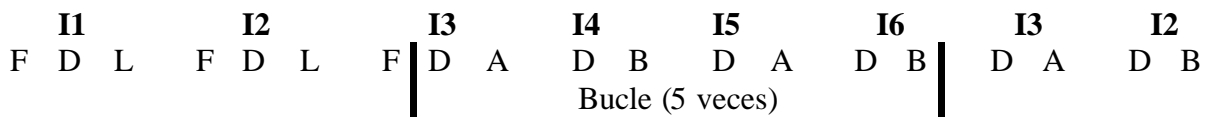


Problema 1

- a.
- | | | |
|------|----------------|---------|
| (I1) | LOAD 0(R0),R1: | 1 vez |
| (I2) | LOAD 1(R0),R2: | 1 vez |
| (I3) | SUB R2,R1,R3: | 6 veces |
| (I4) | BGE 08h: | 6 veces |
| (I5) | ADDI R2,#3,R2: | 5 veces |
| (I6) | BR 04h: | 5 veces |

b. El cuadro siguiente ilustra el número de ciclos realizado para cada instrucción:



Total de ciclos = $7 + 8 \cdot 5 + 4 = 51$ ciclos

Tiempo en segundos = $51 \text{ ciclos} \cdot 0,5 \cdot 10^{-9} \text{ seg/ciclo} = 25,5 \cdot 10^{-9} \text{ seg}$

Solució Problema 2

a) Analitzem cada cas per separat:

- **DA i DE (entrada):** Amb n bits és possible codificar en complement a 2 números en el rang $[-2^{n-1} \dots 2^{n-1}-1]$. Donat que tant la Data Actual com la Data d'Expedició de la targeta poden estar entre 0 i 364, caldrà fer servir un nombre de bits n tal que $364 \leq 2^{n-1} - 1$. La desigualtat s'acompleix per $n \geq 10$. Per tant, **el nombre mínim de bits que han de tenir els senyals DA i DE és de $n=10$ bits.**
- **ERR (sortida):** Tan sols cal que indiqui si s'ha excedit o no el període d'ús de la targeta. Per tant, **n'hi ha prou que ERR sigui un senyal d'un sol bit.**

b) En primer lloc, caldrà esbrinar quants dies han passat des que es va adquirir la targeta multiviatge tot calculant la diferència **DA-DE**. Per dur a terme aquesta operació, podem fer servir sumadors binaris de 10 bits. Donat que tant **DA** com **DE** estan codificats en complement a 2, podem realitzar la resta com la suma de **DA** i l'oposat de **DE**:

$$\mathbf{DA-DE = DA+ (- DE)}$$

Per calcular l'oposat de **DE** cal dur a terme un canvi de signe que, en complement a 2, es redueix a invertir **DE** bit a bit i sumar-li una unitat al bit de menys pes, és a dir:

$$\mathbf{- DE = DE' + 1}$$

i, per tant,

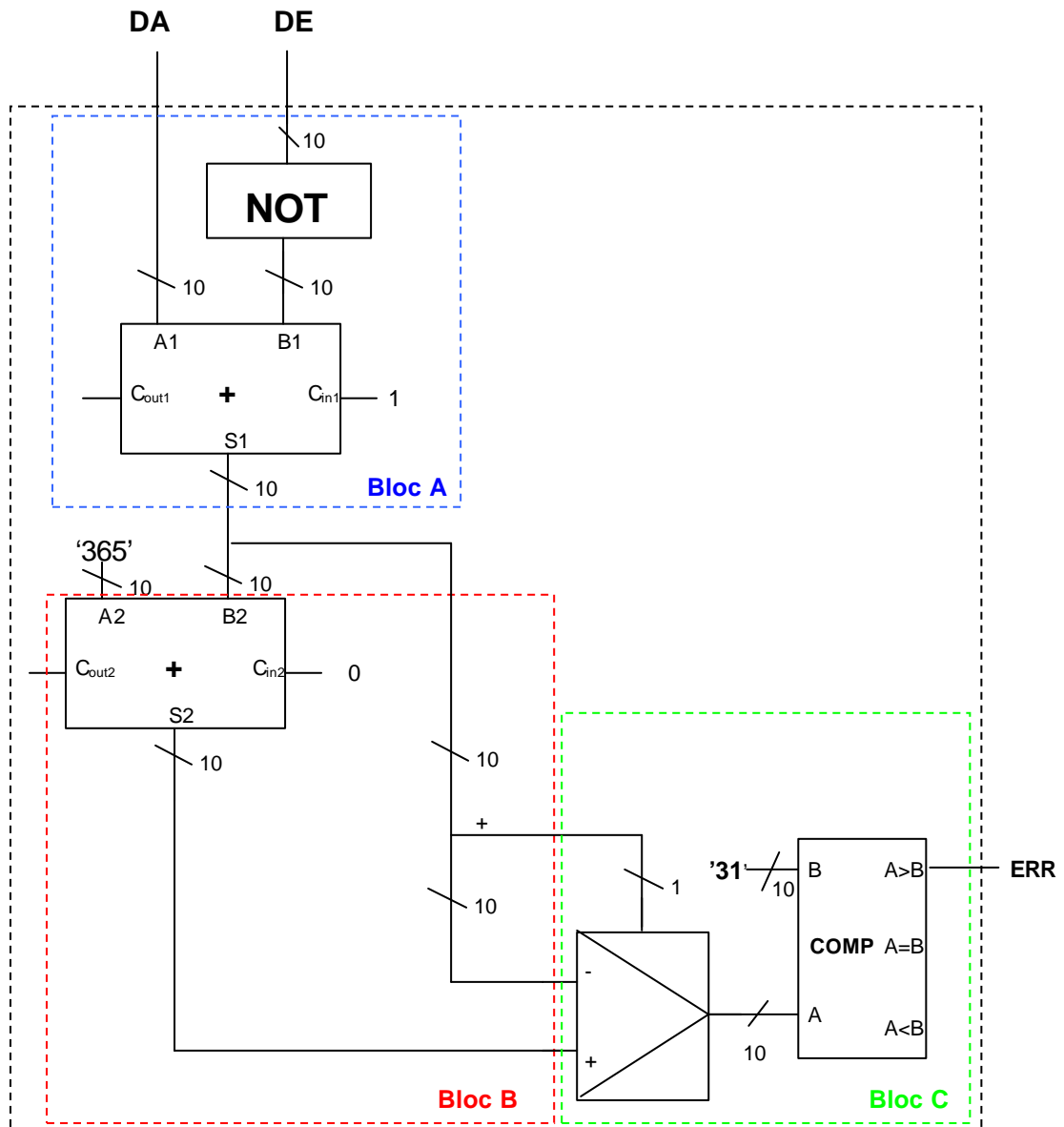
$$\mathbf{DA-DE = DA+ (- DE) = DA + DE' + 1}$$

En d'altres paraules, per calcular la diferència **DA-DE** podem fer la suma de **DA**, **DE'** i 1. L'últim sumand, s'introduirà a través del transport d'entrada (C_{in}) del corresponent bloc sumador. Aquesta operació és la que realitza el subcircuit etiquetat com a '**Bloc A**' de la figura de la pàgina següent.

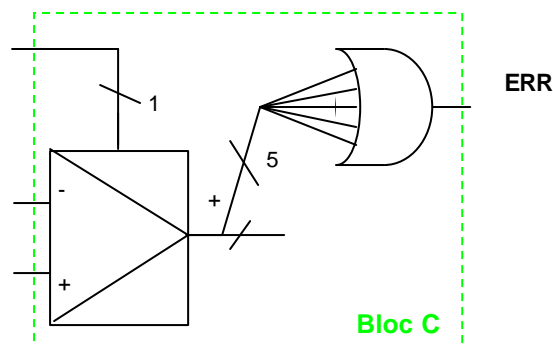
Caldrà tenir en compte dos possibles casos que es poden donar en calcular la sortida del bloc A:

- Si el resultat és positiu o zero, la sortida del bloc A ja és directament la diferència de dies entre les dates actual i d'expedició.
- Si el resultat és negatiu, la diferència real de dies vindrà donada pel resultat de sumar 365 al número negatiu obtingut a la sortida del Bloc A. Per exemple, com es deia a l'enunciat, entre el 6 de gener d'un any i el 25 de desembre de l'any anterior, la diferència DA-DE val $5-358 = -353$. En realitat, però, han passat $-353 + \mathbf{365} = +12$ dies. Aquesta suma, es realitza al subcircuit etiquetat com a '**Bloc B**'.

