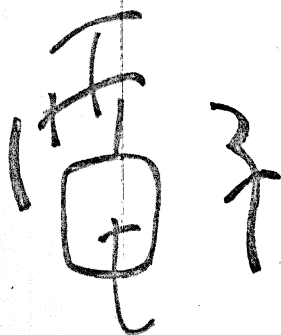


U N S

Departamento de
Electrotecnia

Cátedra de



PROYECTO

título

Instalación de una memoria bipolar de

4x16 palabras y revisión crítica del

procesador.

autor

Aníbal Raúl Gil

tribunal examinador

presentado el

aprobado el

profesor

I N D I C E

título	pág.
1) OBJETIVOS.....	1
2) RESUMEN DEL PROYECTO.....	2
3) INSTALACION DE LA MEMORIA BIPOLAR DE 4x16 PALABRAS.....	3
a) Presentación de dirección a memoria.....	6
b) Escritura.....	7
c) Ubicación.....	10
TARJETA VI.....	10
TARJETA VII.....	12
TARJETAS VIII y IX.....	15
TARJETAS X, Y XI.....	17
4) REVISION CRITICA DEL PROCESADOR.....	19
MODIFICACIONES	
a) instrucciones rotación derecha e izquierda.....	20
b) instrucción salto por memoria.....	22
VERIFICACIONES	
a) instrucción transferir acumulador.....	27
b) instrucción salto por memoria.....	28
c) instrucción parar máquina.....	30
IMPLEMENTACIONES	
a) instrucciones poner y sacar desborde.....	30
TARJETA XII.....	32
5) OBSERVACIONES.....	34
6) FE DE ERRATAS.....	36
7) DIAGRAMA LOGICO DE LA MEMORIA.....	38
8) CONCLUSIONES.....	39
9) BIBLIOGRAFIA.....	40
10) APENDICE.....	41

1) OBJETIVOS

Previendo el avance en tecnología digital que se está produciendo a nivel nacional, se hace imprescindible el aporte humano y científico de la universidad. De tal manera se logrará no sólo la integración al medio sino también la independencia tecnológica.

Con estos objetivos se comenzó el desarrollo y construcción de una minicomputadora digital de propósito general, de la cual se implementaron las unidades de control y aritmética y la interfase con capacidad de interrupción.

La implementación de la memoria bipolar permitió poner a punto las señales de escritura, lectura y presentación de dirección, para emplearlas en la memoria semiconductora MOS de 12K palabras con la que contará la minicomputadora en un futuro próximo.

Se mejoró la performance del procesador con el diseño de dos microinstrucciones y la reimplementación de instrucciones ya existentes.

2) RESUMEN DEL PROYECTO

El conocimiento a fondo de las unidades aritmética y control y la interfase con capacidad de interrupción, construídas en tres proyectos anteriores, constituyó un paso previo a la implementación de este trabajo final.

El proyecto incluyó la instalación de una memoria bipolar de 4x16 palabras y la revisión crítica del procesador, que comprendió la modificación de algunas instrucciones y la implementación de otras.

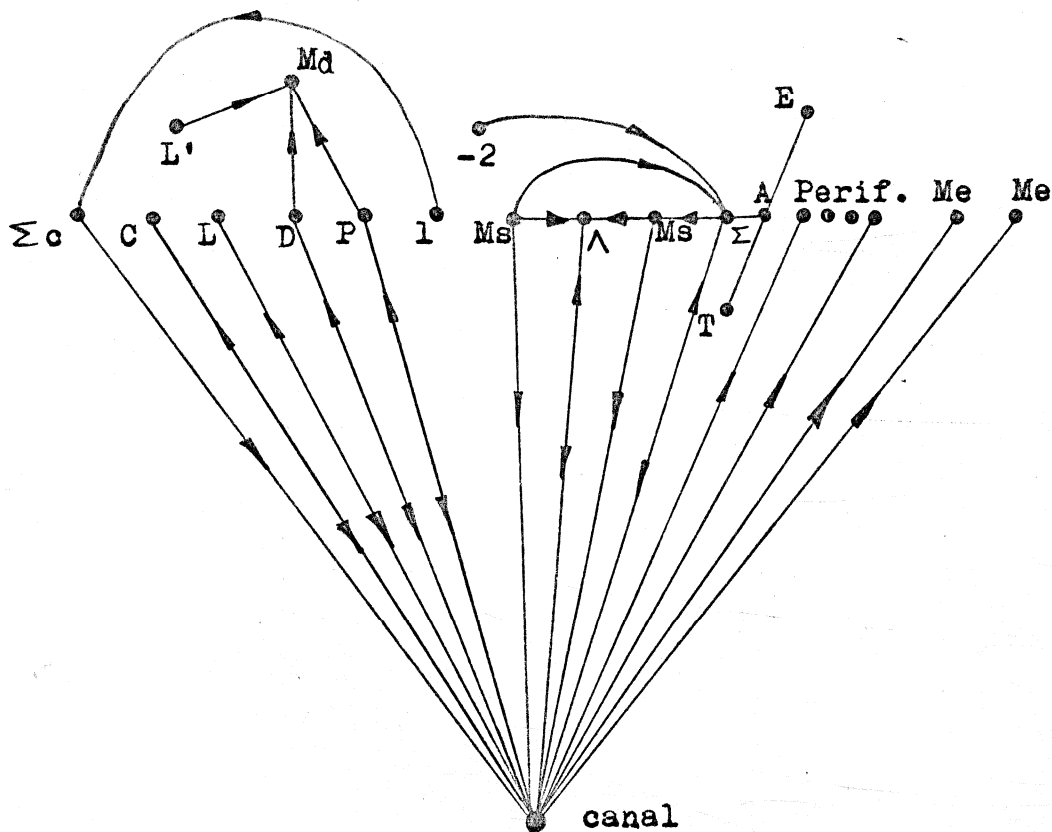
La colocación de la memoria trajo aparejada una verificación de las instrucciones que hacían uso de ella. Surgieron así problemas como retardos de tiempo, compuertas deterioradas, conexiones cambiadas, señales incorrectas y notación equivocada.

Si bien el tiempo empleado en su solución fue mucho, también lo fue la experiencia ganada en la interpretación de señales y detección de fallas, ayudado por el uso de programas de cuatro palabras o menos.

La implementación de modificaciones y nuevas instrucciones se complicó debido a la falta de espacio en las tarjetas y de lugar en la minicomputadora.

3) INSTALACION DE LA MEMORIA BIPOLAR

Con su instalación se cierran los caminos para el flujo de datos e instrucciones en el árbol lagrangiano que representa al sistema. Se tiene de esta forma:



Ma: entrada de dirección a memoria

Ms: salida de memoria

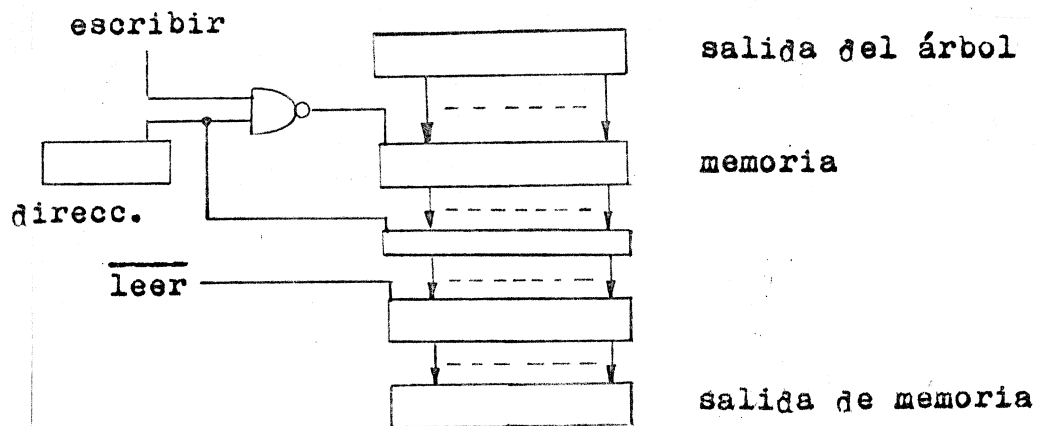
Me: entrada a memoria

L': llaves de dirección

L: llaves de información

Previo a su instalación, las llaves de dirección funcionaban como registro de salida de memoria. Las llaves de información no estaban conectadas, como se indica, para cargar instrucciones en el registro de instrucción presente.

En un diagrama bloque la memoria queda organizada de la siguiente forma:



Para escribir en memoria se presenta dirección y la señal de escribir; la intersección de ambas constituye el reloj de los registros. La información viene del árbol directo a la entrada D de los flip-flops. Para leer se habilitan las salidas Q por medio de la dirección; con la señal leer complemento, se tiene la información en los registros de salida de memoria. Las salidas de ésta entran directamente al árbol.

La presentación de dirección a memoria se hace a través de:

- 1) periféricos.
- 2) registro contador de programa.
- 3) byte cero de P concatenado con D.
- 4) llaves de dirección de consola.

1) La dirección de periféricos se fuerza a través de las líneas de C; aquella da la dirección de la primera instrucción de la subrutina para leer información o perforar resultados.

2) La dirección que aparece en C es la de la próxima dirección a ejecutar. En el frente de la máquina había ya implementado un registro indicador de dirección. La indicación correcta de la dirección presentada corresponde sólo a los bits 1 y 2; para los restantes hay que efectuar una verificación de las compuertas de presentación de dirección como se indicará más adelante.

3) Si hay búsqueda de operando se tiene para:

-direccionado directo:

a) a páginas 0, 1 y 2 se accede con PO concatenado con los bits de página.

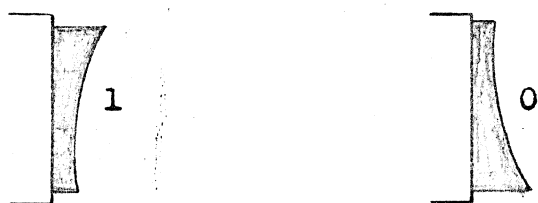
b) a página presente se accede con PO concatenado con D

-direccionado indirecto:

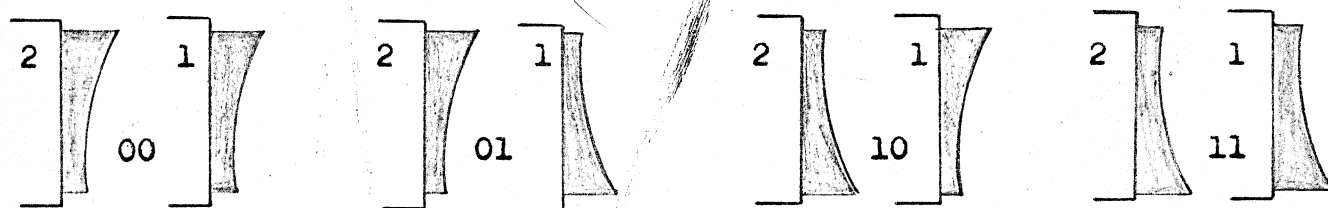
La primera dirección puede corresponder a a) o b); la información obtenida se dispone en PO concatenado con D y ésta será la dirección que se presentará a memoria para sacar el operando.

4) Con las llaves dispuestas en tablero se puede acceder a 2^{15} direcciones. Para la memoria instalada se usan sólo dos llaves ya que cuenta con cuatro registros. Se usan las llaves correspondientes a los bits 1 y 2 de dirección, ya que el bit cero en el registro C está para indicar si se accede al byte 0 ó 1 y en el registro P no tiene utilidad.

Las posiciones que deben adoptar las llaves para la entrada de un uno o un cero son respectivamente:



por lo tanto, las cuatro únicas direcciones pueden ser accedidas de la forma siguiente:



Esta forma de presentar la dirección se debe a que las llaves no entran complementadas a las compuertas de presentación de dirección, como se indica en informes anteriores, sino directamente.

Para escribir en memoria de tablero se coloca la instrucción en las llaves de información (ésta entra complementada al árbol) y una dirección en el registro de dirección. Se acciona luego el pulsador de escritura hacia arriba.

Para leer el contenido de memoria se presenta la dirección al mismo tiempo que aparece la señal de leer, que complementada actúa como reloj de los registros de salida de memoria apareciendo la información en tablero. Como el pulso de lectura no correspondía al indicado en el informe se implementó uno nuevo dispuesto en la tarjeta de presentación de dirección a memoria. Si el acceso a memoria es directo aparecen los pulsos Po y P4; para acceso indirecto Po, P2, P4 donde:

Po: presenta la dirección a memoria e incrementa C en 1.

P2: presenta PO concatenado con los bit de página (0, 1, 2)

Es conveniente analizar estos trenes de pulsos para facilitar la comprensión a posteriores proyectistas.

Con el primer Po Md<1> y Md<2> están en cero; por lo tanto se presenta la dirección del registro cero, dada por Md<1> y Md<2>. Con P4 el registro P presenta dirección y saca el operando. Como se tiene P<1> y P<2> en uno aparecen en Md<1> y Md<2> dos pulsos.

Con el segundo Po se presenta la dirección del registro uno dada por Md<1> y Md<2>. Como P<1> y P<2> se mantienen en uno lógico con P 4 aparecen nuevamente un pulso en Md<1> y otro en Md<2>. El análisis es similar para las otras dos direcciones.

La dirección de tablero no se presentaba correctamente. Se comprobó luego de solucionar el problema de escritura, que es tema de análisis posterior.

Cuando se presentaba una dirección la información extraída correspondía a otra. Condujo a suponer que no escribía bien en memoria; pero quedó descartado ya que revisando los registros éstos contenían la información previamente escrita.

