

PRINCIPIOS PARA EL DISEÑO DE UN
CALCULADOR ELECTRONICO DE ESCRITORIO
A BASE DE CIRCUITOS DIGITALES

Tesis previa a la obtención
del Título de Ingeniero en
la Especialización de
ELECTRONICA Y TELECOMUNICACIONES
de la Escuela Politécnica Nacional

MARCELO VALLEJO T.

Quito
Julio de 1969

Certifico que este trabajo ha sido
realizado en su totalidad por el
señor Marcelo Vallejo T.

Ing. Warren Littlefield

Ing. Warren Littlefield,
CONSULTOR DE TESIS.

Quito, Julio de 1969.

△

MIS

PADRES.

MI AGRADECIMIENTO:

Al señor Ingeniero Warren Littlefield, Director de esta Tesis, por la invaluable ayuda prestada en la realización de este trabajo y durante su docencia en la Cátedra de "Introducción al Diseño de Computadoras Digitales". Si algo vale este trabajo, a él se lo debo.

A todos los miembros del Departamento de Electrónica, Control y Comunicaciones de la Escuela Politécnica Nacional en cuyos laboratorios realicé toda la parte práctica de esta Tesis.

A mi hermana Elsa sin cuyo trabajo dedicado no hubiera podido imprimirla.

INDICE GENERAL

	Pág.
PERFORMANCE DE LA MAQUINA Y ESPECIFICACIONES GENERALES -----	1
Introducción -----	1
Operaciones requeridas -----	1
Selección del número de registros -----	2
Entrada de números -----	3
Exhibición del resultado -----	3
Ventajas -----	3
Representación de los números -----	4
 DIAGRAMA GENERAL DE BLOQUES Y ANALISIS DE LAS OPERACIONES -----	 7
Unidad de entrada -----	8
Unidad aritmética -----	8
Unidad de memoria -----	9
Unidad de control -----	10
Unidad de salida -----	10
Notación usada -----	11
Análisis de la operación Adición -----	12
Análisis de la operación Sustracción -----	16
Análisis de la operación Multiplicación -----	19
Análisis de la operación División -----	24
Análisis de la operación $(a + b) c =$ -----	30
Análisis de la operación $(a + b) c/d =$ -----	30
Análisis de la operación $a \times b = + c \times d = + e \times f =$ -----	32
Análisis de la operación $(a \times b = + c \times d = + e \times f =) \times g =$ -----	35
 DIAGRAMA DESARROLLADO DE BLOQUES -----	 38
Diagrama de bloques de la unidad de entrada -----	39
Unidad formadora de pulsos -----	40
Unidad formadora de niveles -----	42
Codificador -----	42

	Pág.
Circuito complementador de 10 -----	43
Diagrama de bloques de la unidad aritmética -----	44
Contador de entrada -----	47
Flip flop de llevo de CE -----	48
Contador Auxiliar -----	48
Flip flop de llevo de CA -----	49
Compuertas G_1, G_2, G_3, G_4 -----	49
Contador de multiplicación -----	50
Flip flop de llevo de CM -----	51
Contador de División -----	51
Contador de 12 -----	52
Contador auxiliar de división -----	52
Diagrama de bloques de la unidad de memoria -----	54
Registro de entrada -----	55
Unidad de rotación de RE -----	57
Registro auxiliar -----	57
Unidad de rotación de RA -----	58
Registro de resultados -----	58
Diagrama de bloques de la unidad de control -----	59
Subunidad de tiempo -----	60
Subunidad de distribución -----	61
Diagrama de bloques de la unidad de salida -----	63
Selector de registros -----	64
Unidad decodificadora -----	64
Unidad de exhibición -----	64
APENDICE -----	66
Apéndice I	
Convenciones Lógicas -----	67
La serie 350 de MECL -----	67
Compuertas de tres entradas dos salidas -----	70
Paquetes de dos compuertas, dos entradas por com puerta, una salida por compuerta -----	71
Paquetes de dos compuertas, tres entradas por com puerta, una salida por compuerta -----	73
Compuerta de cinco entradas, dos salidas -----	74
Medio sumador -----	75
Flip flop R-S -----	76
Flip flop J-K -----	77

Apéndice II

Diseño de la unidad de entrada	
Diseño de UFP -----	79
Diseño de UFN -----	82
Diseño de COD y CCD -----	83
Diseño de la unidad aritmética	
Diseño de CE, CA, CM y CD -----	86
Diseño de FLE, FLA y FLM -----	89
Diseño de CAD -----	90
Diseño de CDO -----	92
Diseño de la unidad de memoria	
Diseño de RE, RA y RR -----	93
Diseño de URRE y URRA -----	94
Diseño de la unidad de control -----	95
Diseño de SUT -----	103
Diseño de SUD -----	107
Ecuaciones -----	108
Ecuaciones reducidas -----	118
Diseño de la unidad de salida	
Diseño de SR -----	122
Diseño de DEC -----	124
Diseño de UEX -----	126

Apéndice III

Laboratorios sobre tablas de verdad -----	127
Laboratorios sobre la unidad de entrada -----	127
Laboratorios sobre la unidad aritmética -----	128
Laboratorios sobre la memoria -----	128
Laboratorios sobre el control -----	128
Laboratorios sobre la unidad de salida -----	129

BIBLIOGRAFIA -----	130
--------------------	-----

INDICE DE TABLAS

TABLA No.		Pág.
1	Operaciones requeridas -----	1
2	Códigos -----	5
3	Operación Adición -----	12
4	Operación Adición -----	13
5	Operación Sustracción -----	16
6	Operación Sustracción -----	17
7	Operación Multiplicación -----	19
8	Operación Multiplicación -----	22
9	Operación Multiplicación -----	23
10	Operación División -----	24
11	Operación División -----	25
12	COD y CCD -----	83
13	CE, CA, CM, CD -----	87
14	CAD -----	90
15	CDO -----	92
16	SR -----	122
17	DEC -----	124

INDICE DE PLANOS

PLANO No.	
1	Unidad de entrada: UFP, UFNI
2	Unidad de entrada: COD, CCD
3	Unidad aritmética
4	Unidad de memoria
5	Unidad de control: SUT
6	Unidad de control: SUD
7	Unidad de salida.

INDICE DE FIGURAS

FIGURA No		Pág.
1	Diagrama general de bloques -----	8
2	Suma elemental -----	14
3	Suma elemental -----	15
4	Bloqueo de RE -----	21
5	Diagrama de bloques de la unidad de entrada -----	39
6	Diagrama de bloques de la unidad aritmética -----	44
7	Diagrama de bloques de la unidad de memoria -----	54
8	Diagrama de bloques de la unidad de control -----	59
9	Diagrama de bloques de la unidad de salida . -----	63
10	MC356 -----	70
11	MC357 -----	71
12	MC365 -----	71
13	MC356 -----	71
14	MC359 -----	72
15	MC360 -----	72
16	MC361 -----	73
17	MC361 -----	73
18	MC359 -----	73
19	MC362 -----	74
20	MC351 -----	75
21	MC353 -----	76
22	MC352 -----	76
23	MC358 -----	77
24	MC358 -----	78
25	Disparo Schmitt -----	79
26	Multivibrador monoestable -----	80
27	Retardo integrante -----	81
28	CCD -----	86
29	Diagramas de tiempo -----	96
30	Diagramas de tiempo -----	97
31	Diagramas de tiempo -----	98
32	Diagramas de tiempo -----	99
33	Diagramas de tiempo -----	99
34	Diagramas de tiempo -----	99
35	Diagramas de tiempo -----	100
36	Diagramas de tiempo -----	100
37	Diagramas de tiempo -----	102

FIGURA No.		Pág.
38	Diagramas de tiempo -----	103
39	Diagramas de tiempo -----	103

PERFORMANCE DE LA MAQUINA
Y
ESPECIFICACIONES GENERALES

PERFORMANCE DE LA MAQUINA Y ESPECIFICACIONES GENERALES

INTRODUCCION:

El calculador electrónico de escritorio hecho a base de circuitos digitales cuyos principios de diseño van a ser detallados en la presente tesis, realiza básicamente las cuatro operaciones aritméticas, a saber: suma, resta, multiplicación y división; pero va a ser hecho de tal manera versátil y flexible que puede realizar operaciones combinadas.

Para el efecto viene exteriormente equipado con un tablero que contiene las teclas para los dígitos 0, 1, 2, 3,, 9, así como las correspondientes a las operaciones +, -, x, ÷, =, y otras que se irán detallando en el transcurso de la tesis.

1. OPERACIONES REQUERIDAS:

La tabla mostrada abajo es una representación reducida de las operaciones que se desea sean realizadas por el calculador.

TABLA 1

1. - $a + b =$

2. - $a + b + c + d + \dots =$

3. - $a - b =$

4. - $a + b + c - d =$

5. - $a \times b =$

6. - $a \times b \times c \times d \times \dots =$

7. - $(a + b) \times c =$

8. - $a \div b =$

$$9. - \frac{(a+b) c}{d} =$$

$$10. - \begin{array}{r} a \times b = \\ c \times d = + \\ e \times f = + \\ \hline \text{total} \end{array}$$

$$11. - \begin{array}{r} a \times b = \\ c \times d = + \\ e \times f = + \\ \hline \text{subtotal} \\ g \times \text{sub} + \\ \hline \text{total} \end{array}$$

2. SELECCION DEL NUMERO DE REGISTROS

Las operaciones básicas mostradas arriba y base requerida del performance de la máquina, exigen en total tres registros de desplazamiento. Las operaciones numeradas del 1 al 7 inclusive se pueden realizar con un total de dos registros y, las numeradas del 8 al 11 requieren el aumento de un registro.

Se podría diseñar una máquina que realice las operaciones del 1 al 7 y usar solamente dos registros, pero si se quiere versatilidad sin mayor aumento de costo se puede usar un ideal de tres registros. En este caso en realidad lo único que se aumenta es un registro más y básicamente el resto de circuitos digitales no sufren una variación considerable en costo.

No hay ninguna razón técnica que impida trabajar con cuatro o más registros, pero en este caso el factor limitante es económico pues la adición de uno o más registros se justificaría con mayor cantidad de operaciones requeridas del calculador, lo que acarrearía un aumento considerable del número de circuitos digitales y consecuentemente del costo.

Cada registro tiene lugar para 12 dígitos, y cada dígito se codifica en cuatro bits, de modo que cada registro tiene capacidad para 48 bits.

3. ENTRADA DE NUMEROS:

Los números son introducidos en el calculador al presionar las teclas del tablero de mando, y se realiza en forma descendente desde el dígito de mayor valor significativo hasta el de menor valor significativo.

Los dígitos, una vez codificados, son introducidos a través de la unidad de entrada en el registro de entrada RE de derecha a izquierda, excepción hecha de la operación multiplicación, en la que el multiplicador va a un contador de multiplicación como se explicará a su debido tiempo.

4. = EXHIBICION DEL RESULTADO:

El número del resultado de cualquier operación se muestra al operador mediante 12 tubos indicadores numéricos en el panel frontal. Se contará también con un indicador de signo, de modo que el resultado es mostrado en magnitud y signo. Los tubos indicadores numéricos se conectan automáticamente al registro en el que se encuentra el resultado deseado.

5. VENTAJAS:

El calculador electrónico, para ser comercialmente constructible y además práctico, debe ofrecer innegables ventajas sobre sus similares del tipo mecánico, para justificar el aumento en el costo, que indudablemente existe. Esas ventajas son: trabajo silencioso, alta velocidad, gran versatilidad, simplicidad en el uso, mayor confiabilidad, menor costo de mantenimiento pues no tiene partes mecánicas movibles y clara exhibición del resultado.

Una vez terminado este trabajo se calculó que el tiempo máximo para la ejecución de una operación suma es de 0.58 ms y el de la división que es el más complicado de $55,4 \text{ ms}$ de modo que prácticamente la velocidad de operación del calculador está determinada exclusivamente por la capacidad del operador más bien que por la máquina misma.

Todas esas innegables mejoras sobre sus similares, hacen del calculador electrónico muy aconsejable para uso en oficinas, bancos, etc.

REPRESENTACION DE LOS NUMEROS

Se podría usar para representar los dígitos 0, 1, 2, , 9, cualquier código binario ponderado o no, simétrico o no, siempre de cuatro bits que es el mínimo requerido, pero la experiencia enseña que uno de los factores fuertemente determinados en el costo de los circuitos combinacionales y secuenciales es una buena selección del código a usarse.

El código ponderado DCB (decimal codificado en binario) que se muestra en la Tabla 2, ofrece algunas ventajas sobre todo en lo referente a operaciones aritméticas en binario, por el hecho de que el dígito decimal escrito en base 2 tiene la misma representación que en el DCB; por tanto es uno de los candidatos.

TABLA 2

	DCB		EXCESO 3		AIKEN
	<u>PESO = 8 4 2 1</u>				<u>PESO = 2 4 2 1</u>
0	0 0 0 0	0	0 0 1 1	0	0 0 0 0
1	0 0 0 1	1	0 1 0 0	1	0 0 0 1
2	0 0 1 0	2	0 1 0 1	2	0 0 1 0
3	0 0 1 1	3	0 1 1 0	3	0 0 1 1
4	0 1 0 0	4	0 1 1 1	4	0 1 0 0
5	0 1 0 1	5	1 0 0 0	5	1 0 1 1
6	0 1 1 0	6	1 0 0 1	6	1 1 0 0
7	0 1 1 1	7	1 0 1 0	7	1 1 0 1
8	1 0 0 0	8	1 0 1 1	8	1 1 1 0
9	1 0 0 1	9	1 1 0 0	9	1 1 1 1

También aparecen en la TABLA 2 otros 2 posibles códigos: el exceso 3 y el de Aiken. El primero es no ponderado a diferencia del segundo cuyos pesos son 2, 4, 2, 1. Ambos por ser simétricos gozan de una interesante propiedad: el complemento de nueve de cualquier dígito, es formado reemplazando los 0's con 1's y viceversa. Ello facilita enormemente la representación de números negativos y permite que la resta pueda reducirse a una suma (se añade al minuendo el complemento de 9 del sustraendo). A su vez facilita el proceso de división, que es el más complicado, pues éste se realiza a base de restas sucesivas.

Esta es una razón de suficiente peso como para desechar al código DCB como posible candidato. Por otro lado no hay ninguna razón técnica fuertemente determinante que exija escoger entre el código de Exceso 3 y el código de Aiken. Observe que los cinco primeros dígitos del código de Aiken tienen exactamente la misma representación que en DCB; ésta además de mi gusto son las razones que me han hecho escoger la representación de Aiken para la codificación de los dígitos en el trabajo interno del calculador.

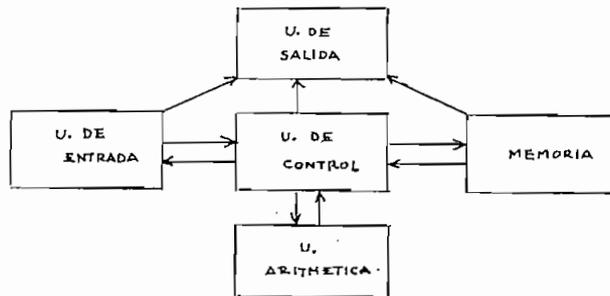
DIAGRAMA GENERAL DE BLOQUES

Y

ANALISIS DE LAS OPERACIONES

DIAGRAMA GENERAL DE BLOQUES

FIG. 1



1. UNIDAD DE ENTRADA:

Sirve para introducir la información al computador. Consta del tablero de teclas, de codificadores y de circuitos digitales que traducen al lenguaje de la máquina toda la información procedente del exterior. Un dígito entra al computador presionando la tecla correspondiente para quedar en la memoria cuando la tecla ha regresado a su posición normal. Esta unidad se encarga de dar a la unidad de control la información necesaria para la ejecución de cualquier operación.

2. UNIDAD ARITMETICA:

Es la que realiza las operaciones aritméticas básicas: suma, resta, multiplicación y división. Para el efecto está equipada con cinco contadores binarios:

CE: contador de entrada

CA: contador auxiliar

CM: contador de multiplicación

CD: contador de división

CAD: contador auxiliar de división

Todas las operaciones que vamos a realizar se reducen a una suma: la resta se realiza sumando el complemento de 9 del sustraendo; la multiplicación mediante sumas sucesivas y, la división a base de restas sucesivas, es decir por medio de adiciones sucesivas del complemento de 9.

Para hacer un compromiso entre economía y velocidad de operación se ha escogido el método dígitos en serie, bits en paralelo: entran dos dígitos de los dos números a sumarse en la unidad aritmética; una vez sumados pasan a la memoria y entran dos dígitos más.

3. UNIDAD DE MEMORIA:

Es una de las partes más importantes de la máquina. Sirve para almacenar los dígitos introducidos al calculador, así como para guardar los resultados parciales antes de ser mostrados al operador. Consta de tres registros:

RE: registro de entrada

RA: registro auxiliar

RR: registro de resultados

Cada registro tiene lugar para 11 dígitos más un lugar para signo. En la mayoría de las operaciones se usan solamente los dos primeros, pero en operaciones tales como división, suma de productos, suma de cuocientes, etc. se incluye el registro de resultados, RR.

La unidad aritmética y la unidad de memoria trabajan en estrecha colaboración.

4. UNIDAD DE CONTROL:

Es la que interpreta las informaciones provenientes de la unidad de entrada a través del tablero de mando y se encarga de formar y de distribuir los pulsos a los terminales apropiados de la máquina para preparar y ejecutar las operaciones.

Para el efecto consta de dos subunidades: la subunidad de tiempo (encargada de formar los pulsos) y la subunidad de distribución (encargada de distribuir los pulsos). Cabe indicar que esta es la parte de más difícil diseño.

5. UNIDAD DE SALIDA:

Sirve para mostrar el resultado en forma de luz proveniente de los tubos indicadores numéricos. Para el efecto viene dotada de 12 tubos indicadores, de codificadores y circuitos digitales adicionales que se encargarán de conectarla al registro en el que se encuentre el resultado.

He aquí un plan general de trabajo para el diseño del calculador electrónico. Con esta visión de conjunto se empezarán a analizar una por una las operaciones que el calculador va a realizar. Ello hará ver el forma clara a qué parte de la máquina afecta cada paso de una operación. De ahí se sacará la información necesaria para conocer los elementos de que debe disponer cada bloque para cumplir con sus funciones específicas. Esto permitirá hacer un diagrama de bloques desarrollado y luego vendrá el arduo trabajo de diseño de cada parte.

1. NOTACION USADA:

RE: el lugar para el dígito n en RE. (n se cuenta de derecha a izquierda)

(RE): el contenido de RE

(X)[']: el complemento de 9 de X

(X)["]: el complemento de 10 de X

(RE)→n: rote el contenido de RE n lugares a la derecha

(RE)←n: rote el contenido de RE n lugares a la izquierda

(RE)→CE: transfiera el contenido de RE a CE

(RE) + (RA)→RA: sume el contenido de RE más el contenido de RA y ponga
el resultado en RA

1 + (RA)→RA: 1 más el contenido de RA y ponga el resultado en RA

P: Ponga

0's : ceros

9's: nueves

ANALISIS DE LAS OPERACIONES

1. ANALISIS DE LA OPERACION ADICION:

Ejemplo 2.347 + 215 = 2.562

TABLA 3

Tecla operada	Registro RE	Registro RA	
	12 11 10 9 8 7 6 5 4 3 2 1	12 11 10 9 8 7 6 5 4 3 2 1	
	+ 0 0 0 0 0 0 0 0 0 0 0 0	+ 0 0 0 0 0 0 0 0 0 0 0 0	Estado inicial
2	+ 0 0 0 0 0 0 0 0 0 0 0 2	+ 0 0 0 0 0 0 0 0 0 0 0 0	P 2 en RE ₁₂ y (RE) ← 1
3	+ 0 0 0 0 0 0 0 0 0 0 2 3	+ 0 0 0 0 0 0 0 0 0 0 0 0	P 3 en RE ₁₂ y (RE) ← 1
4	+ 0 0 0 0 0 0 0 0 0 2 3 4	+ 0 0 0 0 0 0 0 0 0 0 0 0	P 4 en RE ₁₂ y (RE) ← 1
7	+ 0 0 0 0 0 0 0 0 2 3 4 7	+ 0 0 0 0 0 0 0 0 0 0 0 0	P 7 en RE ₁₂ y (RE) ← 1
+	+ 0 0 0 0 0 0 0 0 2 3 4 7	+ 0 0 0 0 0 0 0 2 3 4 7	(RE) + (RA) → RA
	+ 0 0 0 0 0 0 0 0 0 0 0 0	+ 0 0 0 0 0 0 0 2 3 4 7	P 0's en RE
2	+ 0 0 0 0 0 0 0 0 0 0 0 2	+ 0 0 0 0 0 0 0 2 3 4 7	P 2 en RE ₁₂ y (RE) ← 1
1	+ 0 0 0 0 0 0 0 0 0 0 2 1	+ 0 0 0 0 0 0 0 2 3 4 7	P 1 en RE ₁₂ y (RE) ← 1
5	+ 0 0 0 0 0 0 0 0 0 2 1 5	+ 0 0 0 0 0 0 0 2 3 4 7	P 5 en RE ₁₂ y (RE) ← 1
+	+ 0 0 0 0 0 0 0 0 2 1 5	+ 0 0 0 0 0 0 0 2 5 6 2	(RE) + (RA) → RA
	+ 0 0 0 0 0 0 0 0 0 0 0 0	+ 0 0 0 0 0 0 0 2 5 6 2	P 0's en RE

Conclusiones:

De la observación de la Tabla 3 pueden obtenerse los siguientes resultados:

1. - Las teclas de los dígitos deben ser aplastadas secuencialmente, comenzando con el de mayor valor significativo.

2. - En esta operación los dígitos entran al registro de entrada, concretamente a la posición 12 del registro de entrada; el contenido de dicho registro debe ser rotado un lugar a la izquierda. De aquí se obtiene que la tecla de cualquier dígito al ser presionada debe cumplir con esta función.

3. - Al aplastar la tecla de cualquier dígito la unidad de salida debe conectarse al registro RE, durante esta fase de la operación.

4. - Se observa que al aplastar la tecla "+" debe ocurrir: $(RE) + (RA) \rightarrow RA$.
 Esto se lleva a cabo de la siguiente forma: (suponga que estamos analizando el proceso que ocurre al aplastar la segunda tecla "+")

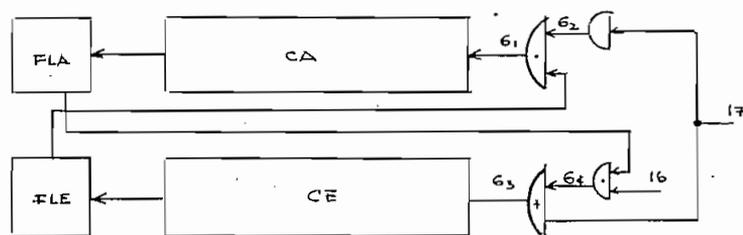
TABLA 4

Tecla operada	REGISTRO RE												REGISTRO RA														
	12	11	10	9	8	7	6	5	4	3	2	1	12	11	10	9	8	7	6	5	4	3	2	1			
5	+	0	0	0	0	0	0	0	0	0	2	1	5	+	0	0	0	0	0	0	0	0	2	3	4	7	P 5 en RE ₁₂ (RE) ← 1
+	5	0	0	0	0	0	0	0	0	0	2	1	7	0	0	0	0	0	0	0	0	0	2	3	4	(RE) ₁ → CE ₁₂ (RA) ₁ → CA (RE) ↔ 1 (RA) ↔ 1	
	5	0	0	0	0	0	0	0	0	0	2	1	2	0	0	0	0	0	0	0	0	0	2	3	4	(CA) → RA ₁₂	
	1	5	0	0	0	0	0	0	0	0	2	4	2	0	0	0	0	0	0	0	0	0	2	3	(RE) ₁ → CE ₁₂ (RA) ₁ → CA (RE) → 1 (RA) → 1		
	1	5	0	0	0	0	0	0	0	0	2	6	2	0	0	0	0	0	0	0	0	0	2	3	(CA) → RA ₁₂		
	2	1	5	0	0	0	0	0	0	0	0	3	6	2	0	0	0	0	0	0	0	0	0	2	(RE) ₁ → CE ₁₂ (RA) ₁ → CA (RE) → 1 (RA) → 1		
	2	1	5	0	0	0	0	0	0	0	0	5	6	2	0	0	0	0	0	0	0	0	0	2	(CA) → RA ₁₂		
	+	0	0	0	0	0	0	0	0	2	1	5	+	0	0	0	0	0	0	0	0	2	5	6	2		

En la Tabla 4 aparece claramente el modo de trabajo: bits paralelo-dígitos serie. Al aplastar la tecla + se transfiere el primer par de dígitos (5 y 7) a la unidad aritmética (CE y CA), y a su vez se rota los registros un lugar a la derecha para dejarlos listos para el próximo paso o para desocupar un lugar en el que se puede recibir información.

La unidad aritmética procesa el primer par de dígitos (en la forma en que se explicará enseguida) y al final de ese proceso se tiene en CA el resultado de la suma $5 + 7 = 2$ y llevo 1. De ahí que el próximo paso sea $(CA) \rightarrow RA_{12}$, mientras el llevo queda en la unidad aritmética para ser tomado en cuenta cuando el próximo par de dígitos sea procesado.