En funció del signe del resultat obtingut al Bloc A, caldrà decidir si la diferència real de dies ve donada per **DA-DE** o bé per **DA-DE+365**. Aquesta funció la duu a terme el multiplexor a l'entrada del '**Bloc C**'. A l'entrada de control s'hi connecta el bit més alt del resultat del Bloc A que indica el signe del número representat. En resum, el que s'obté a la sortida del multiplexor és la resta mòdul 365 de **DA** i **DE**, o sigui, $(\mathbf{DA-DE}) \bmod 365$ que sempre és positiu. Finalment, caldrà posar a 1 el senyal de sortida **ERR** si i només si $(\mathbf{DA-DE}) \bmod 365$ és més gran que 31. La manera més immediata consisteix en comparar la sortida del multiplexor del '**Bloc C**' amb '31' mitjançant un bloc comparador i prendre com a resultat el senyal $A > B$.



Per al cas que ens ocupa, hi ha una manera molt senzilla de comparar si $(DA - DE) \bmod 365$ és més gran que 31 tot comprovant, de manera alternativa, si $(DA - DE) \bmod 365$ és més gran o igual que 32. Per simple inspecció es pot veure que, quan això succeeix, algun dels 5 bits més alts a la sortida del MUX ($b_5 \dots b_9$) ha de ser forçosament 1. Així, el resultat de la comparació, pot obtenir-se mitjançant una suma lògica d'aquest bits amb un porta OR de cinc entrades¹:



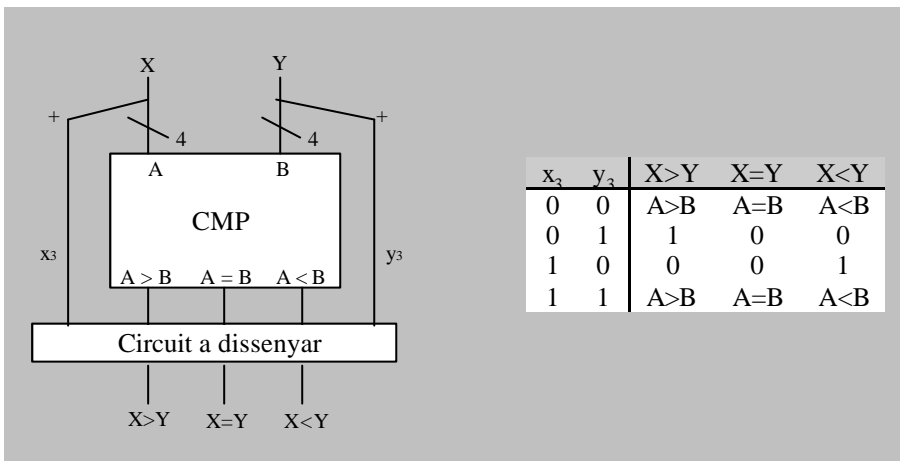
¹ De fet, donat que $(DA - DE) \bmod 365$ és sempre positiu el bit b_9 sempre valdrà 0. No seria necessari, doncs, incloure aquest bit en la suma lògica.

c) El resultat de la comparació depèn dels signes dels dos números (que vénen donats pel seu bit de més pes: 1 pels negatius, 0 pels positius o zero).

Si els dos números són de signes diferents, llavors el més gran és el positiu.

Quan els dos números són del mateix signe, llavors els podem comparar usant un comparador de números naturals. Fixem-nos que el resultat serà correcte també en el cas que els dos números siguin negatius, perquè si $X > Y$ interpretant X i Y en complement a 2, llavors també es compleix que $X > Y$ si els interpretem en binari. Per exemple, si comparem $X=-1$ (1111) amb $Y=-2$ (1110) el comparador trauria un 1 per la sortida $X>Y$.

A continuació es mostra un primer esquema del circuit, i la taula de veritat que descriu el comportament de la part del circuit que queda per implementar, en funció del signe dels dos números.



