

ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA ELECTRICA

INTERFASE  
MICROCOMPUTADOR - MAQUINA VIP DE TARJETAS UNIVAC

IVAN ENDARA

TESIS PREVIA A LA OBTENCION DEL  
TITULO DE INGENIERO ELECTRICO EN  
LA ESPECIALIZACION DE ELECTRONI  
CA Y TELECOMUNICACIONES



Abril, 1982

Certifico que el presente  
trabajo ha sido realizado  
en su totalidad por el se  
ñor Iván Endara Prieto

A handwritten signature in dark ink, appearing to read 'H. Jakobson JM'. The signature is fluid and cursive, with the initials 'JM' at the end.

ING. HERBERT JACKOBSON

DIRECTOR DE TESIS

A DIOS;

A MIS PADRES.

## A G R A D E C I M I E N T O

A la Escuela Politécnica Nacional, su alumnado y personal docente; en especial a los Ingenieros Herbert Jakobson y Malachy J. Foley que han hecho posible la realización de este trabajo.

# I N D I C E

	Pág.
INTRODUCCION: MICROCOMPUTADORES E INTERFASES -----	1
CAPITULO I: ESTUDIO DE LA MAQUINA PERFOVERIFICADORA DE TARJETAS UNIVAC.	
1.1. Modos de funcionamiento -----	5
1.1.1. Interruptores -----	6
1.1.2. Teclás -----	7
1.1.3. Indicadores -----	8
1.2. Diagrama general de bloques y flujo de señales-	9
1.3. Memoria -----	14
1.3.1. Capacidad -----	14
1.3.2. Direccionamiento -----	15
1.3.3. Habilidadación -----	16
1.4. Transferencia de señales entre memorias y regis- tros -----	21
1.4.1. Introducción de datos desde el teclado -----	22
1.4.2. Verificación y carga de datos desde la estación de lectura -----	23
1.4.3. Secuencia automática de salida de datos -----	25
1.5. Primera fase de la secuencia automática de sali- da de datos -----	27

	Pág.
1.5.1. Flujo -----	27
1.5.2. Períodos importantes -----	31
1.6. Diagrama de tiempos -----	32
1.7. Referencia para la simbología utilizada -----	33
1.7.1. Zonas de la máquina -----	36
1.7.2. Ubicación de las placas -----	37
1.7.3. Clases de elementos -----	38
1.7.4. Tecnología RTL y niveles lógicos -----	40
1.7.5. FLIP-FLOPS -----	44
1.7.6. Representación en las láminas del manual -----	47
CAPITULO II: ESTUDIO DEL MICROPROCESADOR Z-80 DESDE EL PUNTO DE VISTA DE INTERFASES	51
2.1. Descripción de los pines de Z80-CPU -----	53
2.2. Características eléctricas -----	50
2.2.1. Características en DC -----	61
2.2.2. Características en AC -----	63
2.3. Diagramas de tiempo para los ciclos de entrada y salida de datos -----	67
2.4. Soportes en Hardware para Z80-CPU -----	69
2.5. El circuito integrado PIO -----	70
2.5.1. Funcionamiento de PIO -----	71
2.5.2. Ventajas de PIO -----	73

2.5.3.	Descripción interna -----	75
2.5.4.	Pines -----	78
2.5.5.	Diagramas de tiempo -----	80
2.5.6.	Características eléctricas; rangos máximos ----	84

### CAPITULO III: INTERFASE

3.1.	Diversas posibilidades para el Interfase -----	88
3.2.	Diagrama de bloques para el método escogido ---	95
3.3.	Funcionamiento de cada bloque -----	97
3.3.1.	Bloque de compatibilidad de niveles eléctricos y control de flujo de datos hacia UNIVAC -----	98
3.3.2.	Bloque de encodificación -----	106
3.3.3.	Bloque de decodificación -----	112
3.3.4.	Bloque de control de flujo A/DE Z80-CPU y den- tro del Interfase -----	114
3.3.4.1	Bloque de control de flujo de datos desde el bloque de encodificación a PROM -----	114
3.3.4.2	Bloque de control de flujo desde el bloque de control Lógico a Z80-CPU -----	115
3.3.4.3	Bloque de control de flujo de datos entre Z80 - CPU y PIO -----	117
3.3.5.	Bloque PROM -----	119

3.3.6.	Bloque para el borrado del registro de entrada de UNIVAC -----	122
3.3.7.	Bloque de alimentación de nueva ficha y de borrado general -----	127
3.3.8.	Bloque PIO -----	128
3.3.9.	Bloque de control lógico -----	131
3.3.9.1	Señales provenientes de Z80-CPU -----	131
3.3.9.2	Direcciones del Interfase -----	132
3.3.9.3	Imposición del estado de espera -----	134
3.3.9.4	Habilitaciones señaladas por UNIVAC -----	135
3.4.	Interconexión de bloques -----	139
3.5.	Programación -----	139
3.5.1..	El status de UNIVAC -----	143

#### CAPITULO IV: CONSTRUCCION PRACTICA

4.1.	Ventajas del <i>WIRE-WRAP</i> -----	145
4.2.	Distribución de los elementos en la <i>PLACA INT</i> <i>FASE</i> -----	147
4.3.	<i>Extender card y probador lógico</i> -----	149
4.4.	Instalación -----	151
4.4.1.	Ruido e interferencia -----	151
4.4.2.	Alimentación y puesta a tierra -----	154
4.4.3.	Conectores y enchufes -----	157



4.4.4.	Las 24 señales en UNIVAC -----	159
4.5.	Pruebas -----	162
4.5.1.	Pruebas con UNIVAC -----	163
4.5.2.	Pruebas con Z80-CPU y la PLACA-INTFASE -----	163
4.5.3.	Pruebas con todo el sistema interconectado ----	165
4.6.	Funcionamiento -----	166
4.7.	Software -----	168
4.8.	Mantenimiento general -----	170
4.9.	Costos -----	171

#### CAPITULO V: COMENTARIOS, RECOMENDACIONES Y CONCLUSIONES

5.1.	Comentarios -----	177
5.2.	Recomendaciones -----	179
5.3.	Conclusiones -----	180
Anexo -----		183
Bibliografía -----		226

I N T R O D U C C I O N

MICROCOMPUTADORES E INTERFASES.

Los microcomputadores son sistemas electrónicos contruídos para el procesamiento de cualquier tipo de información, y se componen de un microprocesador, cierta cantidad de memoria y uno o varios "terminales" o "sistemas periféricos" que sirven para efectuar la entrada/salida de datos en una forma apropiada.

La utilización de los microcomputadores en muchos campos de la vida diaria se ha vuelto común y a veces, imprescindible. Dos factores que han contribuido para que esto suceda son:

- 1. Su relativa facilidad de adquisición;
- 2. Su enorme flexibilidad, que está dada por su capacidad de aceptación y comando de terminales periféricos por medio de los "Interfases" o entre-fases.

Se puede, entonces, conformar sistemas de procesamiento de información con características acordes a casi todo propósito. Esta es la finalidad que persigue el diseño de un Interfase.

En forma general se clasifican a los Interfases en cuatro categorías:

- a) Interfases Independientes, que permiten el acceso a dis

- cos, cintas, memorias masivas en general.
- b) Interfases Interactuantes, para manejo de impresoras, con-  
solas de teclado y video.
- c) Interfases Sensores, utilizados en la adquisición de da-  
tos que provienen de algún proceso físico como puede ser,  
la detección de presión y temperatura de un horno dentro  
de una fábrica, etc. Estas señales son de naturaleza ana-  
lógica en su mayoría.
- d) Interfases de Control, que se utilizan en el manejo auto-  
mático de switches, válvulas, generadores de señales, etc.

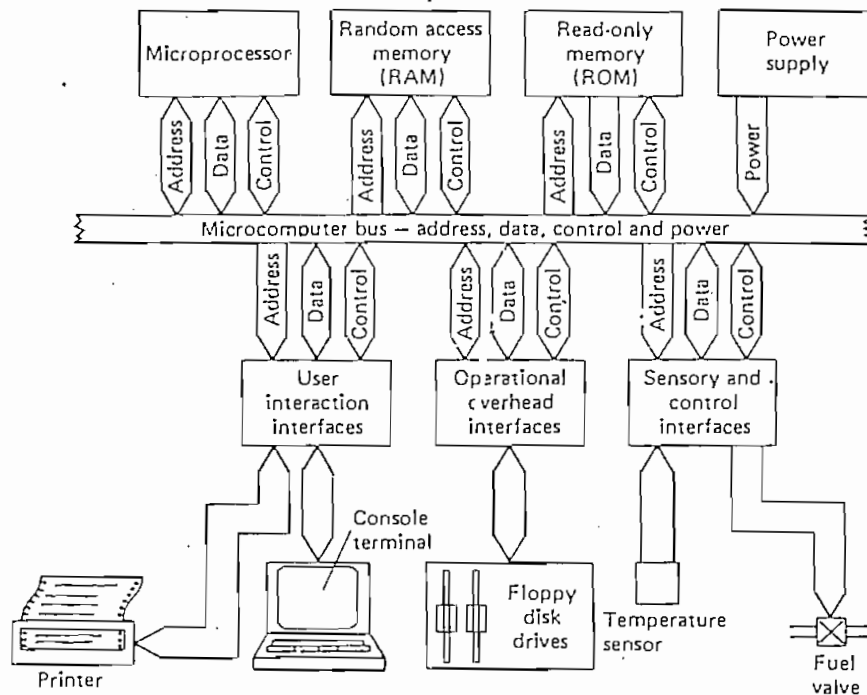


Figura I.1.1. Esquema de un sistema típico: Microcomputador-  
Interfases.