FIG. 2



La Fig. 2 ilustra el principio en el que se basa la unidad aritmética para realizar la suma de un par de dígitos. En la Fig. 2

CE: contador de entrada (cuenta en el código de Aiken)

CA: contador auxiliar (cuenta en el código de Aiken)

FLE: flip-flop de llevo del contador CE

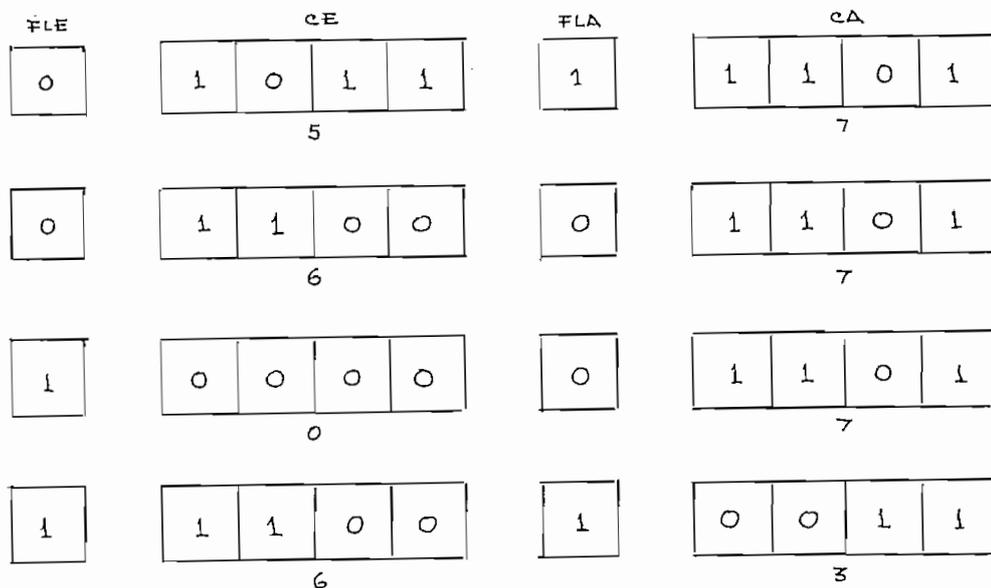
FLA: flip-flop de llevo del contador CA

G_1, G_2, G_3, G_4 : compuertas.

Un pulso aplicado al terminal 16, incrementa CE en uno, sólo si hay FLA.

Un pulso aplicado al terminal 17, incrementa CE en uno siempre, e incrementa CA en uno, sólo si hay FLE.

FIG. 3



Supóngase que deseo sumar $5 + 7$ y llevo 1 de la etapa anterior. La respuesta debe ser $5 + 7 + 1 = 3$ y llevo 1. El 3 debe aparecer en CA y el llevo en FLA. La primera fila de la Fig. 3 me enseña el estado inicial.

Doy un pulso al terminal 16 (vea Fig. 2). Esto incrementará CE en uno puesto que hay un 1 en FLA. El correspondiente estado se muestra en la fila 2 de la Fig. 3, después de que se ha puesto en reset al FLA, pues la información que él tenía ya ha sido asimilada.

Ahora demos 10 pulsos al terminal 17.

Los 4 primeros pulsos incrementan CE en 4, pero no incrementan CA pues no hay todavía FLE. Esa situación (después de dados los 4 primeros pulsos) está representada en la fila 3.

Los restantes 6 pulsos incrementan tanto CE como CA y el resultado final se

ve en la fila 4. Era exactamente lo que deseábamos.

5. - De ahora en adelante interpretaremos como signo + un cero en RE_{12} o RA_{12} y, como signo menos, un nueve en RE_{12} o RA_{12} .

6. - La información que entra en RE, RA, CE, CA, se introduce por el método de "sobreponga" y no por el método "limpie-ponga". Ello implica que se usarán R - S de los flip-flops J - K de que están formados esos elementos.

7. - Al aplastar la tecla +, la unidad de salida debe conectarse a RA

2. ANALISIS DE LA OPERACION SUSTRACCION

Ejemplo 1: $2.318 - 614 = 1.704$

TABLA 5

Tecla operada	REGISTRO RE												REGISTRO RA												
	12	11	10	9	8	7	6	5	4	3	2	1	12	11	10	9	8	7	6	5	4	3	2	1	
	+0	0	0	0	0	0	0	0	0	0	0	0	+0	0	0	0	0	0	0	0	0	0	0	0	Estado inicial
2	+0	0	0	0	0	0	0	0	0	0	0	2	+0	0	0	0	0	0	0	0	0	0	0	0	P 2 en RE_{12} y (RE) ← 1
3	+0	0	0	0	0	0	0	0	0	0	2	2	+0	0	0	0	0	0	0	0	0	0	0	0	P 3 en RE_{12} y (RE) ← 1
1	+0	0	0	0	0	0	0	0	0	2	3	1	+0	0	0	0	0	0	0	0	0	0	0	0	P 1 en RE_{12} y (RE) ← 1
8	+0	0	0	0	0	0	0	2	3	1	8	+0	0	0	0	0	0	0	0	0	0	0	0	P 8 en RE_{12} y (RE) ← 1	
-	+0	0	0	0	0	0	0	2	3	1	8	+0	0	0	0	0	0	0	2	3	1	8	(RE) + (RA) → RA		
	-9	9	9	9	9	9	9	9	9	9	9	+0	0	0	0	0	0	0	2	3	1	8	P 9's en RE		
6	-9	9	9	9	9	9	9	9	9	9	3	+0	0	0	0	0	0	0	2	3	1	8	P (6)' en RE_{12} y (RE) ← 1		
1	-9	9	9	9	9	9	9	9	9	3	8	+0	0	0	0	0	0	0	2	3	1	8	P (1)' en RE_{12} y (RE) ← 1		
4	-9	9	9	9	9	9	9	3	8	5	+0	0	0	0	0	0	0	2	3	1	8	P (4)' en RE_{12} y (RE) ← 1			
+	-9	9	9	9	9	9	9	3	8	5	1+0	0	0	0	0	0	0	1	7	0	3	(RE) + (RA) → RA			
	+0	0	0	0	0	0	0	0	0	0	0	1+0	0	0	0	0	0	0	1	7	0	3	P 0's en RE		
	+0	0	0	0	0	0	0	0	0	0	0	+0	0	0	0	0	0	0	1	7	0	4	(RA) + 1 si FLA		

Ejemplo 2: $- 2.324 + 612 = - 1.712$

TABLA 6

Tecla operada	REGISTRO RE												REGISTRO RA												
	12	11	10	9	8	7	6	5	4	3	2	1	12	11	10	9	8	7	6	5	4	3	2	1	
	+ 0	0	0	0	0	0	0	0	0	0	0	0	+ 0	0	0	0	0	0	0	0	0	0	0	0	Estado inicial
-	+ 0	0	0	0	0	0	0	0	0	0	0	0	+ 0	0	0	0	0	0	0	0	0	0	0	0	(RE) + (RA) → (RA)
	- 9	9	9	9	9	9	9	9	9	9	9	9	+ 0	0	0	0	0	0	0	0	0	0	0	0	P 9's en RE
2	- 9	9	9	9	9	9	9	9	9	9	9	7	+ 0	0	0	0	0	0	0	0	0	0	0	0	P (2)' en RE ₁₂ y (RE) ← 1
3	- 9	9	9	9	9	9	9	9	9	9	7	6	+ 0	0	0	0	0	0	0	0	0	0	0	0	P (3)' en RE ₁₂ y (RE) ← 1
2	- 9	9	9	9	9	9	9	9	7	6	7	6	+ 0	0	0	0	0	0	0	0	0	0	0	0	P (2)' en RE ₁₂ y (RE) ← 1
4	- 9	9	9	9	9	9	9	7	6	7	5	5	+ 0	0	0	0	0	0	0	0	0	0	0	0	P (4)' en RE ₁₂ y (RE) ← 1
+	- 9	9	9	9	9	9	9	7	6	7	5	5	- 9	9	9	9	9	9	7	6	7	5	5	5	(RE) + (RA) ₁₂ → RA
	+ 0	0	0	0	0	0	0	0	0	0	0	0	- 9	9	9	9	9	9	7	6	7	5	5	5	P 0's en RE
6	+ 0	0	0	0	0	0	0	0	0	0	0	6	- 9	9	9	9	9	9	7	6	7	5	5	5	P 6' en RE ₁₂ y (RE) ← 1
1	+ 0	0	0	0	0	0	0	0	0	0	6	1	- 9	9	9	9	9	9	7	6	7	5	5	5	P 1 en RE ₁₂ y (RE) ← 1
2	+ 0	0	0	0	0	0	0	0	0	6	1	2	- 9	9	9	9	9	9	7	6	7	5	5	5	P 2 en RE ₁₂ y (RE) ← 1
+	+ 0	0	0	0	0	0	0	0	0	6	1	2	- 9	9	9	9	9	9	8	2	8	7	7	7	(RE) + (RA) ₁₂ → RA
	+ 0	0	0	0	0	0	0	0	0	0	0	0	- 9	9	9	9	9	9	8	2	8	7	7	7	P 0's en RE

Al analizar las dos tablas anteriores se completan las ideas expuestas en el análisis de la operación adición, y aparecen nuevas funciones de modo que expondremos las siguientes:

Conclusiones:

1. - Al aplastar la tecla de cualquier dígito, este entra a RE₁₂ sea en su forma original, o sea en forma del complemento de nueve, cuando el dígito ha sido alimentado después de aplastar la tecla "-". Es obvio, a partir de esto, que al aplastar la tecla "- " se deben preparar ciertos circuitos para cumplir esta condición.

2. - Al aplastar la tecla "+" o "-" debe ocurrir que

a. - (RE) + (RA) → RA

b. - P 0's en RE si "+" P 9's en RE si "-"

c. - 1 + (RA) si FLA

3. - Al aplastar la tecla de cualquier dígito, para estas dos operaciones, la unidad de salida debe conectarse a RE. Sin embargo cuando el dígito se ha aplastado después de un "-" no aparece en su forma original sino complementada. Habrían dos posibilidades: dejar que el operador traduzca el resultado, con un poco de práctica, si el número es negativo; o hacer que la unidad de salida se conecte a RE de tal forma que no muestre su contenido sino el complemento de su contenido. Esto complica un poco las cosas pero para el presente trabajo ha sido escogida esta última posibilidad.

4. - Al aplastar las teclas "+" o "-" la unidad de salida debe mostrar el resultado almacenado en RA. Si el resultado es negativo valen las mismas consideraciones del punto 3. Esto permite que la máquina pueda realizar sumas y restas, o mejor sumas algébricas de cualquier tipo.

5. - Las operaciones anteriores han sido realizadas sólo con dos términos pero aparece obvio que nada obsta para que puedan haber 3 o más términos, de modo que cabe anotar que la máquina puede realizar la operación:

$a + b + c + \dots$

cualquiera sean los signos de a, b, c,

3. ANALISIS DE LA OPERACION MULTIPLICACION.

Será realizada a base de sumas sucesivas y sólo con factores positivos. Con un poco más de trabajo se puede aumentar un sencillo circuito de conserfvación del signo, pero este ha sido dejado fuera del presente trabajo. En tal caso se trabajaría en la misma forma como la mostrada aquí, y el signo sólo aparecería al final.

Ejemplo: 2.305 x 162 = 373.410

373.410 x 51 = 19'043.910

TABLA 7

Tecla operada	REGISTRO RE										REGISTRO RA														
	12	11	10	9	8	7	6	5	4	3	2	1	12	11	10	9	8	7	6	5		4	3	2	1
	+	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	0	Estado inicial
2	+	0	0	0	0	0	0	0	0	0	0	2	+	0	0	0	0	0	0	0	0	0	0	0	P 2 en RE ₁₂ y (RE) ← 1
3	+	0	0	0	0	0	0	0	0	0	2	3	+	0	0	0	0	0	0	0	0	0	0	0	P 3 en RE ₁₂ y (RE) ← 1
0	+	0	0	0	0	0	0	0	0	2	3	0	+	0	0	0	0	0	0	0	0	0	0	0	P 0 en RE ₁₂ y (RE) ← 1
5	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	0	0	0	0	0	P 5 en RE ₁₂ y (RE) ← 1	
x	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	0	0	0	0	0	(RE) + (RA) → RE	
	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	0	0	0	0	0	P 0's en RA	
1	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	0	0	0	0	0	(RA) ← 1	
	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	0	2	3	0	5	1 x (RE) + (RA) → RA	
6	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	2	3	0	5	0	(RA) ← 1	
	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	0	3	6	8	8	0	6 x (RE) + (RA) → RA	
2	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	3	6	8	8	0	0	(RA) ← 1	
	+	0	0	0	0	0	0	2	3	0	5	+	0	0	0	0	0	3	7	3	4	1	0	2 x (RE) + (RA) → RA	
=	+	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	3	7	3	4	1	0	P 0's en RE	
x	+	0	0	0	0	0	3	7	3	4	1	0	+	0	0	0	0	0	3	7	3	4	1	0	(RE) + (RA) → RE
	+	0	0	0	0	0	3	7	3	4	1	0	+	0	0	0	0	0	0	0	0	0	0	0	P 0's en RA
5	+	0	0	0	0	0	3	7	3	4	1	0	+	0	0	0	0	0	0	0	0	0	0	0	(RA) ← 1
	+	0	0	0	0	0	3	7	3	4	1	0	+	0	0	0	0	1	8	6	7	0	5	0	5 x (RE) + (RA) → RA
1	+	0	0	0	0	0	3	7	3	4	1	0	+	0	0	0	1	8	6	7	0	5	0	0	(RA) ← 1
	+	0	0	0	0	0	3	7	3	4	1	0	+	0	0	0	1	9	0	4	3	9	1	0	1 x (RE) + (RA) → RA
=	+	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	1	9	0	4	3	9	1	0	P 0's en RE

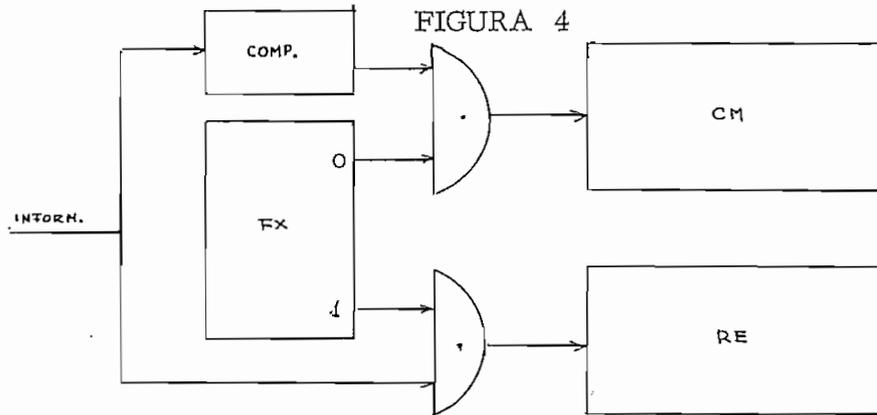
Conclusiones:

1. - La primera fase de la operación, o sea la de introducir el multiplicando, es exactamente igual a la suma o resta y será igual en la división. Esto es obvio puesto que la máquina no sabe aún qué operación va a ser ejecutada.
2. - En el proceso de aplastar la tecla x ocurre en primer lugar $(RE) + (RA) \rightarrow RE$. A primera vista, en el ejemplo anterior, no aparece cuál es el objeto de esta función, sobre todo si sólo fijamos nuestra atención en la primera tecla x aplastada, pero si miramos a la segunda tecla x se verá claramente su objeto.
3. - Se observa que después de aplastada la tecla x, las teclas de los dígitos no tienen la misma función como en la suma o resta; es por tanto obvio que la tecla x debe preparar ciertos circuitos que recuerdan lo ocurrido en el pasado a fin de ejecutar adecuadamente las funciones en el futuro.
4. - Los dígitos del multiplicador no entran al registro de entrada, por tanto éste debe ser bloqueado después de aplastar la tecla x. Esta deberá ser otra de las funciones que debe cumplir esta tecla.

En su lugar, el multiplicador es alimentado a un contador: el contador de multiplicación (GM) que se encarga de contar el número de sumas sucesivas como se explicará enseguida.

El proceso de bloqueo del registro de entrada se detalla en la Figura 4.

Dicha figura no es una representación exacta del circuito que se empleará, pero es buena para explicar el principio de funcionamiento que se sigue.



Al aplastar la tecla "x" se pone en estado cero el flip-flop de multiplicación FX, bloqueando así la compuerta que va a RE₁₂ y permitiendo que la información, en forma de complemento de 10 pase a CM, es decir que si aplasto la tecla 6 aparece en CM un 4 (= 10 - 6).

Cada vez que se ha realizado una suma sucesiva en la unidad aritmética, el contador CM es incrementado en 1, de modo que en el ejemplo anterior, después de la primera suma, CM contendrá un 5, y después de 6 sumas sucesivas contendrá un 0 y llevo "1". Este llevo aparece en FLM (flip-flop de llevo del contador de multiplicación) y él impide que se realicen más sumas sucesivas.

5. - Las sumas sucesivas se llevan a cabo como se muestra en la Tabla 8.

Partamos del ejemplo anterior (Tabla 7) antes de insertar el 6.

TABLA 8

Tecla operada	REGISTRO RE	REGISTRO RA	
	12 11 10 9 8 7 6 5 4 3 2 1	12 11 10 9 8 7 6 5 4 3 2 1	
6	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 2 3 0 5	1 x (RE)+(RA) → RA
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 2 3 0 5	(RA) ← 1
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 2 5 3 5 5	(RE)+(RA) → RA
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 2 7 6 6 0	(RE) + (RA) → RA
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 2 9 9 6 5	(RE) + (RA) → RA
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 3 2 2 7 0	(RE) + (RA) → RA
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 3 4 5 7 5	(RE) + (RA) → RA
	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 3 6 8 8 0	(RE) + (RA) → RA
2	+ 0 0 0 0 0 0 0 0 2 3 0 5	+ 0 0 0 0 0 0 0 0 3 6 8 8 0 0	(RA) ← 1

6. - La información sobre el multiplicando se mantiene en RE hasta antes de aplastada la tecla =, pero una vez aplastada se pierde.

7. - La información sobre el multiplicador se pierde una vez alimentado.

8. - La unidad de salida debe conectarse en la siguiente forma:

- a) Durante la primera fase, hasta que se haya aplastado una tecla de dígito después de la tecla x : al registro RE
- b) Una vez aplastada una tecla de dígito después de la tecla x : al registro RA.

9. - De los ejemplos anteriores se ve que no hay aparentemente razón para no realizar la multiplicación también con números negativos. Quienes así piensan están asistidos de parte de verdad: se puede realizar la multiplicación con números negativos. Observe el siguiente

$$\begin{aligned} \text{Ejemplo: } & -121 \times 321 = -38.841 \\ & (121)^2 = 878 \quad (38.841)^2 = 61.158 \end{aligned}$$

TABLA 9

Tecla operada	REGISTRO RE													REGISTRO RA													
	FLA																										
	12	11	10	9	8	7	6	5	4	3	2	1	12	11	10	9	8	7	6	5	4	3	2	1			
	+	0	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	Estado inicial		
-	+	0	0	0	0	0	0	0	0	0	0	0	0	+	0	0	0	0	0	0	0	0	0	0	(RE) + (RA) → RA		
	-	9	9	9	9	9	9	9	9	9	9	9	0	+	0	0	0	0	0	0	0	0	0	0	P 9's en RE		
1	-	9	9	9	9	9	9	9	9	9	9	8	0	+	0	0	0	0	0	0	0	0	0	0	P (1)' en RE ₁₂ y (RE) ← 1		
2	-	9	9	9	9	9	9	9	9	9	8	7	0	+	0	0	0	0	0	0	0	0	0	0	P (2)' en RE ₁₂ y (RE) ← 1		
1	-	9	9	9	9	9	9	9	8	7	8	8	0	+	0	0	0	0	0	0	0	0	0	0	P (1)' en RE ₁₂ y (RE) ← 1		
x	-	9	9	9	9	9	9	9	8	7	8	8	0	+	0	0	0	0	0	0	0	0	0	0	(RE) + (RA) → RE		
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	9	9	9	P 9's en RA si RE ₁₂		
3	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	9	9	9	(RA) ← 1		
	-	9	9	9	9	9	9	9	8	7	8	8	1	-	9	9	9	9	9	9	9	8	7	7	(RE) + (RA) → RA		
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	8	7	8	1 + (RA) ₁ si FLA		
	-	9	9	9	9	9	9	9	8	7	8	8	1	-	9	9	9	9	9	9	9	7	5	6	(RE) + (RA) → RA		
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	7	5	7	1 + (RA) ₁ si FLA		
	-	9	9	9	9	9	9	9	8	7	8	8	1	-	9	9	9	9	9	9	9	6	3	5	(RE) + (RA) → RA		
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	6	3	6	1 + (RA) ₁ si FLA		
2	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	6	3	6	9	(RA) ← 1	
	-	9	9	9	9	9	9	9	8	7	8	8	1	-	9	9	9	9	9	9	9	6	2	4	7	(RE) + (RA) → RA	
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	6	2	4	8	1 + (RA) ₁ si FLA	
	-	9	9	9	9	9	9	9	8	7	8	8	1	-	9	9	9	9	9	9	9	6	1	2	6	(RE) + (RA) → RA	
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	6	1	2	7	1 + (RA) ₁ si FLA	
1	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	6	1	2	7	9	(RA) ← 1
	-	9	9	9	9	9	9	9	8	7	8	8	1	-	9	9	9	9	9	9	9	6	1	1	5	7	(RE) + (RA) → RA
	-	9	9	9	9	9	9	9	8	7	8	8	0	-	9	9	9	9	9	9	9	6	1	1	5	8	1 + (RA) ₁ si FLA
=	+	0	0	0	0	0	0	0	0	0	0	0	0	-	9	9	9	9	9	9	9	6	1	1	5	8	P 0's en RE

La tabla anterior ha sido hecha en detalle para que se pueda ver claramente las complicaciones que surgen al trabajar con números negativos. Prácticamente en cada paso hay que hacer $1 + (RA)_1$ si FLA, lo que equivale a doble utilización de la unidad aritmética. Todo se podía haber obviado con un circuito de conservación del signo al comienzo. El problema se complica más cuando es el multiplicador el que tiene signo negativo, y más todavía cuando tanto multiplican-

do como multiplicador son negativos. Además como la división sólo se hará con números positivos, por razones que se verá más adelante, cuando se use multiplicación y división combinadas van a surgir graves problemas.

Estas consideraciones hacen que, no en un afán de evitar el problema, sino con la aspiración de construir una máquina versátil y económica pero no perfecta, se haya puesto como condición la de que la operación x se realice sólo con números positivos.

4. ANALISIS DE LA OPERACION DIVISION

Es la más difícil. Se realizará sólo con números positivos. Se ejecuta restando sucesivamente el divisor del dividendo, tantas veces cuantas sean necesarias para obtener un residuo negativo. Esta última resta se compensa con una suma. Hecho esto se desplaza el dividendo un lugar a la izquierda y se continúa con el proceso para obtener el próximo dígito del cociente. El número de restas sucesivas realizadas en cada paso, es el dígito del cociente. Ellas se cuentan en el contador de división (CD).

Ejemplo del principio de la división: $2.712 \div 12 = 226$

TABLA 10

2 712	
- 2 x 12	2 restas sucesivas
<u>-----</u>	
0 312	
312	
- 2 x 12	2 restas sucesivas
<u>-----</u>	
0 72	
72	
- 6 x 12	6 restas sucesivas
<u>-----</u>	
00	

Resultado = 2 2 6

Ejemplo: $2.712 \div 12 = 226$
 $(12)^* = 87$

TABLA 11

Tecla operada	REGISTRO RE	CD	FLA	REGISTRO RA	REGISTRO RR	Estado inicial		
121110987654321			121110987654321		121110987654321			
+	00000000000	0	+	00000000000	+	00000000000		
2	+	00000000002	0	+	00000000000	P 2 en RE ₁₂ y (RE) ← 1		
7	+	00000000027	0	+	00000000000	P 7 en RE ₁₂ y (RE) ← 1		
1	+	00000000271	0	+	00000000000	P 1 en RE ₁₂ y (RE) ← 1		
2	+	00000002712	0	+	00000000000	P 2 en RE ₁₂ y (RE) ← 1		
÷	+	00000002712	0	+	00000002712	(RE) + (RA) → RA		
-	99999999999	0	+	00000002712	+	00000000000	P 9's en RE	
-	99999999999	0	+	02712000000	+	00000000000	(RA) ← 10-4	
1	-	99999999998	0	+	02712000000	+	00000000000	P (1) en RE ₁₂ y (RE) ← 1
2	-	99999999987	0	+	02712000000	+	00000000000	P (2) en RE ₁₂ y (RE) ← 1
=	-	98799999999	0	+	02712000000	+	00000000000	(RE) ← 10-2
-	98799999999	0	+	02712000000	+	02712000000	(RA) → RR y Reset FLA	
-	98799999999	0	1	+	01511999999	+	02712000000	(RE) + (RA) → RA
-	98799999999	1	1	+	01512000000	+	02712000000	1 + (RA) → RA si FLA e Inc
-	98799999999	1	0	+	01512000000	+	01512000000	(RA) → RR y Reset FLA
-	98799999999	1	1	+	00311999999	+	01512000000	(RE) + (RA) → RA
-	98799999999	2	1	+	00312000000	+	01512000000	1 + (RA) → RA si FLA e Inc
-	98799999999	2	0	+	00312000000	+	00312000000	(RA) → RR y Reset FLA
-	98799999999	2	0	-	99111999999	+	00312000000	(RE) + (RA) → RA
-	98799999999	2	0	+	00312000000	+	00312000000	(RR) → RA si FLA
-	98799999999	2	0	+	00312000000	+	00312000000	(CD) RA ₁₂ y (RA) ← 1
-	98799999999	0	0	+	00312000000	+	00312000000	P 0 en CD
-	98799999999	0	0	+	00312000000	+	03120000002	(RA) → RR y Reset FLA
-	98799999999	0	1	+	01920000000	+	03120000002	(RE) + (RA) → RA
-	98799999999	1	1	+	01920000000	+	03120000002	1 + (RA) → RA si FLA e Inc



(... continúa en la página siguiente)

En la Tabla 10 no se muestra como va a llevarse el proceso dentro de la máquina; no se dice tampoco que las restas sucesivas van a realizarse mediante sumas sucesivas del complemento de 9 del divisor. Conviene pues aclarar este tópico. La Tabla 11 es un ejemplo detallado de la operación división.

