

Posición de memoria	contenido
30FA	21
30FB	10
30FC	12
30FD	0
30FE	0
30FF	0
3100	12
3101	45
3102	65
3103	12
3104	45
3105	1
3106	12
3107	0
3108	0
3109	0

registro	contenido
A	0
B	0
C	0
D	0
E	0
H	0
L	0
estado	0
PC	3000
SP	5000

Posición de memoria	contenido
30FA	21
30FB	10
30FC	12
30FD	0
30FE	0
30FF	0
3100	12
3101	45
3102	65
3103	12
3104	45
3105	13
3106	12
3107	0
3108	0
3109	0

registro	contenido
A	13
B	0
C	0
D	0
E	0
H	31
L	04
estado	15
PC	3011
SP	5000

1) La posición de cada dispositivo depende de como se genere su CS. En este circuito, los CS se generan con el 74HC139, de la siguiente forma:

10/ $\bar{m}$	A_{13}	A_{12}	CS RAM CSRAM 3253 3255			
			Y_0	Y_1	Y_2	Y_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

De donde se ve:

- Todos los dispositivos están en memoria ($10/\bar{m} = 0$)

- La RAM está en la posición
 $A_{15} \quad A_{12} \quad A_{11} \quad A_8 \quad A_7 \quad A_4 \quad A_3 \quad A_0$
 000 00000 0000 0000 y siguientes.

Como es de 1K (10 líneas de dirección) ocupará las posiciones

0000 0000 0000 0000 $\rightarrow$ 0000h
 hasta
 0000 0011 1111 1111 $\rightarrow$ 003FFh.

• La ROM está en la posición.

A_{15} A_{12} A_{11} A_8 A_7 A_6 A_5 A_0
 0001 0000 0000 0000 $\rightarrow$ 1000h

y como ocupa 4K (12 líneas de dirección), llega hasta

0001 1111 1111 1111 $\rightarrow$ 1FFFh.

• El 8253 está en:

0010 0000 0000 0000 $\rightarrow$ 2000h

y como ocupa 4 posiciones, llega hasta

0010 0000 0000 0011 $\rightarrow$ 2003h

• El 8255 está en:

0011 0000 0000 0000 $\rightarrow$ 3000h

y llega hasta

0011 0000 0000 0011 $\rightarrow$ 3003h.

Mapa de memoria

RAM	0000h 03FFh.
ROM	1000h 1FFFh
8253	2000h 2003h
8255	3000h 3003h

2) 2.1 Carga en el acumulador el contenido de la posición 3100
 Lo suma con el contenido de la posición 3101, con
 el de la posición 3102, el de la 3103 y la
 3104 y almacena el resultado en la
 posición 3105

2.2

De la manera, sólo cambia el contenido de la
 posición 3105, que pasa a ser

$$\underbrace{12 + 45 + 65 + 12}_{CE} + 45 = 113$$

como los registros son de 8 bits, los acarreo se
 pierden. Quedando: 13 h.

Lo mismo queda en el registro $A = 13h$.

La pareja de registros HL queda con 3104 $\Rightarrow \left. \begin{array}{l} H=31 \\ L=04 \end{array} \right\}$
 La última instrucción que afecta al registro de estado es
 ADD M, que suma $CE + 45 \rightarrow 113 \Rightarrow \left. \begin{array}{l} Z=0 \\ S=0 \end{array} \right\} \begin{array}{l} P=1 \\ CF=1 \end{array} \quad AC=1$

PC queda apuntando a la siguiente instrucción HLT.

Estado: SZOACOPCF

00010101

15

Como los códigos de OP son de 1 byte, excepto a los
 instrucciones que llevan operandos el programa ocupa 17 (11h)
 bytes $\Rightarrow PC = 3011$