Se distinguen dos aspectos en el Interfase:

1. El aspecto funcional, que involucra la forma como actúan los distintos elementos que lo constituyen, principalmente los circuitos integrados y su funcionamiento lógico; así como también el desempeño global del conjunto; y,
2. El aspecto eléctrico o de niveles voltamperimétricos, que involucra los requerimientos de corrientes y voltajes para dichos elementos de acuerdo con sus características; aquí se topan asuntos como la correcta alimentación, tiempos de retardo, ruido, etc.

Estos dos aspectos anotados han sido considerados en el desarrollo del presente trabajo.

El Interfase aquí tratado cae dentro del grupo de los Interfases Interactuantes y acopla a dos sistemas electrónicos de funcionamiento y características diferentes, cuales son:

- Un microcomputador basado en el microprocesador Z80A (designado Z80-CPU para objeto de simplificación a lo largo del presente); y,
- Una máquina perforadora de tarjetas con impresión 1710 VIP UNIVAC (designada UNIVAC).

El Interfase permitirá una comunicación bidireccional de datos, de modo que UNIVAC pueda actuar como terminal de entrada de datos (sea como lectora de tarjetas o por medio de teclado) y/o como terminal de salida de datos en forma de tarjetas perforadas; todo ésto, bajo control automático.

El control automático está determinados por el "software" introducido al microcomputador, en estrecha vinculación con las señales que se produzcan en el mismo Interfase.

Se estudian en primera instancia y por separado ambos sistemas, UNIVAC y Z80-CPU en los capítulos I y II; luego, conociendo sus características particulares y sabiendo de sus funcionamientos, se procede propiamente al diseño del Interfase en el capítulo III.

El capítulo IV considera su construcción práctica.

# C A P I T U L O I

## ESTUDIO DE LA MAQUINA PERFOVERIFICADORA DE TARJETAS UNIVAC

El estudio de la máquina perfoverificadora de tarjetas con interpretación (Verifier Interpret Puncher, VIP) UNIVAC 1710 ha resultado muy interesante, como podrá notarse a través de los diferentes puntos de este capítulo.

Con estos puntos se va profundizando gradualmente en el examen de UNIVAC, para que así, la comprensión de su funcionamiento no ofrezca mayores dificultades.

UNIVAC es una máquina que no ha sido construída para trabajar con Interfases, por ello ha sido necesario analizar la forma cómo podría ser aprovechada justamente para ésto y de ahí el detenimiento que se ha hecho en su estudio.

### 1.1. MODOS DE FUNCIONAMIENTO

Los modos de funcionamiento que tiene UNIVAC son:

- Como perforadora de tarjetas;
- como verificadora de tarjetas;
- como interpretadora de tarjetas.

Para ello cuenta con una serie de interruptores, teclas e in

dicadores, los mismos que se ven en la figura 1.1.

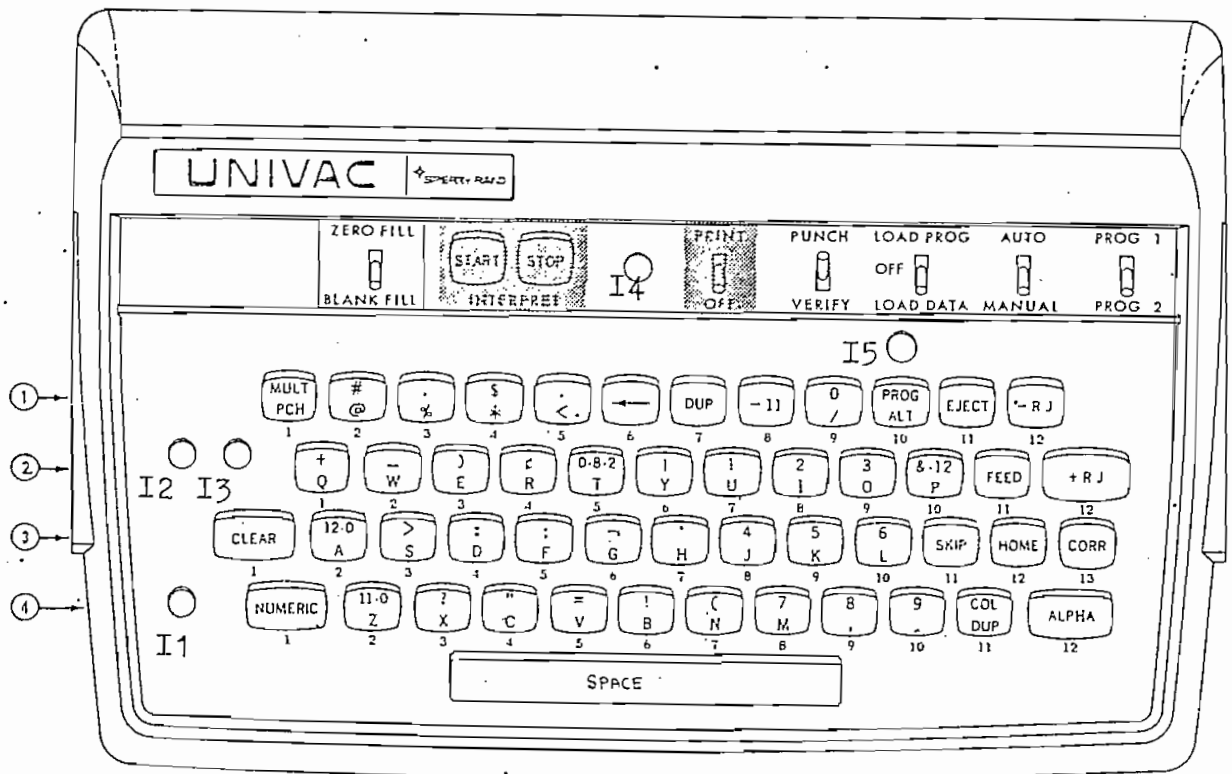



fig. 1.1. TECLADO DE LA 1701 VP Y 1710 VIP  
PARA LA LINEA UNIVAC 9000

 Solamente en la VIP

### 1.1.1. INTERRUPTORES

- ZERO FILL/BLANK FILL: determina si las columnas a la izquierda de un campo con justificación a la derecha\* deberán llenarse con ceros ó con espacios en blanco.
- PRINT/OFF: sirve para conectar o desconectar la unidad impresora, de modo que la impresión pueda o no tener lugar, tan

\* Se recomienda leer el Manual del Operador de UNIVAC.

to si se está perforando como si se está verificando.

- PUNCH/VERIFY: según la posición de este interruptor, la máquina operará como Perforadora o como Verificadora.
- LOAD PROG/OFF/LOAD DATA: permite al operador cargar programas o datos en memoria a partir de fichas\*. Para empezar a perforar es necesario que este interruptor se encuentre en la posición de OFF.
- AUTO/MANUAL: hace que la máquina realice, bajo control manual o bajo control automático, funciones tales como las de salto, duplicación y alimentación.
- PROG 1/PROG 2: mediante este interruptor el operador puede hacer el cambio de un programa a otro con sólo variarlo de posición.

### 1.1.2. TECLAS

Se explica brevemente el funcionamiento de algunas teclas especiales.

- INTERPRET START/STOP: inician o detienen la interpretación de fichas perforadas; para ello, el interruptor de impresión

---

\* Se utilizan indistintamente las palabras "tarjeta" o "ficha", para referirse a las conocidas tarjetas Hollerith.



deberá estar situado en posición de PRINT.

- MULT. PUNCH: permite la perforación múltiple de cuantos datos se deseen en una misma columna.
- COL. DUP: duplica los datos de la memoria, sin afectarla.
- SKIP: permite saltar hasta la primera columna del siguiente campo, borrando las anteriores.
- PROG. ALT: escoge uno de los dos programas de memoria, estando UNIVAC en el modo automático.

### 1.1.3. INDICADORES

Hay cinco indicadores luminosos que dan información sobre las condiciones o *status* de UNIVAC durante su funcionamiento; éstos son:

I1: indicador de condición general de error.

I2: indicador de disparidad o diferencia de caracteres durante la verificación.

I3: indicador de error luego del tercer intento en la verificación de una columna.

I4: indicador de modo alfabético o numérico de los datos que se digitan.

I5: indicador de programa principal o alterno.

Ahora, en orden a indagar cómo ejecuta UNIVAC la serie de operaciones anteriores, se trae a continuación el diagrama de bloques de sus circuitos.

## 1.2. DIAGRAMA GENERAL DE BLOQUES Y FLUJO DE SEÑALES

La figura 1.2. muestra el diagrama general de bloques y flujo de señales de UNIVAC.

Se da a continuación una idea de lo que es cada bloque.

- READER PHOTOCELL DATA: estación lectora de tarjetas. (Se aprecia en la Fig. 3.1.c.).
- READER SPROCKET COUNTER: detector de tarjetas para el sincronismo de la lectura con los ciclos de la máquina.
- READ-VERIFY LOAD DATA OR PROGRAM: circuito lógico que determina el funcionamiento de UNIVAC como lectora, verificadora ó alimentadora de datos - también programas -.
- TRANSLATOR: codificador de caracteres teclados, en doce líneas que corresponden a las doce zonas en que se divide la tarjeta Hollerith. Se detalla en la Fig. 3.1.b.
- ALPHA: indicador de entrada de caracteres en modo alfabético.
- SKIP: salto de columnas en blanco.