Conclusiones:

1. - En esta operación se utilizan ya 3 registros. El único objeto del tercer registro es compensar la última resta sucesiva cuando el residuo es negativo. La misma función se puede lograr solamente con dos registros, y el proyecto original para la presente tesis se realizó en esa forma (estimo innecesario, por lo largo y repetido, detallar en otra tabla el proyecto original). Conformémonos con pensar que la unidad de control, para el caso de dos registros, se complica notoriamente, y que puesto que existen tres registros, siendo el registro RR el adicional con el objeto de darle versatilidad a la máquina para que pueda realizar operaciones combinadas como se explicó en la Tabla 1, por qué no aprovechar este tercer registro para esta operación si ello redundará en reducción de los circuitos de la unidad de control?

2. - La primera fase de la operación, hasta antes de aplastar la tecla + , es similar a la suma, resta y multiplicación. Una vez aplastada la tecla + ocurre: $(RE) + (RA) \rightarrow RA$ y P 9's en RE que era lo mismo que ocurría al aplastar la tecla -. Pero además se rota el (RA) de modo que el primer dígito significativo se sitúe en la posición RA_{10} que ha sido arbitrariamente fijada como tope. Esta misma rotación tiene lugar al aplastar la tecla =, pero con el (RE).

3. - Los dígitos del divisor entran en igual forma que el sustraendo de una resta, es decir en forma de su complemento de 9.

4. - El proceso mismo de división comienza al aplastar la tecla =. Cada vez que se produce una resta sucesiva hay un llevo, el cual incrementa el contenido de CD (contador de división). Siguen las restas sucesivas hasta que no se produce llevo, lo cual es una indicación de que el residuo es negativo y que se debe compensar la última resta sucesiva, proceso éste que se realiza transfiriendo (RR) a RA. Así pues en CD queda el primer dígito del cuociente y este a su vez es almacenado en RA₁₂.

Viene luego una rotación a la izquierda del (RA) y estamos listos para obtener un próximo dígito del cuociente. El proceso es repetido 10 veces (controladas por un nuevo contador: el contador auxiliar de división (CAD) para obtener los 10 dígitos del cuociente.

5. - Se observa que cada vez que se produce un llevo hay que incrementar RA en uno. Este es un largo proceso puesto que en el peor de los casos equivale a una nueva suma. Esto ocurre porque se pone en RE no el verdadero complemento de 10 del divisor que sería lo adecuado, sino el complemento de 9, y entre uno y otro hay la diferencia de una unidad.

Ejemplo: $(3.217)^{\circ} = 6783$

$(3.217)^{\square} = 6.782$

La solución que nace enseguida es la de poner en RE el verdadero complemento de 10 del divisor. Definitivamente esto aceleraría el proceso de división (y el de resta). Pero surge el problema: para formar el verdadero complemento de 10 de un número de algunas cifras hay que conocer el número en total. En el ejemplo anterior, si 7 no fuera la última cifra, en el verdadero complemento de 10 no aparecería correspondientemente el 3 sino el 2. De modo que el diseño de un circuito complementador de 10 (que se debe hacer para el peor caso) entrañaría hacer un circuito complementador para 12 dígitos (48 bits), a más de que el dividendo no se debería poner en RE sino en otro registro para una vez complementado pasarlo a RR, y como siempre ocurre, cuando se trata de hacer una máquina con mejor performance el costo aumenta. En este caso escojo la máquina más lenta pero más barata.

6. - Las funciones de la tecla = después de aplastada la tecla + , son diferentes de las que tenía después de aplastada la tecla x. Por tanto la tecla + debe preparar ciertos circuitos para este cambio de funciones.

7. - La unidad de salida debe conectarse en la siguiente forma:

- a) Antes de aplastar la tecla + : muestra (RE)
- b) Al aplastar la tecla + : muestra (RA)
- c) Antes de aplastar la tecla = : muestra (RE)'
- d) Al aplastar la tecla = : muestra (RA)

5. ANALISIS DE LA OPERACION $(a + b) c =$

Ejemplo: $(1.203 + 1.102) 162 = 373.410$

La primera suma se realiza en exactamente la misma forma explicada anteriormente, así, una vez finalizada tendremos:

+ 0 2 3 0 5 P 0's en RE

Ahora al aplastar la tecla x

x + 0 0 0 0 0 0 0 0 2 3 0 5 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (RE) + (RA) RE

Quedamos en las mismas condiciones que el ejemplo de multiplicación y así el proceso será llevado en la misma forma.

Una vez terminado este proceso y aplastada la tecla = quedamos en la siguiente condición:

RE RA
= + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 + 0 0 0 0 0 3 7 3 4 1 0

Nótese que esta operación puede ser realizada siempre que el resultado parcial de $(a + b)$ sea positivo, puesto que la multiplicación sólo se realiza con números positivos.

El análisis de esta operación nos capacita para afirmar que todavía para ella se requieren sólo dos registros y que las operaciones manuales requeridas son:

$a + b + x c =$

6. ANALISIS DE LA OPERACION $\frac{(a + b) c}{d} =$

Para la presente operación tenemos en el numerador la operación analizada en el ejemplo anterior pero se ha agregado una operación división.

$$\text{Ejemplo} = \frac{(1.203 + 1.102) 162}{25} = 14.936,4$$

La operación del numerador es exactamente igual a la del ejemplo anterior. Una vez terminada vamos a quedar en la siguiente condición

$$= \begin{array}{r} \text{RE} \\ + 000000000000 \\ \text{RA} \\ + 00000373410 \end{array}$$

Al apastar la tecla \div vamos a tener

$$\div \begin{array}{r} + 000000000000 \\ - 999999999999 \\ - 999999999999 \end{array} \begin{array}{r} + 00000373410 \\ + 00000373410 \\ + 03734100000 \end{array} \begin{array}{l} (\text{RE}) + (\text{RA}) \rightarrow \text{RA} \\ \text{P 9's en RE} \\ (\text{RA}) \leftarrow 10 - 6 \end{array}$$

A partir de este punto el proceso de división sigue como el explicado en el análisis de la operación división.

No hace falta recordar que puesto que la división sólo puede realizarse con números negativos, la presente operación y en general cualquier operación combinada en la que entren multiplicación y/o división estará limitada por la misma restricción.

Del presente análisis se desprende que para esta operación se requieren sólo dos registros, pero la información sobre subtotales parciales se pierde. También se observa que las operaciones manuales requeridas son:

$$a + b + \times c = \div d =$$

Esta operación es un buen ejemplo para hacer notar que la misma tecla (por ejemplo la tecla =) aplastada después de una operación x no tiene las mismas funciones que aplastada después de +.

Una vez aplastada la última tecla = tendremos la siguiente condición:

RE	RA	
+ 0 0 0 0 0 0 0 0 0 0 0 0	0 + 0 1 4 9 3 6 4 0 0 0 0	P 0's en RE

Analizando este último resultado, se observa que después del proceso de división, la información sobre los registros no queda en una posición correcta como para realizar adicionales operaciones a posteriori; por tanto, si en una operación combinada entra la división, se debe, de ser posible, dejarla para el último y, si esto no es factible por la naturaleza misma de la operación a realizarse, será necesario reescribir en la máquina el resultado parcial para poder seguir adelante. Se podría obviar este problema añadiendo más teclas de las previstas, pero esto lleva a aumento en el costo y a complicaciones en los circuitos, así como dificultad en el manejo, que para ciertos operadores no capacitados, en manos de quienes va a estar el calculador, no serían aceptables.

7. ANALISIS DE LA OPERACION

a x b =
c x d = +
e x f = +
<hr/> Total

Esta operación se realiza con tres registros.

Cada operación separada se realiza de la misma forma que la explicada bajo el título: "análisis de la operación multiplicación". Así, una vez aplastada la

primera tecla "=", se tiene la siguiente condición:

El registro RE limpio

El registro RA con el resultado de $a \times b$

El registro RR limpio

Conviene en este punto hacer un paréntesis para estudiar las funciones que desempeñan dos nuevas teclas que se han añadido a la máquina, y que sirven para hacerla sumamente versátil:

Tecla "R" : ejecuta las siguientes funciones

1. - $(RR) \rightarrow RE$
2. - $(RE) + (RA) \rightarrow RA$
3. - limpie RE

Es decir ejecuta exactamente lo que la tecla "+" pero previamente transfiere el contenido de RR a RE

Tecla "T": ejecuta las siguientes funciones

1. - $(RA) \rightarrow RR$
2. - Limpie RA

Con la adición de estas dos teclas, se puede intercambiar en variadas formas la información entre los registros. De ahí que sean muy útiles para todo tipo de operación en el que se deba guardar un resultado parcial.

Después de la primera tecla "=" se aplasta la tecla "R", de modo que la condición ahora es:

El registro RE limpio

El registro RA con el resultado de $a \times b$

El registro RR limpio

A continuación aplastamos la tecla "T" y la nueva condición es:

El registro RE limpio

El registro RA limpio

El registro RR con el resultado de $a \times b$

Entonces quedan RE y RA aptos para realizar la segunda parte, de modo que después de aplastada la segunda tecla "=" la condición es:

El registro RE limpio

El registro RA con el resultado de $c \times d$

El registro RR con el resultado de $a \times b$

Al aplastar la tecla "R" quedan:

El registro RE limpio

El registro RA con el resultado de $c \times d + a \times b$

El registro RR con el resultado de $a \times b$

Al aplastar la tecla "T" :

El registro RE limpio

El registro RA limpio

El registro RR con el resultado de $c \times d + a \times b$

Claramente se ve que las operaciones manuales requeridas son:

$$a \times b = RT \quad c \times d = RT \quad e \times f = RT$$

La unidad de salida debe conectarse en la siguiente forma:

- a) Hasta aplastada la tecla x: muestra (RE)
- b) Después de aplastada la tecla x: muestra (RA)
- c) Al aplastar la tecla R: muestra (RA)
- d) Al aplastar la tecla T: muestra (RR)

8. ANALISIS DE LA OPERACION

$$\begin{array}{r} a \times b = \\ c \times d = + \\ e \times f = + \\ \hline \text{Subtotal} \end{array}$$
$$\begin{array}{r} \text{Subtotal} \times g = + \\ \hline \text{Total} \end{array}$$

Esta operación es una muestra de la versatilidad adquirida por la máquina al añadir las dos últimas teclas.

La primera parte, hasta la obtención del subtotal es similar a la operación analizada inmediatamente antes, de modo que la condición es:

El registro RE limpio

El registro RA limpio

El registro RR con el resultado de $e \times f + c \times d + a \times b$

Al aplastar una vez más la tecla R, quedan:

El registro RE limpio

El registro RA con el resultado de $e \times f + c \times d + a \times b$

El registro RR con el resultado de $e \times f + c \times d + a \times b$

Al aplastar la tecla "x" quedan:

El registro RE con el resultado de $e x f + c x d + a x b$

El registro RA limpio

El registro RR con el resultado de $e x f + c x d + a x b$

Al aplastar la tecla "=", quedan:

El registro RE limpio

El registro RA con el resultado de $g x (e x f + c x d + a x b)$

El registro RR con el resultado de $e x f + c x d + a x b$

Al aplastar la tecla "R", quedan:

El registro RE limpio

El registro RA con el resultado de $(e x f + c x d + a x b) +$
 $+ g (e x f + c x d + a x b)$

El registro RR con el resultado de $e x f + c x d + a x b$

Las operaciones manuales requeridas son:

$$a x b = RT \quad c x d = RT \quad e x f = RTR \quad x g = R$$

La unidad de salida debe conectarse siguiendo las mismas normas explicadas en el análisis de la operación anterior

Existe, como aparece evidente desde estos análisis, una gran variedad de operaciones adicionales que la máquina puede implementar, pero éstas resultan muy obvias para analizarlas y además serían por sí mismas objeto de decenas de páginas para detallarlas, páginas sosas por lo repetidas e insustanciosas.

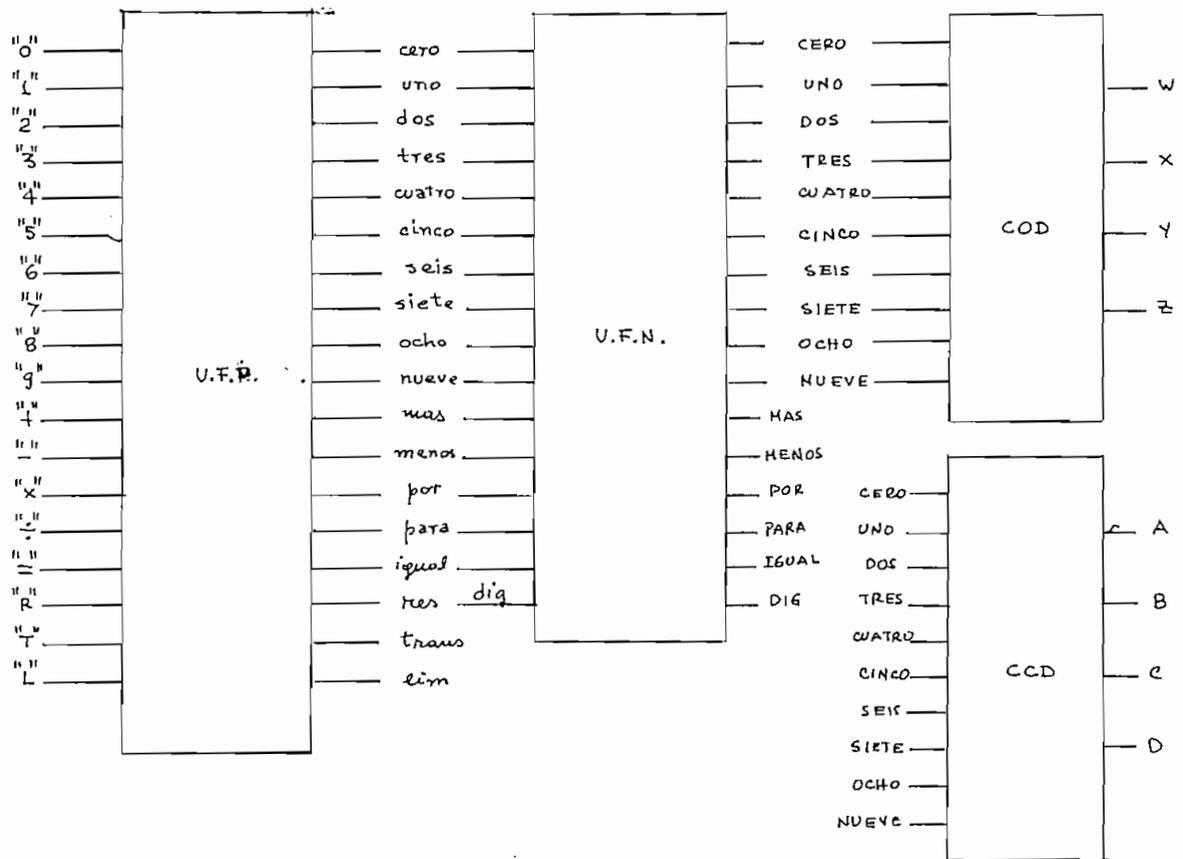
Por tanto, nos es suficiente con las operaciones hasta aquí analizadas, dejando constancia de que la máquina puede realizar mucho más.

A este punto el panorama se ha clarificado y estoy apto para elaborar un diagrama de bloques desarrollado, teniendo en cuenta todas las conclusiones obtenidas del análisis. Luego vendrá el arduo trabajo de elaborar el programa que se está siguiendo, escribir las ecuaciones que significa ese programa y luego el diseño final.

DIAGRAMA DESARROLLADO
DE
BLOQUES

1. DIAGRAMA DE BLOQUES DE LA UNIDAD DE ENTRADA

FIG. 5



Nomenclatura:

1. - UFP = unidad formadora de pulsos
2. - UFN = unidad formadora de niveles
3. - COD = codificador
4. - CCD = circuito complementador de diez

Notación:

1. - Cualquier símbolo escrito entre comillas significa una tecla.

Ejemplos: "3" significa la tecla 3

"=" significa la tecla =

2. - Cualquier símbolo escrito en minúscula significa un pulso.

Ejemplos: tres significa el pulso tres

igual significa el pulso igual

3. - Cualquier símbolo escrito en mayúscula significa un nivel.

Ejemplos: TRES significa el nivel TRES

IGUAL significa el nivel IGUAL

(Cuando hablo de niveles, me refiero a niveles lógicos; ver Apéndice

"CONVENCIONES LOGICAS")

- A. - UNIDAD FORMADORA DE PULSOS (UFP)

Transforma la información que el operador da a la máquina al aplastar una tecla., en un lenguaje que la máquina pueda entender.

Asociado con cada tecla, que no es sino un switch que produce una transición entre -5,2V y OV, va un tipo especial de Buffer, el disparador Schmitt, que se encarga de transformar los niveles -5,2V y OV a otro tipo de niveles lógicos que son empleados por MECL (Motorola Emitter Coupled Logic), que son los circuitos empleados en este trabajo. A continuación hay un multivibrador monoestable que se encarga de formar un pulso, y finalmente un circuito especial de retardo diseñado de modo que, con personas nerviosas que tiemblan al aplastar una tecla, o molestosas, que suelen jugar aplastando muchas veces y muy rápido la misma tecla, la máquina no responda a posteriores aplastadas de una misma tecla, si no ha transcurrido un tiempo "razonable" desde la primera presión de la tecla.

En total, esta unidad forma un pulso, correspondientemente a cada tecla de que dispone el tablero exterior. Este pulso tiene características especiales de amplitud, duración, etc., de modo que puede ser interpretado por el resto del calculador.

Por tanto, los pulsos de que se dispone ahora son: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, mas, menos, por, para, igual, res, trans, lim, correspondientes a cada tecla de igual nombre. Además se forma otro pulso al aplastar cualquier tecla de dígito, cero hasta nueve, éste se ha llamado pulso dig.

B. - UNIDAD FORMADORA DE NIVELES (UFN)

Del análisis de las operaciones, recordará el lector que ciertas teclas preparan circuitos para que en lo posterior trabajen de una manera especial. Es entonces claramente necesario que la máquina recuerde cual tecla ha sido aplastada en el pasado para realizar ciertas funciones de acuerdo a esas anteriores órdenes. Esto sería imposible de conseguir solamente con los pulsos formados por la unidad inmediata anterior, puesto que una vez desaparecido el pulso se pierde toda información. De ahí que sea necesario el formar niveles que indiquen en forma continua cuál tecla se ha aplastado. Estos niveles serán borrados (digámoslo así), cuando ya no se necesiten.

De acuerdo a estas necesidades, la presente unidad se conformará básicamente de biestables del tipo R-S. A cada entrada S (set) va un pulso formado en la unidad anterior para formar el nivel correspondiente. La entrada R (reset) se usará cuando se desee borrar el nivel por ya no ser necesario.

Los niveles formados son:

CERO, UNO, DOS, TRES, CUATRO, CINCO, SEIS, SIETE, OCHO, NUEVE, MAS, MENOS, POR, PARA, IGUAL, DIG.

No se necesita niveles TRANS, RES ni LIM.

C. - CODIFICADOR (COD)

Puesto que se tiene diez dígitos y sólo dos niveles lógicos (representados por 0 o 1), se debe escoger una cierta forma de representación de cada número pa

ra alimentarlo dentro de la máquina en una forma en que ésta entienda. Esta labor la realiza esta unidad. Deberá utilizarse por lo menos 4 bits para la codificación ($2^4 > 10$). Por razones especiales de simetría, etc., discutidas al comienzo del presente trabajo, se escogió el código de AIKEN.

La unidad tendrá consecuentemente 10 entradas, correspondientes a los 10 dígitos a codificarse, y 8 salidas, correspondientes a los dos sexos de los 4 bits en que se codifica cada dígito. Estas salidas serán conocidas en lo posterior, con los nombres W, X, Y, Z y \bar{W} , \bar{X} , \bar{Y} , \bar{Z} (léase: no W, no X, no Y, no Z)

Ejemplo: la codificación para el dígito 5 según el código de Aiken es:

$$\begin{array}{rcccl} \text{dígito} & & \text{WXYZ} & & \overline{\text{WXYZ}} \\ 5 & = & 1011 & & 0100 = 4 \end{array}$$

Observe que $\overline{\bar{W}\bar{X}\bar{Y}\bar{Z}} = (WXYZ)^*$ puesto que $5 + 4 = 9$

D. - CIRCUITO COMPLEMENTADOR DE DIEZ (CCD)

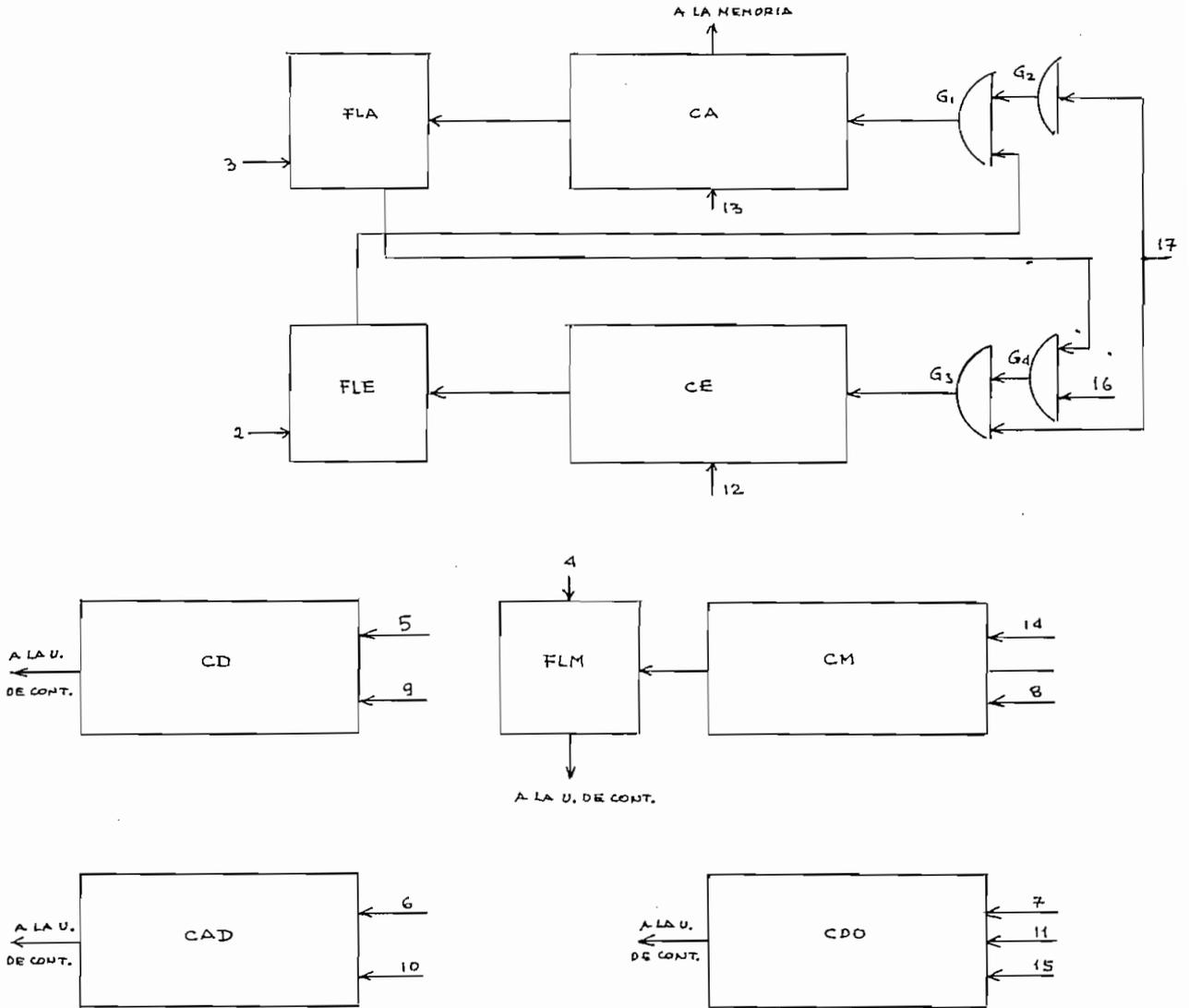
Estudiamos en el análisis de la operación multiplicación, que la información sobre el multiplicador va a CM (contador de multiplicación), pero no en forma original sino en forma de su complemento de 10. Esta unidad se encarga de hacer ese trabajo. Estará consecuentemente formada de circuitos combinacionales y tendrá 8 entradas, correspondientes a W, X, Y, Z, \bar{W} , \bar{X} , \bar{Y} , \bar{Z} y ocho salidas que llamaremos A, B, C, D, \bar{A} , \bar{B} , \bar{C} , \bar{D} . Por definición $ABCD = (WXYZ)''$

Ejemplo:

$$\begin{array}{rcccccccc} \text{dígito} & & \text{W X Y Z} & & \text{A B C D} & & \overline{\bar{A}\bar{B}\bar{C}\bar{D}} \\ 4 & & 0 1 0 0 & & (4)'' = 6 = 1 1 0 0 & & 0 0 1 1 = 3 \end{array}$$

2. DIAGRAMA DE BLOQUES DE LA UNIDAD ARITMETICA

FIG. C



Nomenclatura:

1. - CE: contador de entrada
2. - CA: contador auxiliar
3. - CM: contador de multiplicación
4. - CD: contador de división
5. - CAD: contador auxiliar de división
6. - FLE: flip flop de llevo de CE
7. - FLA: flip flop de llevo de CA
8. - FLM: flip flop de llevo de CM
9. - CDO: contador de doce
10. - G_1, G_2, G_3, G_4 : compuertas

Terminales:

- 2: Ponga FLE en cero
 - 3: Ponga FLA en cero
 - 4: Ponga FLM en cero
 - 5: Limpie CD
 - 6: Limpie CAD
 - 7: Limpie CDO
 - 8: - Incremente CM en uno
 - 9: Incremente CD en uno
 - 10: Incremente CAD en uno
 - 11: Incremente CDO en uno
 - 12: Ponga información en CE
 - 13: Ponga información en CA
-

- 14: Ponga información en CM
- 15: Ponga CDO en doce
- 16: Incremente CE en uno si FLA
- 17: Incremente CE en uno y CA en uno si FLE.