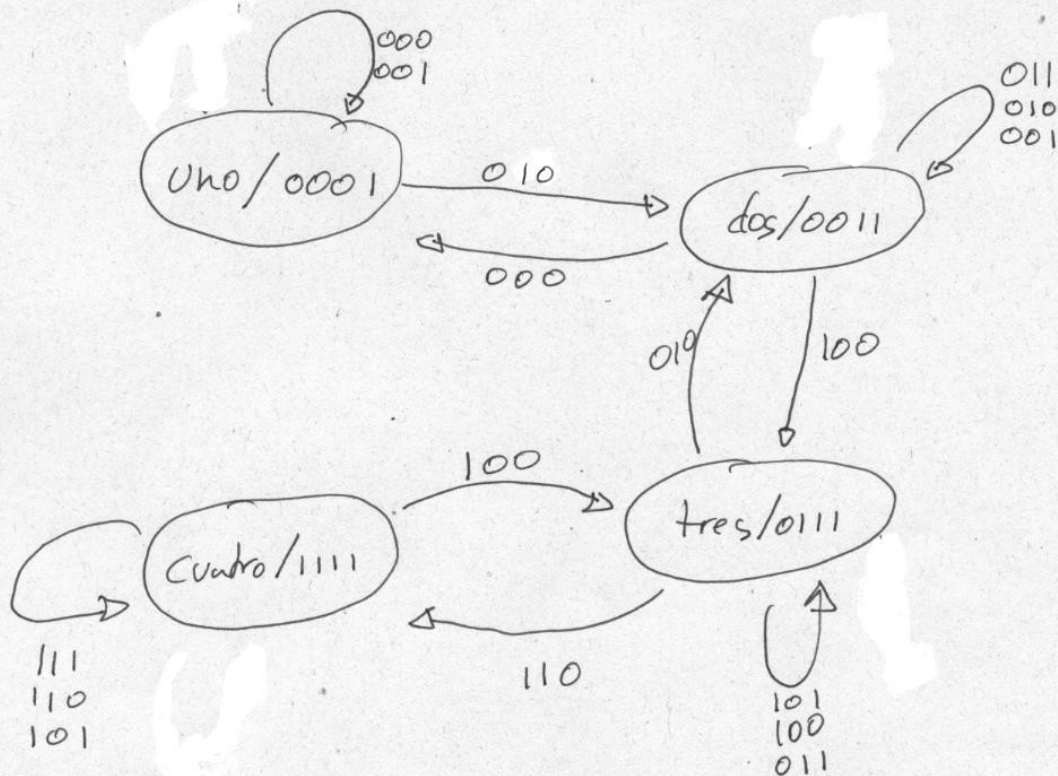
El resto de registros quedan igual

3)

a) El diagrama de estados será:

Entradas: a b c

Salidas: g3 g2 g1 g0



Codificación de estados: $Q_1 Q_0 \rightarrow$ UNO = 00, DOS = 01, TRES = 10, CUATRO = 11

b) Tabla de transiciones

Estado inicial	$Q_1 Q_0$	Entradas a b c	Estado prox	
			$Q_1 Q_0$	
uno	00	000	uno	00
uno	00	001	uno	00
uno	00	010	dos	01
dos	01	000	uno	00
dos	01	001	dos	01
dos	01	010	dos	01
dos	01	011	dos	01
dos	01	100	tres	10

$Q_1 Q_0$	a b c	$Q_1 Q_0$
tres 10	010	dos 01
tres 10	011	tres 10
tres 10	100	tres 10
tres 10	101	tres 10
tres 10	110	cuatro 11
cuatro 11	100	tres 10
cuatro 11	101	cuatro 11
cuatro 11	110	cuatro 11
cuatro 11	111	cuatro 11

Tabla estados - salidas.

Estado	Salidas			
$Q_1 Q_0$	g_3	g_2	g_1	g_0
uno 0 0	0	0	0	1
dos 0 1	0	0	1	1
tres 1 0	0	1	1	1
cuatro 1 1	1	1	1	1

$$G_0 = 1$$

$$G_1 = Q_1 + Q_0$$

$$G_2 = Q_1$$

$$G_3 = Q_1 Q_0$$

Simplificación

D_0 :

$Q_1 = 0$

$Q_0 a \backslash bc$	00	01	11	10
00	0	0	X	0
01	X	X	X	X
11	0	X	X	X
10	0	1	1	1

$Q_1 = 1$

$Q_0 a \backslash bc$	00	01	11	10
00	X	X	0	1
01	0	0	X	1
11	0	1	1	1
10	X	X	X	X

$$D_0 = Q_0 b + Q_0 a + Q_1 b \bar{c}$$

$Q_0 a$

$Q_0 b$

$Q_1 b \bar{c}$

D_1 :