La verificación se desvió a las compuertas de presentación de dirección imponiéndose el siguiente plan de trabajo: se supuso que las entradas al árbol de Po□D, C y LDA correspondían a las compuertas indicadas en el informe. Se fijó luego a que compuertas de presentación de dirección iban C y Po□D restando verificar que llave de dirección entraba. Las conexiones quedaron así:

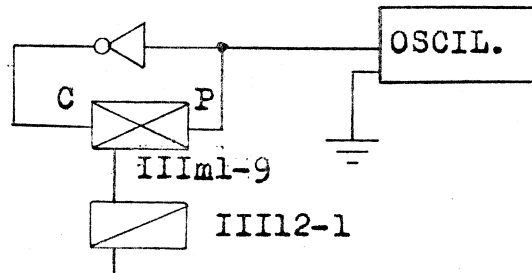
pin	con. ext.	conexión interior		función
5	consola	IIIa3-9	IIIa3-12	LDi<3>
6	"	IIIc1-14	IIIa3-9	LDi<0>
7	"	IIIe3-14	IIIc1-8	LDi<2>
8	"	IIIa3-13	IIIe3-7	LDi<1>

Se recomienda -usando el mismo procedimiento- revisar el resto de las entradas Ldi<i>.

b) ESCRITURA

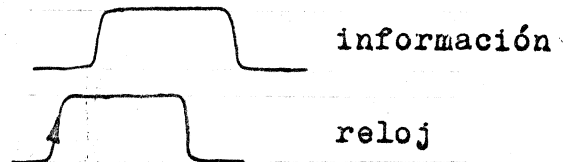
Como desde tablero no escribía se supuso que la información y la entrada de reloj llegaban desplazadas. Para poder visualizar

lo en el osciloscopio se empleó el siguiente circuito:

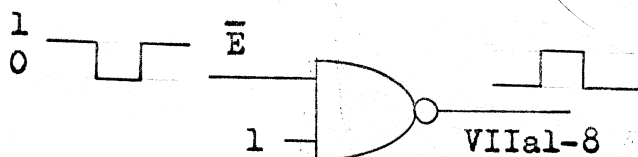


Cada vez que aparecía un pulso volcaba por clear el flip-flop a cero. Cuando por preset se volcaba a uno se accionaba el monostable IIII2-1.

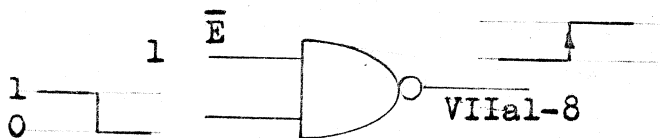
La señal de reloj y la información llegaban así:



Como los flip-flops cambian con el frente positivo no se escribía la información. Se pensó primero retrasar el pulso de reloj empleando una cadena de compuertas nand 7400 cuyos retardos típicos son de 15ns. Se consiguió escribir en memoria; sin embargo cuando escribía en un registro borraba lo que había escrito en el anterior. La explicación de este comportamiento es la siguiente: \bar{E} estaba siempre en uno lógico salvo cuando se pulsaba escribir en memoria de tablero:



Cuando presentaba dirección en el segundo registro en el primero aparecía:

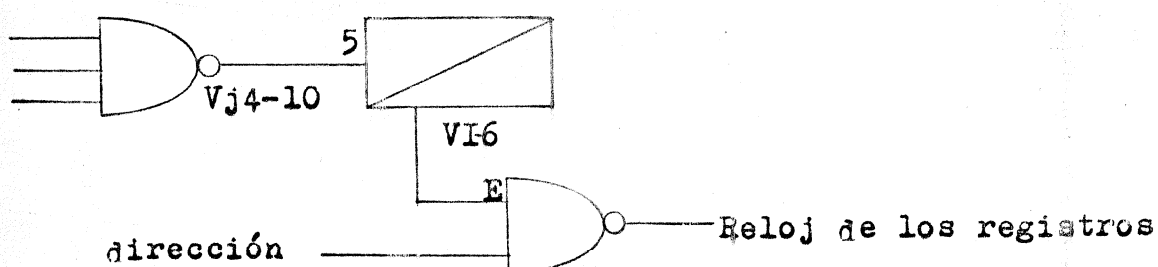


y el frente positivo borraba la información.

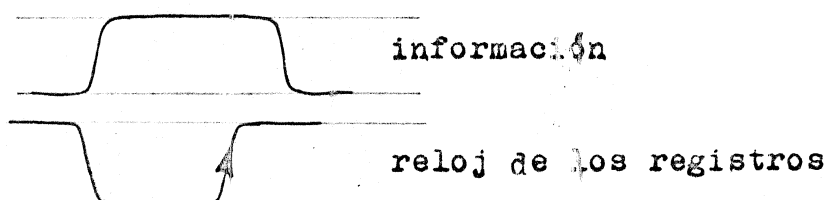
Para obviar este inconveniente hay que mantener la línea de \bar{E} en cero. Por lo tanto en lugar de tomar la salida IIIk6-10 se debió tomar la señal de Vj4-10.

Al hacer este cambio en algunos bits la información no llegaba junto con reloj. Se intercaló entre Vj4-10 y las entradas a

las compuertas que habilitan reloj de memoria un monostable ubicado en la tarjeta VI.

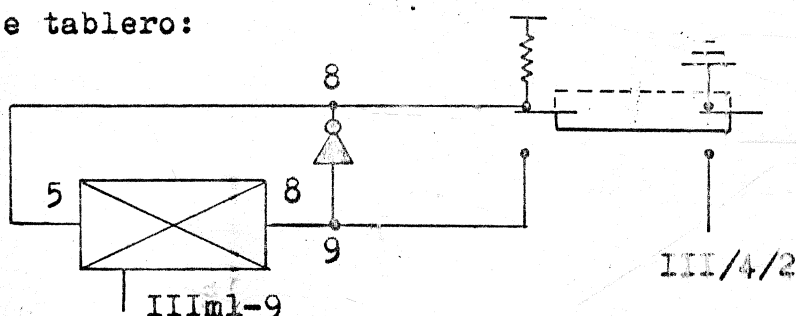


Con esa configuración el reloj y la información aparecían



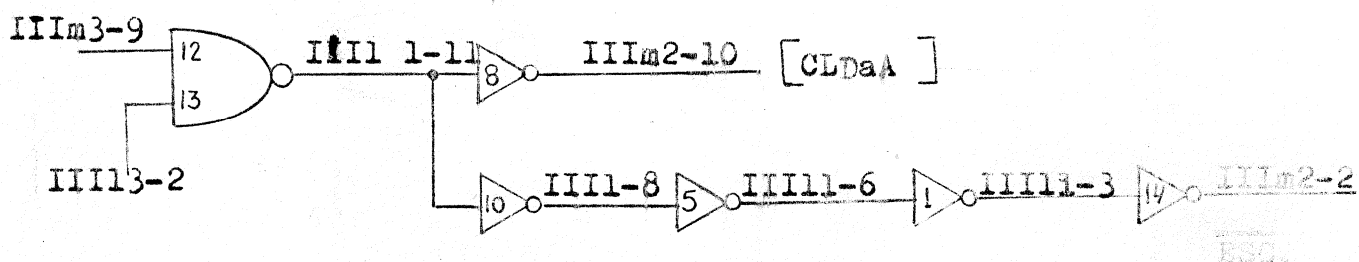
Se obvió la cadena de compuertas 7400 dejando sólo el monostable.

Corregido el problema de simultaneidad, debido a rebotes en la llave se implementó la siguiente modificación para escribir de tablero:



El inversor empleado está colocado en la parte posterior de la consola y continúa a la llave de arranque. De los seis inversores de esa compuerta 7404 se empleó uno sólo. La entrada que viene de III/4/2 corresponde al reloj del procesador.

El control que habilita la entrada, al canal, de los datos y constituye el pulso de escritura es el siguiente:



c) UBICACION

La memoria bipolar consta de cuatro registros de 16 bits compuestos de flip-flops D. Los registros están numerados de 0 á 3. Las tarjetas correspondientes están ubicadas en la parte inferior, en la caja de tarjetas de control de los equipos periféricos. Se identifican con numeración romana conteniendo:

VI: monostable correspondiente a la señal de escritura.

VII: presentación de dirección memoria. Generación del pulso de escritura.

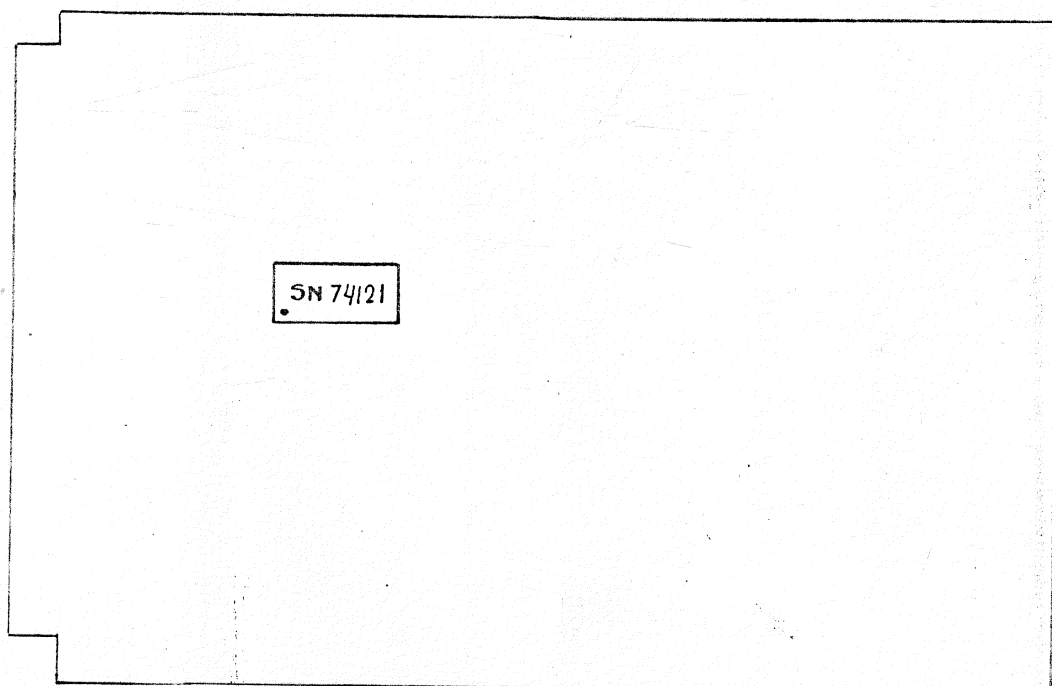
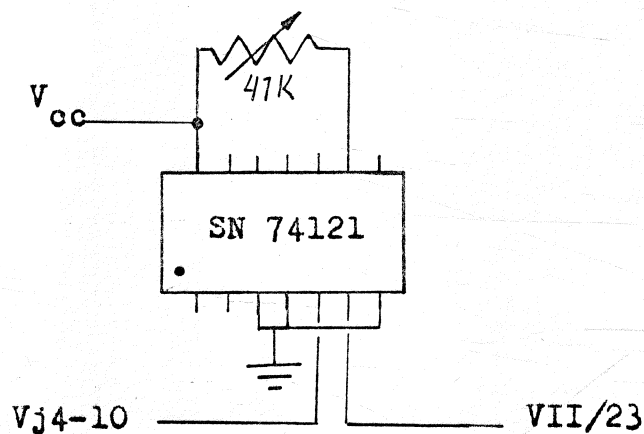
VIII: bit 0 a bit 7 de los cuatro registros. (primer byte)

IX: bit 8 a bit 15 de los cuatro registros. (segundo byte)

X: registro de salida de memoria del primer byte.

XI: registro de salida de memoria del segundo byte.