- DUPLICATER: duplicador de columnas.
- OPERATIONAL CONTROL: control operacional lógico que automatiza el funcionamiento del flujo de datos de acuerdo al programa alimentado previamente a UNIVAC.
- INPUT REGISTER 12 BITS: registro de 12 bits en paralelo, que almacena un carácter que puede provenir del teclado, de la estación lectora ó de DATA REG, de acuerdo con el ciclo que esté tomando lugar en UNIVAC.
- DATA REGISTER 12 BITS: registro de 12 bits para almacenar un carácter que puede provenir de INPUT REG., de la Memoria de Entrada ó de la Memoria de Salida.
- COMPARATOR: comparador de los caracteres que se encuentran en DATA REG e INPUT REG en un momento dado. Sirve en verificación.
- MISMATCH CHARACTER CONTROL: indicador de caracteres iguales, que actúa una vez realizada la comparación mencionada.
- VERIFY NOT CHARACTER CONTROL: indicador para que se realice o no la comparación de caracteres.
- OUTPUT SEQUENCE COUNTER: contador automático utilizado para indicar las diferentes fases que se suceden en la "Secuencia de Salida de Datos".
- PUNCH SPROCKET: sincronizador de la perforación de tarjetas

con los ciclos de la máquina.

- CARD MOTION CONTROL: control del desplazamiento de tarjetas.
- PUNCH ODD: perforador de columnas impares.
- PUNCH EVEN: perforador de columnas pares.
- PROGRAM REGISTER: registro que contiene la información de los programas cargados en UNIVAC; dicha información dá el comportamiento que debe tener OPERATIONAL CONTROL para cada columna de la tarjeta cuando UNIVAC se encuentra en modo automático.
- MEMORY CONTROL: controlador del flujo de datos entre OUTPUT MEMORY ó INPUT MEMORY con DATA REG.
- OUTPUT MEMORY: memoria que contiene los datos de la última ficha leída o perforada; estos datos salen luego hacia DATA REG. Contiene también la información del programa 1.
- INPUT MEMORY: memoria que almacena los datos nuevos provenientes de DATA REG. Contiene también la información del programa 2.
- MEMORY SELECTION CONTROL: seleccionador de una de las dos memorias que posee UNIVAC.
- MEMORY ADDRESS REGISTER: Registro que posee la dirección de una localidad de memoria.
- COLUMN INDEX DISPLAY: indicador que permite la lectura en

### 1.3. MEMORIA

En primera instancia, un estudio sobre la memoria de UNIVAC resulta conveniente, puesto que de esta forma se va a adquirir una idea más clara de las capacidades y limitaciones que presenta esta máquina.

#### 1.3.1. CAPACIDAD

UNIVAC posee dos bloques de memoria como se ha visto en la figura 1.2. El primero, OUTPUT MEMORY o Memoria de Salida de Datos, contiene los datos que saldrán hacia DATA REG. y de allí hacia las bobinas de perforación y registros de impresión. El segundo, INPUT MEMORY o Memoria de Entrada de Datos, almacena los datos que provienen desde el teclado ó la estación lecto-ra. Se los denomina abreviadamente como OUT. MEM. e IN. MEM.

Cada bloque tiene una capacidad de almacenamiento de 80 palabras \* de 16 bits cada una, con un total de 2.560 bits entre ambos.

Siendo que una tarjeta contiene 80 columnas, entonces toda su información puede ser introducida a memoria.

Cada bit está construido físicamente por un núcleo de ferri

---

\* "palabras", "words" o "bytes".

ta, siendo el dato almacenado ahí, no volátil.

Los 16 bits de cada palabra se distribuyen así:

- 12 bits son utilizados para almacenar un carácter alfanumérico. Esto es, los datos contenidos en la memoria, se encuentran codificados de acuerdo al Código IBM para tarjetas Hollerith (en este código, la tarjeta es dividida en 12 zonas, de manera que cualquier columna de dato equivale a una combinación determinada de estas 12 zonas).

Cuando UNIVAC se halla en el modo de Verificación, el bit de la zona 12 del bloque OUT. MEM. es utilizado para guardar el resultado de la comparación efectuada entre los datos que están siendo verificados.

- 4 bits son utilizados para almacenar una columna de programa. En dichos 4 bits se codifican las operaciones que debe realizar OPERATIONAL CONTROL cuando UNIVAC se encuentra en el modo automático. En OUT. MEM. se almacena el programa 1 y en IN. MEM. el programa 2.

### 1.3.2. DIRECCIONAMIENTO

Las 80 localidades que posee cada bloque de memoria son direccionadas en una forma especial.

La figura 1.3.2. representa al Registro de Direccionamiento - de la Memoria con sus circuitos\*. Pueden allí verse a los 7 flip-flops utilizados para realizar el direccionamiento, así como algunas señales adicionales que sirven de control y que son generadas en concórdancia con las secuencias automáticas que toman lugar en UNIVAC. También se incluyen dos tablas que muestran los estados de los 7 flip-flops para el direccionamiento, tanto de los dígitos de las unidades como de las decenas.

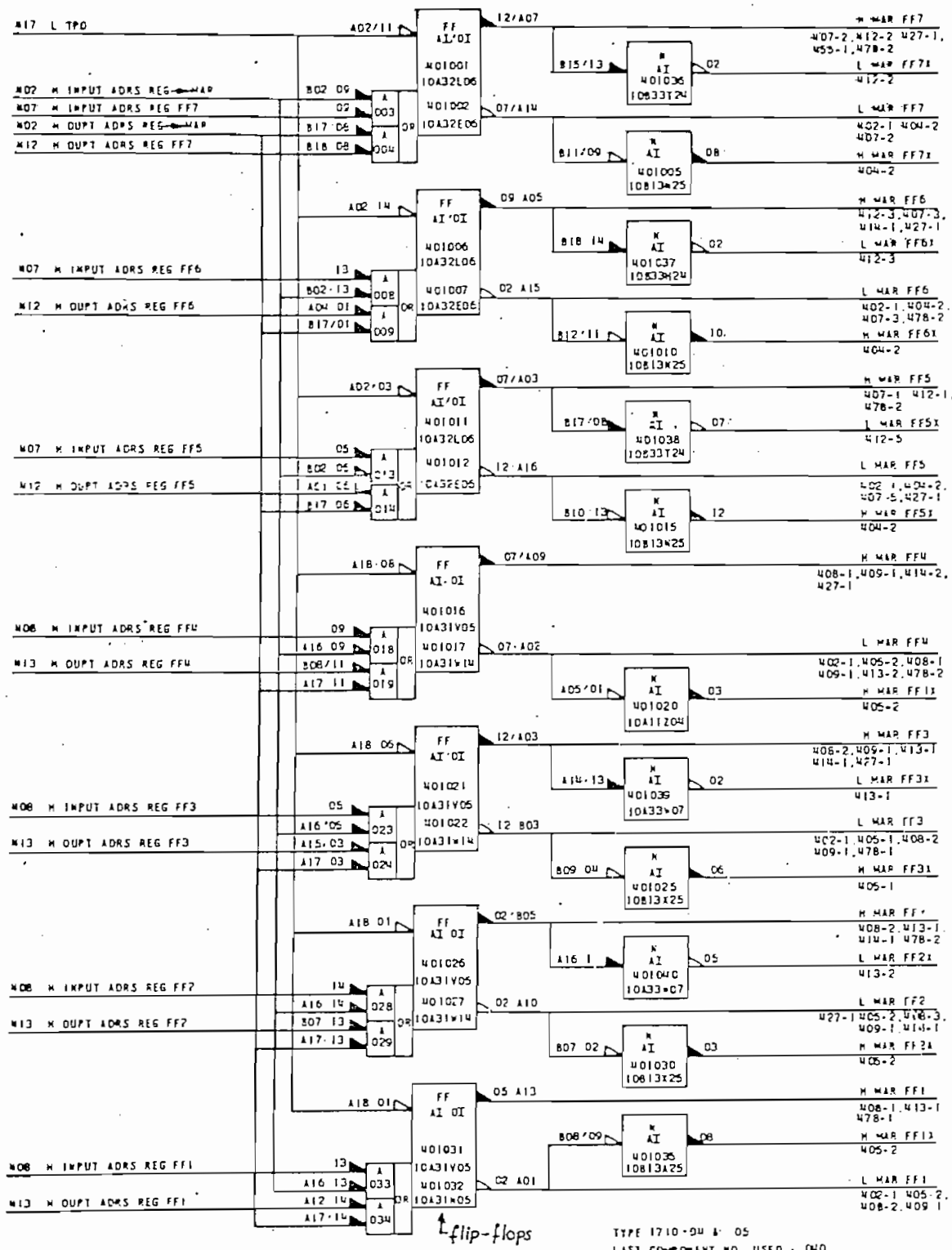
### 1.3.3. HABILITACION

Cada memoria posee 8 grupos -décenas- que son habilitados en forma individual y, a su vez, cada grupo contiene 10 unidades que son habilitadas en grupos de 5 (figuras 1.3.3.1-2). Las líneas que habilitan a tales grupos y unidades dependen de algunas señales de control y temporización.

Se muestra finalmente la ruta de los 12 bits de datos a través de la memoria en la figura 1.3.3.3. donde se observan ciertos bloques que son los que regulan la intensidad de corriente que pasará a través de los núcleos de ferrita señalando los datos.