$$Q_1 = 0$$

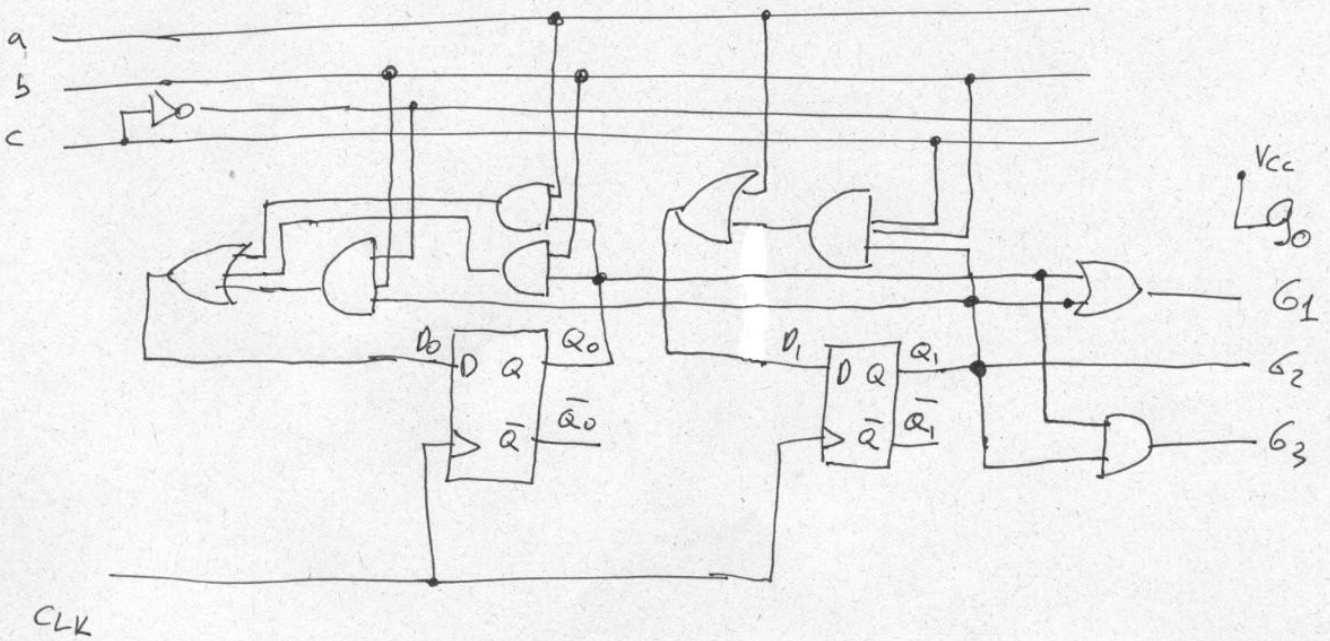
$Q_0 a \backslash bc$	00	01	11	10
00	0	0	X	0
01	X	X	X	X
11	1	X	X	X
10	0	0	0	0