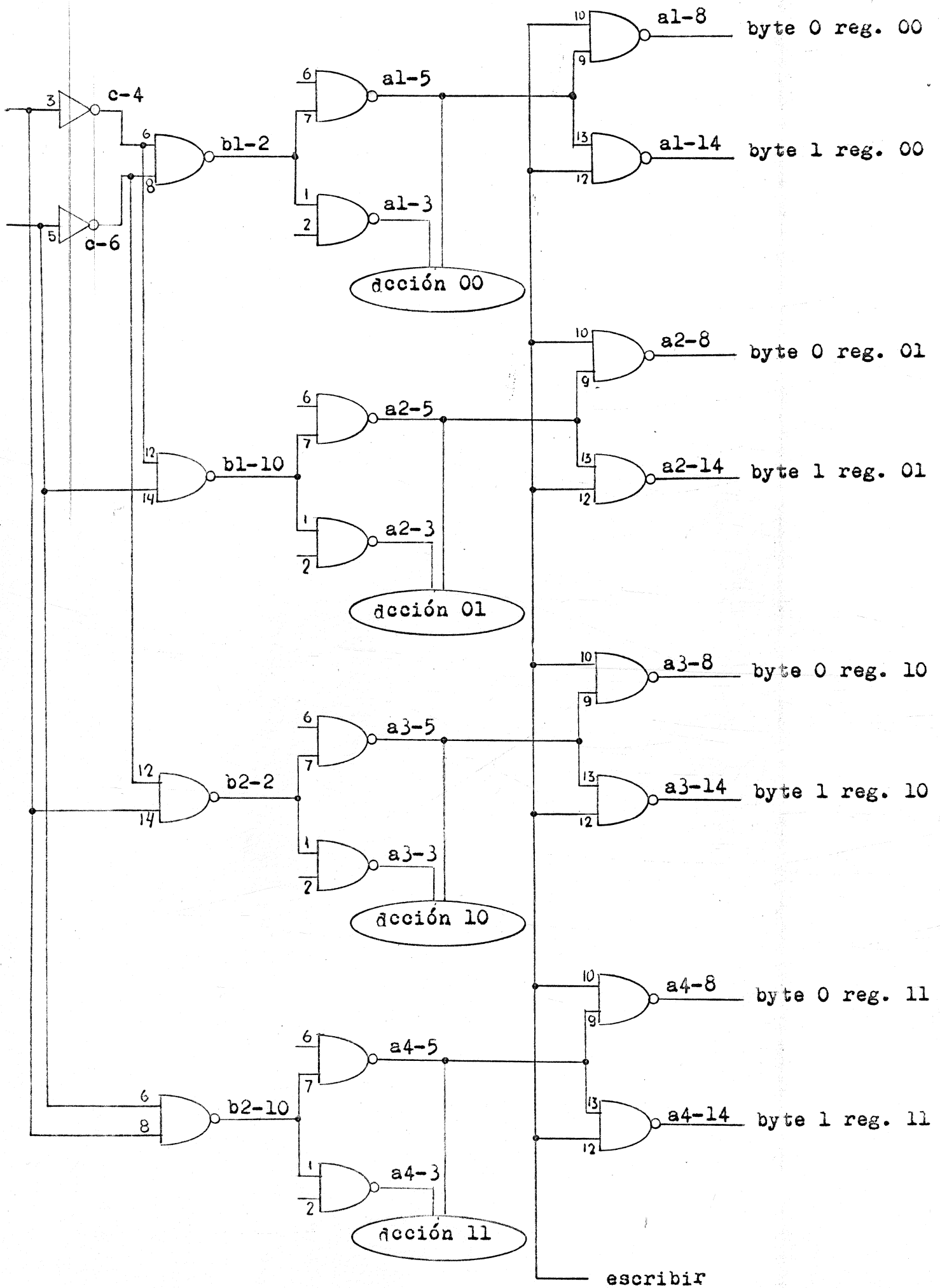
TARJETA VI: monostable para escritura

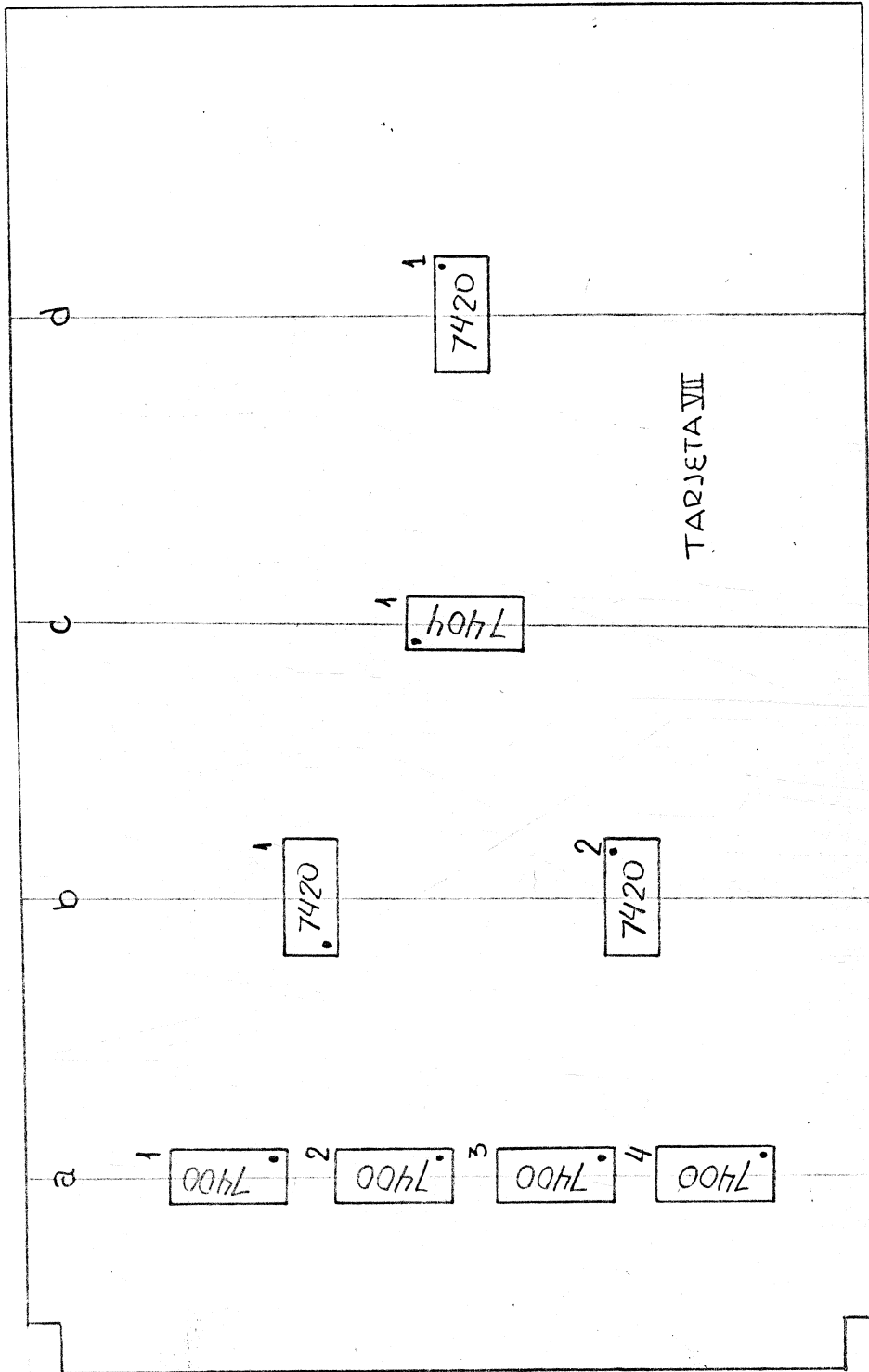
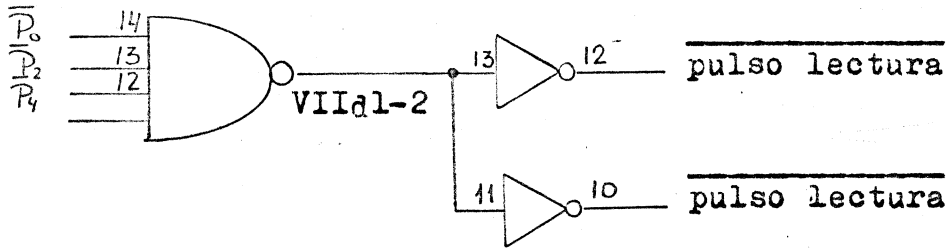


pin	con. exterior	conexion interior	función
1	V/5/5	VI-5	pulso esc.
2			
3			
4		VI-3	A ₁
5	VII/23	VI-6	pulso esc.
6			
7			
8			
9			
10			
11			
12			
13			
14		VI-7	tierra
15			
16			
17			
18			
19		VI-14	alimentación
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			

TARJETA VII: presentación de dirección a memoria

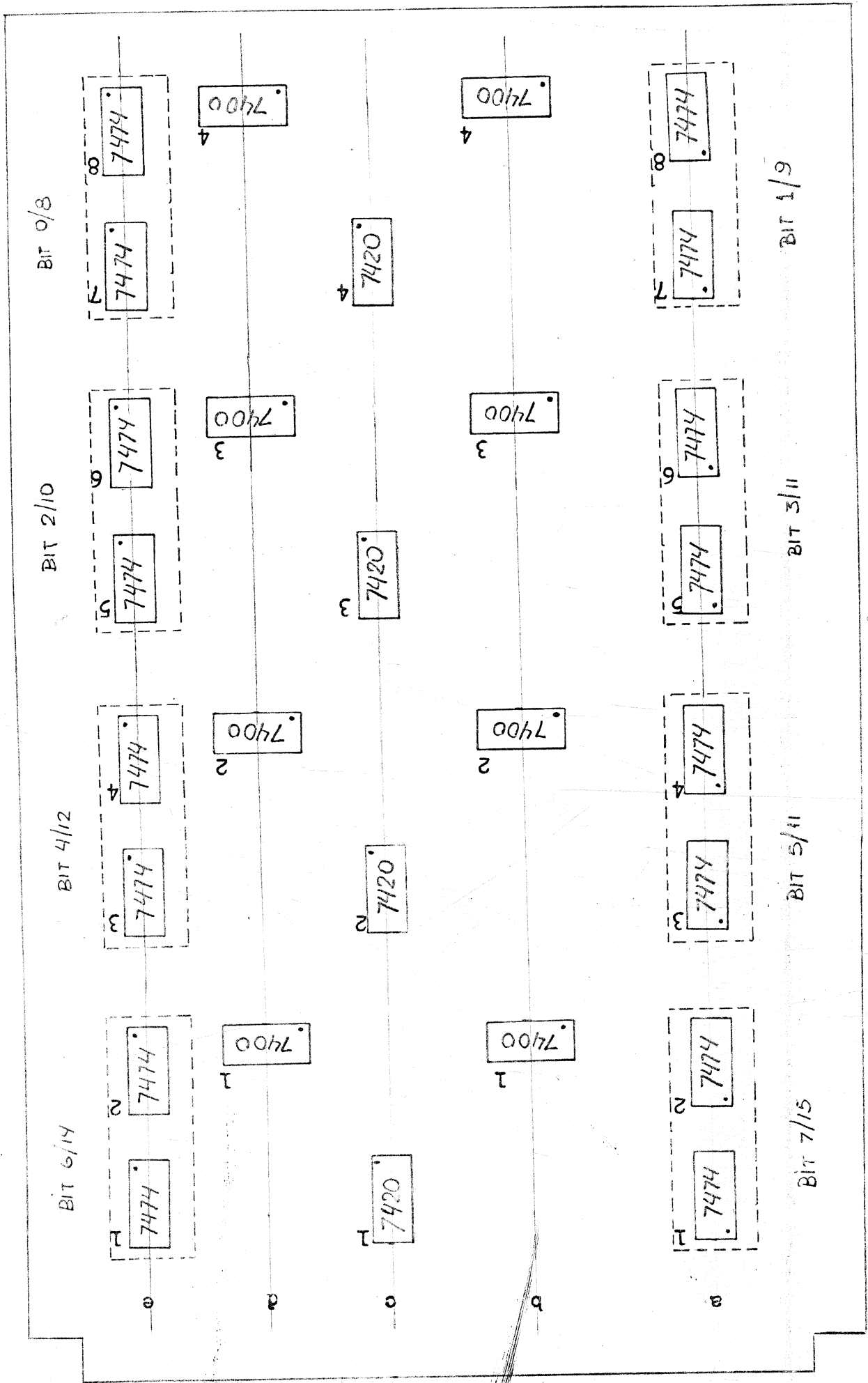
habilita Re





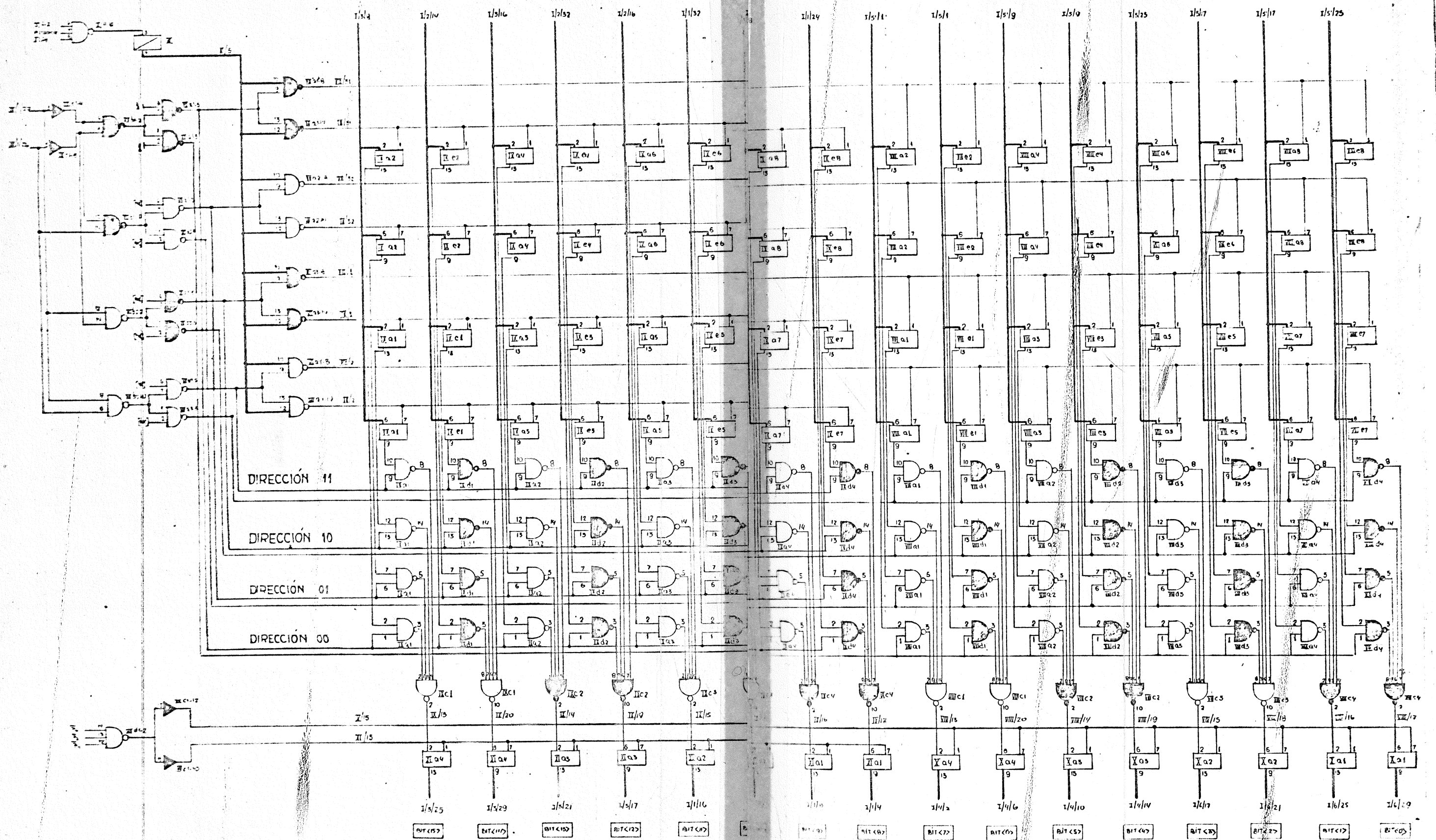
pin	con. exterior	conexión interior	función
1			alimentación
2	VIII/31	VIIa1-8	habilita Re 00
3	VIII/11	VIIa1-5	acción reg. 00
4	IX/11	VIIa1-3	acción reg. 00
5	IX/31	VIIa1-14	habilita Re 00
6	VIII/32	VIIa2-8	habilita Re 01
7	VIII/9	VIIa2-5	acción reg. 01
8	IX/9	VIIa2-3	acción reg. 01
9	IX/32	VIIa2-14	habilita Re 01
10			libre
11			libre
12	X/15	VIIc1-12	<u>lectura</u>
13	XI/15	VIIc1-10	<u>lectura</u>
14	NC	VIIc1-1	<u>Md<3></u>
15	NC	VIIc1-2	<u>Md<3></u>
16	III/1/27	VIIc1-3	<u>Md<2></u>
17	NC	VIIc1-4	<u>Md<2></u>
18	III/1/26	VIIc1-5	<u>Md<1></u>
19	NC	VIIc1-6	<u>Md<1></u>
20	V/3/8	VIIa1-14	<u>Po</u>
21	V/3/18	VIIa1-13	<u>P2</u>
22	V/3/11	VIIa1-12	<u>P4</u>
23			escribir
24	VIII/1	VIIa3-8	habilita Re 10
25	VIII/23	VIIa3-5	acción reg. 10
26	IX/23	VIIa3-3	acción reg. 10
27	IX/1	VIIa3-14	habilita Re 10
28	VIII/2	VIIa4-8	habilita Re 11
29	VIII/22	VIIa4-5	acción reg. 11
30	IX/22	VIIa4-3	acción reg. 11
31	IX/2	VIIa4-14	habilita Re 11
32			tierra