---

\* En el punto 1.7. se da una referencia para la simbología empleada en estas figuras. Por el momento no se requiere ahondar en ello.



decenas

DIGIT	MAR FF		
	7	6	5
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	0

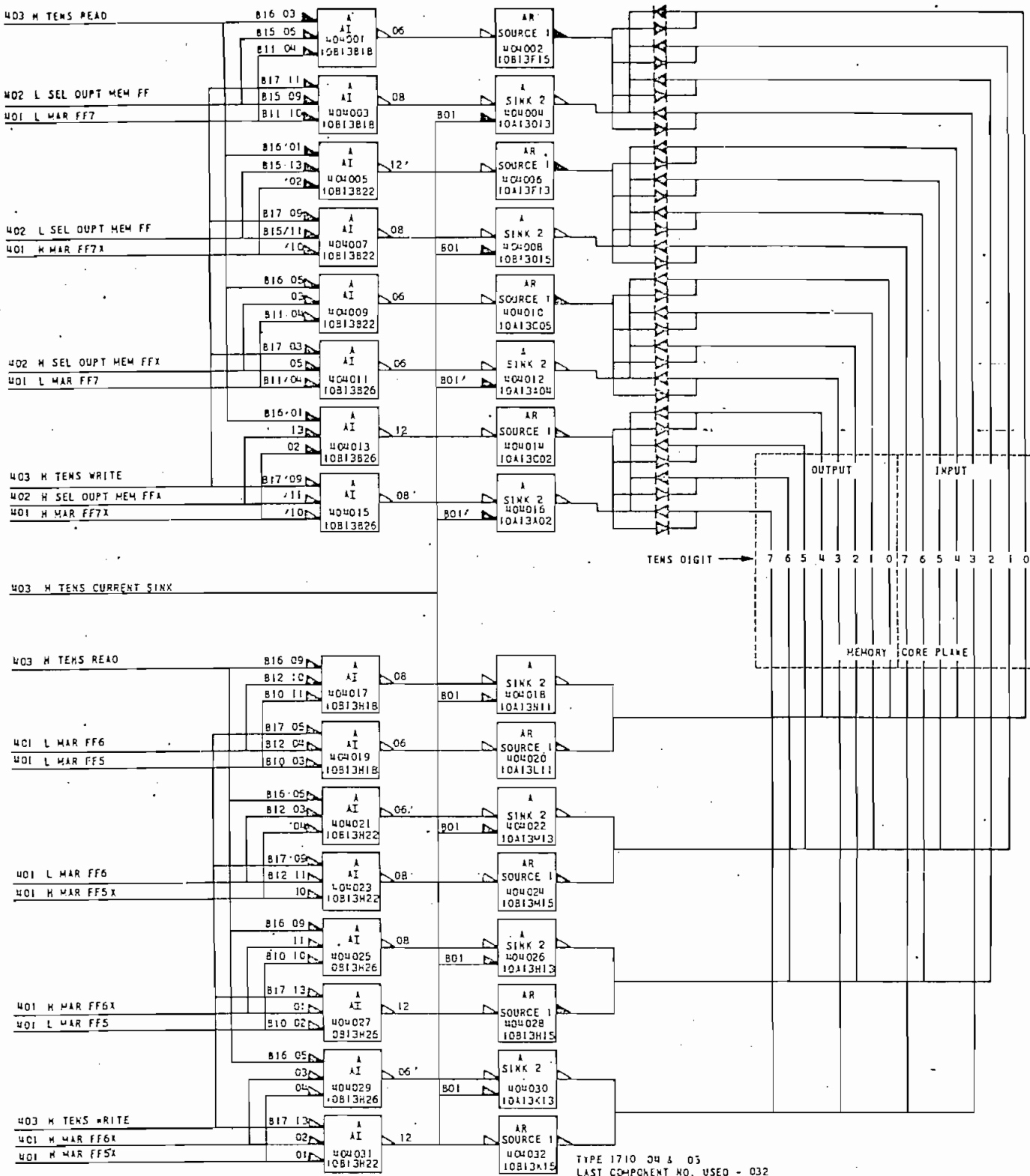
Unidades

DIGIT	UNITS COUNT			
	4	3	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	0
4	0	1	0	0
5	1	0	0	0
6	1	0	0	1
7	1	0	1	1
8	1	1	1	0
9	1	1	0	0

tablas  
1.3.2

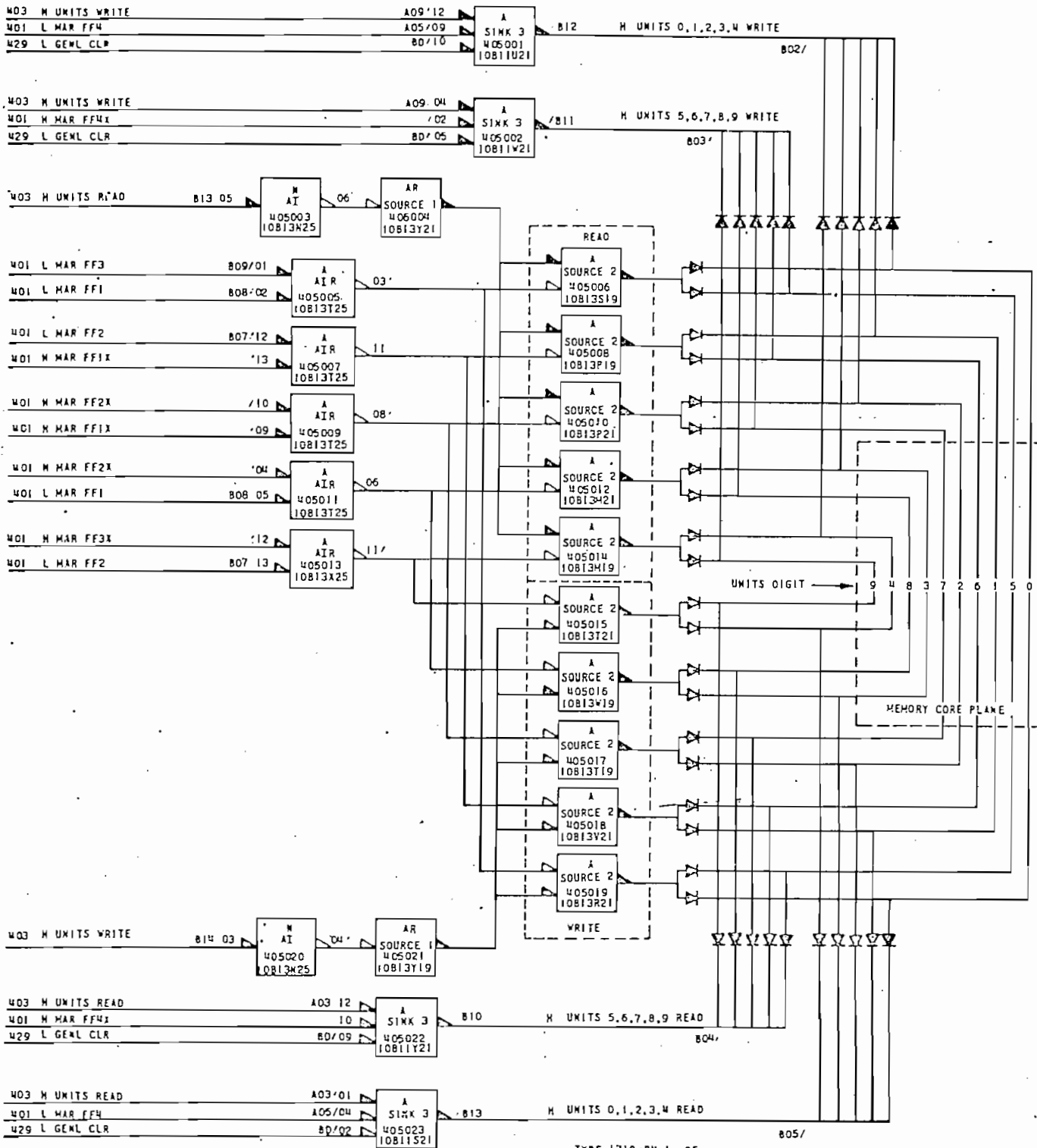
Figura 1.3.2. Circuitos que constituyen el Registro de Direcciónamiento de la Memoria de UNIVAC.





TYPE 1710 04 & 05  
LAST COMPONENT NO. USED - 032

Figura 1.3.3.1. Direccinamiento de los grupos de las decenas de la Memoria.



TYPE 1710-04 & -05  
LAST COMPONENT NO. USED - 023

Figura 1.3.3.2. Selección de las direcciones de unidades de la Memoria.