$$Q_1 = 1$$

$Q_0 a \backslash bc$	00	01	11	10
00	X	X	1	0
01	1	1	X	1
11	1	1	1	1
10	X	X	X	X

$Q_1 bc$

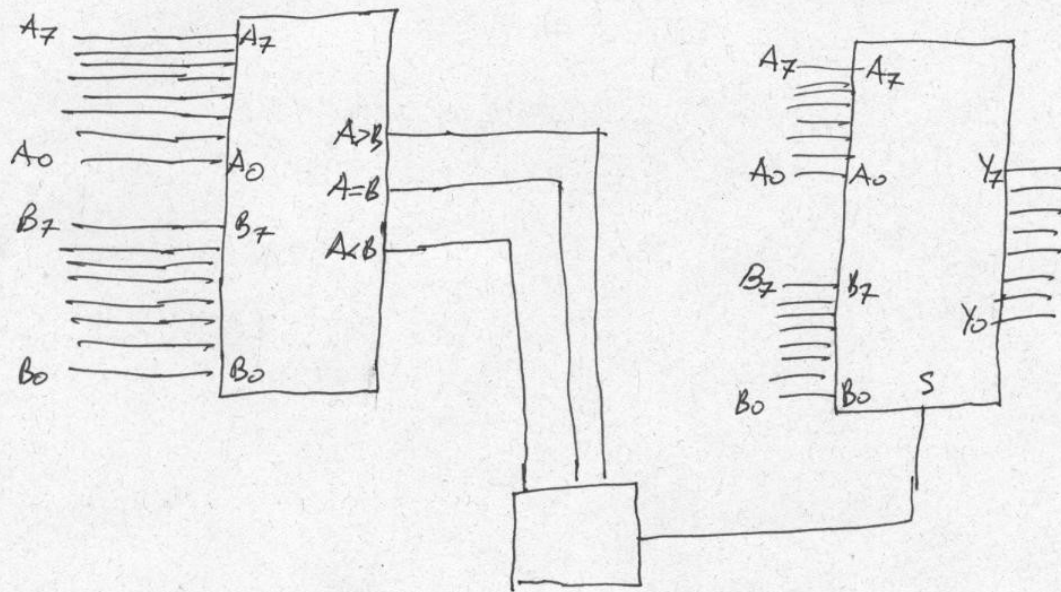
$$D_1 = Q_1 bc + a$$



4) Se necesita

- Saber cual de los dos datos es mayor $\Rightarrow$ comparador
- Seleccionar el mayor de ellos $\Rightarrow$ multiplexor.

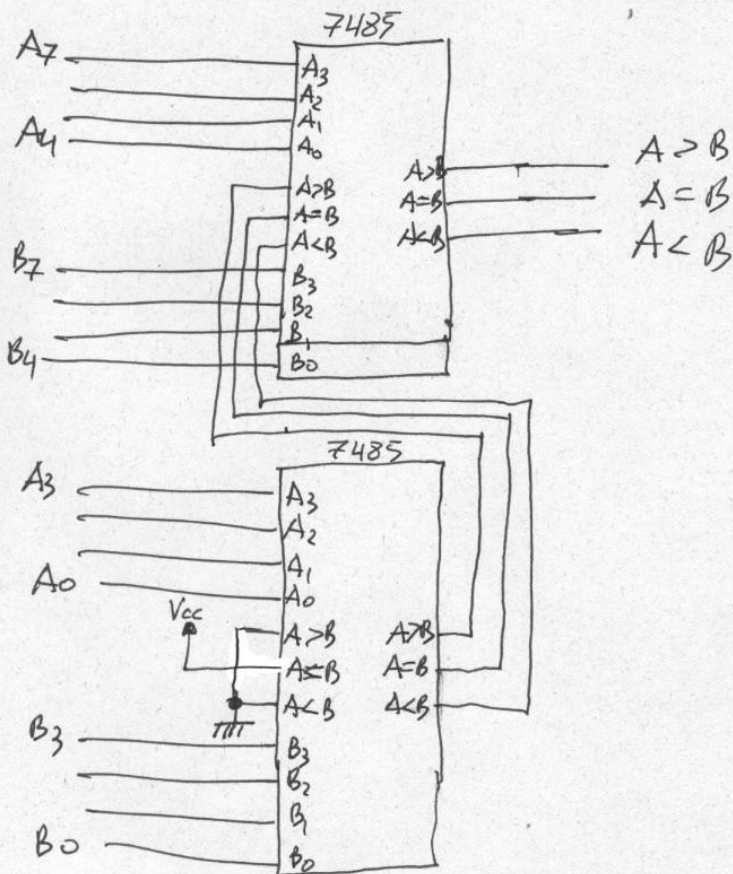
El diagrama de bloques sería:



- Un comparador de dos números de 8 bits.
- Un multiplexor múltiple 2:1 que seleccione 8 líneas (una entrada de control)
- Lógica para generar la línea de control a partir de las salidas del comparador.

El desarrollo de cada bloque:

- a) Comenzando tenemos el 7485 que es un comparador de 2 números de 4 bits. Conectados dos a cascada tenemos:

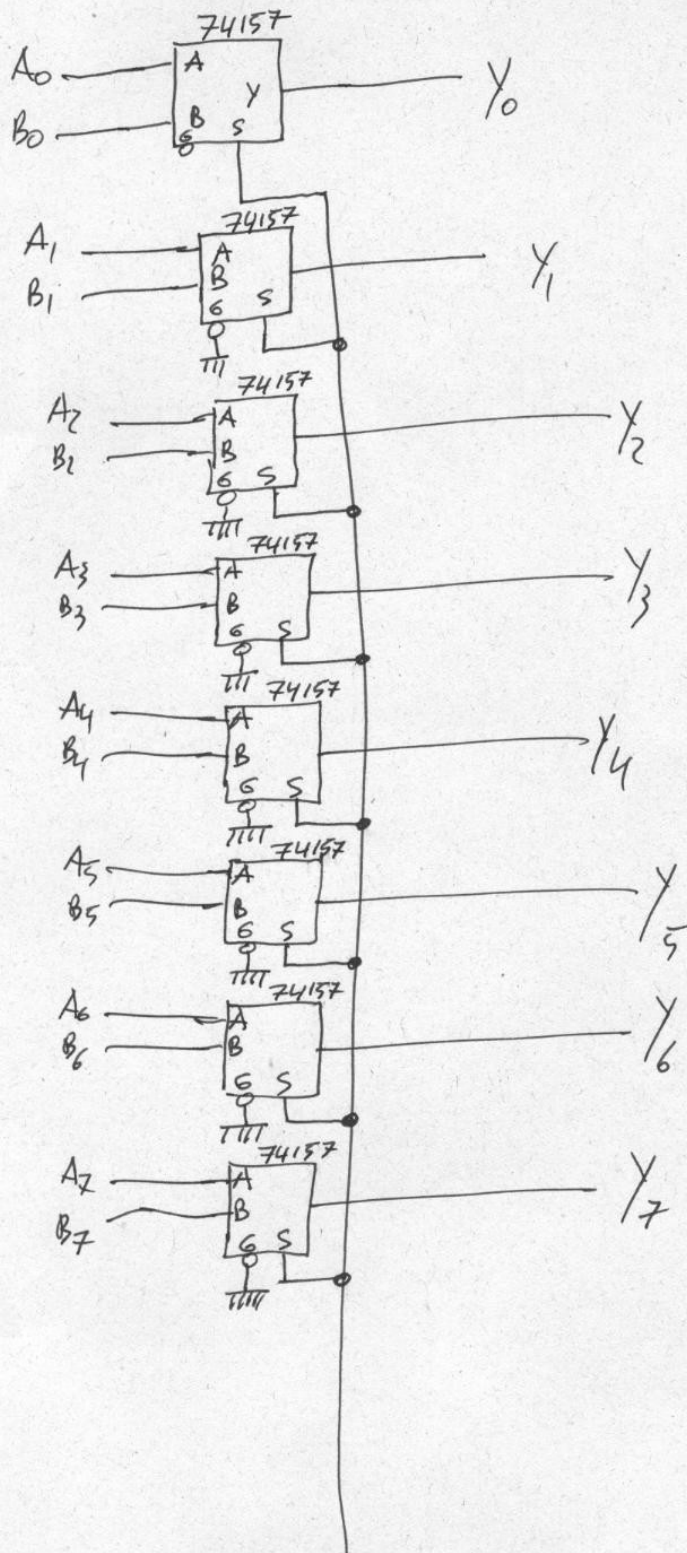


- b) Por otro lado, para generar la señal de control; tenemos

A>B	A=B	A<B	S	
1	0	0	0	→ selecciona A
0	1	0	X	
0	0	1	1	→ selecciona B

de donde se ve que puede ser: $S = A < B$

c) Para el multiplexor múltiple necesitamos 8 mux 2:1
como por ejemplo el 74157



5) El comportamiento de los registros de desplazamiento es:

$S_1 S_0$	
0 0	mantiene
0 1	despl izda
1 0	despl dcha
1 1	Carga paralela

Para enviar el dato, se cargará en el registro de desplazamiento por las entradas paralelo y luego se enviará, desplazando a derecha (empujando por LSB)

Como los datos son de 8 bits, tenemos que unir dos registros para hacer un registro de desplazamiento de 8 bits.

Las entradas de control serán comunes, U2 almacenará los 4 bits más altos, y al desplazar a derecha, los bits van "saliendo" por su línea de menor peso (QA) y entrarán a U4 (4 bits bajos) por la izquierda (entrada serie SL).