Esta unidad está compuesta de 6 contadores. Cuatro de ellos que ya conocemos, CE, CA, CD y CM cuentan en el código de Aiken. Los dos restantes CAD y CDO por razones de ahorro se diseñarán para contar en el código DCB.

La parte más importante de la unidad aritmética y el corazón de la máquina, es la formada por CE, CA, FLE, FLA, G_1 , G_2 , G_3 y G_4 . Sirve en realidad para sumar dos dígitos. Su funcionamiento se explicó ya brevemente en otra parte de este trabajo pero conviene repetirlo algo más explícitamente. Los dos dígitos a ser sumados son puestos en CE y CA mediante los terminales 12 y 13 y proceden generalmente de la memoria (RE_1 y RA_1). Se asume inicialmente FLE en estado cero, FLA en cero y CDO en cero. Suponga que estos dos dígitos son 5 en CE y 7 en CA. Si 10 pulsos son aplicados al terminal 17, los 5 primeros incrementan CE pero no CA pues todavía no hay llevo en FLE. De modo que ahora estamos con cero en CE, uno en FLE, 7 todavía en CA y 0 en FLA. Los cinco pulsos restantes incrementan tanto CE como CA. Una vez aplicados todos los pulsos quedamos con 5 en CE, 1 en FLE, 2 en CA y 1 en FLA. Observe que $5 + 7 = 2$ y llevo 1. El 2 queda en CA y el llevo 1 en FLA.

Ahora puedo pasar el primer resultado parcial para guardarlo en la memoo

ría, a la vez que incrementar CDO en uno para recordar que ya hemos sumado un par de dígitos. Ponemos luego un próximo par de dígitos en CE y CA y en es ta vez se ve claro que lo primero que debo hacer es tomar en cuenta el uno que llevaba. Esto se consigue aplicando un pulso al terminal 16. (Esto se debía hacer también en la primera vez, pero no lo hice para facilitar la explicación).

Sólo me faltaría borrar lo que hay en FLE y FLA para quedar listo en la misma forma que al principio.

El proceso se repite 12 veces, incrementando cada vez CDO en uno, para sumar los 12 pares de dígitos que pueden ser almacenados en RE y RA, de mo do que cuando CDO marca 12 se para el proceso.

A. CONTADOR DE ENTRADA, CE.

Su función acaba de ser explicada. Cuenta en el código de Aiken. Está formado por 4 flip flop o biestables del tipo J-K y por adicionales circuitos combinacionales que lo hacen contar en este código. La información le llega desde la memoria, concretamente desde RE_1 y es puesta a través del terminal 12. (Es evi dente que el terminal 12 no es sólo un terminal, sino 8 terminales correspondientes a las entradas R-S de cada flip flop, pues se está utilizando el método de superposición para poner información dentro de CE).

Puesto que nueva información es puesta cada vez que CE va a ser usado, no se necesita un terminal para limpiarlo.

Es usado en todas las operaciones: suma, resta, multiplicación, división y combinadas.

B. FLIP FLOP DE LLEVO DE CE, FLE.

Es un biestable del tipo J-K conectado de manera que pase de cero a uno cuando el contador de entrada va de 1 1 1 1 a 0 0 0 0, es decir de 9 a 0. Se usa para permitir el paso de pulsos que incrementen CA solamente cuando FLE esté en estado uno.

Cada vez que vaya a ser usado, debe ser limpiado para asegurarse que comienza en estado cero. Esto se consigue mediante el terminal 2. Un pulso aplicado a él, limpia FLE.

C. CONTADOR AUXILIAR, CA.

En él queda la respuesta de la suma de un par de dígitos. Su constitución es exactamente igual a la de CE, es decir está formado por 4 biestables del tipo J-K conectados de tal manera, mediante adicionales circuitos combinacionales, que le hacen contar en el código de Aiken. Generalmente, la información le llega desde la memoria, desde la posición uno de RA, y es puesta a través del terminal 13. Acabada una suma elemental, la información en él contenida es pasada nuevamente a la memoria, generalmente a la posición 12 de RA, aunque en ciertos casos (como sucede al aplastar la tecla x) la información es puesta en RE_{12} . Es incrementado sólo cuando FLE está en uno, por cada pulso que pase por G_1 .

También es empleado, como CE y FLE, en todas las operaciones.

D. FLIP FLOP DE LLEVO DE CA, FLA

Similar a FLE. Va de cero a uno cuando CA pasa de 9 a 0. En él queda el llevo de una suma elemental de un par de dígitos, de ahí que se use como condición para poner, a través de G_4 con un pulso aplicado al terminal 16, el llevo en CE, cuando se haga la suma de un próximo par de dígitos.

Es limpiado con un pulso al terminal 3.

E. COMPUERTAS G_1 , G_2 , G_3 , G_4 .

G_1 y G_4 son compuertas Y (AND)

G_3 es una compuerta O (OR)

G_2 se usa sólo como cambiadora de sexo * (* Referirse a capítulo "Convenciones Lógicas"), para acoplar el sexo alto de los pulsos aplicados al terminal 17 con el sexo bajo que entra a G_1 .

Un pulso al terminal 16 ordena poner el llevo proveniente de la suma elemental inmediata anterior.

Diez pulsos aplicados al terminal 17 ordenan sumar los dos dígitos contenidos en CE y CA. Tanto el pulso que va al terminal 16 como los que se aplica al terminal 17 vienen ordenados desde la unidad de control.

F. CONTADOR DE MULTIPLICACION, CM.

Similar a CE y CA, consta de 4 flip flops J-K conectados para contar en el código de Aiken.

Se utiliza sólo en la operación multiplicación y su labor es contar el número de sumas sucesivas realizadas cuando se le aplica un pulso al terminal 8.

El proceso es como sigue: cuando se ha apastado la tecla x, la información posterior no va, como usualmente sucede, al registro RE, sino que va a CM una vez que el dígito ha sido puesto en forma de complemento de 10. La información, por tanto, viene desde la unidad de entrada, de la salida del circuito complementador de 10 y es puesta por el método de superposición en CM, a través del terminal 14. Note nuevamente que 14 no es un solo terminal, sino 8 terminales, correspondientes a los 4 pares de entradas R-S de los flip flops de CM. Cuando se ha acabado una suma sucesiva se aplica un pulso al terminal 8, lo cual incrementa CM en uno. CM sigue así incrementándose hasta que cuando pasa de 9 a 0 ordena ir de 0 a 1 a FLM y esto sirve como indicación de que no se deben hacer más sumas sucesivas.

Tampoco CM necesita ser limpiado pues nueva información es superpuesta sobre él.

G. FLIP FLOP DE LLEVO DE CM, FLM

Es del tipo J-K. Va de 0 a 1 cuando CM va de 9 a 0. Si FLM = 1 se ordena a la unidad de control que pare de realizar sumas sucesivas.

Al comenzar el trabajo debe ponerse en 0 mediante un pulso aplicado al terminal 4.

H. CONTADOR DE DIVISION, CD.

Su función específica es contar el número de restas sucesivas que se hacen en el proceso de división y que arrancan al aplastar la tecla igual después de puesto el divisor. Es decir, en CD aparecen paso a paso cada uno de los dígitos que forman el cociente de una división. Si se revisa detenidamente el proceso de división, se notará que cada vez que aparece un uno en FLA debe incrementarse CD en uno. Esta es justamente la forma en que se cuentan las restas sucesivas.

Al igual que sus anteriores, consta de 4 flip flops J-K conectados para contar en el código de Aiken. Es usado solamente en procesos en que entre la división. Al comenzar la operación debe asegurarse que comienza en cero y esto se consigue aplicando un pulso al terminal 5. Es incrementado mediante pulsos aplicados al terminal 9.

La información en él contenida debe ser puesta nuevamente en la posición 12 de RA cuando lo ordene la unidad de control conforme se deduce del "análisis de la operación división".

No tiene un flip flop de llevo pues no es necesario ya que nunca se realizarán más de nueve restas sucesivas, sin ser de nuevo limpiado.

I. CONTADOR DE DOCE, CDO.

Cuenta en el código DCB. Cuando se ha realizado una suma elemental de un par de dígitos en la unidad aritmética, antes de proceder a sumar el próximo par, es necesario incrementar CDO en uno con el propósito de cuando se llegue a 12 dar una orden a la unidad de control que haga suspender las sumas elementales, es decir cuando ya se ha sumado los 12 pares de dígitos que pueden caber en RE y RA.

Cuando comience a ser usado debe también ser puesto en cero y ello se consigue con un pulso aplicado al terminal 7. Muchas otras veces conviene asegurarse de que está en estado 12 y para hacerlo basta con dar un pulso al terminal 15. Cada vez que se ha completado una suma elemental viene un pulso que lo incrementa a través del terminal 11. Consta también de 4 biestables J-K.

J. CONTADOR AUXILIAR DE DIVISION, CAD.

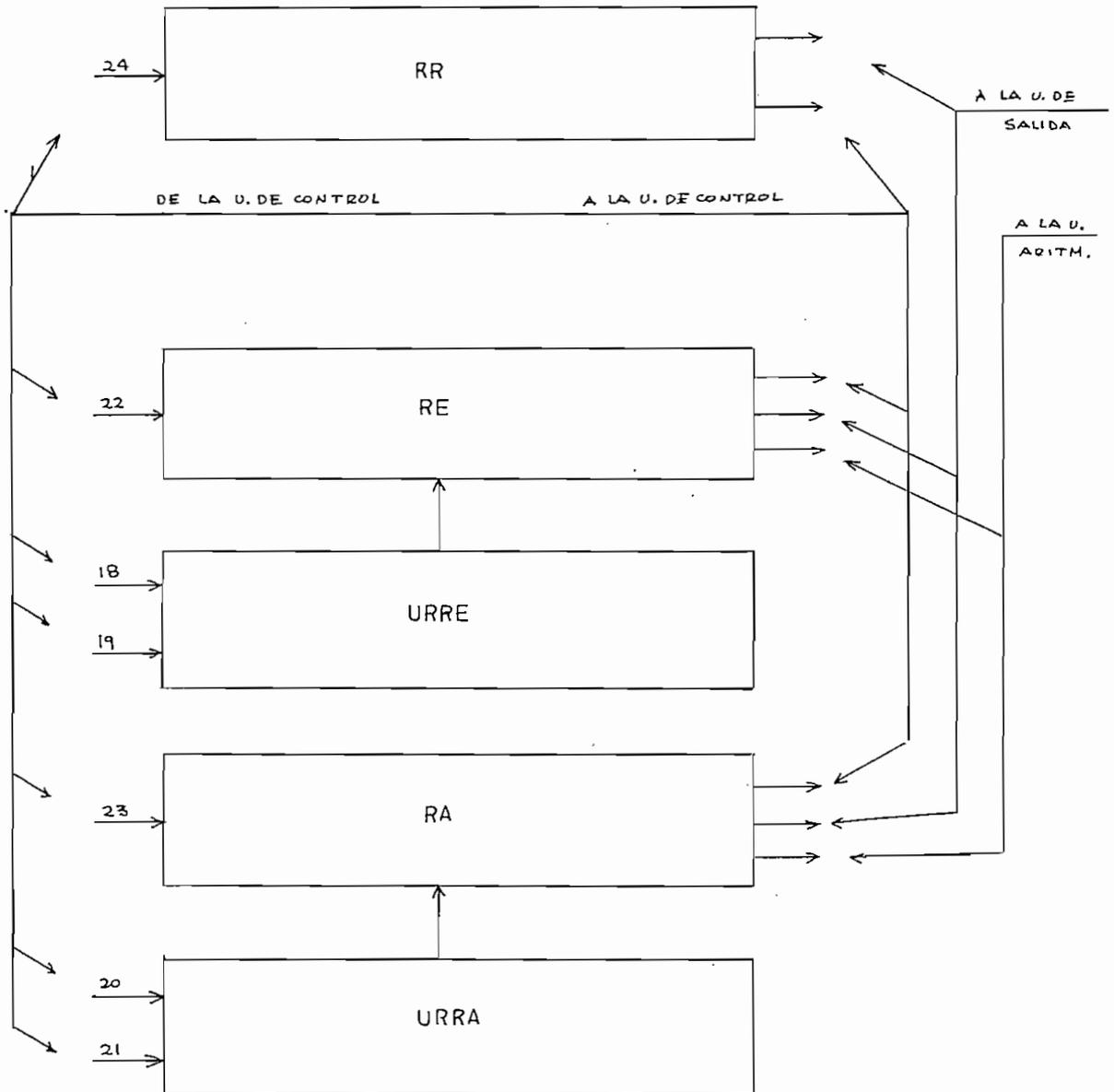
Volviendo nuevamente al análisis de la operación división se puede ver que el proceso de obtener cada dígito del cuociente se debe repetir durante 10 veces, de modo que en todos los casos el cuociente constará de 10 cifras. Quien cumple este cometido es el contador auxiliar de división. Viene para el objeto equipado con 4

biestables J-K conectados para contar en CDB. Debe ser limpiado al comienzo aplicando un pulso al terminal 6 y, es incrementado cada vez que se ha obtenido un dígito del cuociente, mediante un pulso aplicado al terminal 10.

Cuando llega a estado 10, ordena a la unidad de control que no se obtengan más dígitos del cuociente.

DIAGRAMA DE BLOQUES DE LA UNIDAD DE MEMORIA.

FIG. 7



Nomenclatura:

- RE: registro de entrada
- RA: registro auxiliar
- RR: registro de resultados
- URRE: unidad de rotación de RE
- URRA: unidad de rotación de RA

Terminales:

- 18: Rote el contenido de RE un lugar a la izquierda
- 19: Rote el contenido de RE un lugar a la derecha
- 20: Rote el contenido de RA un lugar a la izquierda
- 21: Rote el contenido de RA un lugar a la derecha
- 22: De información a RE
- 23: De información a RA
- 24: De información a RR

A. REGISTRO DE ENTRADA, RE.

Está, al igual que RA, formado por 12 lugares para dígitos y a su vez cada lugar de dígito por 4 bits, lo que da una capacidad de 48 bits.

Correspondientemente está formado por 48 flip flops del tipo J-K, que son elementos capaces de guardar información (De ahí que toda esta unidad se denomine con el nombre de memoria).

Como pueden ver de los análisis de todas las operaciones, RE se encarga generalmente de almacenar cada dígito cuya tecla es aplastada, excepción hecha del multiplicador en la operación multiplicación, de modo que, por regla general, la información llega a RE procedente de la unidad de entrada pero siempre comandada por la unidad de control que es la que ordena qué cosa y a qué tiempo debe ponerse en RE. Esto es realizado a través del terminal 22. Pero también llega a RE muy variado tipo de información. Por ejemplo: limpie RE, o ponga 9's en RE, o ponga (CE) en RE₁₂, o ponga (CA) en RE₁₂, o (RR) RE. Todo aquello se realiza a través del terminal 22 y siempre acatando las órdenes de la unidad de control.

Fácil es comprender que el terminal 22 no es un solo terminal. En realidad son 96, correspondientes a los 48 pares de terminales R-S de cada biesta-ble de RE. De éstos, los más usados son los de RE₁₂, pues por allí generalmente entra la mayor información. Asimismo tiene 96 salidas correspondientes a los dos estados de los 48 biestables. Estas van tanto a la unidad de salida (cuando quiere exhibirse el resultado que contiene RE), como a la unidad aritmética y comandado por la unidad de control (cuando se desea hacer cualquier operación, como suma, resta, etc.) como a la unidad de control (por ejemplo en la división cuando se desea poner el divisor marginado en la posición RE₁₀). De estas 96 salidas las más usadas son las de RE₁ como se puede recordar de los análisis.

Es usado en absolutamente todas las operaciones que realiza la máquina.

B. UNIDAD DE ROTACION DE RE, URRE.

Toma la información contenida en RE y la coloca nuevamente en él una vez convenientemente rotada sea uno o más lugares a la derecha o izquierda de acuerdo al número de pulsos aplicados ya sea al terminal 19 o 18.

Constará para ello de apropiados circuitos combinacionales (48 del mismo tipo) para cumplir este objetivo. En el diseño se verá que son 48 medio sumadores (Ver convenciones lógicas).

Conviene tal vez notar que se habla de rotar "lugares". Por ejemplo un pulso aplicado al terminal 18 hace que aquella que contiene RE_{11} va a RE_{12} , lo que contiene RE_{12} va a RE_1 , lo de RE_1 a RE_2 , para aclarar lo que alguien podría pensar, o sea rotar un bit.

C. REGISTRO AUXILIAR, RA.

Es completamente similar a RE. Consta de los mismos elementos, tiene las mismas entradas, las mismas salidas y cumple aproximadamente funciones semejantes, excepto que a él no va la información cuando se aplasta un dígito. También como RE, recibe información de muy distinta índole (como limpie RA, (CA) RA_{12} , (RR) RA, etc.) todo a través de su terminal 23 y comandado por la unidad de control. Las entradas más usadas son las de RA_{12} y las salidas más usadas, las de RA_1 . Envía información tanto a la unidad de salida (cuando quiere exhibirse el resultado contenido en él), a la unidad de control (cuando quiere marginarse el dividen

do a la posición RA_{10}), a la unidad aritmética (al contador auxiliar) y a URRRA (cuando se debe rotar su contenido). Se emplea en todas las operaciones.

D. UNIDAD DE ROTACION DE RA, URRRA.

Exactamente igual que URRE. Vale todo lo dicho para ella. Un pulso aplicado al terminal 20 rota (RA) un lugar a la izquierda, un pulso aplicado al terminal 21 rota (RA) un lugar a la derecha.

E. REGISTRO DE RESULTADOS, RR.

A diferencia de RA y RE consta de biestables del tipo J-K y no tiene una adicional unidad de rotación pues no se requiere que sea un registro de desplazamiento o de rotación, sino que solamente es usado para almacenamiento. No es usado en la operación suma, ni en la resta, ni en la multiplicación.

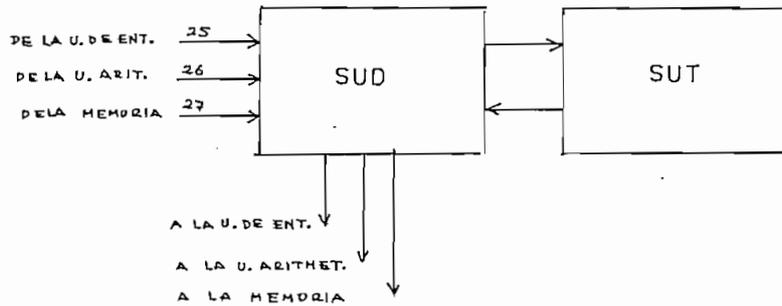
A semejanza de sus anteriores, tiene lugar para 12 dígitos, es decir para 48 bits.

Por el terminal 24 (que en realidad son 96 terminales) le llega información siempre comandada por la unidad de control y procedente ya sea de RA (cuando se ordena (RA) RR), o procedente de la unidad de entrada (cuando se desea limpiarlo).

El a su vez entrega sus conocimientos, a través de 96 terminales, ya sea a la unidad de salida (si se desea mostrar su contenido) o a la unidad de control (para que ésta la transfiera a RA o a RE).

DIAGRAMA DE BLOQUES DE LA UNIDAD DE CONTROL.

FIG. 8



Nomenclatura:

SUD: Subunidad de distribución

SUT: Subunidad de tiempo

Terminales:

25: ponga información en SUD desde la unidad de entrada.

26: ponga información en SUD desde la unidad aritmética

27: ponga información en SUD desde la unidad de memoria.

Viendo el diagrama de bloques de esta unidad, cualquiera exclamaría ¡fácil!. El que eso pronuncie se equivoca en mucho. Prácticamente todo el trabajo que entraña diseñar un calculador de este tipo, se concentra en formar esta

parte de la máquina, sobre todo la parte referente a SUD.

A. SUBUNIDAD DE TIEMPO, SUT.

Se encarga de formar pulsos que deben ser entregados a las distintas partes del calculador en instantes de tiempo adecuados y a través de terminales adecuados. Estos pulsos deben tener características especiales de duración y de separación entre ellos de modo que puedan cumplir eficazmente la labor a ellos encomendada.

Para mejor comprensión, y adelantando criterios, se observó, una vez terminado un diseño, que SUT debe formar en total 26 clases distintas de pulsos, que se han llamado sucesivamente

$tt_0, tt_1, tt_2, tt_3, t_0, t_1, t_2, t_3, t_4, \dots, t_{18}, t_{19}, t_{20}, t_{21}$

Estos pulsos deben salir de SUT en secuencias especiales. Por ejemplo unas veces sale una secuencia tt_0, tt_1, tt_2, tt_3 , en otras sale sólo una secuencia t_{16}, t_{17} y bajo determinadas condiciones sale una secuencia

$t_0, t_1, t_2, t_3, t_4 - t_{13}, t_{14}, t_{15}$. En esta anterior, por ejemplo, los pulsos marcados $t_4 - t_{13}$, 10 en total, salen por un mismo terminal y sirven para que aplicados al terminal 17 de la unidad aritmética realicen una suma elemental de los dígitos contenidos en CE y CA. Anteriormente es posible que se haya usado el pulso t_3 por ejemplo para poner los dígitos en CE y CA desde RE_1 y RA_1 , y así por el estilo.

Desde esta explicación se puede comprender que básicamente estará formado por un reloj maestro que forma pulsos y que pueda ser arrancado o parado a voluntad de acuerdo a órdenes de SUD, y también por algún tipo de contadores y de circuitos combinacionales que formen, o mejor separen, las distintas secuencias y, finalmente por otro tipo de elementos que fijen las condiciones bajo las cuales debe salir una o más determinadas secuencias.

Cabe sin embargo anotar que no es SUT la que se encarga de repartir los pulsos que ella forma. Simplemente SUT recibe alguna orden de SUD, forma los pulsos y los entrega otra vez a SUD para que sea ella la que se encargue de distribuirlos.

B. SUBUNIDAD DE DISTRIBUCION, SUD.

Realmente es imposible hablar de qué subbloques está compuesta SUD, porque todo se reduce a una cantidad inmensa de compuertas AND, OR, NAND, NOR, NOT con entradas convenientes. Se contó, una vez terminado el presente trabajo, que SUD está compuesto por 644 compuertas, tomando en cuenta las de dos, tres o cinco entradas sean AND, OR, etc. Todo su diseño se hace en base de ecuaciones. Por ejemplo: si se ha aplastado una tecla de dígito, después de aplastada la tecla x y hay un uno en FLM, se debe parar el reloj para no seguir con más sumas sucesivas. Esto equivale a decir: si hay el nivel DIG y hay el nivel POR y hay FLM se debe inhibir el reloj, o sea aplicar un pulso a la entrada de

inhibición del reloj. Escrito como ecuación:

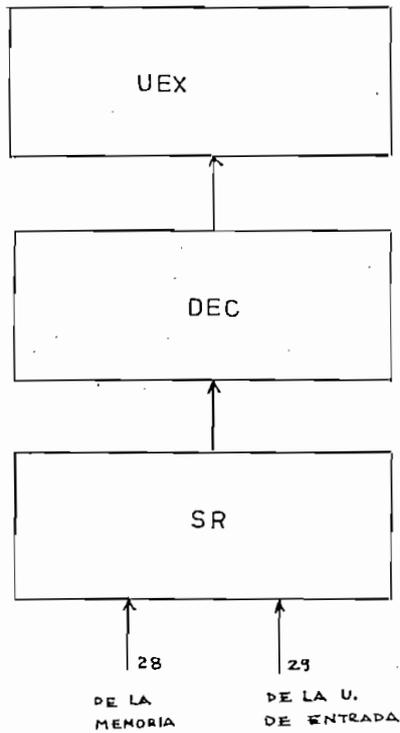
$$J_{INH} = \text{DIG. POR. FLM. } t_{17}$$

Entonces se debe tener una compuerta AND de 4 entradas para realizar lo indicado antes. Pero no sólo bajo estas condiciones se para el reloj sino también dependiendo de muchos otros factores. Luego habrá que usar compuertas OR para tenerlas en cuenta. Cuando de este tipo hay 500 o más ecuaciones se comprende lo arduo que hay que pensar y lo duro que hay que trabajar.

Pero la única finalidad es la siguiente: acatar órdenes de la unidad de entrada, analizar las condiciones existentes en la unidad aritmética y en la memoria. Recopilado esto, dar una orden a SUT para que ésta devuelva las secuencias adecuadas de pulsos que se repartirán a las distintas partes del calculador.

DIAGRAMA DE BLOQUES DE LA UNIDAD DE SALIDA.