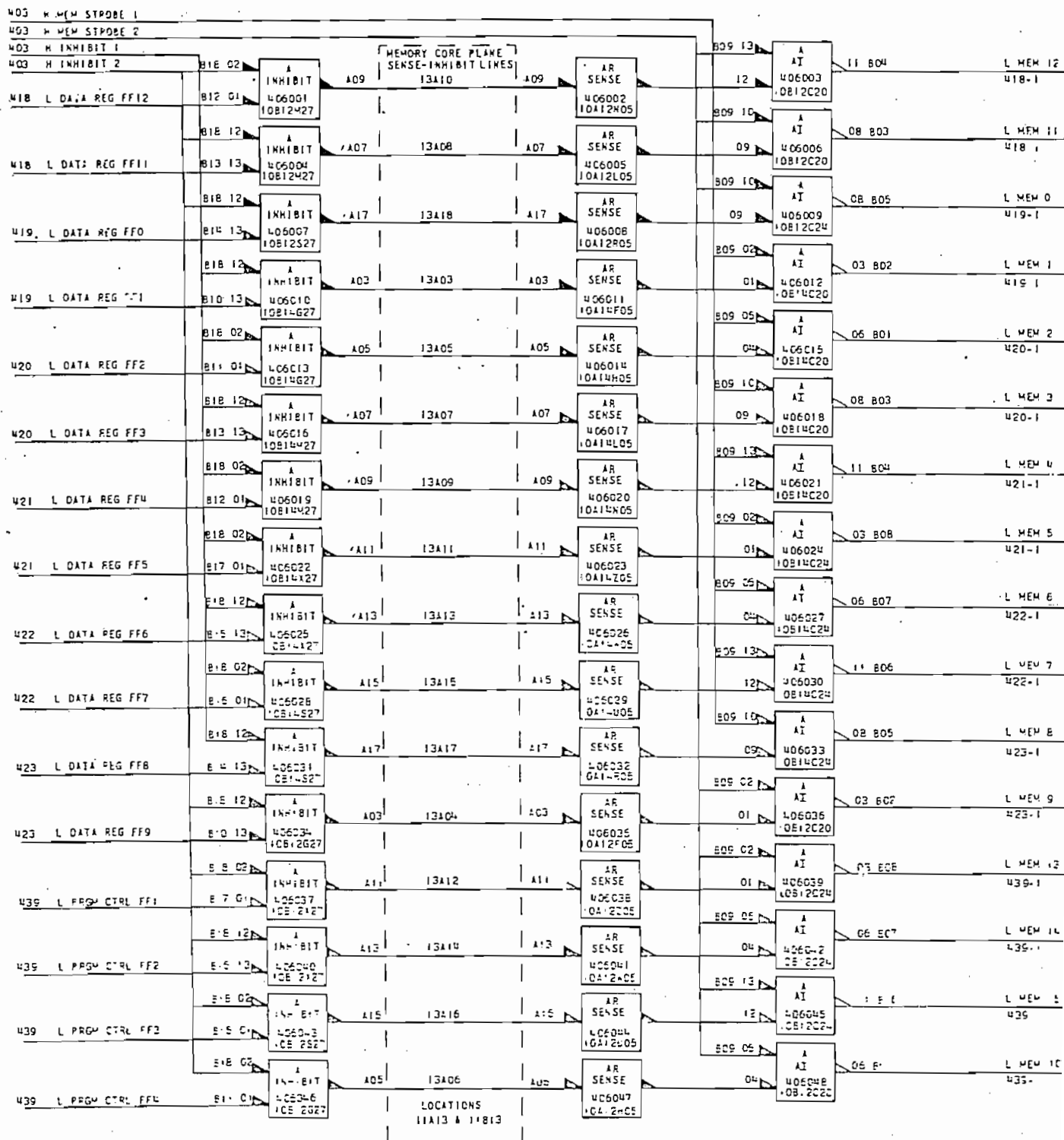


Figura 1.3.3.3. Ruta de los datos a/de Memoria -habilitación-

En conclusión puede decirse que el acceso directo a la memoria de UNIVAC no es muy sencillo. El lograr artificialmente el direccionamiento y la habilitación de la memoria resulta ser una tarea algo complicada, por lo cual conviene más bien estudiar la forma en que entran y salen los datos por los diferentes registros internos; ésto se lo ha hecho en el siguiente punto.

#### 1.4. TRANSFERENCIA DE SEÑALES ENTRE MEMORIAS Y REGISTROS

El paso de los datos a través de los registros y memorias de UNIVAC se realiza en forma secuencial, siguiendo un orden preciso. Las operaciones se suceden principalmente con intervención de circuitos monoestables, biestables y contadores; estos circuitos se encuentran sincronizados en su mayoría con una señal maestra de reloj de UNIVAC.

En la figura 1.2. también se observan los dos registros de datos que intervienen en la transferencia de señales conjuntamente con la memoria, éstos son DATA REGISTER e INPUT REGISTER; ambos contruídos a base de flip-flops y que resultarían ser un sitio muy favorable para lograr un acceso a los datos desde el exterior.

Algunas operaciones automáticas que se efectúan en UNIVAC ( y que se encuentran descritas en el Documento de Servicio de UNIVAC 1710) aclaran la forma como se efectúa la mencionada trans

ferencia de señales, siendo las de mayor interés las siguientes:

#### 1.4.1. INTRODUCCION DE DATOS DESDE EL TECLADO

Al presionar una tecla, se da inicio a una serie de hechos que se cumplen en muy corto tiempo internamente en UNIVAC, sea para introducir el dato digitado en IN. MEM., ó, si el modo en que se encuentra es el de Verificación, efectuar dicha verificación. En estas operaciones intervienen los registros de datos, de direccionamiento, contadores y los dos bloques de memoria.

La forma como se efectúen dichas operaciones, depende del modo en que se encuentre UNIVAC, esto es, en modo de Perforación ó en modo de Verificación.

- a) En el modo de Perforación.- La operación de introducción de datos consta de dos "ciclos básicos de memoria" (un ciclo básico de memoria tiene 8 períodos de 5  $\mu$ seg. cada uno, por lo tanto dura 40  $\mu$ seg.)\*. El primer ciclo reemplaza el dato de IN. MEM. con el dato que acaba de ser digitado e incrementa en uno el índice de columnas; el segundo ciclo chequea el programa almacenado en la memoria cuando UNIVAC se encuentra en modo automáti

---

\*En el punto 1.6. se explica sobre el "ciclo básico de memoria"

co de operación (más detalles sobre esta operación constan en la tabla 3.1.).

- b) En el modo de Verificación.- Utiliza también dos ciclos básicos de memoria; el primero realiza la comparación e incrementa el direccionamiento del registro; el segundo chequea el programa almacenado en memoria para el control automático de la operación. Si ocurriera una disparidad, entonces el direccionamiento del registro no es incrementado y son permitidas dos tentativas más para la verificación. Si luego de dichas dos tentativas persistiera el error, el carácter digitado reemplazará al dato que se encontraba en IN. MEM.

Si bien es cierto que los ciclos de memoria se suceden con mucha rapidez, sin embargo el inicio de los mismos ocurre solamente una vez que el circuito monoestable de retardo utilizado para asegurar la estabilidad del dato digitado y así evitar los problemas de redisparos, rebotes, sobretiros, etc. , haya regresado a su estado original luego de unos 12 mseg. de haber sido presionada la tecla.

#### 1.4.2. VERIFICACION Y CARGA DE DATOS DESDE LA ESTACION DE LECTURA

En esta operación, los datos se introducen a través de los

fotodiodos de la estación de lectura y provienen de tarjetas perforadas que se desean verificar ó desde tarjetas perforadas cuyos datos se desean cargar a memoria para luego duplicarlos en otras tarjetas y/o interpretarlos.

En la operación de lectura se utilizan cuatro ciclos de memoria. El primero almacena en IN. MEM: el dato leído, realizando previamente una comparación entre el dato guardado anteriormente en IN. MEM. con el dato que acaba de leerse; el segundo almacena el resultado de dicha comparación en OUT. MEM. (sóloamente en el modo de Verificación), incrementa la dirección de MEMORY ADDRESS REG. y establece el inicio de la operación de "control de post-lectura".

El control de post-lectura viene a ser una secuencia de conteo efectuada por medio de flip-flops, que toma inicio el momento en que ha atravesado por la estación lectora una columna cualquiera de la tarjeta que está siendo leída.

El ciclo de memoria para la primera columna ocurre cuando el contador-sincronizador interno de UNIVAC alcanza la cuenta de 46 y los ciclos de memoria para las subsiguientes columnas ocurren cada vez que dicho contador alcanza la cuenta de 16.

Los ciclos básicos de memoria tercero y cuarto, son usados para generar 25 contabilizaciones adicionales de columnas luego de que ya han sido leídas las 80 columnas de la tarjeta;

hasta que acabe de pasar ésta por la estación visible.

#### 1.4.3. SECUENCIA AUTOMATICA DE SALIDA DE DATOS

El momento en que se digita una de las teclas FEED ó EJECT, ocurre internamente en UNIVAC una secuencia que automáticamente transfiere todo el contenido de IN. MEM. a OUT. MEM., inicia la movilización de la tarjeta hacia la estación visible, luego continúa con su perforación, hasta acabar con la expulsión de la tarjeta hacia el cajetín de salida.

Dicha secuencia puede iniciarse automáticamente cuando UNIVAC se halla bajo el modo automático.

Se ha dividido a esta secuencia en cuatro fases que se describen a continuación.

- a) La primera fase transfiere los datos de IN. MEM. a OUT. MEM. para lo cual utiliza dos de los siete contadores básicos (flip-flops de conteo); y continúa ininterrumpidamente hasta que las 80 columnas han sido totalmente transferidas.

Esta fase opera en forma idéntica en los modos de Perforación y de Verificación. Para la Interpretación, la transferencia puede darse desde IN. MEM. hacia OUT. MEM.



desde OUT. MEM. hacia OUT. MEM., dependiendo de la posición del switch PROGM/AUTO.

En el punto 1.5. se analiza a esta fase con más detenimiento.

- b) La segunda fase se inicia inmediatamente después de completada la primera. Aquí, la tarjeta inicia su recorrido hacia la estación de lectura. La dirección del registro de salida es incrementada en uno cada vez que los contadores alcanzan la cuenta de 4. La dirección del registro de salida sirve a su vez de contador para determinar si una tarjeta se encuentra en la estación de lectura o no.

Esta fase se desarrolla a través de 4 ciclos básicos de memoria.

- c) La tercera fase sirve para sincronizar la perforación y la impresión de datos en la tarjeta que se desplaza a través de la estación visible. Los datos salen de OUT. MEM.
- d) La cuarta y última fase toma lugar una vez finalizada la anterior. Sirve para remover la tarjeta desde la estación de perforación hasta el cajetín. Se desarrolla a través de 4 ciclos básicos de memoria.

## 1.5. PRIMERA FASE DE LA SECUENCIA AUTOMATICA DE SALIDA DE DATOS

Como ya se anotó, durante esta fase se vierte todo el contenido de la Memoria de Entrada en la Memoria de Salida.