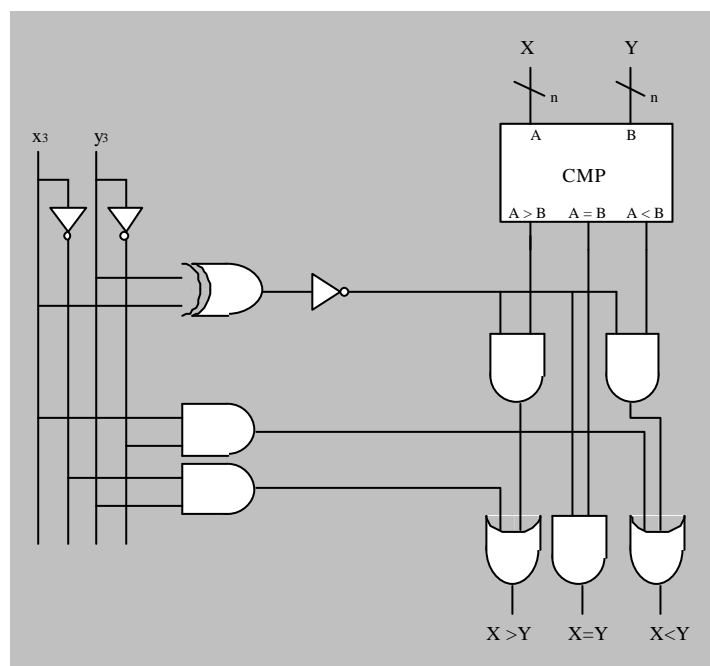
D'aquestes taules es dedueixen les següents igualtats:

$$[X > Y] = x_3' y_3 + (x_3 \oplus y_3)' \cdot [A > B]$$

$$[X < Y] = x_3 y_3' + (x_3 \oplus y_3)' \cdot [A < B]$$

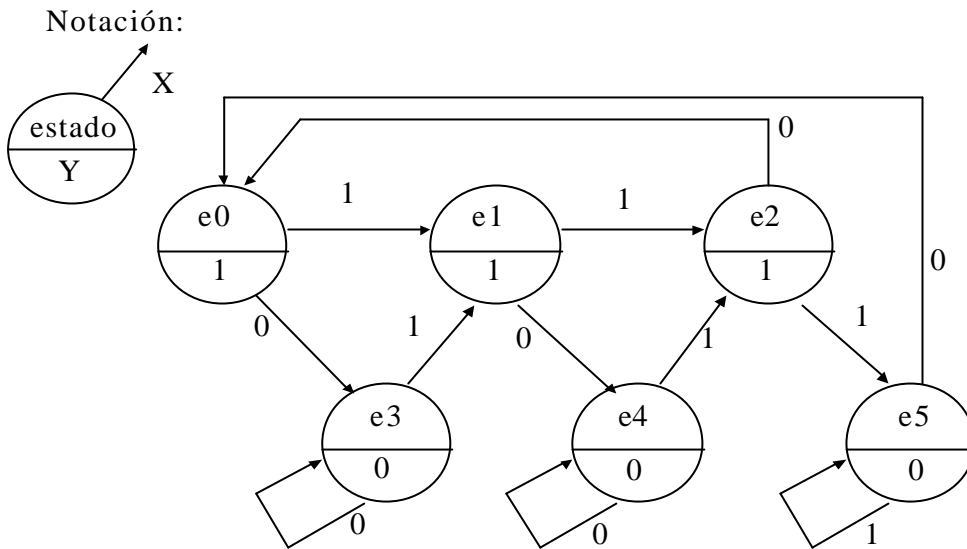
$$[X = Y] = (x_3 \oplus y_3)' \cdot [A = B]$$

A continuació es mostra el circuit complet:



SOLUCIÓN P3:

a)



b)

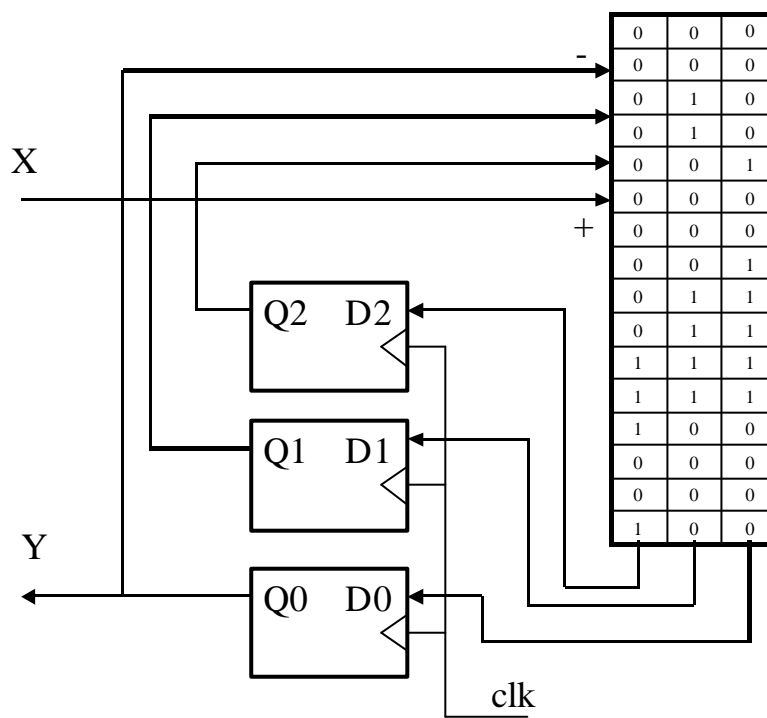
estado	codificación
e0	001
e1	011
e2	111
e3	000
e4	010
e5	100