FIG. 9



Nomenclatura:

- SR: selector de registros
- DEC: unidad decodificadora
- UEX: unidad de exhibición

Terminales:

- 28: Ponga información en SR desde la memoria
- 29: Ponga información en SR desde la unidad de entrada.

A. SELECTOR DE REGISTROS, SR.

Según cuál haya sido la última o las últimas teclas aplastadas (información que proviene desde la unidad de entrada, a través del terminal 29), y según cuál sea el contenido de la memoria (lo cual se hace conocer a SR mediante el terminal 28), esta unidad procede a escoger el registro cuyo resultado va a ser exhibido, ya sea RE, RA o RR, y también a escoger la forma en que el resultado debe ser exhibido, es decir si el contenido del registro escogido debe ser mostrado en forma normal o en forma de su complemento de nueve.

Así seleccionado el material, la información es pasada a la próxima unidad, DEC.

B. UNIDAD DECODIFICADORA, DEC.

Está compuesta por 12 decodificadores parciales iguales, correspondientes a los 12 dígitos a ser mostrados. Cada decodificador parcial tiene 4 entradas (correspondientes a los 4 bits en que se han codificado los dígitos) y 10 salidas, correspondientes a los dígitos 0, 1, 2,, 9.

C. UNIDAD DE EXHIBICION, UEX.

Compuesta asimismo de 12 subbloques iguales. Cada subbloque está formado por un juego de lámparas colocadas en un mismo bulbo, y tiene 10 terminales de entrada. Cuando se ha activado por ejemplo el terminal 5, se prende el co

rrespondiente juego de lámparas que forma un cinco. Sin embargo, cada entrada no es activada directamente desde el decodificador, pues los voltajes no serían suficientes para prender las lámparas y luego las corrientes tomadas constituirían excesiva carga para los decodificadores.

De ahí que se tenga necesidad de circuitos adicionales a transistores para, aprovechando sus características de corte o saturación, se active o no determinada entrada.

A P E N D I C E

particular. Esto equivale a decir que cambiamos de una parte a otra la definición de lógica "1" y lógica "0". Cuando se entiende de esta manera basta sólo un poco de experiencia para facilitar mucho el trabajo.

La presente convención se aplicará a todos los componentes de la serie MC350, excepción hecha de MC354 que no es una unidad lógica sino una unidad de polarización.

Es necesario tener un resistor de carga (2K) en todas las salidas de las compuertas. Si se hace compuertaje por emisor sólo se necesita un resistor de carga para toda la línea cuyos emisores se han unido.

Todos los elementos tienen un alto fan-out, alrededor de 15, consecuencia de la baja impedancia de salida de los seguidores de emisor. La única excepción es MC365 que tiene un mayor fan-out (alrededor de 25), pues es una unidad para manejar líneas de 50

Convenciones de voltajes y de polaridades:

Para el adecuado trabajo de los componentes MC350 se debe conectar:

el pin 3 al voltaje $V_{CC} = 0$ volts

el pin 2 al voltaje $V_{EE} = -5,2$ volts

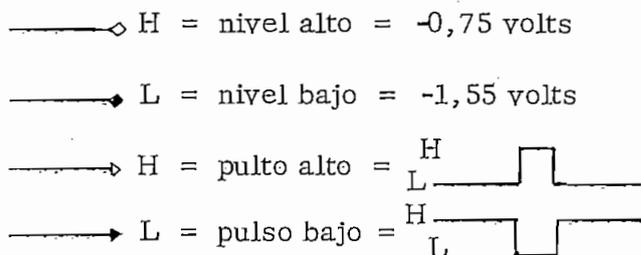
el pin 1* al voltaje $V_{BB} = -1,15$ volts

* Sólo para las unidades que necesitan esta polarización. Las excepciones son

352, 355, 358, 362A y 364.

Este voltaje V_{BB} generalmente es obtenido desde el pin 1 de MC354, la unidad de polarización que mantiene unas muy constantes características de voltaje DC sobre un amplio rango de temperatura 0 a +75°C.

Todas las entradas no usadas deben ser conectadas a V_{EE}



Para todas las compuertas, si se quiere realizar la operación lógica OR se debe tener inserciones altas a las entradas. Si se desea realizar la operación lógica AND se debe tener inserciones bajas a las entradas.

La misma regla es válida para el compuertaje por emisor.

Un círculo antes de una salida indica salida invertida (o cambiada de sexo).

Para OR se define alto como lógica 1 a la entrada

Para AND se define bajo como lógica 1 a la entrada.

Muchos de los elementos de la serie MC350 tienen dos salidas, de modo que realizan a la vez las operaciones NAND y AND, o las operaciones NOR y OR, pero

puesto que la distinción se hace con sexos, en el medio de cada compuerta se marcará con un solo signo para indicar la operación lógica realizada

- + para OR
- . para AND
- para NOT

Se indicará en el transcurso de este capítulo aquellas compuertas que tienen un resistor de carga dentro del paquete.

B. COMPUERTAS DE 3 ENTRADAS, 2 SALIDAS.

1. - MC356: Resistores de carga en ambos sexos de la salida
2. -MC357: No tiene resistores de carga en ningún sexo de la salida
3. - MC365: No tiene resistores de carga en ningún sexo de la salida

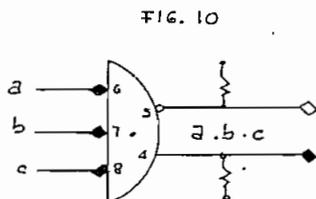
Cualquiera puede ser empleada para operación AND, OR, NOT o aislador.

Pueden usarse igualmente bien con pulsos o con niveles o combinado. Abajo se muestran algunos ejemplos y las correspondientes tablas de verdad.

OPERACION AND.

Define entradas bajas como lógica 1.
Se tiene ambos sexos de la salida.

MC356

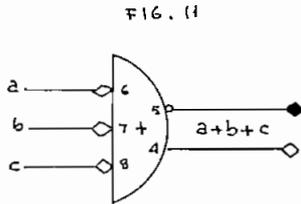


a	b	c	Pin 4	Pin 5
H	H	H	H	L
H	H	L	H	L
H	L	H	H	L
H	L	L	H	L
L	H	H	H	L
L	H	L	H	L
L	L	H	H	L
L	L	L	L	H

OPERACION OR.

Define entradas altas como lógica 1.
Se tiene ambos sexos de la salida.

MC357 o MC365

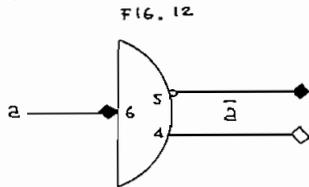


a	b	c	Pin 4	Pin 5
L	L	L	L	H
L	L	H	H	L
L	H	L	H	L
L	H	H	H	L
H	L	L	H	L
H	L	H	H	L
H	H	L	H	L
H	H	H	H	L

OPERACION NOT.

Defina como usted quiera. Trabaja sólo con niveles.
Ambos sexos de la salida.

MC365

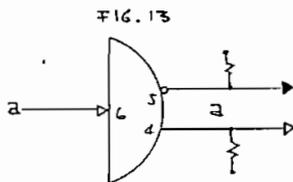


a	Pin 4	Pin 5
L	L	H
H	H	L

OPERACION AISLADOR

Defina como usted quiera. Esta no es una operación lógica. La salida es la misma que la entrada pero se tienen ambos sexos. Trabaja con pulsos o niveles.

MC356



a	Pin 4	Pin 5
L	L	H
H	H	L

C. PAQUETES DE DOS COMPUERTAS, 2 ENTRADAS POR COMPUERTA, UNA SALIDA POR COMPUERTA.

1. - MC359: Resistores de carga en ambos emisores de salida (pin 5 y pin 6)
2. - MC360: Resistor de carga en un solo emisor de salida (pin 6)
3. - MC361: No tiene resistores de carga en ninguno de los emisores.

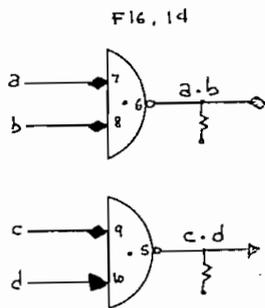
Cualquiera puede ser usada para operación AND, OR, NOT o aislador.

Pueden usarse pulsos o niveles. Abajo se muestran algunos ejemplos y sus tablas de verdad.

OPERACION AND.

Define entradas bajas y salidas altas como lógica 1.

MC359



a	b	Pin 6	c	d	Pin 5
H	H	L	H	H	L
H	L	L	H	L	L
L	H	L	L	H	L
L	L	H	L	L	H

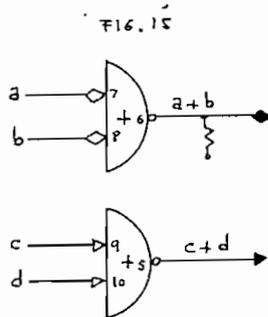
Si se juntan los emisores, la salida resultante será:

$$a \cdot b + c \cdot d$$

OPERACION OR.

Define entradas altas y salidas bajas como lógica 1.

MC360



a	b	Pin 6	c	d	Pin 5
L	L	H	L	L	H
L	H	L	L	H	L
H	L	L	H	L	L
H	H	L	H	H	L

Si se juntan los emisores el resultado será:

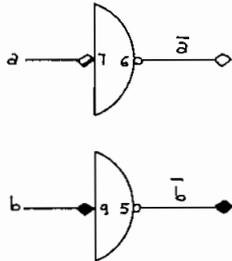
$$(a + b) \cdot (c + d)$$

OPERACION NOT

Defina como usted quiera. Trabaje sólo con niveles.

MC361

FIG. 16



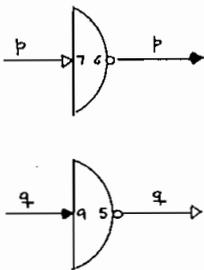
a	Pin 6
L	H
H	L

b	Pin 5
H	L
L	H

Observe lo que sucede cuando se trabaja con pulsos

MC361

FIG. 17



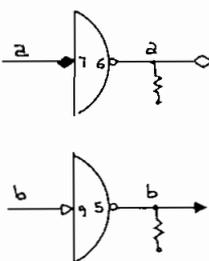
Entonces no está haciendo la operación NOT porque en lógica de pulsos, no pulso significa "no existe pulso" y lo que aquí conseguimos es un pulso pero de otro sexo.

OPERACION AISLADOR.

Defina como quiera. Sirve para cambiar de sexo.

MC359

FIG. 18



a	Pin 6
H	L
L	H

b	Pin 5
L	H
H	L

D. PAQUETE DE DOS COMPUERTAS, TRES ENTRADAS POR COMPUERTA, UNA SALIDA POR COMPUERTA.

MC362A: Resistores de carga en ambos emisores de salida. Debido a este hecho

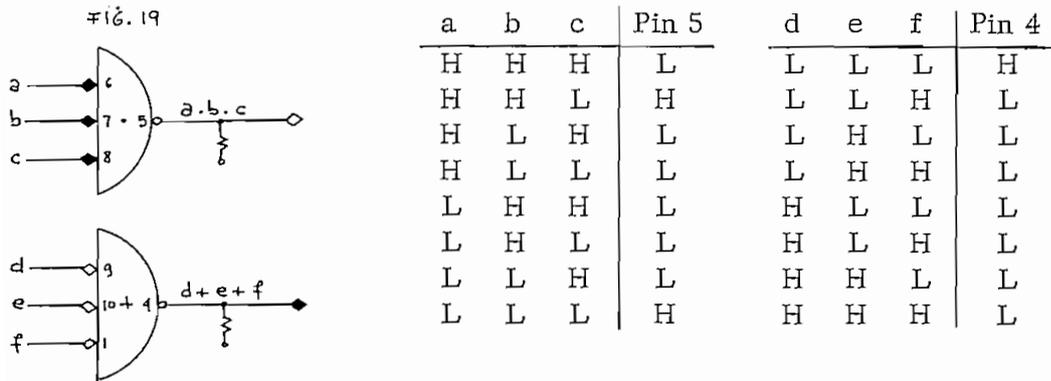
el compuertaje por emisor está severamente limitado con MC362A.

Funciona exactamente igual que MC359 pero cada compuerta tiene una entrada más.

Para operación AND define entradas bajas y salida alta como lógica 1.

Para operación OR define entradas altas y salidas bajas como lógica 1.

MC362A



E. COMPUERTA DE CINCO ENTRADAS, DOS SALIDAS.

MC351: Resistores de carga en ambos sexos de la salida.

Por esta razón el compuertaje por emisor está limitado, si se lo hace con otras MC351.

Funciona exactamente igual que MC356 pero tiene dos entradas más.

Para operación AND define entradas bajas como lógica 1

Para operación OR define entradas altas como lógica 1.

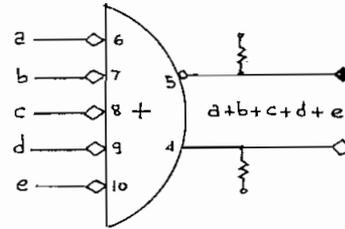
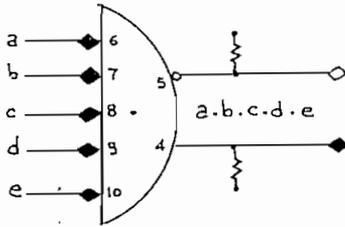
Se tiene ambos sexos de la salida.

No se adjunta la tabla de verdad correspondiente (por ser demasiado larga), pero se muestra gráficamente ambos tipos de conexión.

MC351

MC351

FIG. 20



F. MEDIO SUMADOR (MC353)

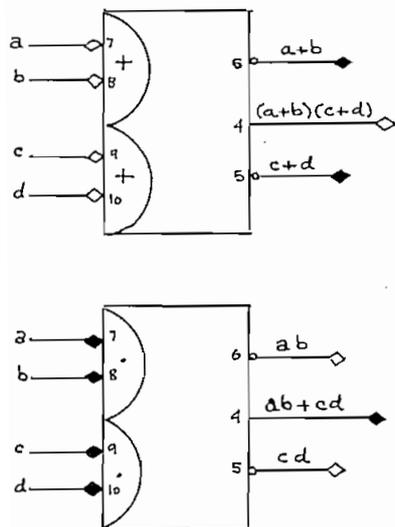
En realidad equivale a 3 compuertas de 2 entradas en un solo paquete. No necesariamente reduce el precio total pues es más caro que 3 compuertas de dos entradas. Realiza una función lógica que ocurre muy frecuentemente, de modo que se presta mucho para reducir el número total de paquetes necesarios para realizar una determinada función, cuando se usa adecuadamente. Puede ser considerado como un medio sumador (en el verdadero sentido) cuando se usa en conjunto con un flip flop J-K MC358.

(Ver tabla en página siguiente)

NOTA: Cuando las cuatro entradas son bajas se produce un estado espureo (SL = super bajo = - 2,3 volts). En el calculador bajo diseño nunca se produce este caso. Generalmente este caso suele considerarse sólo un nivel bajo

normal pero el tiempo de recuperación puede ser diferente del especificado.

FIG. 21



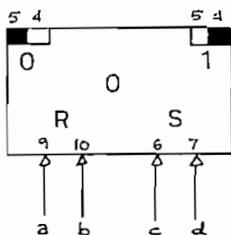
a	b	c	d	Pin 4	Pin 6	Pin 5
L	L	L	L	SL	H	H
L	L	L	H	L	H	L
L	L	H	L	L	H	L
L	L	H	H	L	H	L
L	H	L	L	L	L	H
L	H	L	H	H	L	L
L	H	H	L	H	L	L
L	H	H	H	H	L	L
H	L	L	L	L	L	H
H	L	L	H	H	L	L
H	L	H	L	H	L	L
H	L	H	H	H	L	L
H	H	L	L	L	L	H
H	H	L	H	H	L	L
H	H	H	L	H	L	L
H	H	H	H	H	L	L

G. FLIP FLOP R-S (Mc352)

D - C flip flop de set - reset. SENTIVO A NIVELES.

Define pulsos altos a las entradas como lógica 1. Se tiene ambos sexos de las dos salidas. Este tipo tiene dos entradas R y dos S que han sido internamente conectadas en forma OR, pero externamente aisladas.

FIG. 22



a	b	c	d	Q
0	0	0	0	q
0	0	-	1	1
0	0	1	-	1
1	-	0	0	0
-	1	0	0	0
1	-	1	-	N.D

o en otra forma

R	S	Q
0	0	q
0	1	1
1	0	0
1	1	N.D

Convenciones.

- = 0 o 1, cualquiera

N.D = no definido

q = estado inicial del flip flop

Q = estado final del flip flop

R = se refiere a cualquiera de las entradas 9 o 10

S = se refiere a cualquiera de las entradas 6 o 7

H. FLIP FLOP J-K (MC358 o MC364)

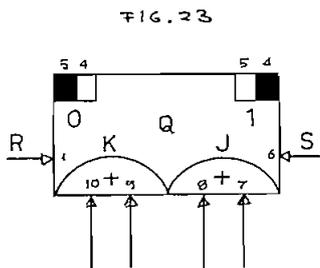
Operación Normal J - K: Define como lógica 1 una transición de bajo a alto a la entrada .

Se tiene ambos sexos de los niveles de las dos salidas.

K = se refiere a cualquiera de las entradas a o b

J = se refiere a cualquiera de las entradas c o d

Tiene adicionales entradas R -S para operación en esta forma.



Operación J -K

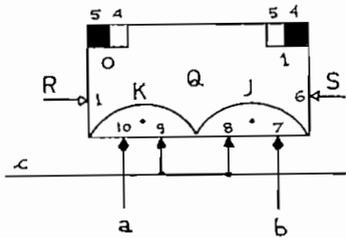
J	K	Q
0	0	q
0	1	0
1	0	\bar{q}
1	1	q

Operación R -S

R	S	Q
0	0	q
0	1	1
1	0	0
1	1	N.D.

Operación con Reloj: Defina como lógica 1 niveles bajos de entrada a a o b y un pulso (señal dinámica) bajo a la entrada c. Usado de esta manera provee un tipo de lógica que funciona en el filo posterior de los pulsos.

Fig. 24



b	a	c	Q
-	-	0	q
H	H	1	q
H	L	1	0
L	H	1	1
L	L	1	\bar{q}

A P E N D I C E II

DISEÑO DE CADA UNIDAD

1. DISEÑO DE LA UNIDAD DE ENTRADA

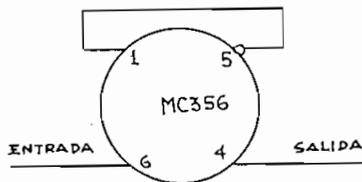
A. U. F. P.

Las teclas "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "+", "-", "x", "÷", "=", "R", "T", "L", en realidad son switches que producen una transición entre -5,2 volts y 0 volts cuando son aplastadas y la transición inversa cuando son depresionadas. Naturalmente estos no son los niveles lógicos de MECL, de ahí que lo primero sea transformarlos.

De esto se encarga un DISPARO SCHMITT. Este es básicamente un circuito regenerativo que cambia abruptamente de estado cuando la entrada cruza niveles de DC específicos llamados los niveles de disparo.

Usaremos un tipo de disparo Schmitt que puede ser fabricado conectando MC356 de una manera especial. Abajo se muestra el diagrama.

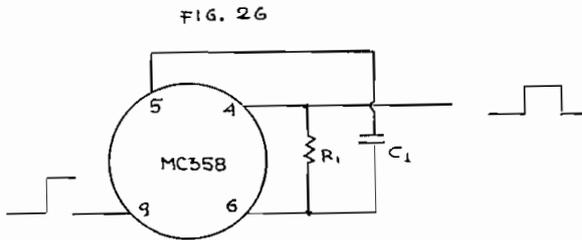
FIG. 25



Su construcción es posible gracias a los dos polos de la salida de MC356 y gracias a la existencia del pin 1 de polarización. Cuando se realimenta la salida del pin 5 al pin 1, el amplificador diferencial la compara con la de la entrada al pin 6 y propor

ciona a la salida el resultado diferencialmente comparado. Suele tener características muy deseables de inmunidad contra ruido. A veces es usado en la conversión de niveles DEC a MECL.

El próximo paso constituye "formar un pulso" aprovechando la transición anterior, de ahí que se requiera un MULTIVIBRADOR MONOESTABLE. La descripción de tal tipo de circuitos puede ser encontrada en cualquier libro de técnica de impulsos. Nos limitaremos a indicar el método de formar fácilmente un monoestable con un flip flop J K de MECL y adicionales elementos como son una resistencia y un condensador. Esto se puede encontrar en el manual de Motorola "Motorola Digital Integrated Circuits". De las curvas existentes en dicho manual



o con la fórmula que se da a continuación se puede calcular la duración del pulso.

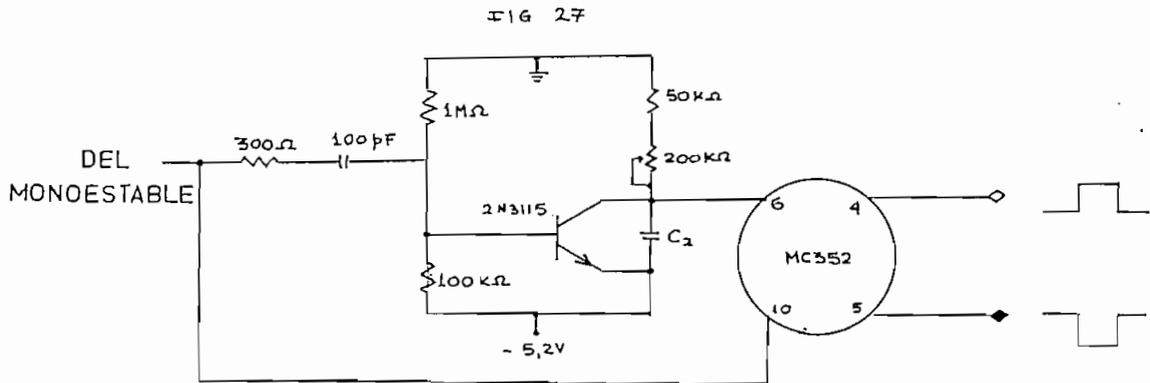
$$t_{ON} = 20 \text{ ns} + 1.4 R_1 (C_1 + 5 \text{ pF})$$

Por razones de estabilidad debe limitarse

R_1 a un valor menor que 175 K .

Con el fin de proteger la máquina contra personas que tiemblan al aplastar las teclas, o contra personas molestosas que juegan aplastando muchas veces la misma tecla, o inclusive contra teclas defectuosas que no hacen un solo contacto, se ha previsto que la máquina no reaccione a posteriores aplastadas de la tecla si no ha transcurrido 0,2 seg. Para el objeto se ha provisto de un RETARDO INTEGRANTE que acoplado con el monoestable anterior dar el resultado deseado. Su

configuración se muestra abajo.



Se requiere un pulso de determinada duración a la entrada pues por tener el 2N3115 un finito beta (alrededor de 100), toma un tiempo finito descarga C_2 . La duración del pulso de entrada determina el mayor tamaño de C_2 . De ahí que haya sido necesario formar un monoestable de 100 u seg para descargar C_2 .

El valor de C_2 puede ser obtenido de

$$T_{ret} = R_2 C_2 \ln \frac{5,2}{1,15}$$

Así se ha obtenido:

$$C_2 = 0,56 \mu F, \quad R_1 = 3 K, \quad C_1 = 0,022 \mu F$$

El diagrama completo de U. F. P. se muestra en el PLANO No. 1 al final de este trabajo.

Finalmente, para muchas aplicaciones solamente se requiere saber si ha sido aplastada una tecla de dígito, sin interesar específicamente cuál. De ahí

que se haya formado el pulso dig usando dos compuertas OR MC351 y compuerta-
je por emisor para tal cometido.

B. U. F. N.

Cada uno de los pulsos formados por U. F. P. se usan para poner en 0 o en 1 correspondientes flip flop RS MC352. No se ha procedido así, sin embargo, con los pulsos res, lim ni trans por no necesitarse esos niveles, pero se ha añá-
dido el nivel DIG producido por un flip flop R S a cuya entrada S va el pulso dig.

Qué sucedería sin embargo si se aplastara (ocurriría con un molesto) sucesivamente o a la vez las teclas +, -, x, +, e =, o cualquier combinación de ellas?

La máquina se comportaría de un modo extraño, no definido. De ahí que para dar alguna protección contra este hecho se use a las entradas de cada flip flop no el pulso directamente, sino compuertado con la negación de los otros niveles no aplastados. Una excepción es hecha con los casos IGUAL PARA e IGUAL POR que si suelen ocurrir, siempre que primero haya POR o PARA y luego el IGUAL.

En realidad esta parte no entraña mayor dificultad en su diseño. Su dia-
grama completo aparece en el PLANO No. 1.

Así pues, tenemos ya creados los siguientes pulsos y niveles:

cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, dig, mas, menos,
por, para, igual, lim, res, trans.

CERO, UNO, DOS, TRES, CUATRO, CINCO, SEIS, SIETE, OCHO, NUEVE, DIG,
MAS MENOS, POR PARA, IGUAL.

C. GOD, CCD.

Abajo se muestra una tabla que contiene los dígitos, sus correspondientes niveles, su código de Aiken y su complemento de diez.