Los datos van pasando uno a uno por entre los dos registros principales, en períodos predeterminados dentro de cada ciclo de memoria.

Se ofrece de esta manera una buena oportunidad para extraer datos o introducirlos en UNIVAC desde el Interfase, con el auxilio de algunas señales de sincronización provenientes de los registros que controlan esta fase.

Seguidamente se analiza más esta fase.

### 1.5.1. FLUJO

En la tabla 1.5.1. se describe el desarrollo de esta primera fase de la secuencia automática de salida, desarrollo que ha sido traducido en diagramas de bloques de más fácil comprensión, como se muestran en las figuras 1.5.1.1-2.

Los 10 diagramas ahí indicados equivalen a los 10 pasos que toman lugar durante esta fase; así:

	1ST CYCLE	2ND CYCLE	3RD CYCLE	4TH CYCLE	LOOK AHEAD CYCLE	DATA CYCLE
ASTM- CHPCHDUS	FIELD OPERATION INITIATED BY THE SETTING OF START DUPT SEQ FF (451) ALL INPUT OPERATIONS SUSPENDED (434-003)					
TP 0	DUPT SEQ CTRS = (-----) (452,453)	DUPT SEQ CTRS = (-----1) (452,453)	DUPT SEQ CTRS = (-----21) (452,453)	DUPT SEQ CTRS = (-----2-) (452,453)	DUPT SEQ CTR = (-----2-) (452,453) SET INPUT CYCLE FF (437-011) (INITIATES LOOK AHEAD CYCLE) (INITIATES NEW CYCLE (403-004))	DUPT SEQ CTR = (-----) (452,453) CLR INPUT CYCLE FF CYCLE ALERT FF IS C (437-010) AND THEN TO FIRST CYCLE OF P OR PERFORM AUTOMATI OPERATIONS DETECTED PRGM LOOK AHEAD CTC
TP 1		DUPT ADRS REG TO MAR (402-022) (ADDRESSES INPUT NEW EXCEPT WHEN PRGM SW SET TO PRGM 1 DURING INTERPRET MODE (402-010))	DUPT ADRS REG TO MAR (402-022) (ADDRESSES DUPT NEW (402-006))		INPUT ADRS REG TO MAR (402-018) CHECK PRGM SW (430) FOR PRGM LOCATION (402-003)	
TP 2		GENERATE READ NEW STROBE (403-011) INPUT NEW TO DATA REG (418 THRU 423)	GENERATE READ NEW STROBE (403-011)		GENERATE READ NEW STROBE (403-011) SELECTED PRGM NEW TO DATA REG (418 THRU 423) AND PRGM REG (438)	
TP 3		DATA REG TO INPUT REG (426-014)	INCREMENT DUPT ADRS REG (415-004)			
TP 4			CLR DATA REG (426-005)	RP MODE, INITIATE COL 1 PRGM SEARCH WHEN DUPT ADRS REG = 1 (432-001) WHICH SETS INPUT CYCLE ALERT FF (437-002) AND INPUT PRGM CYCLE FF (437-022)		
TP 5			INPUT REG TO DATA REG (426-007)	CLR START DUPT SEQ FF WHEN DUPT ADRS REG = 1 (451-003)	CLR PRGM OPERATIONAL CTRL (441-001)	
TP 6		GENERATE WRITE NEW STROBE (403-015) (DATA TO INPUT NEW)	GENERATE WRITE NEW STROBE (403-015) (DATA TO DUPT NEW) CLR INPUT REG (426-015)	SET PUSH FF WHEN DUPT ADRS REG = 1 (451-014) ENERGIZE THE PUNCH CLUTCH (476-009) AND ACTIVATE THE CARD PATTERN (476-007)	GENERATE WRITE NEW STROBE (403-015) SET PRGM OPERATIONAL CTRL (441) WHEN IN 470 MODE	
TP 7	SET DUPT SEQ CTR FF 1 AT T.E. OF TP 7 (451-005, 453- 016) INITIATE NEW CYCLE (403- 002)	SET DUPT SEQ CTR FF 2 AT T.E. OF TP 7 (454-005, 453- 012) CONTINUE NEW CYCLE	CLR DUPT SEQ CTR FF 1 AT T.E. OF TP 7 (454-005, 453- 015)	CLR DUPT SEQ CTR FF 2 AT T.E. OF TP 7 (454-005) 453-011) IF PUSH FF IS CLEARED, PRO- CEED TO FIRST CYCLE OF THIS PHASE. IF PUSH FF IS SET, DUPT SEQ CTR FF 2 REMAINS SET AND THE LOOK AHEAD CYCLE OF THIS PHASE OCCURS.	CLR INPUT PRGM CYCLE FF AT T.E. OF TP 7 (437-012) CLR INPUT CYCLE ALERT FF IF FIELD DEFINITION CTRL FF IS CLEARED (437-005)	

Tabla 1.5.1. Primera fase de la Operación Automática de Salida de Datos.

### 1.5.2. PERIODOS IMPORTANTES

De entre los diez pasos anteriores, el décimo presta mayor facilidad para la transferencia de datos. Es en este momento en que se escriben los datos en la Memoria de Salida.

Si el modo en que se encuentra UNIVAC es el de perforación, dichos datos son perforados en tarjetas; si se encuentra en el modo de *carga de datos*, entonces dichos datos son los que fueron leídos de la tarjeta anterior y almacenados en la Memoria de Entrada. Así pues, este paso (10) es tomado muy en cuenta para el diseño del Interfase.

Para el caso de introducción de datos desde el Interfase hacia el Registro de Datos, es necesario que este registro se encuentre previamente borrado (todos sus bits iguales a cero) con el fin de evitar "*choques*" o interferencias entre datos diferentes. Este borrado deberá ser efectuado artificialmente por el Interfase, ya que como se ha visto, queda cargado este registro con el contenido de la Memoria de Entrada en el paso 10.

El período en que se tiene mayor facilidad para conseguir - el borrado del Registro de Datos cuando se desea introducir nuevos datos a través del Interfase, viene a ser el sexto (TP6 en la tabla 1.5.1.). Este período contiene a los pasos 4 y 10.

Durante este período se puede lograr el mencionado borrado sin alterar significativamente la normal realización de toda la secuencia, cosa que entrañaría mucho riesgo.

Si en el paso 4 se produjera el borrado del Registro de Entrada como se produce en el paso 10, entonces se obtendría en definitiva el borrado del Registro de Datos en el paso 10, como puede deducirse de la figura 1.5.1.2.

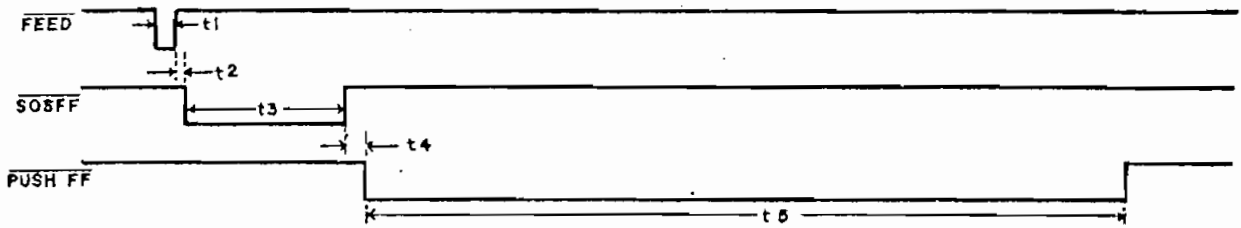
Para el caso de lectura de datos desde UNIVAC hacia el Interfase, no hace falta efectuar este borrado ni ninguna otra operación adicional.

#### 1.6. DIAGRAMA DE TIEMPOS

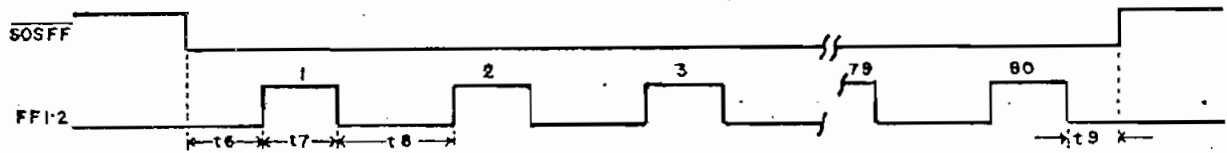
Los períodos y ciclos básicos de memoria a los que se hacía referencia en los puntos anteriores, pueden visualizarse en forma más clara en la figura 1.6.

En esa figura se tiene en la parte superior a la señal maestra de reloj de UNIVAC denominada "H CLOCK", la cual es generada por un circuito oscilador de frecuencia igual a 200 KHz. Este oscilador entrega pulsos cuadrados de 5  $\mu$ seg. de período.

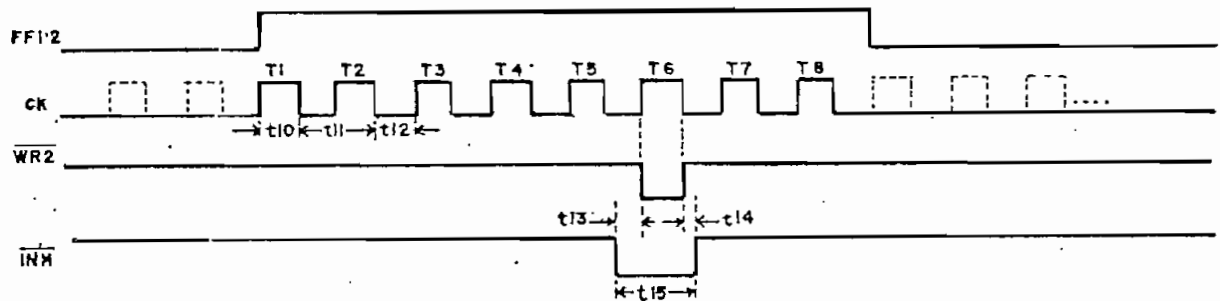
Un tren de 8 pulsos -desde  $t_0$  hasta  $t_7$ - constituye un "cá-



Inicio ( $\overline{FEED}$ ), desarrollo de la Primera Fase ( $\overline{SOSFF}$ ) y de las Fases 2-3 y 4 ( $\overline{PUSH FF}$ ) de la Secuencia Automática de Salida de Datos.