TARJETAS VIII/IX



pin	con. exterior		con. interior		función
1	VII/24	VII/27			Re reg. 10
2	VII/28	VII/21			Re reg. 11
3					alimentación
4	I/5/1	I/2/14	VIIIe1/2-2y6	IX ídem	inf.bit<6>/<14>
5	I/5/9	I/2/32	VIIIe3/4-2y6	IX "	inf.bit<4>/<12>
6	I/5/17	I/1/32	VIIIe5/6-2y6	IX "	inf.bit<2>/<10>
7	I/5/25	I/1/24	VIIIe7/8-2y6	IX "	inf.bit<0>/<8>
8					tierra
9	VII/7	VII/8	pin 6 de SN 7400		acción reg. 01
10	VII/25	VII/26	pin 13 de SN 7400		acción reg. 10
11	VII/3	VII/4	pin 1 de SN 7400		acción reg. 00
12	VII/29	VII/30	pin 9 de SN 7400		acción reg. 11
13	X/31	XI/31	VIIIc1-2	IXc1-2	salida bit<7>/<15>
14	X/25	XI/25	VIIIc2-2	IXc2-2	salida bit<5>/<13>
15	X/8	XI/8	VIIIc3-2	IXc3-2	salida bit<3>/<11>
16	X/3	XI/3	VIIIc4-2	IXc4-2	salida bit<1>/<9>
17	X/2	XI/2	VIIIc4-10	IXc4-10	salida bit<0>/<8>
18	X/9	XI/9	VIIIc3-10	IXc3-10	salida bit<2>/<10>
19	X/24	XI/24	VIIIc2-10	IXc2-10	salida bit<4>/<12>
20	X/30	XI/30	VIIIc1-10	IXc1-10	salida bit<6>/<14>
21	VIII/9	IX/9			acción reg. 01
22	VIII/12	IX/12			acción reg. 11
23	VIII/10	IX/10			acción reg. 10
24	VIII/11	IX/11			acción reg. 00
25					tierra
26	I/5/17	I/2/8	VIIIa7/8-2y6	IX ídem	inf.bit<1>/<9>
27	I/5/25	I/2/16	VIIIa5/6-2y6	IX ídem	inf.bit<3>/<11>
28	I/5/9	I/3/16	VIIIa3/4-2y6	IX ídem	inf.bit<5>/<13>
29	I/5/1	I/3/8	VIIIa1/2-2y6	IX ídem	inf.bit<7>/<15>
30					alimentación
31	VII/2	VII/5			Re reg. 00
32	VII/6	VII/9			Re reg 01

7) DIAGRAMA LOGICO DE LA MEMORIA BIPOLAR



a

1

7474

2

7474

3

7474

4

7474

TARJETA X

a

1

7474

2

7474

3

7474

4

7474

TARJETA XI

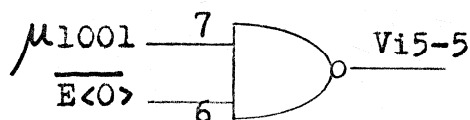
pin	con. exterior		con. interior		función
1					tierra
2	VIII/17	IX/17	Xa1-6	XIa1-6	entrada bit<0>/<8>
3	VIII/16	IX/16	Xa1-2	XIa1-2	entrada bit<1>/<9>
4	I/6/29	I/1/4	Xa1-9	XIa1-9	Q bit<0>/<8>
5			Xa1-10	XIa1-10	\bar{Q} bit<0>/<8>
6			Xa1-12	XIa1-12	\bar{Q} bit<1>/<9>
7	I/6/25	I/1/8	Xa1-13	XIa1-13	Q bit<1>/<9>
8	VIII/15	IX/15	Xa2-6	XIa2-6	entrada bit<3>/<11>
9	VIII/18	IX/18	Xa2-2	XIa2-2	entrada bit<2>/<10>
10	I/6/17	I/1/16	Xa2-9	XIa2-9	Q bit<3>/<11>
11			Xa2-10	XIa2-10	\bar{Q} bit<3>/<11>
12			Xa2-12	XIa2-12	\bar{Q} bit<2>/<10>
13	I/6/21	I/1/2	Xa2-13	XIa2-13	Q bit<2>/<10>
14					libre
15			pines 1 y 7 7474		reloj
16					libre
17					libre
18					libre
19					libre
20	I/4/14	I/3/17	Xa3-9	XIa3-9	Q bit<4>/<12>
21			Xa3-10	XIa3-10	\bar{Q} bit<4>/<12>
22			Xa3-12	XIa3-12	\bar{Q} bit<5>/<13>
23	I/4/10	I/3/21	Xa3-13	XIa3-13	Q bit<5>/<13>
24	VIII/19	IX/19	Xa3-6	XIa3-6	entrada bit<4>/<12>
25	VIII/14	IX/14	Xa3-2	XIa3-2	entrada bit<5>/<13>
26	I/4/6	I/3/25	Xa4-9	XIa4-9	Q bit<6>/<15>
27			Xa4-10	XIa4-10	\bar{Q} bit<6>/<15>
28			Xa4-12	XIa4-12	\bar{Q} bit<7>/<14>
29	I/4/2	I/3/29	Xa4-13	XIa4-13	Q bit<7>/<14>
30	VIII/20	IX/13	Xa4-6	XIa4-6	entrada bit<6>/<15>
31	VIII/13	IX/20	Xa4-2	XIa4-2	entrada bit<7>/<14>
32					alimentación

4) REVISION CRITICA DEL PROCESADOR

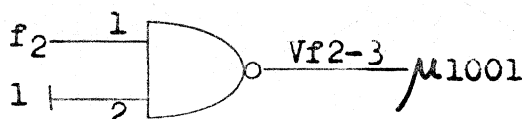
Un problema ineludible, previo a todas las modificaciones, fue solucionar la falla que impedía el correcto funcionamiento del contador.

El planteo básico seguido fue el siguiente: revisar las entradas (D, clear, preset) de cada uno de los flip-flops que componen el contador. De esta forma se fue rastreando hasta comprobar que $\overline{E<0>}$ correspondiente a la señal $\overline{E5(b)}$ proveniente de la tarjeta IV presentaba siempre un uno lógico, debido a que existían conexiones sueltas por malas soldaduras.

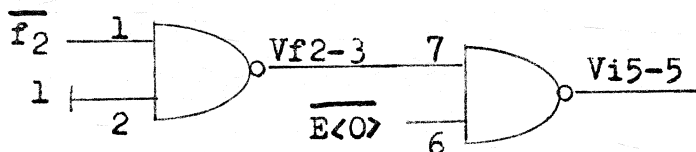
Solucionado este inconveniente se observó que el contador sólo funcionaba cuando $\overline{E<0>}=0$. Como su variación interesa sólo para la microinstrucción sb, que no estaba presente en el momento de la prueba, el paso inmediato fue comprobar que valor entraba al pin 7 de la compuerta Vi5-5



Como en ella había una variación de 2V cuando en realidad debía haber un cero condujo a verificar su decodificación. El problema se presentó en la compuerta Vf2-3; una compuerta 7400 se usaba como inversora dejando uno de los pines al aire, produciendo en la salida la variación de tensión mencionada. Se clavó esa entrada a uno lógico:



en esas condiciones se tiene:



no

Cuando se hace presente la microinstrucción sb en vf2-3 aparece un cero lógico, y por lo tanto, para cualquier valor que tome $\overline{E<0>}$ en Vi5-5, se tiene un uno lógico.

De esta manera quedó solucionado el funcionamiento del contador.

La revisión del procesador implicó:

modificaciones:

- a) en las microinstrucciones $ri(\mu 0110)$ y $rd(\mu 0111)$ el bit transporte T debe copiar al bit signo.
- b) la instrucción $sm(0101)$ debe ejecutar lo siguiente:

$$\begin{aligned} M[z] \geq 0 & \quad M[z] \leftarrow M[z] - 2 \\ M[z] < 0 & \quad C \leftarrow C + 2 \end{aligned}$$

verificaciones:

- a) de la instrucción ta
- b) de la instrucción sm
- c) de la microinstrucción pm

implementaciones:

- a) de las microinstrucciones $pd(\mu 1011)$ y $sd(\mu 1010)$

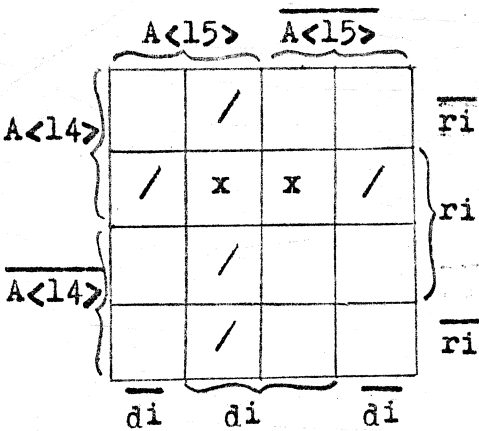
MODIFICACIONES

a) ROTACION IZQUIERDA Y ROTACION DERECHA

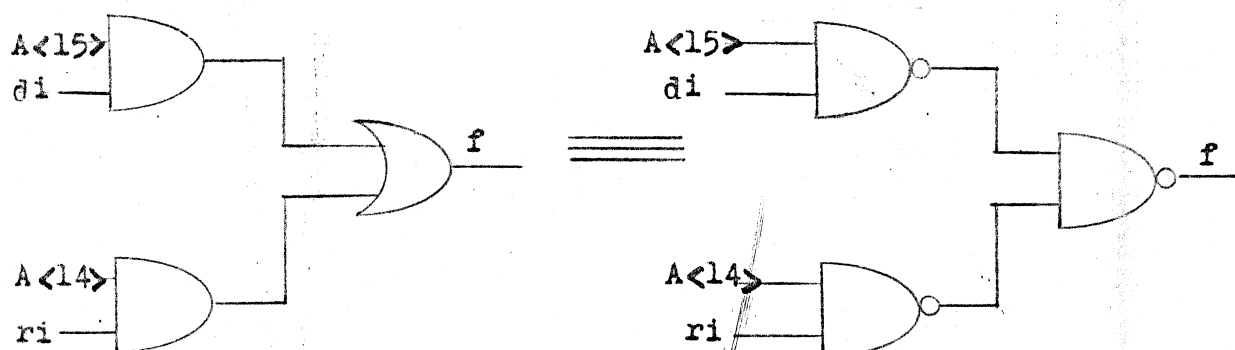
La modificación implementada fue necesaria para poder detectar desborde cuando se opera con un número rotado. En la rotación izquierda el bit T debe copiar el mismo valor que $A<15>$. Para la rotación derecha T debe copiar $A<0>$.

La implementación hecha es la siguiente:

$A<14>$	$A<15>$	\overline{di}	ri	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	x
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	x
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	x
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	x



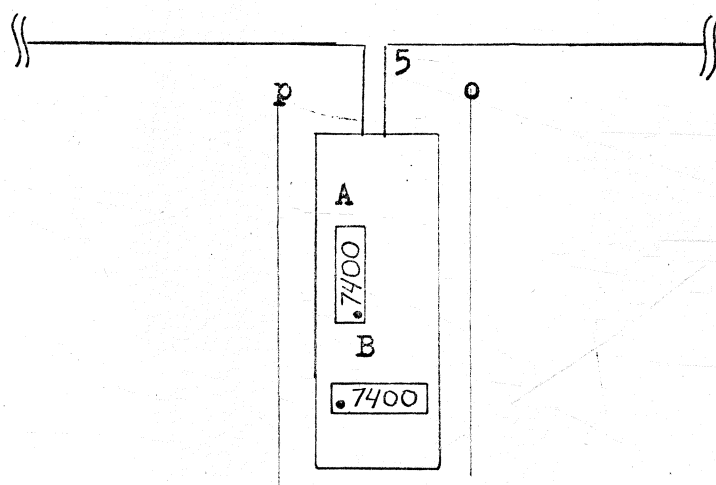
$$f_A = A<15> \cdot \overline{di} + A<14> \cdot ri$$



Se procede de idéntica forma para la rotación derecha quedando la función:

$$f_B = T.dd + A<0>.rd$$

La modificación se hizo teniendo en cuenta lo que ya se había construido. Para evitar un excesivo cableado los módulos empleados se ubicaron sobre una pequeña tarjeta colocada sobre la tarjeta IV. Las conexiones se muestran en la fig. 1.