Dígito	Nivel	Código Aiken				Complemento de diez			
		W	X	Y	Z	A	B	C	D
0	CERO	0	0	0	0	0	0	0	0
1	UNO	0	0	0	1	1	1	1	1
2	DOS	0	0	1	0	1	1	1	0
3	TRES	0	0	1	1	1	1	0	1
4	CUATRO	0	1	0	0	1	1	0	0
5	CINCO	1	0	1	1	1	0	1	1
6	SEIS	1	1	0	0	0	1	0	0
7	SIETE	1	1	0	1	0	0	1	1
8	OCHO	1	1	1	0	0	0	1	0
9	NUEVE	1	1	1	1	0	0	0	1

TABLA 12

De la TABLA 12 se deduce que:

$$W = \text{CINCO} + \text{SEIS} + \text{SIETE} + \text{OCHO} + \text{NUEVE}$$

$$X = \text{CUATRO} + \text{SEIS} + \text{SIETE} + \text{OCHO} + \text{NUEVE}$$

$$Y = \text{DOS} + \text{TRES} + \text{CINCO} + \text{OCHO} + \text{NUEVE}$$

$$Z = \text{UNO} + \text{TRES} + \text{CINCO} + \text{SIETE} + \text{NUEVE}$$

$$A = \text{UNO} + \text{DOS} + \text{TRES} + \text{CUATRO} + \text{CINCO}$$

$$B = \text{UNO} + \text{DOS} + \text{TRES} + \text{CUATRO} + \text{SEIS}$$

$$C = \text{UNO} + \text{DOS} + \text{CINCO} + \text{SIETE} + \text{OCHO}$$

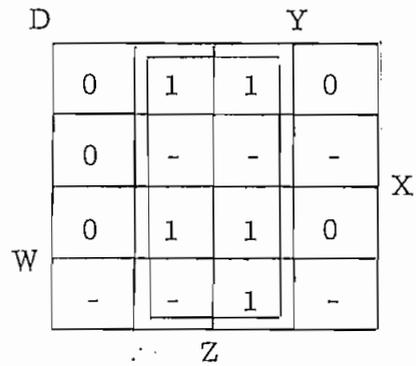
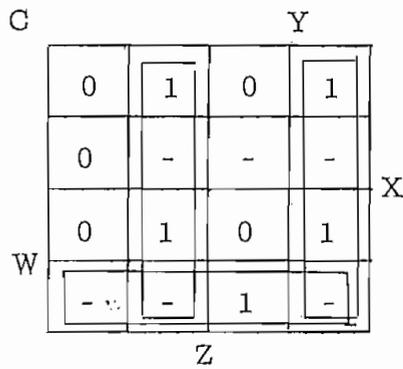
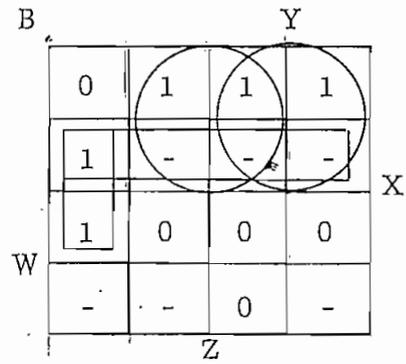
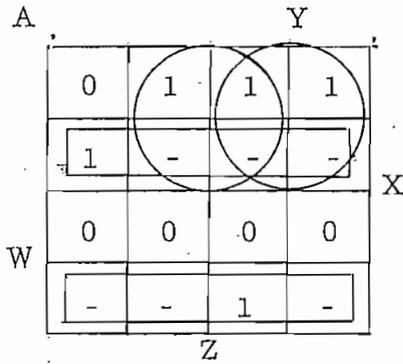
$$D = \text{UNO} + \text{TRES} + \text{CINCO} + \text{SIETE} + \text{NUEVE}$$

Observe de las ecuaciones anteriores que $Z = D$, esto nos permite una apreciable simplificación que es la única que podemos hacer. Directamente desde estas ecuaciones se obtienen los diagramas de COD y de CCD. Estos se muestran en el PLANO No. 2.

De ahora en adelante usaremos W X Y Z para el dígito codificado en forma de Aiken y A B C D para el complemento de 10. Vale la pena anotar que $\bar{W} \bar{X} \bar{Y} \bar{Z}$ será el complemento de 9.

Sin embargo existe otra posibilidad para el diseño de CCD que consiste en suponer ya la existencia de W X Y Z, o sea de COD y usar estas entradas en lugar de los niveles CERO, UNO,etc. para el diseño de CCD. Esto se presenta para una interesante comparación y de ahí que haya decidido mostrar esta otra forma de atacar al problema.

Los correspondientes mapas de Karnaugh para A, B, C y D en función de W, X, Y y Z son:



- = 0 o 1, cualquiera

De donde se obtiene:

$$A = W\bar{X} + \bar{W}X + \bar{W}Y + \bar{W}Z$$

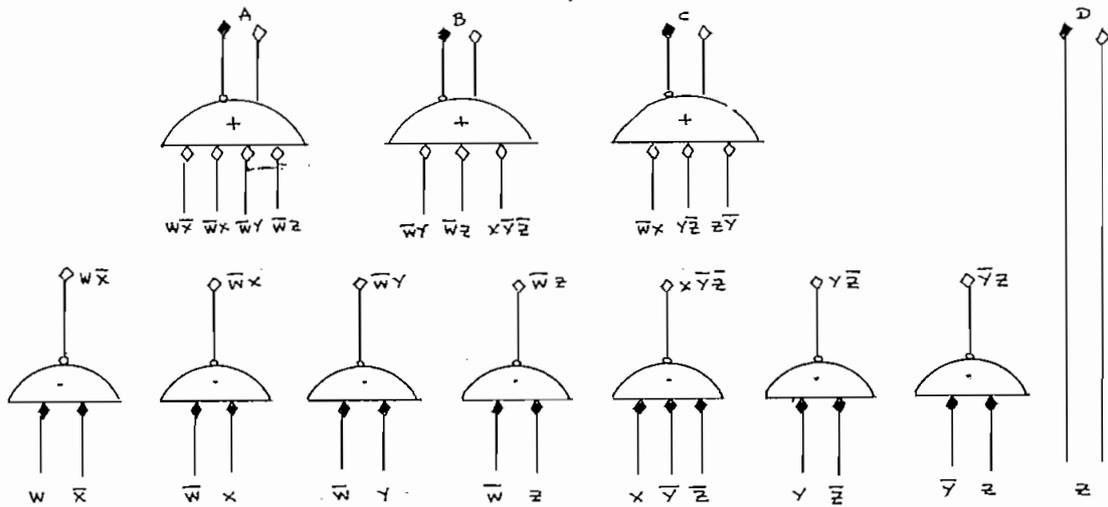
$$B = \bar{W}Y + \bar{W}Z + X\bar{Y}\bar{Z}$$

$$C = W\bar{X} + Y\bar{Z} + Z\bar{Y}$$

$$D = Z$$

Lo que da un diagrama como el mostrado enseguida:

FIG. 28



Obviamente este método es mucho más caro (10 compuertas, 23 entradas) que el anterior que sólo usaba 3 compuertas y 15 entradas. Sirva esto para demostrar cómo un diseño inteligente influye notablemente para hacer menor el costo.

2. DISEÑO DE LA UNIDAD ARITMETICA

A. CE, CA, CM, CD

Estos cuatro contadores se construyen de idéntica forma. Su diseño se muestra a continuación. Serán realizados con 4 flip flops J-K. La TABLA 13 muestra una columna en la cual se muestra el estado inicial de los 4 flip flops del contador. La columna próxima, situada a la derecha de la anterior, muestra el estado al que entran los flip flops después de que se ha dado un pulso al respectivo terminal que ordena incrementar en uno el contador. Las próximas columnas indican qué se debe poner a las entradas J y K de cada flip flop para que realicen

el cambio deseado (del estado inicial al estado nuevo).

A continuación se han dibujado los correspondientes mapas de Karnaugh (ocho en total) para cada entrada. De ahí se obtienen las ecuaciones simplificadas que caracterizan al contador.

TABLAS 13

W X Y Z	W X Y Z	$J_W^K_W$	$J_X^K_X$	$J_Y^K_Y$	$J_Z^K_Z$
0 0 0 0	0 0 0 0	0 -	0 -	0 -	0 -
0 0 0 1	0 0 1 0	0 -	0 -	1 -	- 1
0 0 1 0	0 0 1 1	0 -	0 -	- 0	1 -
0 0 1 1	0 1 0 0	0 -	1 -	- 1	- 1
0 1 0 0	1 0 1 1	1 -	- 1	1 -	1 -
1 0 1 1	1 1 0 0	- 0	1 -	- 1	- 1
1 1 0 0	1 1 0 1	- 0	- 0	0 -	1 -
1 1 0 1	1 1 1 0	- 0	- 0	1 -	- 1
1 1 1 0	1 1 1 1	- 0	- 0	- 0	1 -
1 1 1 1	0 0 0 0	- 1	- 1	- 1	- 1

J_W

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

J_X

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

$$J_Y$$

		Y		
	0	1	-	-
	1	-	-	-
W	0	1	-	-
	-	-	-	-
		Z		

$$J_Z$$

		Y		
	1	-	-	-
	1	-	-	-
W	1	-	-	1
	-	-	-	-
		Z		

$$K_W$$

		Y		
	-	-	-	-
	-	-	-	-
W	0	0	1	0
	-	-	0	-
		Z		

$$K_X$$

		Y		
	-	-	-	-
	1	-	-	-
W	0	0	1	0
	-	-	-	-
		Z		

$$K_Y$$

		Y		
	-	-	1	0
	-	-	-	-
W	-	-	1	0
	-	-	1	-
		Z		

$$K_Z$$

		Y		
	-	1	1	-
	-	-	-	-
W	-	1	1	-
	-	-	1	-
		Z		

$$J_W = \overline{W}X$$

$$J_X = YZ$$

$$J_Y = Z + \overline{W}X$$

$$J_Z = 1$$

$$K_W = XYZ$$

$$K_X = YZ + \overline{W}X$$

$$K_Y = Z$$

$$K_Z = 1$$

Podemos ahora aplicar ciertos criterios simplificantes, que dependen en mucho de la habilidad de cada uno, o de la intuición, pues no hay reglas generales que se puedan aplicar. Por ejemplo, observe que cuando X cambia de 1 a 0 (primera columna), cambia de estado W. Esto nos permite simplificar el circuito pues ahora sólo voy a aplicar a K_W y a J_W la salida baja de X. Cuando X va de 1 a 0 se produce una transición de L a H que aplicada a K_W y a J_W hace cambiar de estado, de acuerdo con la tabla de verdad del flip flops J -K.

Observe también en los mapas de Karnaugh de J_X y de K_X que podemos tomar también $J_X = JZ + \bar{W}X = K_X$. Esto puede ser implementado con un medio sumador MC353 y permite una reducción del número de paquetes.

Los diagramas completos de CE, CA, CM y CD se muestran en el PLANO No. 3. El objeto de las compuertas adicionales que se han incluido junto con CE y CA ha sido discutido ya antes.

B. FLA FLE, FLM.

Tanto FLA como FLE como FLM son flip flops de llevo. Ellos van de 0 a 1 cuando los contadores respectivos van de 9 a 0, o sea de 1111 a 0000, es decir cuando el flip flop de mayor valor significativo, por decirlo así, va de 1 a 0, entonces, puesto que vamos a usar flip flops del tipo J -K para FLE, FLA y FLM, por qué no aprovechar la transición de L a H que se produce cuando el biestable de mayor valor significativo va de 0 a 1, en la salida baja del estado 1 (pin 4), para usarla como entrada J del flip flop de llevo?.

La forma de conexión de FLE, FLA y FLM se muestra en el PLANO No. 3.

C. CAD.

Su diseño es acarreado de manera idéntica al de CE, CA, CM y CD, excepto que cuenta en otro código, el DCB. Debe contar solamente hasta 10, pues como dijimos antes, CAD es el encargado de contar el número de dígitos del cociente de una división.

TABLA 14

	W X Y Z	W X Y Z	$J_W K_W$	$J_X K_X$	$J_Y K_Y$	$J_Z K_Z$
0	0 0 0 0	0 0 0 1	0 -	0 -	0 -	1 -
1	0 0 0 1	0 0 1 0	0 -	0 -	1 -	- 1
2	0 0 1 0	0 0 1 1	0 -	0 -	- 0	1 -
3	0 0 1 1	0 1 0 0	0 -	1 -	- 1	- 1
4	0 1 0 0	0 1 0 1	0 -	0 0	0 -	1 -
5	0 1 0 1	0 1 1 0	0 -	- 0	1 -	- 1
6	0 1 1 0	0 1 1 1	0 -	- 0	- 0	1 -
7	0 1 1 1	1 0 0 0	1 -	- 1	- 1	- 1
8	1 0 0 0	1 0 0 1	- 0	0 -	0 -	1 -
9	1 0 0 1	1 0 1 0	- 0	0 -	1 -	- 1
10	1 0 1 0	- - - -	- -	- -	- -	- -

J_W		Y				X
		0	0	0	0	
W	0	0	1	0	Z	
	-	-	-	-		
	-	-	-	-		

J_X		Y				X
		0	0	1	0	
W	-	-	-	-	Z	
	-	-	-	-		
	0	0	-	-		

$$J_Y$$

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

$$J_Z$$

		Y		
1	-	-	1	
1	-	-	1	X
-	-	-	-	
W	1	-	-	
		Z		

$$K_X$$

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

$$K_W$$

		Y		
-	-	-	-	
-	-	-	-	X
-	-	-	-	
W	0	0	-	
		Z		

$$K_Y$$

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

$$K_Z$$

		Y		
-	1	1	-	
-	1	1	-	X
-	-	-	-	
W	-	1	-	
		Z		

$$J_W = XYZ \quad J_X = YZ \quad J_Y = Z \quad J_Z = 1$$

$$K_W = Y \quad K_X = YZ \quad K_Y = Z \quad K_Z = 1$$

o también $K_W = XYZ$. Observe que $J_W = K_W$, $J_X = K_X$, $J_Y = K_Y$ y $J_Z = K_Z$.

Esto nos hace rápidamente ver que:

Z cambia cuando viene un pulso

Y cambia cuando Z va de 1 a 0

X cambia cuando Y va de 1 a 0

W cambia cuando X va de 1 a 0.

Así como el circuito se simplifica mucho (como se puede ver en el PLANO No. 3) y no es necesaria ninguna compuerta adicional.

D. CDO.

También cuenta hasta 12 en DCB. La tabla 15 muestra el estado inicial y final, después de venir un pulso.

TABLA 15

	M N P Q	M N P Q	
0	0 0 0 0	0 0 0 1	Ahora ya no hay necesidad de dibujar los mapas de Karnaugh, pues inmediatamente se observa que:
1	0 0 0 1	0 0 1 0	
2	0 0 1 0	0 0 1 1	Q cambia cuando viene un pulso
3	0 0 1 1	0 1 0 0	P cambia cuando Q va de 1 a 0
4	0 1 0 0	0 1 0 1	N cambia cuando P va de 1 a 0
5	0 1 0 1	0 1 1 0	M cambia cuando N va de 1 a 0
6	0 1 1 0	0 1 1 1	De modo que CDO es exactamente igual a CAD.
7	0 1 1 1	1 0 0 0	
8	1 0 0 0	1 0 0 1	Su configuración se muestra en el PLANO No. 3.
9	1 0 0 1	1 0 1 0	
10	1 0 1 0	1 0 1 1	
11	1 0 1 1	1 1 0 0	
12	1 1 0 0	- - - -	

3. DISEÑO DE LA UNIDAD DE MEMORIA

A. RE, RA, RR.

Tanto RE como RA y RR tienen doce lugares para dígitos, que se comienzan a contar desde 1, de derecha a izquierda, es decir que la posición 1 corresponde al dígito de menor valor significativo y la posición 12 al de mayor valor significativo. Conviene señalar que las últimas posiciones, 9, 10, 11 y 12 generalmente no se usarán, pues rara vez se realizan operaciones con tantos dígitos.

Cada posición tiene lugar para 4 bits correspondientes a W, X, Y y Z en que puede codificarse un dígito. El diseño no tiene nada de particular:

RE se compone de 48 flip flops J-K MC358

RA igualmente de 48 flip flops J-K MC358

RR, a diferencia, de 48 flip flops R-S MC352

RE y RA son registros de desplazamiento, o mejor de rotación, en ambas direcciones, derecha e izquierda. RR sólo de almacenamiento.

Se ha empleado una nomenclatura especial que conviene anotarla. Usaremos ejemplos:

REW_{10} significa "el flip flop correspondiente al bit W de la posición 10 del registro RE"

RRX_2 significa "el flip flop correspondiente al bit X de la posición 2 del registro RR."

A su vez, cada biestable tiene 4 salidas (en realidad sólo 2, de los pines 4 y 5) correspondientes a los dos sexos de los dos estados del flip flop.

Por ejemplo $\overline{\text{RAZ}}_8$ significa "la salida del sexo bajo del estado 0 del flip flop que guarda el bit Z en la posición 8 de RA".

Las entradas se han marcado similarmente. Por ejemplo:

J_{REY}_5 significa "la entrada J del flip flop Y de la posición 5 de RE".

S_{RRX}_1 significa "la entrada S del flip flop X de la posición 1 de RR".

Los diagramas se muestran en el PLANO NO. 4.

B. URRE, URRRA.

Con respecto a la rotación debo aclarar que es rotación de "lugares" o de "posiciones", lo que quiere decir que cada bit sigue su línea; mejor dicho el bit W sólo puede pasar a ocupar el puesto W sea de la posición superior o inferior, dependiendo de la orden de rotar a la izquierda o a la derecha.

Ejemplo: con una orden "rote RE un lugar a la izquierda", el contenido de REW_6 es pasado a REW_7 ; el de REX_6 a REX_7 , el de REY_6 a REY_7 y el de REZ_6 a REZ_7 , ocurriendo algo similar con las otras posiciones.

En general, si definimos:

$(\text{RE}) \longleftarrow 1 = \text{izqe}$ (el pulso izqe)

$(\text{RE}) \longleftrightarrow 1 = \text{dere}$ (el pulso dere)

$(\text{RA}) \longleftarrow 1 = \text{izqa}$ (el pulso izqa)

$(\text{RA}) \longrightarrow 1 = \text{dera}$ (el pulso dera)

se debe cumplir que:

$$J_{\text{RE}i_j} = \text{RE}i_{j-1} \text{ izqe} + \text{RE}i_{j+1} \text{ dere}$$

$$K_{REi_j} = \overline{REi_{j-1}} \text{ izqe} + \overline{REi_{j+1}} \text{ dere}$$

$$J_{RAi_j} = RAi_{j-1} \text{ izqa} + RAi_{j+1} \text{ dera}$$

$$K_{RAi_j} = \overline{RAi_{j-1}} \text{ izqa} + \overline{RAi_{j+1}} \text{ dera}$$

en donde $i = W, X, Y, Z$

$j = 1, 2, \dots, 12$ y $0 = 12$ y $13 = 1$

Las dos primeras nos dan lo necesario para realizar URRE y las dos últimas para URRRA.

Cada ecuación, con un determinado valor de i y de j será implementada con un medio sumador MC352 (no con compuertas, con el fin de reducir el número de paquetes).

Así pues, reemplazando todos los valores de i y de j se deduce que URRE se compone de 96 medios sumadores y URRRA de otros tantos.

Los diagramas de URRE y URRRA se muestran en el PLANO No. 4 .

4. DISEÑO DE LA UNIDAD DE CONTROL.

Consta de tres partes bien definidas. Comenzaré escribiendo una especie de programa que debe realizar la máquina cuando se apresta cada tecla. Cada parte irá acompañada de su diagrama de tiempo, que indica el número de pulsos y su secuencia. En esta primera parte será de mucha utilidad el capítulo "ANÁLISIS DE LAS OPERACIONES". El segundo tema será traducir el programa en ecuaciones que indiquen qué pulso o nivel debe ir a cada uno de los terminales de la máquina, en forma de álgebra de Boole. Finalmente se reunirán y reducirán las ecuaciones, y, a partir de ellas, se implementará el circuito final. Conviene anotar que con el fin

de no repetir muchas veces los mismos títulos, se conservará durante todo ese capítulo las mismas letras y los mismos numerales.

A. AL APLASTAR UNA TECLA DE DIGITO AL COMENZAR EL TRABAJO O DESPUES DE APLASTADA LA TECLA "+".

1. - Active el reloj encargado de dar los pulsos y prepare circuitos para una secuencia adecuada de pulsos.
2. - Inserte el dígito asociado en RE₁₂
3. - (RE) ← 1 : pulso al terminal 18.
4. - Limpie los flip flops CERO, UNO,, NUEVE.
5. - Limpie el flip flop DIG y pare (inhiba reloj)

Abajo se muestra el correspondiente diagrama de tiempo.

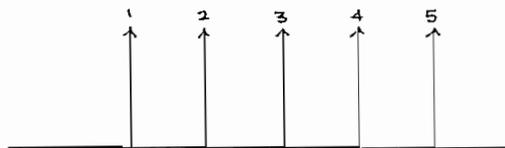


B. AL APLASTAR UNA TECLA DE DIGITO DESPUES DE APLASTADA LA TECLA "-".

1. - Active el reloj encargado de dar los pulsos y prepare circuitos para una secuencia adecuada de pulsos.
2. - Inserte el complemento de nueve del dígito asociado en RE₁₂
3. - (RE) ← 1: pulso al terminal 18
4. - Limpie los flipflops CERO, UNO,, NUEVE.
5. - Limpie el flip flop DIG y pare (inhiba reloj)

Abajo el correspondiente diagrama de tiempo.

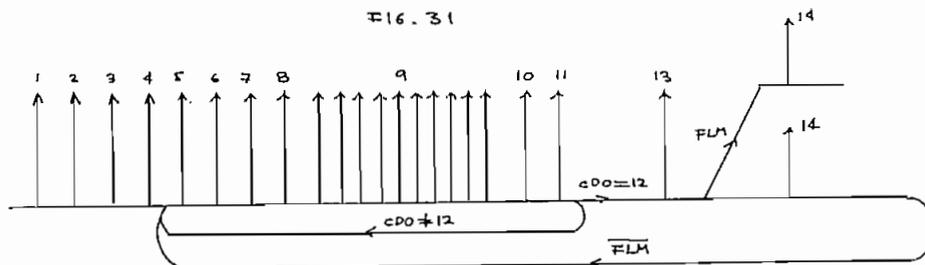
Fig. 30



C. AL APLASTAR UNA TECLA DE DIGITO DESPUES DE APLASTADA LA TECLA "x"

1. - Active el reloj y prepare circuitos para una secuencia adecuada de pulsos.
Limpie FLM: pulso al terminal 4.
2. - Ponga el dígito asociado, en forma de complemento de 10, en CM: pulso a los terminales 14.
3. - (RA) \longleftarrow 1 : pulso al terminal 20.
4. - Limpie CDO: pulso al terminal 7. Limpie CERO, UNO,, NUEVE.
5. - (RE)₁ \longrightarrow CE, (RA)₁ \longrightarrow CA.
6. - (RE) \longrightarrow ¹ : pulso al terminal 19. (RA) \longrightarrow 1 : pulso al terminal 21.
7. - Ponga llevo en CE si lo hay: pulso a 16.
8. - Ponga FLA en cero: pulso a 3.
9. - (CE) + (CA) \longrightarrow CA: diez pulsos a 17.
10. - (CA) \longrightarrow RA₁₂.
11. - Ponga FLE en cero: pulso a 2
Incremento CDO en uno: pulso a 11.
12. - Vuelva a 5 si CDO no marca 12. Siga si CDO marcha 12.
13. - Incremento CM en uno: pulso a 8

14. - Limpie CDO y vuelva a 5 si $\overline{\text{FLM}}$. Limpie DIG y pare si FLM.



D. AL APLASTAR UNA TECLA DE DIGITO DESPUES DE APLASTADA LA TECLA " + " .

Exactamente igual que B.

E. AL APLASTAR LA TECLA " + " .

1. - Active el reloj y prepare circuitos para una secuencia adecuada de pulsos.

Limpie CDO: pulso a 7.

2. - $(\text{RE})_1 \longrightarrow \text{CE}$, $(\text{RA})_1 \longrightarrow \text{CA}$.

3. - $(\text{RE}) \longrightarrow 1$: pulso a 19. $(\text{RA}) \longrightarrow 1$: pulso a 21.

4. - Ponga llevo en CE si lo hay: pulso a 16

5. - Ponga FLA en cero: pulso a 3

6. - $(\text{CA}) + (\text{CE}) \longrightarrow \text{CA}$: diez pulsos a 17.

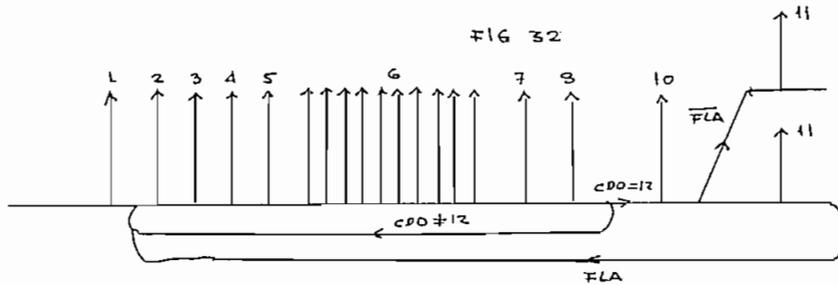
7. - $(\text{CA}) \longrightarrow \text{RA}_{12}$.

8. - Ponga FLE en cero: pulso a 2. Incremente CDO: pulso a 11.

9. - Vuelva a 2 si CDO no marca 12. Siga si CDO marca 12.

10. - Limpie RE.

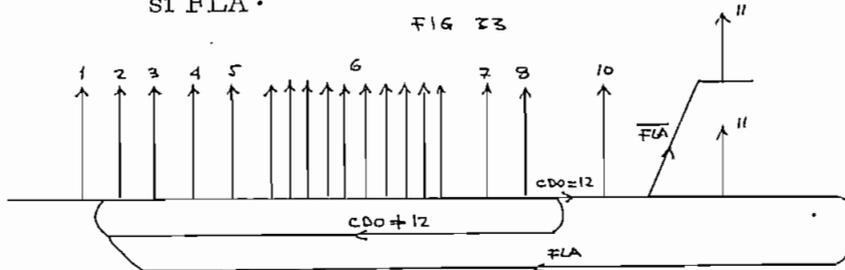
11. - Limpie CDO y vuelva a 2 si FLA. Pare si $\overline{\text{FLA}}$.



