar a fer el procés a la inversa i així anar fent mentre no es para el neteja parabrises (OFF).
- Quan es desactiva el neteja parabrises (OFF), l'escombreta ha de quedar en la posició horitzontal abans de quedar desactivat (MD = ME = 0).

Es demana:

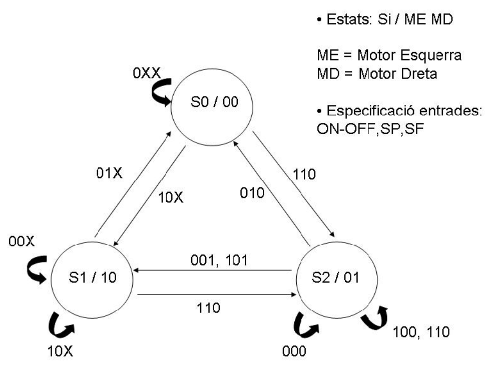
a. Dissenyeu el diagrama d'estats del sistema seqüencial tenint en compte el diagrama de blocs i el funcionament del neteja parabrises que es proposen **(2 Punts)**.

b. Implementeu el circuit seqüencial utilitzant biestables tipus D (actius per flanc de pujada) i una memòria ROM de la grandària que sigui convenient **(2 Punts)**.

NOTES:

- Cal justificar totes les respostes.
- Hem de suposar que disposem d'un senyal de CK amb un període molt més curt que el temps que triga l'escombreta per fer el recorregut complert. (Nota de l'autor: aquest comentari és per ajudar i simplificar).

1. El diagrama d'estats de *Moore* del control del neteja-parabrises pot ser el següent:



2. Taula d'estats a partir de l'autòmat.

Estat actual	Entrades	Estat següent	Sortides
S0	0 X X	S0	0 0
S0	1 0 X	S1	0 0
S0	1 1 0	S2	0 0
S0	1 1 1	XX	0 0
S1	0 0 X	S1	1 0
S1	0 1 X	S0	1 0
S1	1 0 X	S1	1 0
S1	1 1 0	S2	1 0
S1	1 1 1	XX	1 0
S2	0 0 0	S2	0 1
S2	0 0 1	S1	0 1
S2	0 1 X	S0	0 1
S2	1 0 0	S2	0 1
S2	1 0 1	S1	0 1
S2	1 1 0	S2	0 1
S2	1 1 1	XX	0 1

3. Simplificació.

No hi ha estats equivalents: NO ES POT SIMPLIFICAR.

4. Número de biestables necessaris.

Per emmagatzemar els 3 estats es necessitaran 2 biestables.

5. Assignació dels estats.

Estat	$Q_1\,Q_0$
S0	0 0
S1	0 1
S2	1 0
X	1 1

6. Tipus de biestables.

Cal utilitzar biestables de tipus **D** (Q_1 i Q_0) actius per flanc de pujada.

7. Taula d'exitacions i funcions de sortida.

Estat Actual	Entrada	Estat Següent	Sortida ME	Sortida MD	FF_1	FF_0
Q_1 Q_0	ON/SP/SF	Q_1^+ Q_0^+			D_1	D_0
0 0	0 0 0	0 0	0	0	0	0
0 0	0 0 1	0 0	0	0	0	0
0 0	0 1 0	0 0	0	0	0	0
0 0	0 1 1	0 0	0	0	0	0
0 0	1 0 0	0 1	0	0	0	1
0 0	1 0 1	0 1	0	0	0	1
0 0	1 1 0	1 0	0	0	1	0
0 0	1 1 1	X X	0	0	X	X
0 1	0 0 0	0 1	1	0	0	1
0 1	0 0 1	0 1	1	0	0	1
0 1	0 1 0	0 0	1	0	0	0
0 1	0 1 1	0 0	1	0	0	0
0 1	1 0 0	0 1	1	0	0	1
0 1	1 0 1	0 1	1	0	0	1
0 1	1 1 0	1 0	1	0	1	0
0 1	1 1 1	X X	1	0	X	X
1 0	0 0 0	1 0	0	1	1	0
1 0	0 0 1	0 1	0	1	0	1
1 0	0 1 0	0 0	0	1	0	0
1 0	0 1 1	0 0	0	1	0	0
1 0	1 0 0	1 0	0	1	1	0
1 0	1 0 1	0 1	0	1	0	1
1 0	1 1 0	1 0	0	1	1	0
1 0	1 1 1	X X	0	1	X	X
1 1	0 0 0	X X	X	X	X	X
...	...	...	...	...	...	...
1 1	1 1 1	X X	X	X	X	X

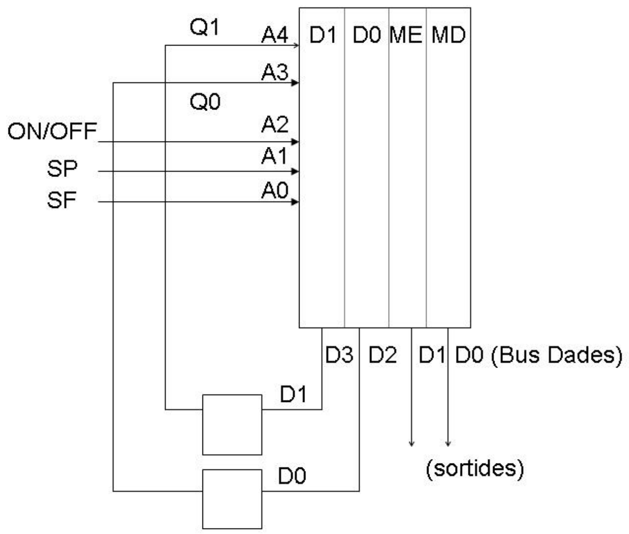
8. Implementació.

A partir d'aquí només queda fer la implementació del circuït seqüencial tenint en compte la memòria ROM adient i els 2 biestables que cal utilitzar.

Per a determinar la memòria ROM cal considerar:

- Nombre de línies del bus d'adreces: senyals d'estat (Q_1, Q_0) + entrades (ON, SP, SF) = 2 + 3 = 5.
- 2^5 = 32 posicions de memòria.
- Nombre de línies del bus de dades (capacitat de cada una de les cel·les de memòria): sortides (ME, MD) + entrades de control dels biestables (D_1, D_0) = 2 + 2 = 4.

Per tant, la memòria, serà de 32 posicions de 4 bits cadascuna



2 BIESTABLES TIPUS D (actius per flanc de pujada)