La prueba inicial con este diseño no resultó. Luego de revisar las conexiones se comprobó que no había correspondencia entre la notación y las salidas de los decodificadores correspondientes a las microinstrucciones presentes. Hecha la corrección se tiene:

microinstrucción	salida
rd	Vel-8
ri	Vel-12
dd	Vf2-11
di	Vel-6

Posteriormente se presentó el problema de que cuando se rotaba a izquierda la información se copiaba en A<15>, T y A<0> al mismo tiempo. Se detectó la falla usando el siguiente programa:

en acumulador se cargó 0100 0000 0000 0000

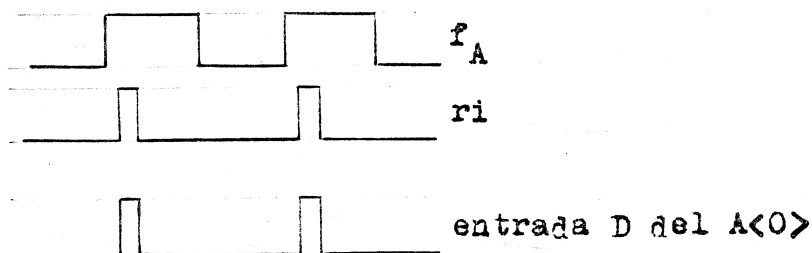
00 ri if

01 ca en forma directa con el contenido de la dirección 11

10 si directo a la acción 00

11 0100 0000 0000 0000

observándose que a la entrada D del bit A<0> llegaba fA en lugar de hacerlo A<15> de la forma siguiente:



Por la misma causa cuando había desplazamiento derecha se borraba la información que pasaba de A<15> a A<14>.

b) SALTO POR MEMORIA

Se reimplementó para poderlo usar como registro índice.

La primera modificación pensada

$$\begin{array}{ll} M[z] \geq 0 & M[z] \leftarrow M[z] - 1 \\ M[z] < 0 & C \leftarrow C + 2 \end{array}$$

permitía reducir el número de instrucciones de un lazo. Sea un ejemplo demostrativo: se quieren llevar 10 números de una posición de memoria 100 a 109 a otra 200 a 209; el programa constaría de las siguientes instrucciones:

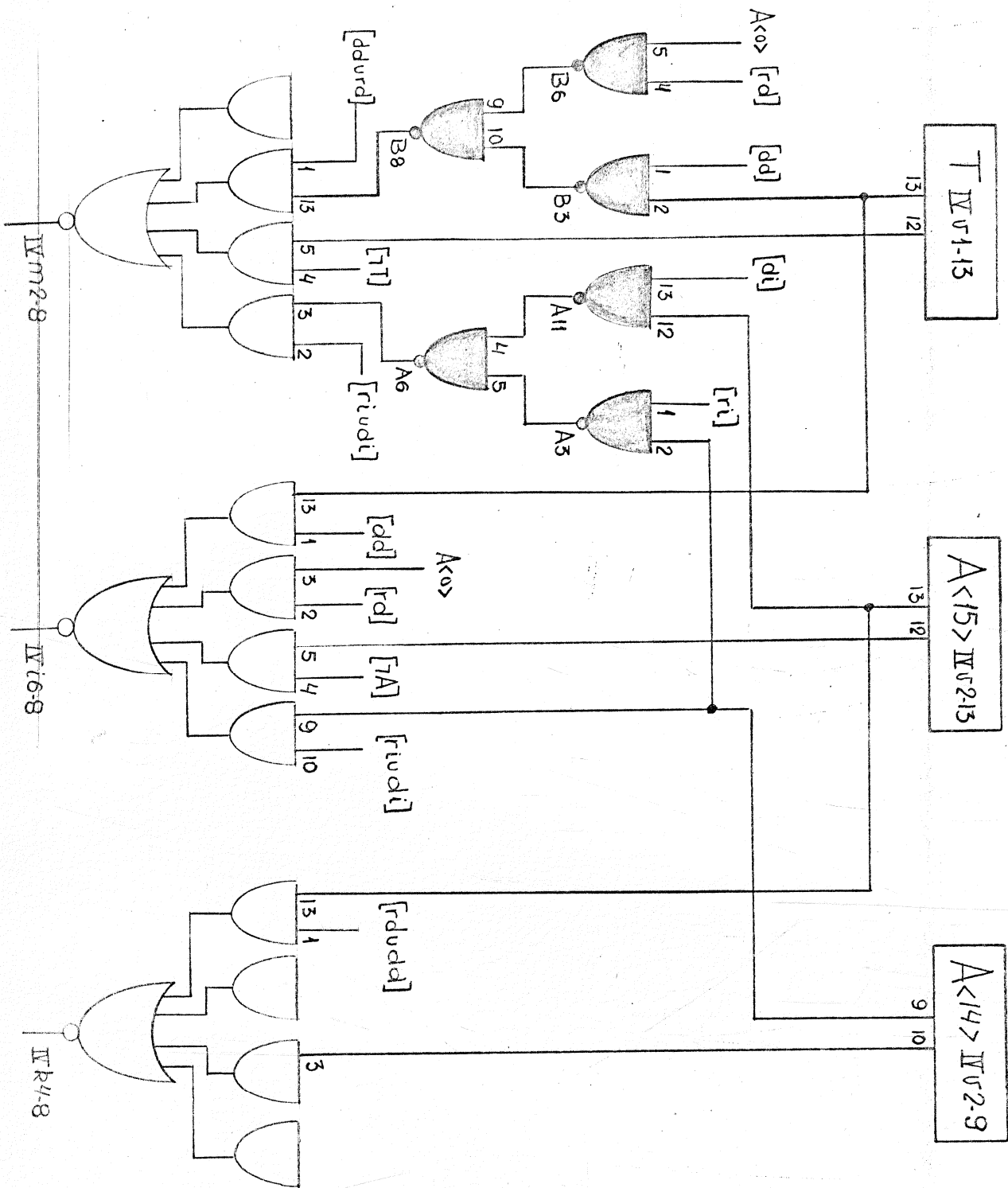
$z_1 = 99$

$z_2 = 200$

$z_3 = -9$

ca con el contenido de z_1
 au if
 ta con el contenido de z_1
 ca indirectamente con el contenido de z_1
 ta indirectamente con el contenido de z_2
 ca con el contenido de z_2

fig. 1



compuertas incluidas.

```

    au  if
    ta  con el contenido de z2
    sm
    si

```

con la modificación se haría lo mismo de la forma siguiente:

```

    ca  indirectamente con el contenido de z1
    ta  indirectamente con el contenido de z2
    sm  con el contenido de z1
    sm  con el contenido de z2
    sm
    si

```

Sin embargo esto acarreará un inconveniente en el direccionado indirecto; supóngase que se quiere realizar el mismo programa anterior. La presentación de dirección a memoria se hace con los bits 1 á 7, por lo tanto si se resta uno solamente, se accede al mismo lugar de memoria -byte uno - en el siguiente paso.

Lo que puede hacerse es: restar uno cuando cuenta y restarle dos cuando es acceso indirecto.

Una posible solución sería usar elbit 8 del registro de instrucción presente: un estado para cuenta y otro para el acceso. Otra posibilidad sería ordenar los saltos por memoria, llevar el número de saltos para dirección y restár dos y cuando llega salto por memoria para cuenta restar uno. Estas dos formas se descartarán optándose por la siguiente:

$$\begin{array}{ll}
 M[z] \geq 0 & M[z] \leftarrow M[z] - 2 \\
 M[z] < 0 & C \leftarrow C + 2
 \end{array}$$

y cuando se quiere hacer cuenta se pone el doble de ciclos que se van a ejecutar.

De esta forma la instrucción del grupo II se transformó pasando a integrar el grupo III; y la del grupo III al II. Los pulsos correspondientes a la fase de operación son:

grupo	instrucción	pulso	función
II	$M[z] \geq 0$	$P_5 - P_8$	habilita el control de sumar -2 a memoria en el sumador y la salida

grupo	instrucción	pulso	función
		\overline{P}_8	del mismo al canal escribe en memoria.
III	$M[z] < 0$	P_5 P_7 y P_8 \overline{P}_7 y \overline{P}_8	salta a P_7 habilitan al árbol la compuerta de salida del banco sumador de C actúan como reloj de C

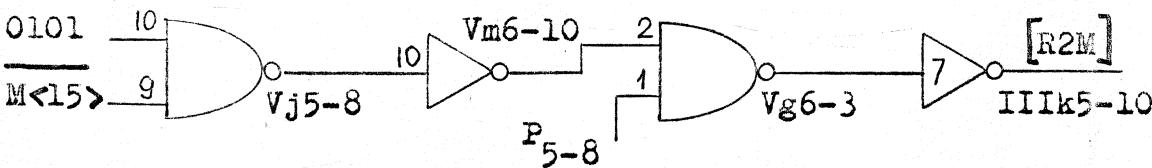
Veamos como funciona el sumador SN 7480 para comprender el conexionado. Haciendo un análisis detallado para suma y resta entre los distintos registros se concluye:

reg. A_2	suma: se complementan los impares y $C_{nl} = 0$ resta: " " " " y $C_{nl} = 1$
reg. A_1	suma: " " " pares y $C_{nl} = 0$ resta: " " " " y $C_{nl} = 1$
reg. B_1	suma: " " " " y $C_{nl} = 0$ resta: " " " imp. y $C_{nl} = 1$
reg. B_2	suma: " " " " y $C_{nl} = 0$ resta: " " " pares y $C_{nl} = 1$

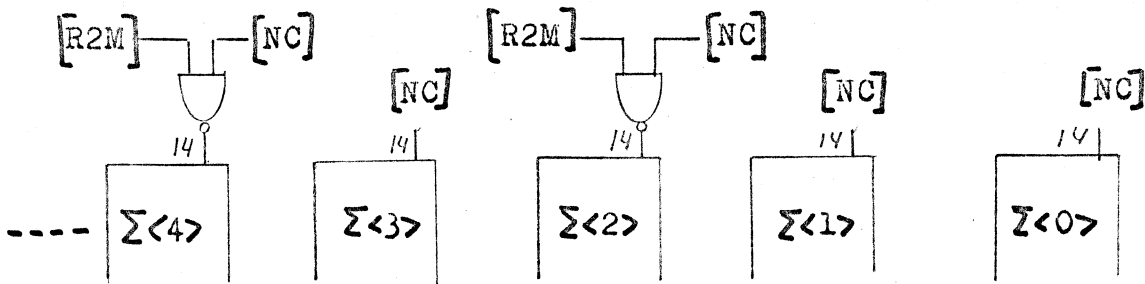
Para restar dos a memoria se necesitaría modificar el di seño hecho; para evitarlo se suma -2 (111....10) y se comple- mentan los impares del registro A_2 .

.....1	1	1	1	0
.....0	1	0	1	1

La señal de sumar uno a memoria se transforma en



Y se tiene así:

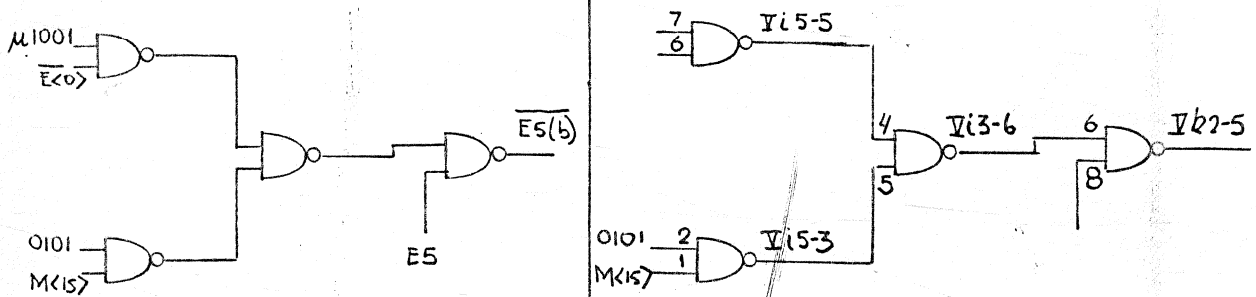


y la tabla de modificaciones correspondiente:

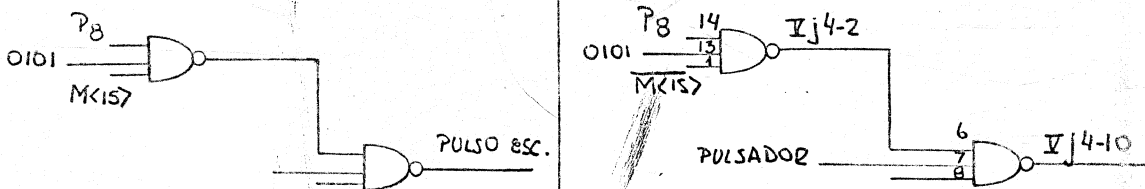
IVb1-3	de IVa2-14 → IVc2-14	IVq3-3	de IVg2-14 → IVr2-14
IVb1-6	NC	IVg3-6	de IVs2-14 → IVt3-14
IVb1-11	de IVd2-14 → IVe2-14	IVg3-8	de IVn2-14 → IVv3-14
IVb1-8	de IVf2-14 → IVg2-14	IVg3-11	NC
IVf1-6	de IVh2-14 → IVp3-14	IVf1-3	NC

Se cambiaron además los siguientes controles:

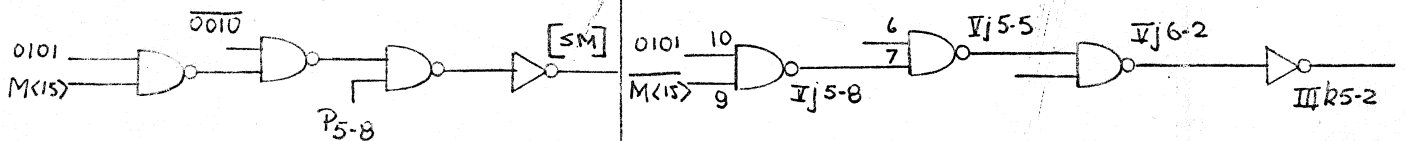
a) para las instrucciones del tipo III



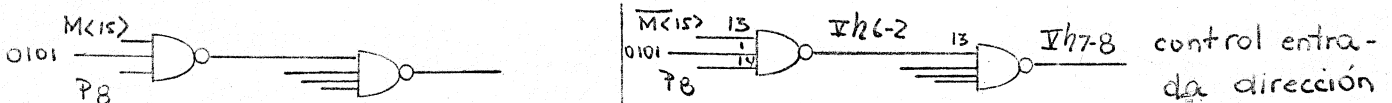
b) para escribir en un registro una vez presentada la información



c) para comandar el banco de operaciones



d) control de entrada de dirección



El correcto funcionamiento de memoria se obtuvo luego de hacer una verificación de las instrucciones que tenían acceso a ella y que se verá en los párrafos siguientes.