X	Q2	Q1	Q0	Q2+	Q1+	Q0+
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	X	X	X
0	1	1	0	X	X	X
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	0	0
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	1	0	0

Tabla de salidas:

Q2	Q1	Q0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	X
1	1	0	X
1	1	1	1

Circuito:



Problema 4

a.

	Dirección	Instrucción en LM	M[@]: LM	M[@] en hexadecimal
	00h	-----	0000 0000 0000 1010	000Ah
	01h	-----	0000 0000 0000 0101	0005h
	02h	-----	0000 0000 0000 0000	0000h
I1	03h	LOAD 0(R0),R1	0000 1000 0000 0000	0800h
I2	04h	LOAD 1(R0),R2	0001 0000 0000 0001	1001h
I3	05h	ADDI R0,#0,R3	1101 1000 0000 0000	D800h
I4	06h	SUBI R2,#0,R0	1100 0010 0000 0001	C201h
I5	07h	BLE 0Bh	1001 1000 0000 1011	980Bh
I6	08h	ADD R3,R1,R3	1101 1011 0010 0100	DB24h
I7	09h	SUBI R2,#1,R2	1101 0010 0000 1001	D209h
I8	0Ah	BR 06h	1000 0000 0000 0110	8006h
I9	0Bh	STORE R3,2(R0)	0101 1000 0000 0010	5802h

b.

	I1			I2			I3		I4		I5	
Estado	FETCH	DECO	LOAD	FETCH	DECO	LOAD	FETCH	DECO	ARIT	DECO	ARIT	DECO
Ld_IR	1	0	0	1	0	0	1	0	1	0	1	0
Ld_PC	1	0	0	1	0	0	1	0	1	0	1	0
Ld_R@	0	1	0	0	1	0	0	1	0	1	0	1
Ld_RA	0	1	0	0	1	0	0	1	0	1	0	1
Ld_RZ	0	0	1	0	0	1	0	0	1	0	1	0
Ld_RN	0	0	1	0	0	1	0	0	1	0	1	0
ERd	0	0	1	0	0	1	0	0	1	0	1	0
\bar{L}/E	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{PC}/@$	0	X	1	0	X	1	0	X	0	X	0	X
CRf	X	1	X	X	1	X	X	1	2	1	2	1
OPERAR	X	X	0	X	X	0	X	X	1	X	1	X
M@	03h	X	00h	04h	X	01h	05h	X	06h	X	07h	X
Mout	0800h	X	000Ah	1001h	X	0005h	D800h	X	C201h	X	980Bh	X
Min/Rx	X	0000h	X	X	0000h	X	X	0000h	0000h	0005h	0000h	0000h
Din/ALU	X	X	000Ah	X	X	0005h	X	X	0000h	X	0005h	X
PC	03h	04h	04h	04h	05h	05h	05h	06h	06h	07h	07h	08h
IR	X	0800h	0800h	0800h	1001h	1001h	1001h	D800h	D800h	C201h	C201h	980Bh
R@	X	X	00h	00h	00h	00h	00h	00h	00h	00h	06h	06h
RA	X	X	0000h	0000h	0000h	0000h	0000h	0000h	0000h	0000h	0005h	0005h
RZ	X	X	X	0	0	0	0	0	0	1	1	0
RN	X	X	X	0	0	0	0	0	0	0	0	0
R0	0	0	0	0	0	0	0	0	0	0	0	0
R1	X	X	X	000Ah	000Ah	000Ah	000Ah	000Ah	000Ah	000Ah	000Ah	000Ah
R2	X	X	X	X	X	X	0005h	0005h	0005h	0005h	0005h	0005h
R3	X	X	X	X	X	X	X	X	X	0000h	0000h	0000h