Desarrollo de la Primera Fase mostrando los Terceros Ciclos Básicos de memoria ( $\overline{FF1.2}$ )



Tercer Ciclo Básico de Memoria y escritura del dato en memoria ( $\overline{WR2}$ )

$t_1 \approx 0.5 \mu s$	$t_6 = 80 \mu s$	$t_{11} = 5 \mu s$
$t_2 = 0.03 \mu s$	$t_7 = 40 \mu s$	$t_{12} = 2.5 \mu s$
$t_3 = 12.79 ms$	$t_8 = 120 \mu s$	$t_{13} = 2.2 \mu s$
$t_4 = 5 \mu s$	$t_9 = 25 \mu s$	$t_{14} = 0.3 \mu s$
$t_5 \approx 1.1 s$	$t_{10} = 2.5 \mu s$	$t_{15} = 5 \mu s$

Figuras 1.6.2. Diagrama de tiempos de ciertas señales generadas en la Secuencia Automática de Salida de Datos de UNIVAC.

### 1.7.1. ZONAS DE LA MAQUINA

Se muestran en la figura 1.7.1. En la zona 10 o "Módulo de Placas"\* es donde van a centrarse las conexiones del Interfa<sub>se</sub>.

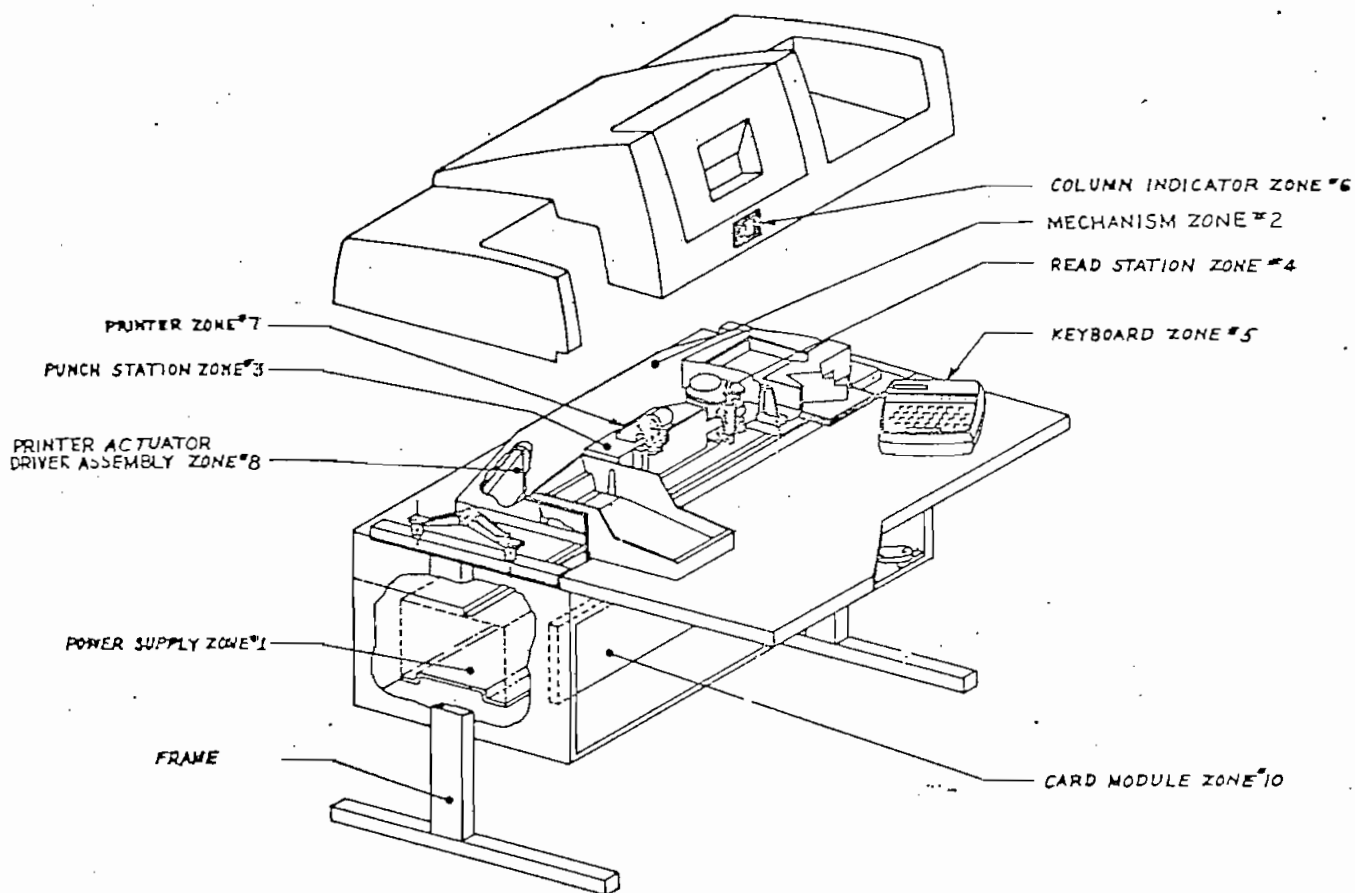


Figura 1.7.1. Localización de las zonas de la máquina UNIVAC 1710

\* Placas o cartas conteniendo a los circuitos electrónicos en este caso.

## 1.7.2. UBICACION DE LAS PLACAS

La ubicación de las placas dentro del Módulo, así como el funcionamiento específico de cada una de ellas, se muestra en la figura 1.7.2. Las placas encerradas en líneas cortadas son

1	KEYBOARD CABLE
2	UPPER: EVEN PUNCH CABLE      LOWER: KEYBOARD CABLE
3	UPPER: ODD PUNCH CABLE      LOWER: READ STATION CABLE
4	UPPER: COLUMN INDICATOR CABLE      LOWER: PRINTER CABLE
5	3616924 SHORT CARD FEATURE
6	3616906 INPUT, DATA, PUNCH REGISTERS (7-8-9)
7	3616906 INPUT, DATA, PUNCH REGISTERS (1-2-3)
8	3616906 INPUT, DATA, PUNCH REGISTERS (0-5-6)
9	3616206 INPUT, DATA, PUNCH REGISTERS (12-11-4)
10	5030394 DATA PATH CONTROLS
11	3616502 MEMORY TIMING AND DRIVE
12	3616501 MEMORY SENSE AND INHIBIT (9-16)
13	3616500 MEMORY STORAGE MODULE
14	3616501 MEMORY SENSE AND INHIBIT (1-8)
15	5030368 RIGHT JUSTIFY EXSP. SGL COL
16	5030588 OPERATIONAL CONTROLS
17	5030826 INTERSPERSED FEATURE
18	5030836 RIGHT JUSTIFY GATES
19	5030379 DATA ENTRY CONTROLS
20	5032152 VERIFICATION ERROR CTRL. CLICK
21	5030821 KB SPT. READ SPT. START O. S. C.
22	5030370 PROGRAM CONTROL
23	5030324 TIMING
24	3618174 PTR. SPT PRT RELAY ODD EVEN
25	3618163 X& FEED, PUNCH CONTROL
26	3616905 OUPY SEQ CTR 1-4, AUX IN CTRLS
27	5030375 MEM. SEL. SHIFT, ADV. PATTERN
28	3617485 FEED, INTERLOCK, PHOTOCELLS
29	3616917 CLUTCH, BRAKE, EJECT, PADDLE
30	3618161 OUTPUT SEQUENCE COUNTER 5-7
31	3616902 UNITS ADDRESS REGISTER
32	3616903 TENS ADDRESS REGISTER
33	3616922 OUTPUT ADDRESS REGISTER
34	3616926 READ CTRL, OUPY ADRS REG CTRL
35	5032382 PRINT CONTROLS AND DELAYS (ODD)
36	5032382 PRINT CONTROLS AND DELAYS (EVEN)
37	3616921 PRINTER COMPARATOR (ODD)
38	3616921 PRINTER COMPARATOR (EVEN)
39	3616929 PTR XLATE, DATA REC O, PUN FF
40	3616930 PTR CTR, CONSTANT DATA CTRLS
41	3618162 READ CTR. INTERPRET CONTROLS
42	3618168 OUTPUT CTRLS
43	3618158 MOD CHECK ACCUMULATOR, CONTROLS
44	3618166 MOD CHECK MULTIPLICAND, CONTROLS
45	3618149 MOD CHECK MULTIPLIER, CONTROLS
46	3618152 (3616933) MOD 10 (11) CHECK DIGIT OUTPUT
47	F.E.D. DIAGNOSTIC MODULE LOCATION A
48	F.E.D. DIAGNOSTIC MODULE LOCATION B

Figura 1.7.2. Ubicación de las Tarjetas en el "Bus".



opcionales, esto es, pueden ser insertadas o no en el "bus" del Módulo y cumplen funciones especiales.