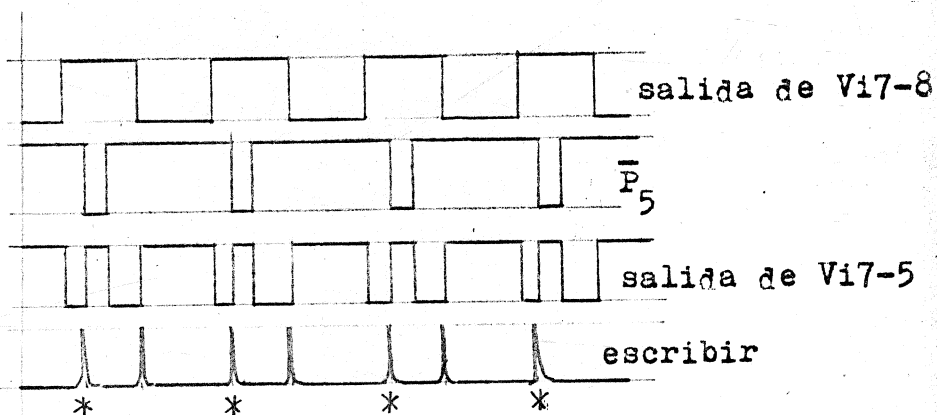
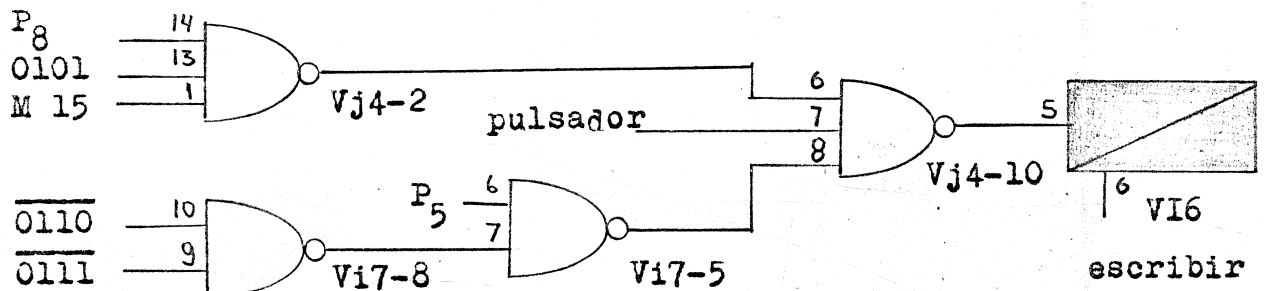
VERIFICACIONES

a) TRANSFERIR ACUMULADOR

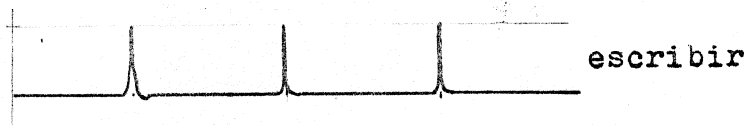
Para comprobar el funcionamiento de la instrucción ta (0110) se hizo el siguiente programa:

```
00  if  if
01  ca  con el contenido de la dirección 00
10  ta  a la dirección 00
11  si  a la dirección 10
```

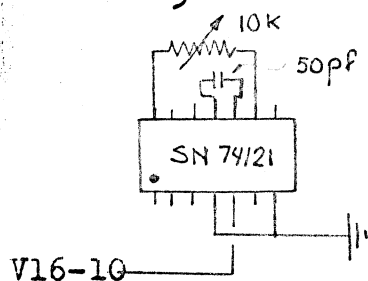
Cuando se presentaba la instrucción ta el contenido de la dirección 00 se borraba. Se detectó que:



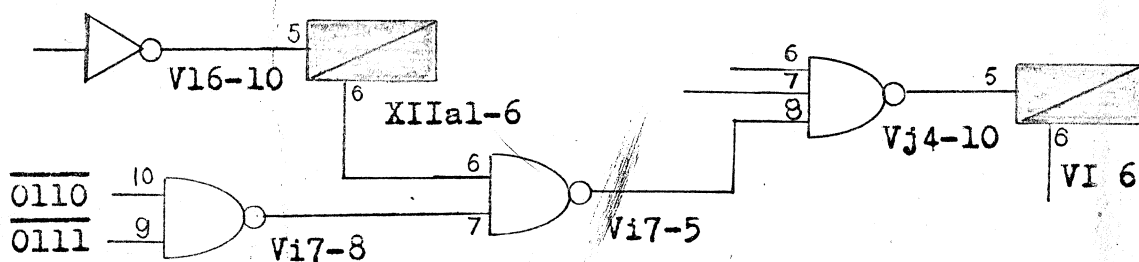
la señal de escribir era incorrecta debido a que a la compuerta Vi7-5 entraba el complemento de P_5 originando los pulsos * que borraban la información del registro 00. Se cambió apareciendo entonces



Sin embargo, esta señal y la presentación de dirección llegaban desplazadas; se retrásó P_5 con un monostable con esta configuración



ubicado sobre la tarjeta XII en la parte posterior de la máquina.



b) SALTO POR MEMORIA

El análisis fue hecho antes del cambio siguiente

$$M[z] \geq 0 \quad M[z] \leftarrow M[z] - 2$$

$$M[z] < 0 \quad C \leftarrow C + 2$$

es decir, se verificó teniendo

$$M[z] < 0 \quad M[z] \leftarrow M[z] + 1$$

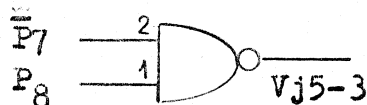
$$M[z] \geq 0 \quad C \leftarrow C + 2$$

o sea, en un paso intermedio entre la implementación original y la implementación final.

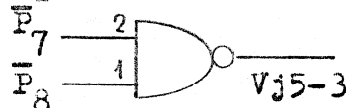
Se cumplió en dos etapas:

$$1) M[z] \geq 0 \quad C \leftarrow C + 2$$

En los cuatro registros de memoria se cargó la instrucción sm; primero se ensayó por pulsos observándose la falta de uno de ellos. Se colocó la máquina en modo continuo y se comprobó que faltaba P_8 a la salida de la compuerta Vj5-3, debido a que no entra su complemento al pin uno, como se muestra en la figura



corregido quedó:



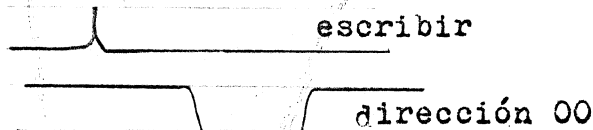
$$2) M[z] < 0 \quad M[z] \leftarrow M[z] + 1$$

En esta etapa fue necesario hacer uso del siguiente programa

00 if if
 01 ta a la dirección 00
 10 sm sobre la dirección 00
 11 si a la dirección 01

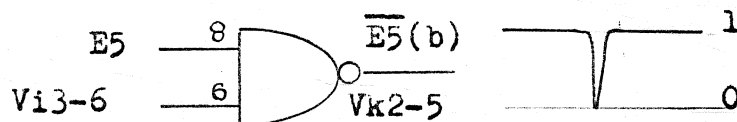
se cargó en el acumulador if if

La primera impresión conujo a pensar que la presentación de dirección y la señal de escritura no llegaban juntas, tal como ocurría con la instrucción ta. Sin embargo como el desfase era de aproximadamente 200ns

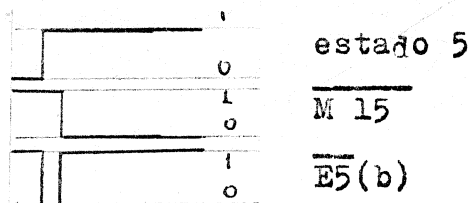


se intuyó que alguna conexión era incorrecta.

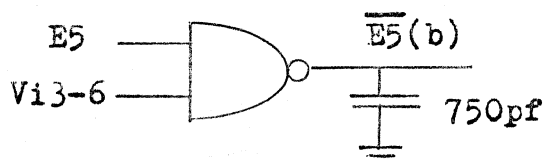
El pulso de información que debía aparecer al sumar uno a memoria no era tal. Se revisaron sus circuitos y se verificó si también pasaba el pulso ancho P_{5-8} , notando para esto último que el contador pasaba del estado cinco al estado siete, debido a que en $Vk2-5$ aparecía una señal que volcaba los flip-flops.



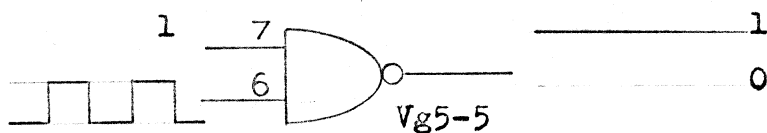
Esta señal se generaba por la intersección de $\overline{M<15>}$ y E5.



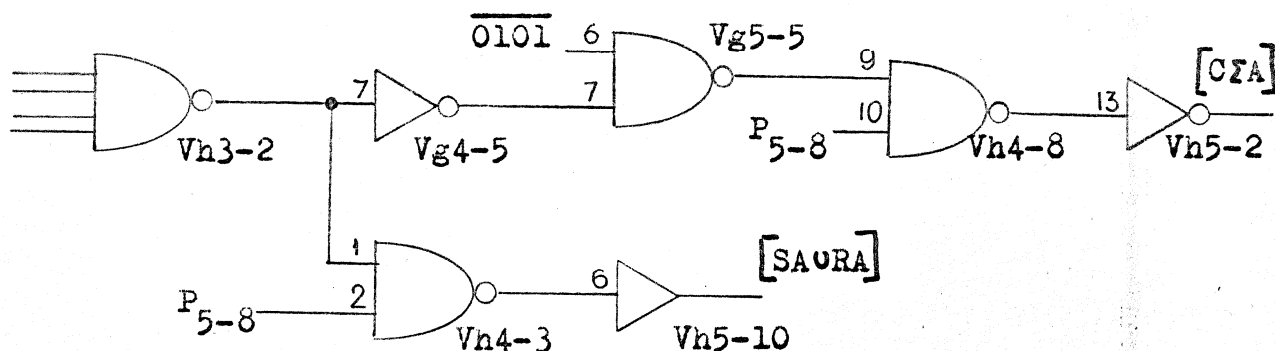
Se intentó retrasar E5 con una cadena de compuertas; como no fue suficiente, se colocaron tres condensadores de 250pf.



A pesar de solucionarlo, no escribía en memoria. Se revisó el control de entrada al sumador comprobándose que la salida de la compuerta $Vg5-5$ estaba clavada a 1 lógico, pese a que sus entradas eran



Se cambió el módulo Vg5 y se revisaron las conexiones [CZA] y [SAURA], comprobándose que estaban intercambiadas. Estas señales corresponden ahora a

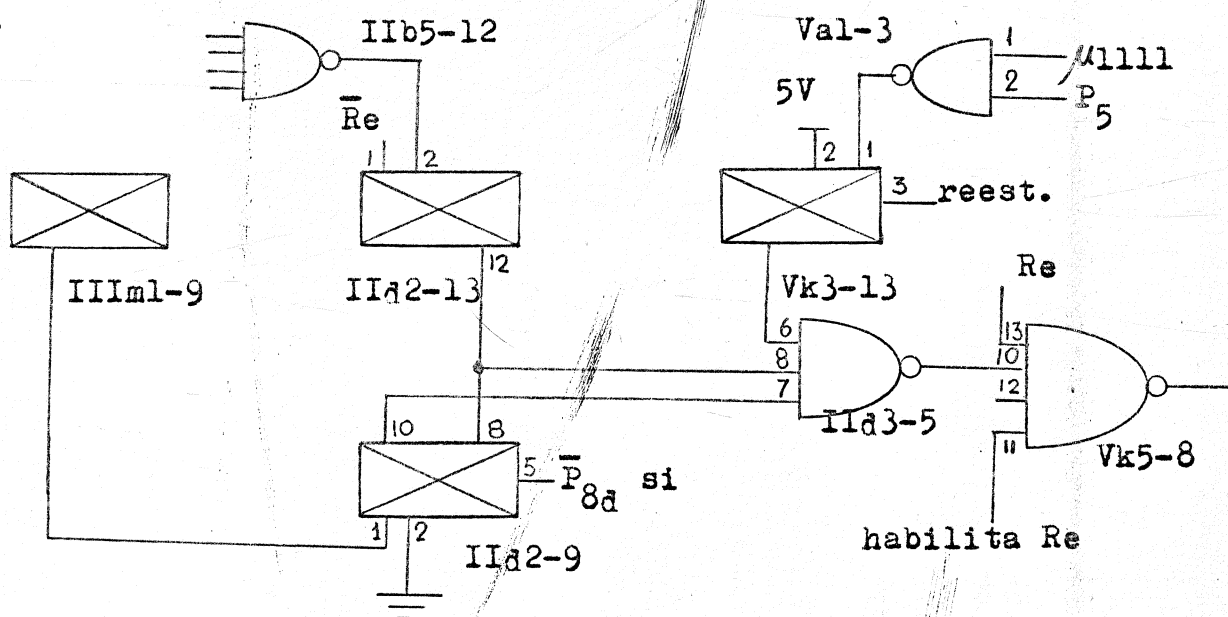


Posteriormente se hizo la modificación

$$\begin{aligned} M[z] \geq 0 & \quad M[z] \leftarrow M[z] - 2 \\ M[z] < 0 & \quad C \leftarrow C + 2 \end{aligned}$$

c) PARAR MAQUINA

Al poner en marcha la minicomputadora el flip-flop IIa2-9 se volcaba a uno lógico(Q), impidiendo el obedecimiento de la microinstrucción pm. Para evitar este inconveniente con el pulso de escritura se vuelca el flip-flop a cero lógico(Q). De esta manera cuando aparezca pm se detendrá el reloj de la máquina.