F. AL APLASTAR LA TECLA "-" .

1 a 10. - Exactamente como los pasos 1 a 10 de E.

11. - Limpie CDO (pulso a 7) y vuelva a 2 si FLA. Ponga 9's en RE y pare si FLA.



G. AL APLASTAR LA TECLA "x".

1 a 6. - Exactamente como los pasos 1 a 6 de E.

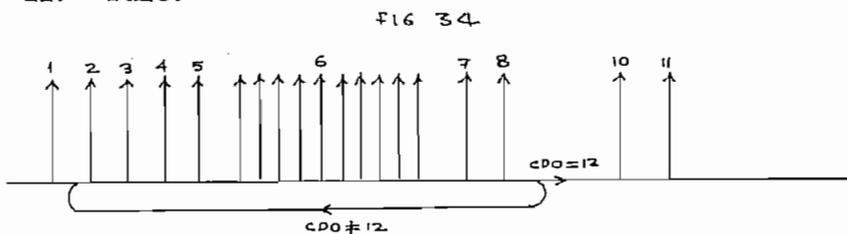
7. - (CA) \longrightarrow RE₁₂

8. - Ponga FLE en cero: pulso a 2. Incremente CDO: pulso a 11.

9. - Vuelva a 2 si CDO no marca 12. Siga si CDO marca 12.

10. - Limpie RA.

11. - Pare.



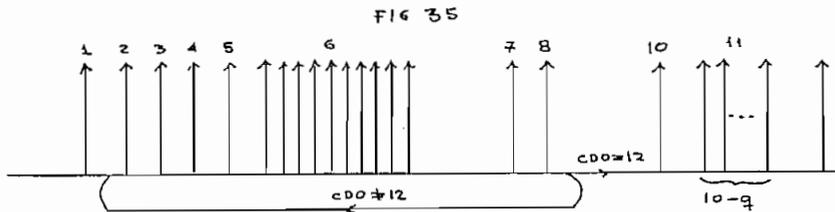
H. AL APLASTAR LA TECLA "÷".

1 a 9. - Como 1 a 9 de E.

10. - Ponga 9's en RE.

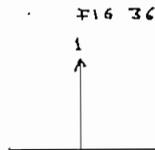
11. - (RA) ← 10 - q y pare: (10 - q) pulsos a 20 y un pulso pare.

q = número de dígitos del dividendo.



I. AL APLASTAR LA TECLA "=" DESPUES DE APLASTADA LA TECLA "x".

1. - Limpie RE.. Limpie POR.



J. AL APLASTAR LA TECLA "=" DESPUES DE APLASTADA LA TECLA "÷".

1. - Active el reloj encargado de dar los pulsos, prepare circuitos para una secuencia adecuada de pulsos y, limpie CAD: pulso a 6.

2. - (RE) ← 10 - r : (10 - r) pulsos al terminal 18.

r = número de dígitos del divisor.

3. - Limpie CD: pulso a 5.

4. - (RA) → RR.

5. - Limpie CDO: pulso a 7.

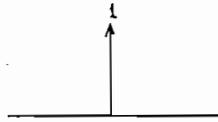
6. - (RE)₁ → CE, (RA)₁ → CA.

7. - (RE) \longrightarrow 1 : pulso a 19. (RA) \longrightarrow 1: pulso a 21.
 8. - Ponga llevo en CE si lo hay: pulso a 16.
 9. - Ponga FLA en cero: pulso a 3.
 10. - (CA) + (CE) \longrightarrow CA: 10 pulsos a 17.
 11. - (CA) \longrightarrow RA₁₂.
 12. - Ponga FLE en cero: pulso a 2. Incremente CDO: pulso a 11.
 13. - Vuelva a 6 si CDO no marca 12. Siga si CDO marca 12.
 14. - Siga con 15 si FLA. Vaya a 27 si \overline{FLA}
 15. - Incremente CD en uno: pulso a 9.
 16. - Limpie CDO: pulso a 7
 17. - Limpie CE y (RA) \longrightarrow CA.
 18. - (RE) \longrightarrow 1 : Pulso a 19. (RA) \longrightarrow 1: pulso a 21.
 19. - Ponga llevo en CE si lo hay: pulso a 16.
 20. - Ponga FLA en cero: pulso a 3.
 21. - (CA) + (CE) \longrightarrow CA: diez pulsos a 17.
 22. - (CA) \longrightarrow RA₁₂
 23. - Ponga FLE en cero: pulso a 2. Incremente CDO: pulso a 11.
 24. - Vuelva a 17 si CDO no marca 12. Siga si CDO marca 12.
 25. - (RA) \longrightarrow RR.
 26. - Limpie CDO: pulso a 7, y, vuelva a 6.
 27. - (RR) \longrightarrow RA.
 28. - Prepare otra secuencia de pulsos.
-

L. AL APLASTAR LA TECLA "T".

1. - (RA) \longrightarrow RR

FIG. 38



M. AL APLASTAR LA TECLA "L".

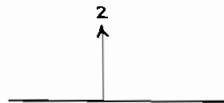
1. - Limpie RE

2. - Limpie RA

Limpie RR

Ponga CDO en 12: pulso a 15.

FIG. 39



DISEÑO DE SUT.

De los diagramas de tiempo se deduce que se repiten a menudo ciertas secuencias de pulsos. Por ejemplo para poner información dentro de la máquina, es decir cuando aplasto una tecla de dígito, generalmente se necesita una secuencia de 5 pulsos. El primero es el que producimos al aplastar la tecla; nos falta producir los 4 restantes a los que llamaremos tt_0, tt_1, tt_2, tt_3 . En la suma, por ejemplo, se requiere una secuencia de 16 pulsos, que se repetirán 12 veces. 10 de esos pulsos deben salir por un mismo terminal (son los que aplicados al terminal 17, ordenan $(CE) + (CA) \longrightarrow CA$). A esos pulsos los llamaremos $t_0, t_1, t_2, t_3, t_4 - t_{13}, t_{14}, t_{15}$.

Al aplastar la tecla "x", también se necesita la secuencia anterior, pero al final se requiere 2 pulsos adicionales. Los llamaremos t_{16} , t_{17} . En el caso de aplastar "=" después de haber aplastado ":" se requiere otra secuencia de 4 pulsos: t_{18} , t_{19} , t_{20} , t_{21} .

De ahí que SUT esté formada por un reloj, con entradas empiece y pare, 4 contadores, uno para cada secuencia, compuertas para obtener los pulsos por terminales separados y algunos circuitos adicionales que imponen las condiciones bajo las cuales debe salir una u otra secuencia.

Con el fin de hacer más claro el asunto en cuestión, voy a usar el circuito diseñado (PLANO No. 5) y con él puedo explicar la forma de trabajo. Obvio es suponer que en la realidad, procedí para el diseño de manera inversa.

El reloj, que es lo primero que debo tomar en cuenta, se forma mediante un biestable MC352, R-S, un paquete de dos compuertas MC359 y adicionales elementos híbridos que son 2 resistencias y 2 condensadores. Este tipo de circuito fue desarrollado por Thomas Chaney en la Universidad de Washington.

El circuito se muestra en el PLANO No. 5. Cada resistencia conecta una salida con la entrada opuesta y cada condensador la salida de un lado con la entrada del mismo lado. En el un lado se intercepta este método de conexión con el paquete MC359 que tiene el único fin de proveer una entrada para INHIBIR el reloj. Cuando el condensador conectado al lado alto "on" se ha descargado, lo suficiente, este lado del flip flop no puede quedar más en alto y a su vez el otro lado está libre para ir a alto mediante la acción del resistor conectado al lado "on". El nivel crítico al

que los cambios ocurren es cuando el voltaje en el condensador pasa por $-1,15$ volts. Se puede deducir que aproximadamente el período está dado por $T = T_1 + T_2 = 1,1 (R_1 C_1 + R_2 C_2)$

Observe que en los monoestables teníamos un $t_{ON} = 1,4 RC$. La razón de la reducción de $1,4 RC$ a $1,1 RC$ es porque el voltaje a través de los capacitores no tiene su valor final al tiempo de conmutación de este circuito (no existe plena recuperación).

Sin problemas de ruido, el pulso de ancho mínimo que se puede ejecutar en esta forma es de alrededor de 50 nseg, pero para actuar con seguridad he escogido un ancho del pulso de 150nseg. El período de los pulsos tiene como factor limitante el camino más largo que debe atravesar un pulso en el peor de los casos. Esto es debido a que cada compuerta o cada elemento en general produce un retardo en la transmisión. De ahí que en realidad el diseño de estos valores se deba hacerlo una vez terminado el trabajo. Se observó que el camino más largo era no mayor de 6 elementos. El retardo promedio siendo de 15 n seg, se tiene un retardo total de 90 nseg, que en este caso es menor aún que el ancho del pulso, de modo que este no constituye factor limitante para nosotros. Sin embargo conviene tener el pulso suficientemente espaciado del anterior, de modo que para el presente trabajo he escogido una relación 10/1 entre el período y el ancho del pulso.

Calculados los parámetros con estos valores se obtuvo:

$$R_1 = 3K \quad C_1 = 47 \text{ pF}$$

$$R_2 = 3K \quad C_2 = 470 \text{ pF}$$

Se ha dejado algo de flexibilidad al intercalar un potenciómetro de 5 K .

A la salida del reloj, y con el propósito de obtener una onda más limpia, se ha colocado un monoestable con un $t_{ON} = 150$ n seg. El pulso producido a la salida lo llamaremos p .

Los biestables E F G H forman un contador en DCB que cuenta hasta 15. Su diseño es absolutamente similar al de CAD o CDO.

\mathcal{U} y \mathcal{K} forman otro contador en DCB, de cero a tres. \mathcal{C} es el biestable de llevo de este contador. \mathcal{T} y \mathcal{M} forman otro contador de 0 a 3 en DCB y \mathcal{L} es su flip flop de llevo. V forma otro contador de 0 a 1. T y U son biestables que imponen condiciones. INH es otro biestable que cuando está en estado 1 para al reloj y cuando está en cero lo deja funcionar. El pulso p incrementa a todos los contadores, pero uno solo a un tiempo y dependiendo de las condiciones, y al mismo tiempo p es compuertado con las salidas de dichos contadores.

Expondremos algunos ejemplos:

- a) Supongamos que \mathcal{C} está en cero, T en cero, \mathcal{L} en uno, que CDO marca doce (lo que equivale a decir $M = 1$ $N = 1$), que DIG está en 1 y damos un pulso que pone INH en cero.

Bajo esas condiciones: el contador E F G H no trabaja porque \overline{MN} es cero; V no trabaja porque $\mathcal{C} = 0$; \mathcal{T} \mathcal{M} no trabaja porque $\mathcal{L} = 1$; pero \mathcal{U} \mathcal{K} si trabaja porque $DIG = 1$ $\mathcal{C} = 0$ $T = 0$ $MN = 1$.

Entonces, al venir el primer pulso obtendremos tt_0 porque $\mathcal{U} = 0$ $\mathcal{K} = 0$, y a su vez en el filo posterior de p cambiará el contador \mathcal{U} \mathcal{K} a 01. El próximo pulso pasará por tt_1 y pondrá \mathcal{U} \mathcal{K} en 10. Un siguiente p pasará por

t_2 y pondrá w_k en 11. Un último p pasará por t_3 y pondrá w_k en 00, y ζ en 1, con lo cual adicionales pulsos no pasarán.

b) Similarmente asoma:

$$\zeta = 1 \quad M = 0$$

$$T = 0 \quad N = 0$$

$$\Pi = 1$$

El contador w_k no trabaja porque $\zeta = 1$

El contador V no trabaja porque $MN = 0$

El contador Π no trabaja porque $\Pi = 1$

Sólo trabaja el contador E F G H. Así pues obtendremos una secuencia

$t_0 \dots t_{15}$. Se ha arreglado las compuertas de modo que $t_4 - t_{13}$ salgan por un solo terminal. Esta secuencia seguirá repitiéndose hasta que CDO marque 12 ($M = 1 \quad N = 1$) o se pare antes el reloj.

Definidos en esta forma todos los pulsos de que disponemos, el próximo paso es

DISEÑO DE SUD.

Comenzaremos traduciendo todo el programa anterior en forma de ecuaciones Booleanas. El próximo paso es escribir las ecuaciones reunidas, y simplificarlas para optimizar el costo. Como se acotó anteriormente usaré las mismas letras y numerales que en el programa.

Durante todo el resto del trabajo regirán los siguientes símbolos:

$i = W, X, Y, Z$

$j = 1, 2, 3, \dots, 12$ y $0 = 12$ y $1 = 13$

$m = \text{CERO, UNO, } \dots, \text{NUEVE}$

$d = M, N, P, Q$

$n = A, B, C, D$ y $(ABCD) = (WXYZ)''$

Ejemplos:

K_{RAi_j} significa las entradas K de los flip flops W X Y Z de las posiciones 1, 2, 3 hasta 12 del registro RA.

R_{CMn} significa las entradas R de los flip flops A B C D de CM.

- A.
1. - $K_{INH} = \text{dig}$ $K_T = \text{dig}$ $S_N = \text{dig}$
 $K_{\mathcal{U}} = \text{dig}$ $S_M = \text{dig}$ $S_{\mathcal{L}} = \text{dig}$
 2. - $R_{REi_{12}} = i \text{ tt}_0 \overline{\text{MENOS POR PARA}}$
 $S_{REi_{12}} = i \text{ tt}_0 \overline{\text{MENOS POR PARA}}$
 3. - $K_{REi_j} = REi_{j-1} \text{ tt}_1 \overline{\text{MENOS POR PARA}}$
 $J_{REi_j} = REi_{j-1} \text{ tt}_1 \overline{\text{MENOS POR PARA}}$
 4. - $K_m = \text{tt}_2 \overline{\text{MENOS POR PARA}}$
 5. - $K_{DIG} = \text{tt}_3 \overline{\text{MENOS POR PARA}}$
 $J_{INH} = \text{tt}_3 \overline{\text{MENOS POR PARA}}$
- B.
1. - $K_{INH} = \text{dig}$ $K_T = \text{dig}$ $S_N = \text{dig}$
 $K_{\mathcal{U}} = \text{dig}$ $S_M = \text{dig}$ $S_{\mathcal{L}} = \text{dig}$
 2. - $R_{REi_{12}} = i \text{ tt}_0 \text{ MENOS}$
 $S_{REi_{12}} = \bar{i} \text{ tt}_0 \text{ MENOS}$

$$3. - K_{REi_j} = \overline{REi_{j-1}} \text{ tt}_1 \text{ MENOS}$$

$$J_{REi_j} = REi_{j-1} \text{ tt}_1 \text{ MENOS}$$

$$4. - K_m = \text{tt}_2 \text{ MENOS}$$

$$5. - K_{DIG} = \text{tt}_3 \text{ MENOS}$$

$$J_{INH} = \text{tt}_3 \text{ MENOS}$$

$$C. \quad 1. - K_{INH} = \text{dig} \quad K_T = \text{dig} \quad S_N = \text{dig} \quad K_{FLM} = \text{dig}$$

$$K_v = \text{dig} \quad S_M = \text{dig} \quad S_u = \text{dig}$$

$$2. - R_{CMn} = \bar{n} \text{ tt}_0 \text{ POR}$$

$$S_{CMn} = n \text{ tt}_0 \text{ POR}$$

$$3. - K_{RAi_j} = \overline{RAi_{j-1}} \text{ tt}_1 \text{ POR}$$

$$J_{RAi_j} = RAi_{j-1} \text{ tt}_1 \text{ POR}$$

$$4. - R_d = \text{tt}_2 \text{ POR} \quad K_m = \text{tt}_2 \text{ POR}$$

$$5. - R_{CEi} = \overline{REi_1} \text{ t}_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CEi} = REi_1 \text{ t}_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$R_{CAi} = \overline{RAi_1} \text{ t}_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CAi} = RAi_1 \text{ t}_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$6. - K_{REi_j} = REi_{j+1} \text{ t}_1 \quad K_{RAi_j} = RAi_{j+1} \text{ t}_1$$

$$J_{REi_j} = REi_{j+1} \text{ t}_1 \quad J_{RAi_j} = RAi_{j+1} \text{ t}_1$$

$$7. - 16 = t_2$$

$$8. - K_{FLA} = t_3$$

$$9. - 17 = t_4 - t_{13}$$

$$10. - R_{RAi_{12}} = \overline{CAi} \text{ t}_{14} \text{ POR DIG}$$

$$S_{RAi_{12}} = GAi_{t_{14}} \text{ POR DIG}$$

$$11. K_{FLE} = t_{15}$$

$$K_Q = J_Q = t_{15}$$

$$12. K_H = J_H = p \overline{M N}$$

$$13. K_{CMD} = J_{CMD} = t_{16} \text{ POR DIG}$$

$$14. R_d = t_{17} \overline{FLM} \text{ POR DIG}$$

$$K_{DIG} = t_{17} \text{ FLM POR}$$

$$J_{INH} = t_{17} \text{ FLM POR DIG}$$

$$D. \quad 1. K_{INH} = \text{dig} \quad K_T = \text{dig} \quad S_N = \text{dig}$$

$$K_{\alpha} = \text{dig} \quad S_M = \text{dig} \quad S_{\mu} = \text{dig}$$

$$2. R_{REi_{12}} = tt_0 \text{ PARA } \overline{IGUAL}$$

$$S_{REi_{12}} = \bar{i} \text{ } tt_0 \text{ PARA } \overline{IGUAL}$$

$$3. K_{REi_j} = \overline{REi_{j-1}} \text{ } tt_1 \text{ PARA } \overline{IGUAL}$$

$$J_{REi_j} = REi_{j-1} \text{ } tt_1 \text{ PARA } \overline{IGUAL}$$

$$4. K_m = tt_2 \text{ PARA } \overline{IGUAL}$$

$$5. K_{DIG} = tt_3 \text{ PARA } \overline{IGUAL}$$

$$J_{INH} = tt_3 \text{ PARA } \overline{IGUAL}$$

$$E. \quad 1. K_{INH} = \text{mas} \quad K_T = \text{mas} \quad S_{\mu} = \text{mas}$$

$$J_{\alpha} = \text{mas} \quad R_d = \text{mas} \quad K_{MENOS} = \text{mas}$$

$$2. R_{CEi} = REi_1 \text{ } t_0 \text{ } (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CEi} = REi_1 \text{ } t_0 \text{ } (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$R_{CAi} = \overline{RAi}_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CAi} = RAi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$3. K_{REi_j} = \overline{REi}_{j+1} t_1$$

$$J_{REi_j} = REi_{j+1} t_1$$

$$K_{RAi_j} = \overline{RAi}_{j+1} t_1$$

$$J_{RAi_j} = RAi_{j+1} t_1$$

$$4. 16 = t_2$$

$$5. K_{FLA} = t_3$$

$$6. 17 = t_4 - t_{13}$$

$$7. R_{RAi_{12}} = \overline{CAi} t_{14} \text{ MAS}$$

$$S_{RAi_{12}} = CAi t_{14} \text{ MAS}$$

$$8. K_{FLE} = t_{15}$$

$$K_Q = J_Q = t_{15}$$

$$9. K_H = J_H = p \overline{MN}$$

$$10. R_{REi_j} = t_{16} \text{ MAS}$$

$$11. R_\alpha = t_{17} \text{ MAS FLA}$$

$$J_{INH} = t_{17} \text{ MAS } \overline{FLA}$$

F. 1. $K_{INH} = \text{menos}$ $K_T = \text{menos}$ $S_{\mu} = \text{menos}$

$J_\alpha = \text{menos}$ $R_a = \text{menos}$ $K_{MAS} = \text{menos}$

$$2. R_{CEi} = \overline{REi}_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CEi} = REi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$R_{CAi} = \overline{RAi}_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CAi} = RAi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$3. K_{REi_j} = \overline{REi_{j+1}} t_1$$

$$J_{REi_j} = REi_{j+1} t_1$$

$$K_{RAi_j} = \overline{RAi_{j+1}} t_1$$

$$J_{RAi_j} = RAi_{j+1} t_1$$

$$4. 16 = t_2$$

$$5. K_{FLA} = t_3$$

$$6. 17 = t_4 - t_{13}$$

$$7. R_{RAi_{12}} = \overline{CAi} t_{14} \text{ MENOS}$$

$$S_{RAi_{12}} = CAi t_{14} \text{ MENOS}$$

$$8. K_{FLE} = t_{15}$$

$$K_Q = J_Q = t_{15}$$

$$9. K_H = J_H = p \overline{MN}$$

$$10. R_{REi_j} = t_{16} \text{ MENOS}$$

$$11. R_{\alpha} = t_{17} \text{ MENOS FLA}$$

$$S_{REi_j} = t_{17} \text{ MENOS FLA}$$

$$J_{INH} = t_{17} \text{ MENOS FLA}$$

$$G. 1. K_{INH} = \text{por} \quad K_T = \text{por} \quad S_{\mu} = \text{por} \quad K_{MENOS} = \text{por}$$

$$J_{\nu} = \text{por} \quad R_{\alpha} = \text{por} \quad K_{MAS} = \text{por}$$

$$2. R_{CEi} = \overline{REi_1} t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$S_{CEi} = REi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

$$R_{CAi} = \overline{RAi_1} t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$

- $$S_{CAi} = RAi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$$
3. $K_{REi_j} = REi_{j+1} t_1$
 $J_{REi_j} = REi_{j+1} t_1$
 $K_{RAi_j} = RAi_{j+1} t_1$
 $J_{RAi_j} = RAi_{j+1} t_1$
 4. $16 = t_2$
 5. $K_{FLA} = t_3$
 6. $17 = t_4 - t_{13}$
 7. $R_{REi_{12}} = \overline{CAi} t_{14} \text{ POR } \overline{DIG}$
 $S_{REi_{12}} = CAi t_{14} \text{ POR } \overline{DIG}$
 8. $K_{FLE} = t_{15}$
 $K_Q = J_Q = t_{15}$
 9. $K_H = J_H = p \overline{MN}$
 10. $R_{RAi_j} = t_{16} \text{ POR } \overline{DIG}$
 11. $J_{INH} = t_{17} \text{ POR } \overline{DIG}$
- H. 1. $K_{INH} = \text{para}$ $K_T = \text{para}$ $S_{11} = \text{para}$ $K_{MENOS} = \text{para}$
 $J_{\alpha} = \text{para } \overline{IGUAL}$ $R_{\alpha} = \text{para}$ $K_{MAS} = \text{para}$
2. $R_{CEi} = \overline{REi}_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 $S_{CEi} = REi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 $R_{CAi} = \overline{RAi}_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 $S_{CAi} = RAi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 3. $K_{REi_j} = \overline{REi}_{j+1} t_1$
-

$$J_{REi_j} = REi_{j+1} t_1$$

$$K_{RAi_j} = RAI_{j+1} t_1$$

$$J_{RAi_j} = RAI_{j+1} t_1$$

4. $16 = t_2$
5. $K_{FLA} = t_3$
6. $17 = t_4 - t_{13}$
7. $R_{RAi_{12}} = CAI t_{14} \text{ PARA}$
 $S_{RAi_{12}} = CAi t_{14} \text{ PARA}$
8. $K_{FLE} = t_{15}$
 $K_Q = J_Q = t_{15}$
9. $K_H = J_H = p \overline{MN}$
10. $S_{REi_j} = t_{16} \text{ PARA}$
11. $J_T = t_{17} \text{ PARA IGUAL}$
 $K_{RAi_j} = RAI_{j-1} p^T \overline{RAW}_{10} \overline{RAX}_{10} \overline{RAY}_{10} \overline{RAZ}_{10}$
 $J_{RAi_j} = RAI_{j-1} p^T \overline{RAW}_{10} \overline{RAX}_{10} \overline{RAY}_{10} \overline{RAZ}_{10}$
 $J_{INH} = p^T (\overline{RAW}_{10} + \overline{RAX}_{10} + \overline{RAY}_{10} + \overline{RAZ}_{10})$
 $K_T = p^T (\overline{RAW}_{10} + \overline{RAX}_{10} + \overline{RAY}_{10} + \overline{RAZ}_{10})$

I. 1. $R_{REi_j} = \text{igual POR}$
 $K_{POR} = \text{igual}$

J. 1. $K_{INH} = \text{igual PARA}$ $J_T = \text{igual PARA}$ $S_N = \text{igual PARA}$
 $K_{\nu} = \text{igual PARA}$ $S_M = \text{igual PARA}$ $S_{\mu} = \text{igual PARA}$
 $K_U = \text{igual PARA}$ $R_{CADI} = \text{igual PARA}$

2. $K_{REi_j} = REi_{j-1} p T REW_{10} REX_{10} REY_{10} REZ_{10}$
 $J_{REi_j} = REi_{j-1} p T REW_{10} REX_{10} REY_{10} REZ_{10}$
 $K_T = p T (REW_{10} + REX_{10} + REY_{10} + REZ_{10})$
3. $R_{CDi} = tt_0$ PARA IGUAL
4. $R_{RRi_j} = RAi_j tt_1$ PARA IGUAL
 $S_{RRi_j} = RAi_j tt_1$ PARA IGUAL
5. $R_\alpha = tt_2$ PARA
6. $R_{CEi} = \overline{REi_1} t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 $S_{REi} = REi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 $R_{CAi} = \overline{RAi_1} t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
 $S_{CAi} = RAi_1 t_0 (\overline{IGUAL} + \overline{PARA} + \overline{U})$
7. $K_{RAi_j} = RAi_{j+1} t_1$
 $J_{RAi_j} = RAi_{j+1} t_1$
 $K_{REi_j} = REi_{j+1} t_1$
 $J_{REi_j} = REi_{j+1} t_1$
8. $16 = t_2$
9. $K_{FLA} = t_3$
10. $17 = t_4 - t_{13}$
11. $R_{RAi_{12}} = \overline{CAi} t_{14}$ PARA
 $S_{RAi_{12}} = CAi t_{14}$ PARA
12. $K_{FLE} = t_{15}$
 $K_Q = J_Q = t_{15}$