En recepción, también se unen U1 y U3 para formar un registro de 8 bits. Como también se va a recibir el dato desplazando a la derecha, los datos van entrando por la izquierda (entrada SL)

El dato se cargará en U2-U4 cuando se active el pulso en DATO. Luego, habrá que desplazar durante 8 pulsos del reloj, y a partir de ahí, mantener el dato. Para contar los 8 pulsos, hacemos que el contador se pare al llegar a 8, conectada QD a ENT, ENP

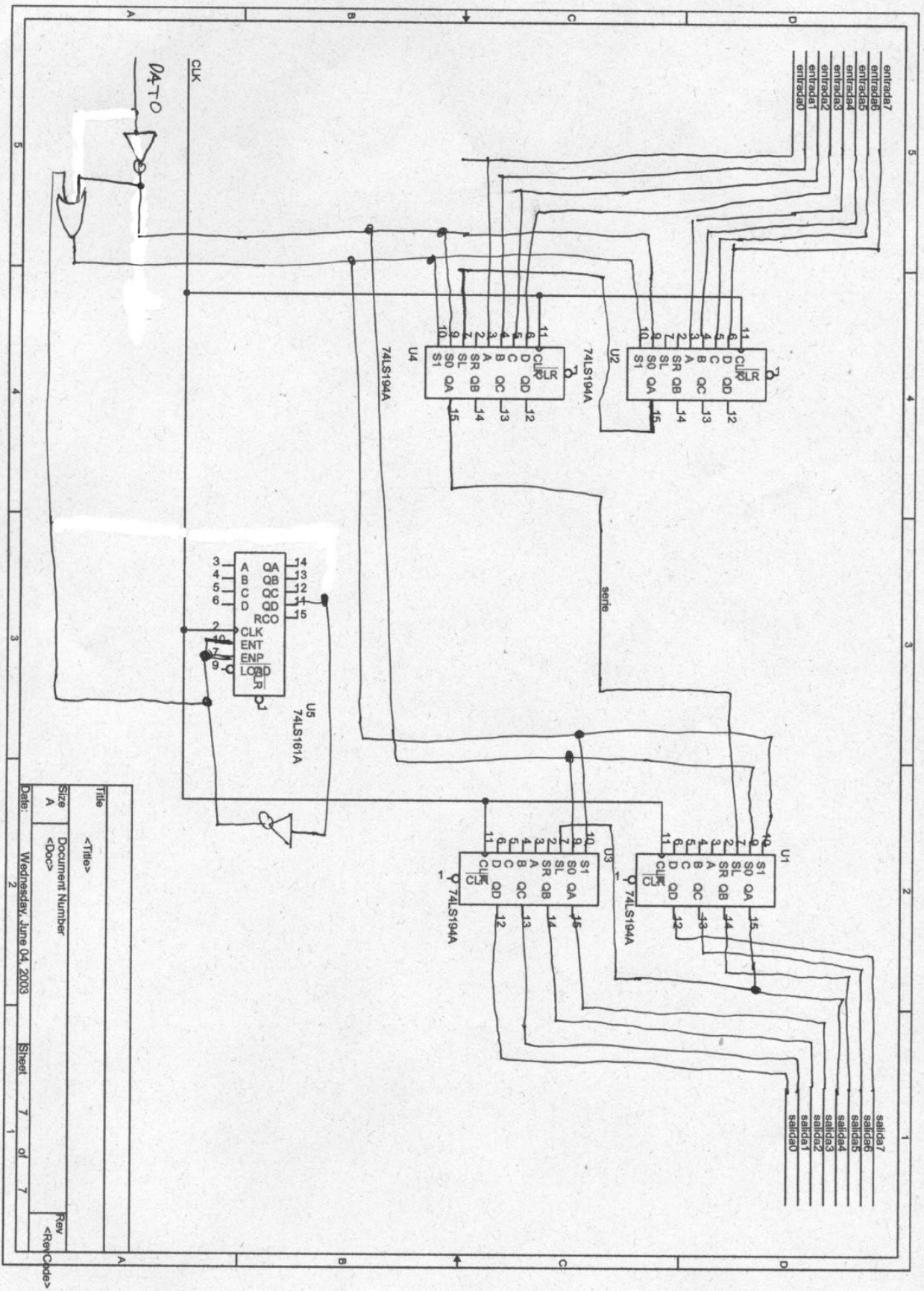
a través de un negador.

La generación de las señales de control será:

DATO	Q_0	S_1, S_0	observación
0	X	11	se carga el dato a el registro U_2-U_4
1	0	10	durante 8 pulsos, se desplaza a dcha
1	1	00	después de los 8 pulsos se mantiene el dato indefinidamente

$$S_1 = \overline{\text{DATO}} + \overline{Q_0}$$

$$S_0 = \overline{\text{DATO}}$$



Title		<Title>	
Size		Document Number	
A		<Doc>	
Date:	Wednesday, June 04, 2003	Sheet	7 of 7
Rev	<RevCode>		