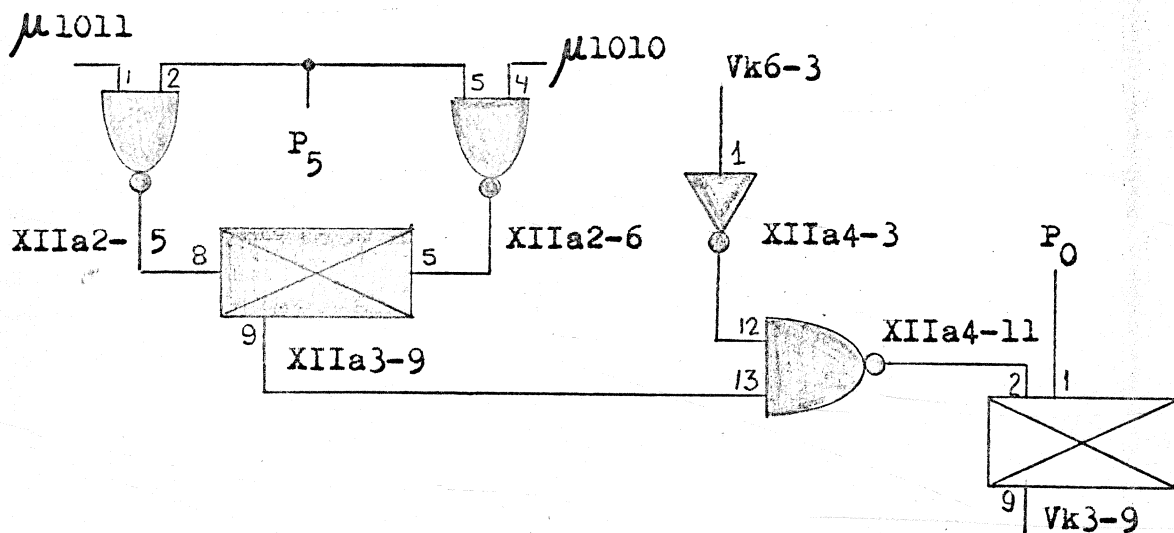
IMPLEMENTACION

a) PONER DESBORDE Y SACAR DESBORDE

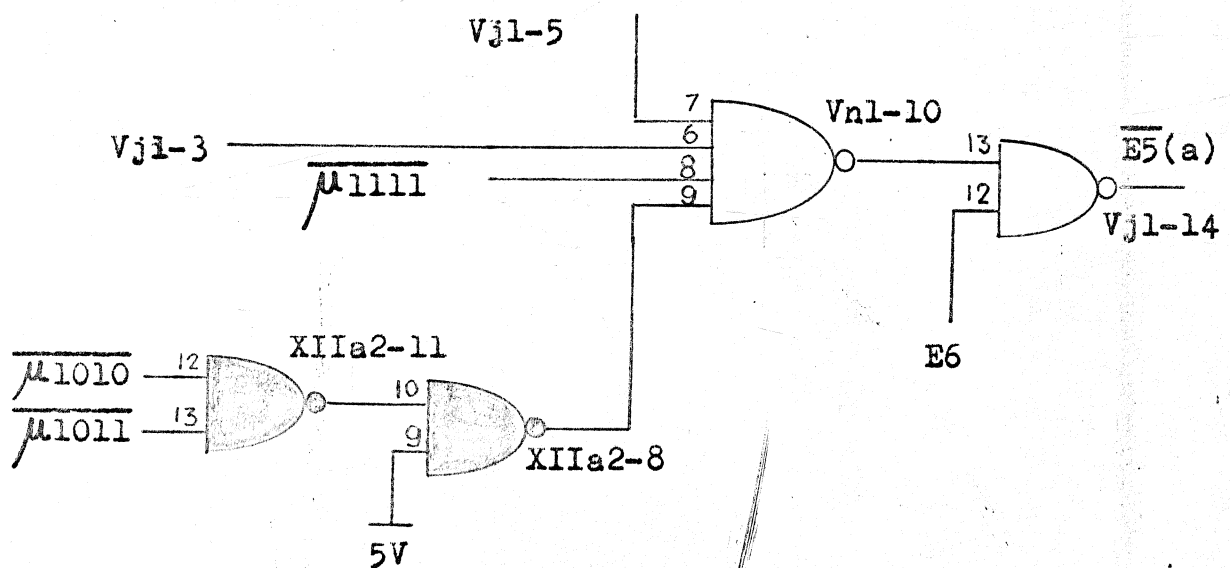
De las primitivas implementaciones a ejecutar: di lógico sin desborde (para generación de números al azar) y tratar a desborde como un equipo periférico (puede ser una futura mejora) se concluyó que sería conveniente implementar dos microinstrucciones: poner desborde(pd) y sacar desborde (sd).

Son instrucciones del grupo I y el código empleado fue:
sd(μ 1010) y pd(μ 1011).

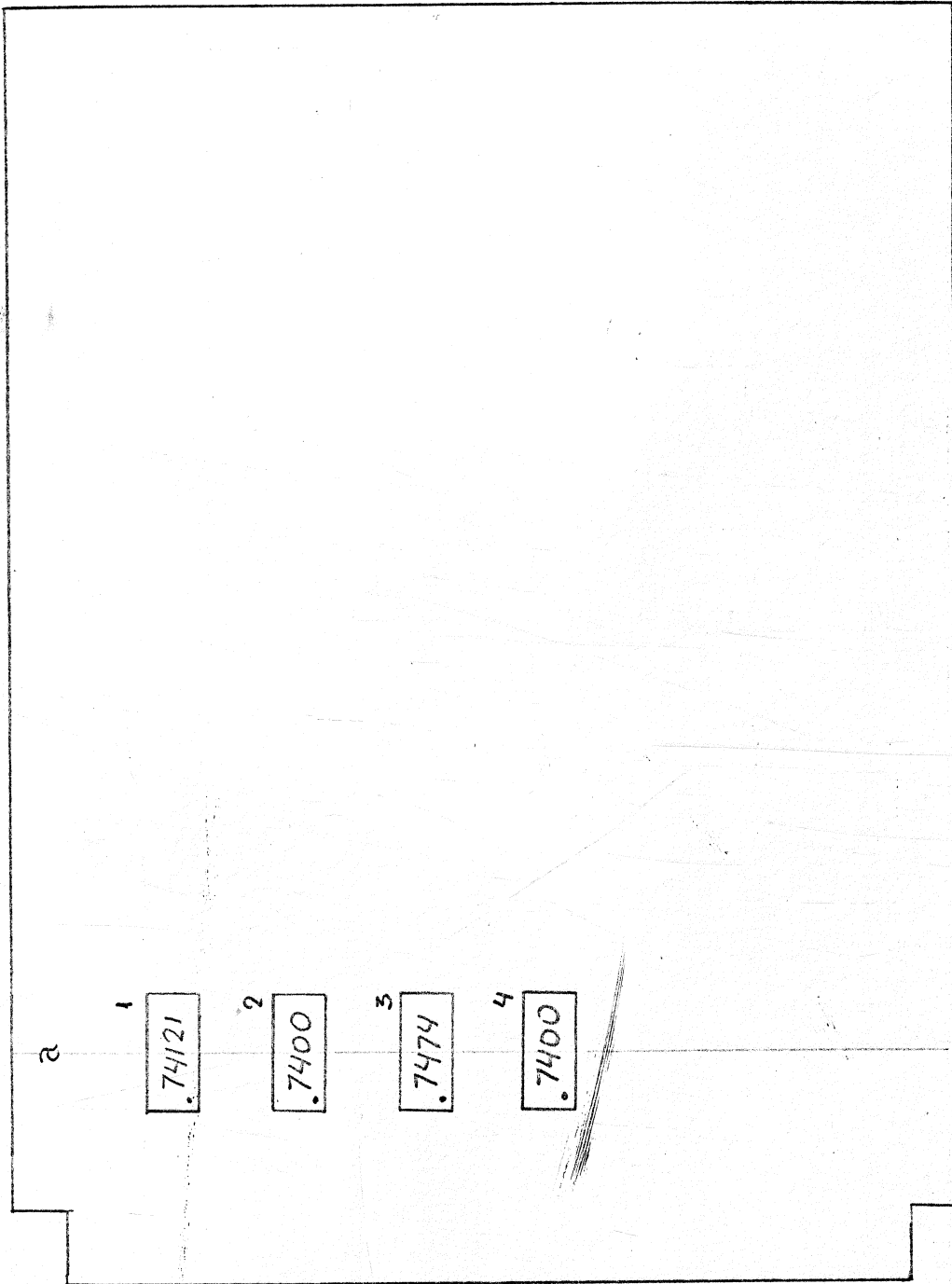
Funcionamiento: si está puesto pd, al producirse desborde para la máquina, reestableciéndose desde el tablero; cuando en el registro de instrucción presente se tiene sd, pese a que se produzca desborde, el reloj de la máquina no se anula. Se diseñó la siguiente lógica:



y para el funcionamiento por compás y generar una señal que vuelque el contador al estado cero, se agregó:



TARJETA XII



pin	con. exterior	con. interior	función
1			alimentación
2	V/2/19	XIIa1-6	Q(monostable)
3		XIIa1-1	\bar{Q} (monostable)
4			tierra
5	V/2/29	XIIa1-5	B(entrada mon.)
6	V/6/22	XIIa2-12	$\mu 1010$
7	V/6/18	XIIa2-13	$\mu 1011$
8	V/4/10	XIIa2-8	$\mu 1010 + \mu 1011$
9	V/6/17	XIIa2-1	$\mu 1011$
10	V/3/32	XIIa2-2 y 5	P_5
11	V/6/19	XIIa2-4	$\mu 1010$
12	XII/18	XIIa3-9	Q (flip-flop)
13			
14			
15			
16			
17	V/4/25	XIIa4-11	ent. D ff desb.
18	XII/12	XIIa4-13	Q (flip flop)
19	V/4/24	XIIa4-1	log. de desborde
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			

OBSERVACIONES

La experiencia adquirida en estos meses de trabajo permite arribar a algunas conclusiones que posibilitarán la implementación de normas para futuras construcciones.

a) Es muy importante indicar numéricamente las entradas y salidas de las compuertas. Es un trabajo que se debe hacer en forma paralela a la implementación física.

b) Se hace necesario, previo a la colocación de módulos en una tarjeta, una planificación de su ubicación a fin de evitar excesiva densidad y cruces de cables.

c) Se deben emplear tarjetas de tamaño normal para facilitar el manipuleo y detección de fallas.

d) En lo posible, las conexiones exteriores entre tarjetas, también deben estudiarse para evitar excesivos cableados.

e) Se deben usar cables de distintos colores, especialmente en conexiones exteriores, a fin de poder seguirlos y verificar sus conexiones.

f) Verificar las soldaduras.

g) Tener cuidado en la interconexión de tarjetas, especialmente si están a relativa larga distancia.

h) Dejar indicado las compuertas que no se usan.

i) Si se emplean compuertas 7400 como inversores es necesario poner las entradas restantes a uno lógico. En caso contrario aparecen señales espúreas que inducen a error.

j) Sería muy útil poder contar con un probador de circuitos integrados a fin de asegurarse un correcto funcionamiento antes de su uso.

k) Es conveniente marcar en el diagrama que indica la posición de los circuitos, el pin 1 del módulo para poder identificar fácilmente sus entradas y salidas, de la forma siguiente:



l) Para comprobar si en un determinado punto del circuito las señales son correctas, es necesario plantear primero las señales que deberían aparecer y compararlas.

ll) Se deben indicar en los informes los detalles que facilitan su comprensión.

m) Elaboración de programas para detectar fallas.