13. $K_H = J_H = p \overline{MN}$
14. Ver 15 o 27
15. $K_{CKZ} = J_{CDZ} = t_{16}$ FLA IGUAL PARA \bar{U}
16. $R_\alpha = t_{17}$ FLA IGUAL PARA \bar{U}
 $J_U = t_{17}$ FLA IGUAL PARA \bar{U}
17. $R_{CEi} = t_0$ IGUAL PARA U
 $R_{CAi} = \overline{RAi}_1 t_0$ IGUAL PARA U
 $S_{CAi} = RAi_1 t_0$ IGUAL PARA U
18. $K_{REi_j} = \overline{REi}_{j+1} t_1$
 $J_{REi_j} = REi_{j+1} t_1$
 $K_{RAi_j} = \overline{RAi}_{j+1} t_1$
 $J_{RAi_j} = RAi_{j+1} t_1$
19. $16 = t_2$
20. $K_{FLA} = t_3$
21. $17 = t_4 - t_{13}$
22. $R_{RAi_{12}} = \overline{CAi} t_{14}$ PARA
 $S_{RAi_{12}} = CAi t_{14}$ PARA
23. $K_{FLE} = t_{15}$
 $K_Q = J_Q = t_{15}$
24. $K_H = J_H = p \overline{MN}$
25. $R_{RRi_j} = \overline{RAi}_j t_{16}$ IGUAL PARA U \overline{FLA}
 $S_{RRi_j} = RAi_j t_{16}$ IGUAL PARA U \overline{FLA}

26. $R_{\alpha} = t_{17}$ IGUAL PARA $\overline{U \overline{FLA}}$
 $K_U = t_{17}$ IGUAL PARA $\overline{U \overline{FLA}}$
27. $R_{RAi_j} = \overline{RRi_j} \quad t_{16}$ IGUAL PARA $\overline{U \overline{FLA}}$
 $S_{RAi_j} = \overline{RRi_j} \quad t_{16}$ IGUAL PARA $\overline{U \overline{FLA}}$
28. $K_{\mu} = t_{17}$ IGUAL PARA $\overline{U \overline{FLA}}$
29. $R_{RAi_{12}} = t_{18}$ CDi
 $S_{RAi_{12}} = t_{18}$ CDi
30. $K_{CADZ} = J_{CADZ} = t_{19}$
31. $K_{RAi_j} = \overline{RAi_{j-1}} \quad t_{20}$
 $J_{RAi_j} = \overline{RAi_{j-1}} \quad t_{20}$
32. $K_{\nu} = t_{21}$ $\overline{CADW \overline{CADY}}$
 $K_{IGUAL} = t_{21}$ $\overline{CADW \overline{CADY}}$
 $K_{PARA} = t_{21}$ $\overline{CADW \overline{CADY}}$
 $R_{REi_j} = t_{21}$ $\overline{CADW \overline{CADY}}$
 $J_{INH} = t_{21}$ $\overline{CADW \overline{CADY}}$
- K. 1. $R_{REi_j} = \overline{RRi_j}$ res
 $S_{REi_j} = \overline{RRi_j}$ res
2. Se cumple automáticamente, por la forma como se ha hecho la conexión en la unidad de entrada.
- L. 1. $R_{RRi_j} = \overline{RAi_j}$ trans
 $S_{RRi_j} = \overline{RAi_j}$ trans
- M. 1. $R_{REi_j} = \lim$

$$R_{RAi_j} = \lim$$

$$R_{RRi_j} = \lim$$

$$S_M = S_N = \lim$$

Ustedes ven de todo este cúmulo de ecuaciones, que existen muchos pulsos que van a la misma entrada; por ejemplo los pulsos a K_{INH} aparecen casi en la totalidad de los pasos anteriores. Conviene por lo tanto tener una sola ecuación para K_{INH} y similarmente para el resto de terminales, es decir obtener un grupo compacto de ecuaciones, mediante simple unión OR de todas las ecuaciones que aparecen. El siguiente paso es necesariamente aplicar los criterios simplificantes del álgebra de Boole para reducir las ecuaciones y optimizar el costo de los circuitos.

Los dos pasos se realizaron separadamente, pero a continuación sólo expongo la solución final, con el objeto de evitar páginas y páginas de ecuaciones que ya no dicen mucho.

ECUACIONES REDUCIDAS.

$$K_{INH} = \text{dig} + \text{mas} + \text{menos} + \text{por} + \text{para} + \text{igual PARA}$$

$$J_{INH} = t_{t_3} (\text{MENOS} + \overline{\text{POR}} \overline{\text{PARA}} + \text{PARA} \overline{\text{IGUAL}}) + t_{t_{17}} \text{POR} (\overline{\text{DIG}} + \text{FLM} + \overline{\text{FLA}} \\ (\text{MAS} + \text{MENOS}) + pT (\text{RAW}_{10} + \text{RAX}_{10} + \text{RAY}_{10} + \text{RAZ}_{10}) + \text{CADW} \\ \cdot \text{CADY } t_{t_{21}}$$

$$K_{y_6} = \text{dig} + \text{igual PARA} + t_{t_{21}} \overline{\text{CADW CADY}}$$

$$J_{y_6} = \text{mas} + \text{menos} + \text{por} + \text{para} \overline{\text{IGUAL}}$$

$$K_T = \text{dig} + \text{mas} + \text{menos} + \text{por} + \text{para} + pT (\overline{RAW}_{10} + \overline{RAX}_{10} + \overline{RAY}_{10} + \overline{RAZ}_{10} + \overline{REW}_{10} + \overline{REX}_{10} + \overline{REY}_{10} + \overline{REZ}_{10})$$

$$J_T = \text{PARA} (\text{igual} + t_{17})$$

$$S_M = S_N = \text{dig} + \text{igual} \text{ PARA}$$

$$S_{11} = \text{dig} + \text{mas} + \text{menos} + \text{por} + \text{para} + \text{igual} \text{ PARA}$$

$$R_{REi_{12}} = tt_0 \overline{i} \overline{\text{MENOS}} \overline{\text{POR}} \overline{\text{PARA}} + i (\overline{\text{MENOS}} + \overline{\text{PARA}} \overline{\text{IGUAL}}) + t_{14} \overline{\text{CAi}} \overline{\text{POR}} \overline{\text{DIG}} + \\ + R_{REi_{j=12}}$$

$$S_{REi_{12}} = tt_0 \overline{i} (\overline{\text{MENOS}} + \overline{\text{PARA}} \overline{\text{IGUAL}}) + i \overline{\text{MENOS}} \overline{\text{POR}} \overline{\text{PARA}} + t_{14} \overline{\text{CAi}} \overline{\text{POR}} \overline{\text{DIG}} + \\ + S_{REi_{j=12}}$$

$$R_{REi_j} = \text{lim} + \text{igual} \text{ POR} + t_{16} (\text{MAS} + \text{MENOS}) + t_{21} \text{CADW CADY} + \text{RRI}_j \text{ res}$$

$$S_{REi_j} = t_{16} \text{PARA} + t_{17} \text{MENOS} \overline{\text{FLA}} + \text{RRI}_j \text{ res}$$

$$K_{REi_j} = \overline{REi_{j-1}} \quad tt_1 (\overline{\text{MENOS}} + \overline{\text{POR}} \overline{\text{PARA}} + \overline{\text{PARA}} \overline{\text{IGUAL}}) + pT \overline{REW}_{10} \overline{REX}_{10} \overline{REY}_{10} \\ \overline{REZ}_{10} + \overline{REi_{j+1}} \quad t_1$$

$$J_{REi_j} = \overline{REi_{j-1}} \quad tt_1 (\overline{\text{MENOS}} + \overline{\text{POR}} \overline{\text{PARA}} + \overline{\text{PARA}} \overline{\text{IGUAL}}) + pT \overline{REW}_{10} \overline{REX}_{10} \overline{REY}_{10} \\ \overline{REZ}_{10} + \overline{REi_{j+1}} \quad t_1$$

$$R_{RAi_{12}} = t_{14} \overline{\text{CAi}} (\text{MAS} + \text{MENOS} + \text{POR} \text{ DIG} + \text{PARA}) + t_{18} \overline{\text{CDi}} + R_{RAi_{j=12}}$$

$$S_{RAi_{12}} = t_{14} \overline{\text{CAi}} (\text{MAS} + \text{MENOS} + \text{POR} \text{ DIG} + \text{PARA}) + t_{18} \overline{\text{CDi}} + S_{RAi_{j=12}}$$

$$R_{RAi_j} = t_{16} \overline{\text{POR}} \overline{\text{DIG}} + \overline{\text{RRI}_j} \quad t_{16} \text{PARA} \overline{\text{IGUAL}} \overline{\text{U}} \overline{\text{FLA}} + \text{lim}$$

$$S_{RAi_j} = \overline{\text{RRI}_j} \quad t_{16} \text{PARA} \overline{\text{IGUAL}} \overline{\text{U}} \overline{\text{FLA}}$$

$$K_{RAi_j} = \overline{RAi_{j-1}} (tt_1 \overline{\text{POR}} + t_{20} + pT \overline{RAW}_{10} \overline{RAX}_{10} \overline{RAY}_{10} \overline{RAZ}_{10}) + \overline{RAi_{j+1}} \quad t_1$$

$$J_{RAi_j} = \overline{RAi_{j-1}} (tt_1 \overline{\text{POR}} + t_{20} + pT \overline{RAW}_{10} \overline{RAX}_{10} \overline{RAY}_{10} \overline{RAZ}_{10}) + \overline{RAi_{j+1}} \quad t_1$$

$$K_m = tt_2 (\overline{\text{MENOS}} + \overline{\text{POR}} + \overline{\text{PARA}} + \overline{\text{IGUAL}})$$

$$K_{DIG} = tt_3 (\overline{MENOS} + \overline{POR PARA} + \overline{PARA IGUAL}) + t_{17} \overline{FLM POR}$$

$$R_{CMn} = \bar{n} tt_0 \overline{POR}$$

$$S_{CMn} = n tt_0 \overline{POR}$$

$$R_{\infty} = \text{mas} + \text{menos} + \text{por} + \text{para} + tt_2 (\overline{POR} + \overline{PARA}) + \\ + t_{17} \overline{FLA (MAS + MENOS) + IGUAL PARA (FLA \bar{U} + \overline{FLA U}) + \overline{FLM DIG}}$$

$$R_{CEi} = t_0 (\overline{IGUAL PARA U} + \overline{REi_1})$$

$$S_{CEi} = t_0 \overline{REi_1 POR IGUAL PARA U}$$

$$R_{CAi} = t_0 \overline{RAi_1}$$

$$S_{CAi} = t_0 \overline{RAi_1}$$

$$K_{FLM} = \text{dig}$$

$$K_{FLA} = t_3$$

$$K_{FLE} = t_{15}$$

$$K_Q = J_Q = t_{15}$$

$$K_H = J_H = p \overline{MN}$$

$$K_U = \text{igual PARA} + t_{17} \overline{IGUAL PARA U FLA}$$

$$J_U = t_{17} \overline{IGUAL PARA \bar{U} FLA}$$

$$K_{\perp} = t_{17} \overline{IGUAL PARA \bar{U} \overline{FLA}}$$

$$K_{MAS} = \text{menos} + \text{por} + \text{para}$$

$$K_{MENOS} = \text{mas} + \text{por} + \text{para}$$

$$K_{POR} = \text{igual}$$

$$K_{PARA} = t_{21} \overline{CADW CADY}$$

$$R_{RRi_j} = \text{lim} + \overline{RAi_j} (\text{trans} + tt_1 \overline{PARA IGUAL} + t_{16} \overline{PARA IGUAL U \overline{FLA}})$$

$$S_{RRi_j} = R_{Ai_j} (\text{trans} + t_{t_1} \text{ PARA IGUAL} + t_{t_{16}} \text{ PARA IGUAL U } \overline{\text{FLA}})$$

$$R_{CDi} = t_{t_0} \text{ PARA IGUAL}$$

$$K_{CMD} = J_{CMD} = t_{t_{16}} \text{ DIG POR}$$

$$K_{IGUAL} = t_{t_{21}} \text{ CADW CADY}$$

$$K_{CDZ} = J_{CDZ} = t_{t_{16}} \text{ IGUAL PARA } \overline{\text{U FLA}}$$

$$R_{CADi} = \text{igual PARA}$$

$$K_{CADZ} = J_{CADZ} = t_{t_{19}}$$

$$16 = t_2$$

$$17 = t_4 - t_{13}$$

Estas son las ecuaciones que caracterizan a SUD. Su diagrama completo de conexiones aparece al final de la tesis, en el PLANO No. 6. Dicho plano tiene algunas filas, y se ha construido de tal manera que cada entrada a una compuerta de una fila ha sido ya formada por compuertas de filas inferiores (Existen, sin embargo poquísimas excepciones a esta regla, por razones de espacio). Cuando a la salida de alguna compuerta hay dibujados uno o dos o más rectángulos, esto significa que se ha conectado a uno o más MC365 con el único objeto de darle el fan-out suficiente como para soportar el número de cargas que va a manejar.

DISEÑO DE LA UNIDAD DE SALIDA.

Se compone de 12 bloques iguales, correspondientes a los 12 dígitos, que serán exhibidos. En lo que sigue a continuación no se detallará sino el contenido de uno de los bloques pues el resto son similares. Los pasos a seguirse son lógi-

cos: primero debemos escoger qué registro vamos a mostrar, luego decodificar su contenido y finalmente mostrarlo. De ahí que empezaremos con SR, seguirá DEC y al último UEX.

A. SR

Recopilando toda la información que existe sobre la unidad de salida en cada una de las conclusiones que aparecen al final del análisis de cada operación, se puede compendiar dichos datos en la siguiente tabla:

TABLA 16

Tecla aplastada	Registro mostrado
+	RA
-	RA
x	RE
÷	RA
=	RA
T	RR
R	RA
dígito no después de x	RE
dígito después de x	RA

Tenemos información permanente (es decir hasta que no se haya aplastado otra tecla) de las siguientes: +, -, x, ÷, R, esta última por ser equivalente a aplastar +. Pero no tenemos información permanente sobre las otras porque: el flip flop IGUAL es limpiado al finalizar su función, cosa similar ocurre con DIG y finalmente sobre T no se conoce nada. De ahí que convenga inventar los siguientes nuevos flip flops.

IG tal que $S_{IG} = \text{igual } \overline{\text{POR}}$

$R_{IG} = \text{mas} + \text{menos} + \text{por} + \text{para} + \text{dig} + \text{trans}$

DI tal que $S_{DI} = \text{dig}$

$R_{DI} = \text{mas} + \text{menos} + \text{por} + \text{para} + \text{igual} + \text{trans}$

y TRANS tal que $S_{TRANS} = \text{trans}$

$R_{TRANS} = \text{mas} + \text{menos} + \text{por} + \text{para} + \text{igual} + \text{dig}$

Anotemos además que se exhibirá cierto registro en su forma normal, si en la posición 12 hay un 0, pero en su forma de complemento de 9 si en la posición 12 hay un 9, puesto que como se anunció al principio ésta es una posición de signo. Por eso conviene aclarar que no se puede ocupar para una determinada operación más que 11 dígitos de resultados. Si ocurre lo contrario la máquina no funcionará correctamente.

Traduciendo esta información y la de la tabla 16 en forma de ecuación, y si definimos

EX = el valor exhibido o mostrado

se debe cumplir para $i = W, X, Y, Z$ y $j = 0, 1, \dots, 12$ que:

$$\begin{aligned} EX_{i,j} = & RR_{i,j} \text{ TRANS } \overline{RRW}_{12} \overline{RRX}_{12} \overline{RRY}_{12} \overline{RRZ}_{12} + \\ & + \overline{RR}_{i,j} \text{ TRANS } RRW_{12} RRX_{12} RRY_{12} RRZ_{12} + \\ & + RA_{i,j} (\text{MAS} + \text{MENOS} + \text{DI POR} + \text{IG} + \text{PARA}) \overline{RAW}_{12} \overline{RAX}_{12} \overline{RAY}_{12} \overline{RAZ}_{12} + \\ & + \overline{RA}_{i,j} (\text{MAS} + \text{MENOS} + \text{DI POR} + \text{IG} + \text{PARA}) RAW_{12} RAX_{12} RAY_{12} RAZ_{12} + \\ & + RE_{i,j} (\text{POR} + \text{DI } \overline{\text{POR}}) \overline{REW}_{12} \overline{REX}_{12} \overline{REY}_{12} \overline{REZ}_{12} + \\ & + \overline{RE}_{i,j} (\text{POR} + \text{DI } \overline{\text{POR}}) REW_{12} REX_{12} REY_{12} REZ_{12} \end{aligned}$$

Los dos últimos renglones se pueden simplificar a

$$\begin{aligned}
 &+ REi_j (POR + DI) \overline{REW}_{12} \overline{REX}_{12} \overline{REY}_{12} \overline{REZ}_{12} + \\
 &+ \overline{REi}_j (POR + DI) REW_{12} REX_{12} REY_{12} REZ_{12}
 \end{aligned}$$

Estas son las ecuaciones con las cuales se ha trazado el diagrama de SR que aparece en el PLANO No. 7.

B. DEC

Procedemos a la inversa que con COD. Ahora a partir de W X Y Z debo obtener el dígito. Esto fácilmente se ve en la Tabla 17.

TABLA 17

	W X Y Z		
0	0 0 0 0		0 = $\overline{W} \overline{X} \overline{Y} \overline{Z}$
1	0 0 0 1		1 = $\overline{W} \overline{X} \overline{Y} Z$
2	0 0 1 0		2 = $\overline{W} \overline{X} Y \overline{Z}$
3	0 0 1 1		3 = $\overline{W} \overline{X} Y Z$
4	0 1 0 0	Entonces	4 = $\overline{W} X \overline{Y} \overline{Z}$
5	1 0 1 1		5 = $W \overline{X} Y Z$
6	1 1 0 0		6 = $W X \overline{Y} \overline{Z}$
7	1 1 0 1		7 = $W X \overline{Y} Z$
8	1 1 1 0		8 = $W X Y \overline{Z}$
9	1 1 1 1		9 = $W X Y Z$

Pero puesto que hay casos que nunca ocurren, cuales son:

$$\begin{array}{ll}
 \overline{W} X \overline{Y} Z & W \overline{X} \overline{Y} \overline{Z} \\
 \overline{W} X Y \overline{Z} & W \overline{X} \overline{Y} Z \\
 \overline{W} X Y Z & W \overline{X} Y \overline{Z}
 \end{array}$$

Aprovechándolos se puede conseguir alguna reducción, de modo que las ecuaciones finales son:

$$\begin{aligned} 0 &= \overline{X} \overline{Y} \overline{Z} \\ 1 &= \overline{X} \overline{Y} Z \\ 2 &= \overline{X} Y \overline{Z} \\ 3 &= \overline{W} \overline{Y} Z \\ 4 &= \overline{W} X \\ 5 &= W \overline{X} \\ 6 &= W \overline{Y} \overline{Z} \\ 7 &= W \overline{Y} Z \\ 8 &= W Y \overline{Z} \\ 9 &= X Y Z \end{aligned}$$

La construcción de estas ecuaciones se muestra para uno solo de los 12 bloques iguales en el PLA NO No. 7.

C. UEX.

Se usarán lámparas de 60 mA a 28 V. Estas alumbrarán ya sea en forma de 0 o de 1, 2, 3, , o 9. En total se usarán 120 lámparas, colocadas en 12 grupos de 10 lámparas cada uno.

El circuito adicional que se emplea, y que se muestra para un solo grupo en el PLANO No. 7, ha sido diseñado en la siguiente forma:

Cuando de los decodificadores hay una salida alta (-0,75 V) el primer transistor 2N4402 se pone en corte, el segundo transistor 2N4402 en saturación y como consecuencia la lámpara se prende. Pero si del decodificador viene -1,15 V, por efecto del diodo, el primer transistor se satura y consecuentemente el segundo se pone en corte y la lámpara se apaga. La razón de tener dos etapas y no una solamente es con el fin de no cargar demasiado a los decodificadores. Finalmente las resistencias adicionales se han calculado de modo que con un mínimo beta de 30 para el 2N4402 el circuito funcione sin problemas. No hubo necesidad de poner en el colector del transistor de salida una resistencia limitadora de corriente.

Se requiere una fuente de -28 V que de BA pero su diseño, al igual que la fuente V_{EE} , queda fuera del propósito de la presente tesis.

APENDICE III

PARTE PRACTICA REALIZADA.

A continuación explico escuetamente todas las porciones de la máquina que se construyeron. Fueron muy alagadoras experiencias pues me hicieron conocer algunos problemas adicionales que no aparecen cuando se trabaja sobre el papel, pero en general no se presentaron dificultades fuertes. Cabe anotar que debido a la limitación en cuanto a componentes existentes, generalmente había que desarmar lo construido para probar otra parte de la máquina.

1. TABLAS DE VERDAD.

Con el fin de familiarizarme suficientemente con los elementos se pro

APENDICE III

PARTE PRACTICA REALIZADA.

A continuación explico escuetamente todas las porciones de la máquina que se construyeron. Fueron muy alagadoras experiencias pues me hicieron conocer algunos problemas adicionales que no aparecen cuando se trabaja sobre el papel, pero en general no se presentaron dificultades fuertes. Cabe anotar que debido a la limitación en cuanto a componentes existentes, generalmente había que desarmar lo construido para probar otra parte de la máquina.

1. TABLAS DE VERDAD.

Con el fin de familiarizarme suficientemente con los elementos, se procedió a obtener prácticamente las tablas de verdad de los siguientes elementos que son los que tenemos en el laboratorio de Electrónica: MC351G, MC352G, MC353G, MC356G, MC358AG, MC362G. Se probó además el correcto funcionamiento de la unidad de polarización MC354G. Comparadas las tablas así obtenidas, con las que aparecen en el capítulo "CONVENCIONES LOGICAS" se logró completo acuerdo.

2. SOBRE LA UNIDAD DE ENTRADA.

Construí y probé el adecuado funcionamiento del disparador Schmitt de entrada y del monoestable siguiente, y se acopló esto con un flip flop que podía haber sido cualquiera de los allí existentes (MAS o MENOS, etc.).

3. SOBRE LA UNIDAD ARITMETICA.

Por ser la parte más importante de la máquina, fue construída totalmente, CE CA, CD, CM, CDO, CAD, FLE, FLA, FLM, G_1 , G_2 , G_3 , G_4 . Naturalmente parte por parte, debido a las limitaciones antes mencionadas. Sin embargo se logró probar como un todo, aquello que es el corazón de la máquina: el bloque formado por CE FLE CA, FLA, G_1 , G_2 , G_3 , G_4 en conexión con un aearable con entrada de inhibición, que estaba encargado de dar los 10 pulsos, más otro contador encargado de contar los 10 pulsos, y el monoestable y disparador Schmitt de la unidad de entrada. Esta fue la parte práctica más relevante y alagadora. Trabajó perfectamente como se planificó.

4. SOBRE LA MEMORIA.

Implementé un registro de desplazamiento para una sola línea de trabajo (en otras palabras sólo para la línea W) y exclusivamente para 4 bits. Pero no era necesario hacer más pues todo es una repetición exacta de otras líneas con mucho más bits.

5. SOBRE EL CONTROL

Se trabajó sobre SUT: un aearable con entrada de inhibir, que actuaba sobre un monoestable y un contador similar al EFGM para separar los pulsos. Sobre todo estaba interesado en conseguir los 10 pulsos que se usan en una suma elemental.

6. SOBRE LA UNIDAD DE SALIDA.

Realicé UEX solamente para una lámpara. En realidad no valía la pena hacer 120 veces lo mismo.

He ahí mi trabajo.

BIBLIOGRAFIA

WARREN MacARTHUR LITTLEFIELD. The Design of a Tape Macromodule. A Thesis Submitted to the Department of Engineering and the Committee on the Graduate Division of Stanford University in Partial Fulfillment of the Requirements for the Degree of Engineering, June 1967.

MOTOROLA SEMICONDUCTOR PRODUCTS, INC. Motorola Digital Integrated Circuits MSL, Technical Data, Motorola Semiconductor Products, Inc., U.S.A 1967.

E. J. McCLUSKEY. Introduction to the Theory of Switching Circuits, McGraw-Hill Book Company, Inc., New York, 1965.

YAOHAN CHU. Digital Computer Design Fundamentals, McGraw-Hill Book Company, Inc., New York, 1962.

JACOB MILLMAN, HERBERT TAUB.. Pulse, Digital and Switching Waveforms, McGraw-Hill Book Company, Inc., Tokyo, 1965.

HEWLETT PACKARD COMPANY. Operating and Service Manual Model 214 A
Pulse Generator, Hewlett Packard Company, Palo Alto, California, 1967.

HEWLETT PACKARD COMPANY. Operating and Service Manual Model 140 A
Oscilloscope, Hewlett Packard Company, Palo Alto, California, 1963.

ADVANCE ELECTRONICS LIMITED. Transistorised Timer Counter TC 2 A
Instruction Manual, Advance Electronics Limited, England, 1964.