FE DE ERRATAS

función/ubicación	incorrecto	correcto
Vk1-6	P_2	E 2
Vn2-11	P_2	E 2
Vj1-12	E 5	E 6
Vi5-5	$E<0>$	$\overline{E<0>}$
Vh4-5	$\mu 0001$	$\mu 0001$
Vh4-14	0001	0001
[Re A]	V13-3	V13-8
Vh3-2	$\mu 0010$	$\mu 0011$
$\overline{M<15>}$	Vj6-3	Vm1-10
[SAURA]	Vh5-2	Vh5-10
[M + 1]	Vj5-5	Vm6-12
di	IIIIm6-2	IIIj6-14
ri	IIIj6-14	IIIIm6-2
Vf2-8	$E<0>$	$\overline{E<0>}$
P<12>	Vd7-13	Ve7-13
P<13>	Vd7-9	Ve7-9
P<14>	Ve6-13	Vf6-13
P<15>	Ve6-9	Vf6-9
P_2	V16-12	Vn8-12
0000	Vf1-12	Vf1-8
0001	Vf1-2	Vf1-6
0100(f_1)	Vc1-12	Vc1-8
0100(f_2)	Vel-2	Vel-6
0101(f_1)	Vb2-6	Vb2-2
0110(f_2)	Vel-8	Vel-12
0111(f_2)	Vel-12	Vel-8
1010(f_1)	Vc1-6	Vc1-12
1111(f_1)	Va2-8	Va2-12
II/2/9	IP	\overline{IP}
LDa<4>	IIIa6-3	IIIa6-7
LDa<5>	IIIb2-3	IIIb2-7
LDa<6>	IIIa2-3	IIIa2-7
LDa<7>	IIIe6-3	IIIe6-7

función/ubicación	incorrecto	correcto
III/1/25	acción mem.<1>	acción mem.<0>
III/1/26	acción mem.<2>	acción mem.<1>
III/1/27	IIIa3-2 [acción mem.<0>	IIIa3-10 acción mem.<2>
III/1/28	IIIa3-10	IIIa3-2
III/4/1	pulso lectura	[CLDaA]
III/4/3	IIIi2-8 [pulso lectura	IIIi2-8 pulso escritura
III/5/25	IIIi4-7	IIIi4-12
III/5/26	IIIi4-12	IIIi4-7
III/5/29	IIIi3-9	IIIi3-12
III/5/30	IIIi3-12	IIIi3-7
III/5/31	IIIi3-9	IIIi3-12
III/5/32	IIIi3-13	IIIi3-7
III/6/1	IIIe4-6	IIIa4-12
III/6/2	IIIe4-13	IIIa4-6
III/6/3	IIIa4-6	IIIe4-13
III/6/4	IIIa4-13	IIIe4-6
III/6/5	IIIa3-9	IIIa3-12
III/6/6	IIIc1-14	IIIa3-9(LDi<0>)
III/6/7	IIIc3-14	IIIc1-8(LDi<2>)
III/6/8	IIIa3-13	IIIe3-7(LDi<1>)
III/6/14	IIIe1-2	IIIe1-3
III/6/15	IIIe1-4	IIIe1-5
III/6/19	IIIa1-2	IIIa1-3
III/6/20	IIIa1-4	IIIa1-5
III/6/24	IIIb1-2	IIIb1-3
III/6/25	IIIb1-4	IIIb1-5
III/6/29	IIIa1-2	IIIa1-3
III/6/30	IIIa1-4	IIIa1-5
V/2/10	Vh5-10	Vh5-2
V/2/15	Vh5-2	Vh5-10
V/3/6	Vk5-9	Vk5-10
V/3/13	Vl6-5	Vl6-12
V/3/17	Vl6-10	Vl6-8
pulso fin esc.	[Vb1-12 [Va1-12	Va1-12 Vb1-12

8) CONCLUSIONES

Para la interpretación y entendimiento de este informe es imprescindible conocer como funcionan las unidades de control, aritmética e interfase.

Al principio el camino se hizo áspero por la falta de experiencia en el manejo de circuitos y el desconocimiento de las señales del procesador.

A medida que se avanzaba se comenzó a adquirir práctica en la interpretación y detección de fallas. La experiencia ganada condujo a la elaboración de determinadas normas, útiles para la construcción, que se detallan en el punto 5) .

El empleo de programas de cuatro palabras o menos ayudó considerablemente. Las instrucciones que los integran son todas con acceso directo, salvo aquellas en las que se indica expresamente el acceso indirecto.

En las modificaciones implementadas se trató de evitar un excesivo cableado que aumentara el ya existente y de usar las compuertas libres halladas.

Si bien este proyecto no tuvo un gran desarrollo en diseño lógico, y el trabajo invertido en él no se aprecia exteriormente, facilitará en gran medida la implementación de trabajos posteriores.

9) BIBLIOGRAFIA

a) "Apuntes de diseño lógico"

Ing. Santos Jorge, 1973

b) "Unidad aritmética de una minicomputadora digital de propósito general"

Ing. Luisoni Lidia.

Proyecto final, UNS, 1974

c) "Unidad de control de una minicomputadora digital de propósito general"

Ing. González Néstor

Proyecto final, UNS, 1974

d) "Interfase con capacidad de interrupción de una minicomputadora digital de propósito general"

Ing. Barrios Ricardo

Proyecto final, UNS, 1974

e) "A pocket guide to Hewlett Packard"

Hewlett Packard

f) "Small computer handbook"

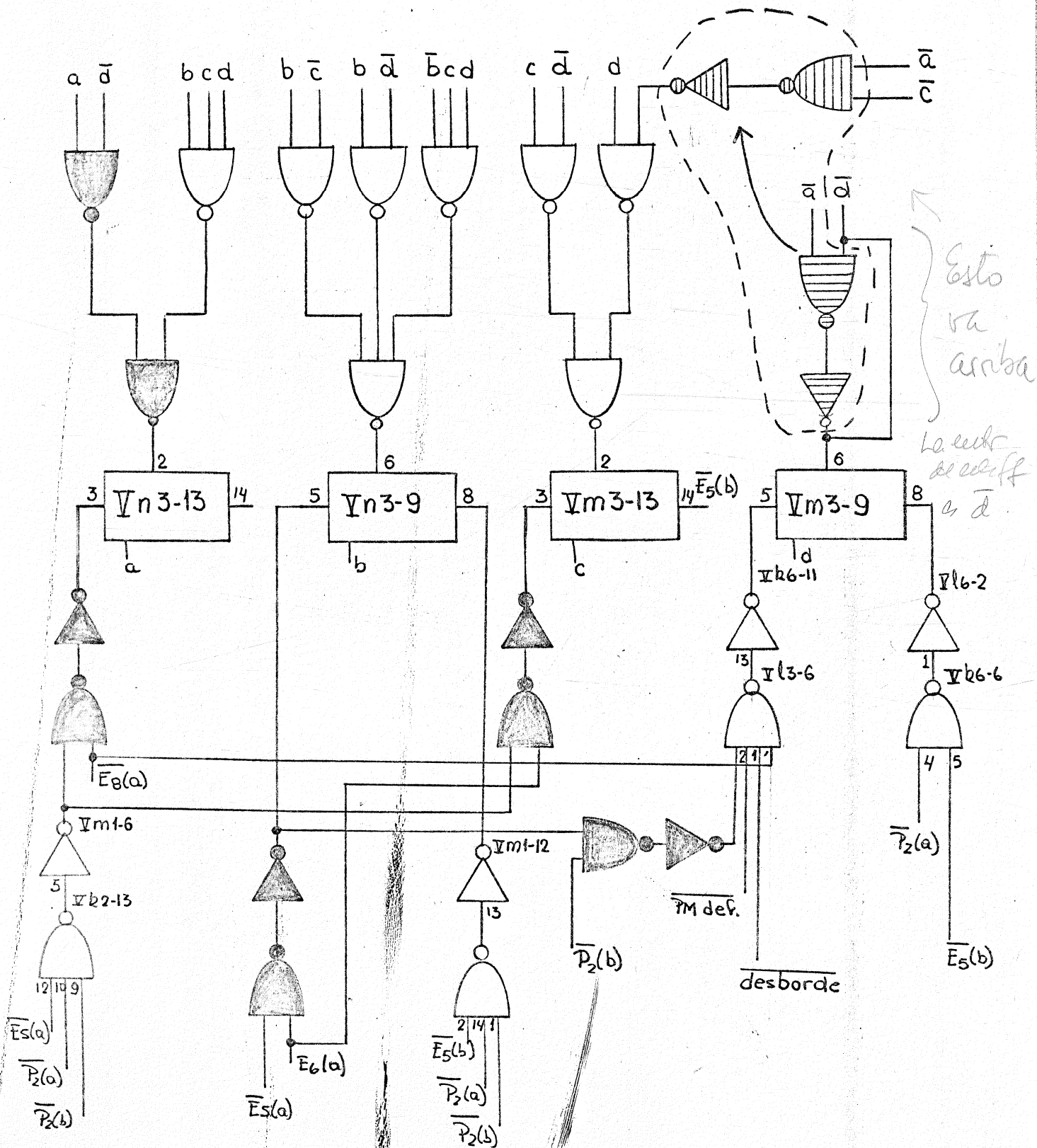
Digital Equipment Corporation, 1972

10) APÉNDICE

Los ciclos de lectura/escritura de la memoria semiconductora MOS son de 200ns. Como las instrucciones que tienen acceso a memoria inician el ciclo de escritura con \bar{P}_5 (\bar{P}_8) disponibles hasta el próximo pulso (P_0) de menos de 170ns - debido a los retrasos en las compuertas - no estando preparada para un nuevo acceso.

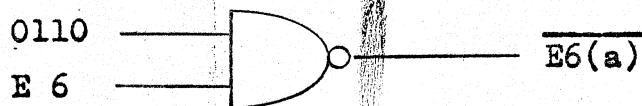
Se presentan dos posibilidades para su solución:

1) modificar el contador agregando otro pulso a las instrucciones sm y ta como se indica a continuación.



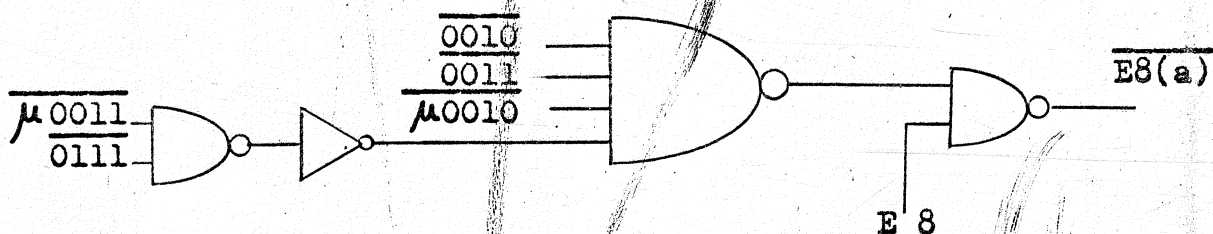
- compuertas empleadas
 compuertas que se deben intercambiar
 compuertas que hay que agregar

Para la instrucción ta(0110) el contador del estado 6 debe pasar al estado cero. Se genera la señal de control

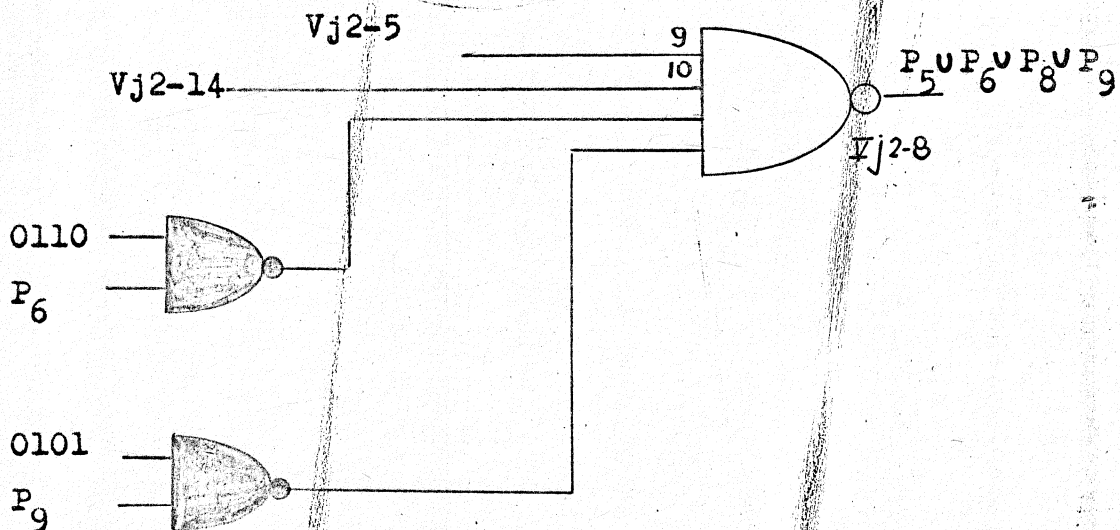


y pasa a ser una instrucción del grupo VI.

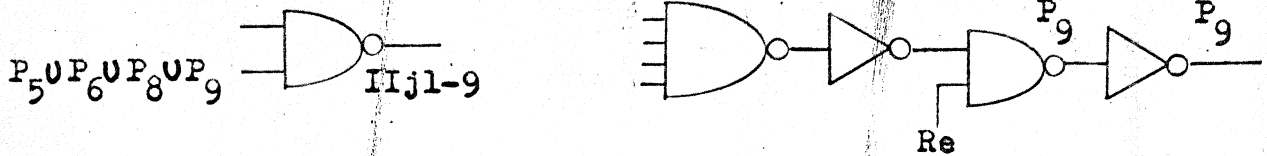
Las instrucciones del grupo II dejaban correr el contador y automáticamente pasaban al estado cero. Ahora se debe generar una señal que vuelque el estado de los flip-flops. A esta señal se puede agrupar la instrucción del grupo IV (ss).



Para el funcionamiento por compás queda



Para la generación de pulsos y compases se agrega P_6 y P_9



necesitando, para esta modificación, un decodificador del estado nueve y del pulso nueve.

Si se encara la modificación del contador se debe implementar una puesta a cero del mismo desde tablero.

2) La otra posibilidad es construir un circuito que al presentarse la instrucción sm y ta detenga por un pulso el reloj del procesador.-