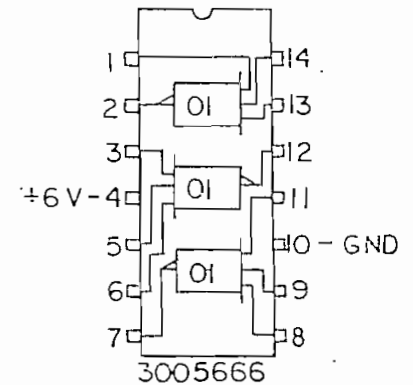
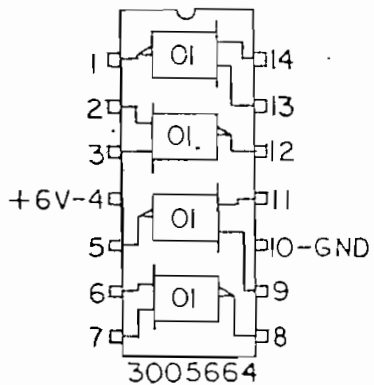
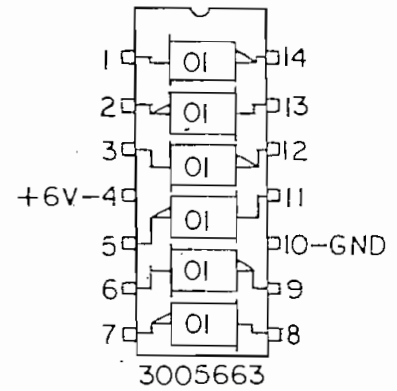
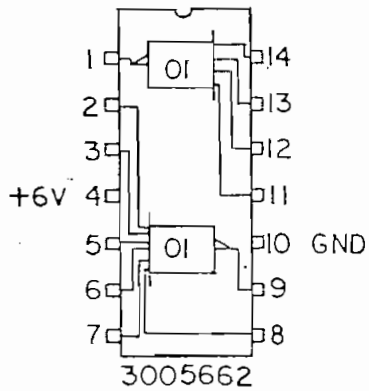
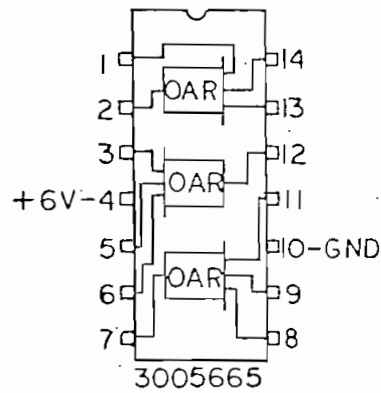
### 1.7.3. CLASES DE ELEMENTOS

En las placas se tienen circuitos discretos e integrados. Los circuitos integrados son casi en su totalidad *de la primera generación*, es decir, son construidos en base a la tecnología RTL (Resistor Transistor Logic).

Los seis tipos fundamentales de "chips"\* que existen en UNIVAC se muestran en la figura 1.7.3.1. Nótese los pines de alimentación  $V_{CC}$  y GND; así también el valor de  $V_{CC} = +6$  V.

---

\* "chips"., I.C., o circuitos integrados, son palabras que se usan indistintamente.



NOTE:  
1. ALL SCHEMATICS ARE SHOWN TOP VIEW.

Figura 1.7.3.1. Clases de circuitos integrados utilizados en UNIVAC.

Los bloques nominados "OI" que se aprecian en dicha figura, tienen su equivalencia en una compuerta NOR - ver figuras 1.7.3.2 y 1.7.3.3.

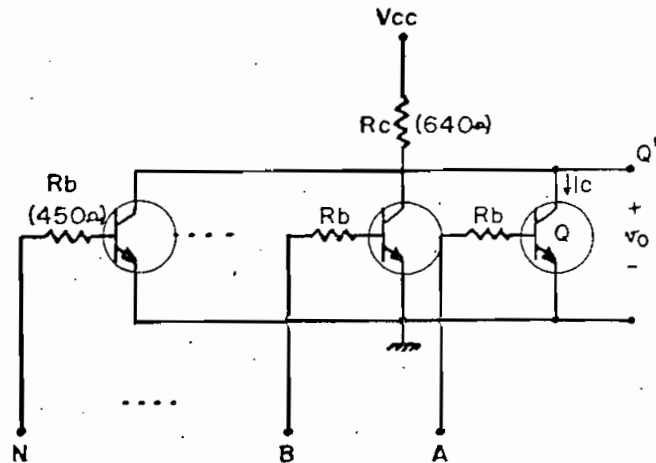


Figura 1.7.3.2. Circuito equivalente al bloque OI

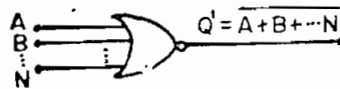


Figura 1.7.3.3. Compuerta NOR equivalente al circuito anterior.

#### 1.7.4. TECNOLOGIA RTL Y NIVELES LOGICOS

Como se ha visto, los chips de UNIVAC giran en torno al transistor y la resistencia.

Se han tratado de obtener en forma experimental los valores

numéricos del circuito equivalente al bloque OI de la figura 1.7.3.1. Llegándose a la conclusión -ya expuesta- de que se trata de un circuito perteneciente a la tecnología RTL y con valores de resistencias aproximados a los que se encuentran entre paréntesis en la figura 1.7.3.2.

El valor de los niveles lógicos teóricamente es:

$$1 \text{ lóg.} = +.6 \text{ V}$$

$$0 \text{ lóg.} = 0 \text{ V}$$

Los valores de corrientes se mantienen relativamente altos en comparación con los de la tecnología TTL utilizada más comúnmente.

De la figura 1.7.3.2., para Q en saturación - ON - :

$$I_C = \frac{V_{CC} - V_{ce}}{R_C}$$

$$I_C \approx \frac{6V - 0.15V}{640}$$

$$I_C \approx 9.1 \text{ mA}$$

para Q abierto - OFF - :  $I_C \approx 0 \text{ mA}$

De algunas mediciones realizadas dentro de las placas en ple

no funcionamiento, se han sacado en promedio los siguientes resultados:

$$5.7V \leq V_{OH} \leq 5.9V$$

$$7 \text{ mA} \leq I_{OH} \leq 12 \text{ mA}$$

$$0.1V \leq V_{OL} \leq 0.2V$$

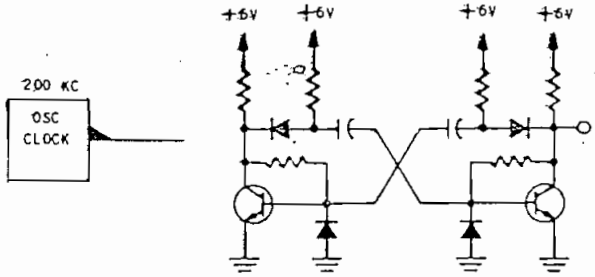
$$--- \leq I_{OL} \leq ---$$

Las diferentes funciones aritmético-lógicas que efectúan las compuertas AND, OR, NAND; las funciones de los flip-flops, registros, etc., son realizadas por circuitos construidos sóla mente en base a los chips mencionados, conectándose a ellos elementos discretos externos como son resistencias, diodos y condensadores.

Puede anotarse que, utilizando la tecnología TTL por ejemplo, podría reducirse considerablemente el número de placas, el consumo de potencia y además, simplificarse mucho su construcción.

Los circuitos que conforman al "oscilador-reloj maestro", al "oscilador secundario para duplicación automática de columnas" y a los "inversores con tiempos de retardo" largo y corto, se dan a continuación.

CLOCK OSCILLATOR



A FREE RUNNING MULTIVIBRATOR WHOSE OUTPUT ① IS ALWAYS OSCILLATING WHEN MACHINE POWER IS ON.

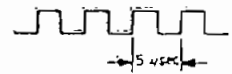
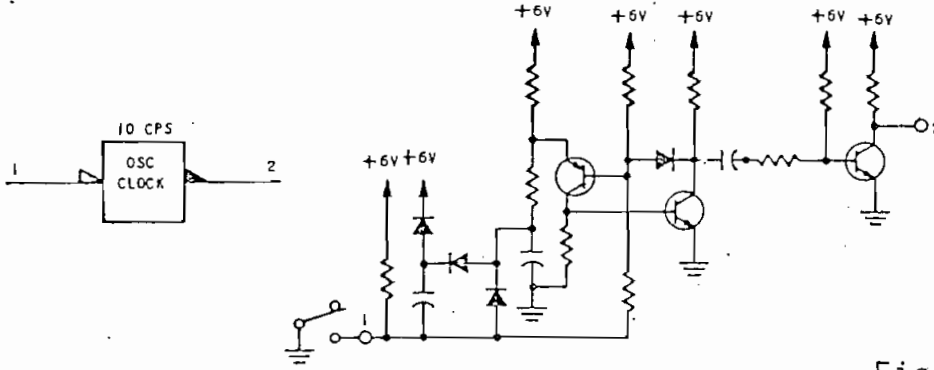


Figura 1.7.4.1.

COLUMN DUPLICATE OSCILLATOR



A SWITCH CONTROLLED LOW FREQUENCY OSCILLATOR CAPABLE OF EMITTING A SINGLE SP- PULSE OR A STRING OF HIGH PULSES AT ②

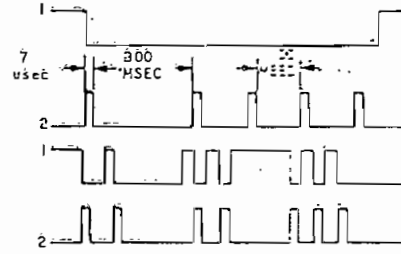
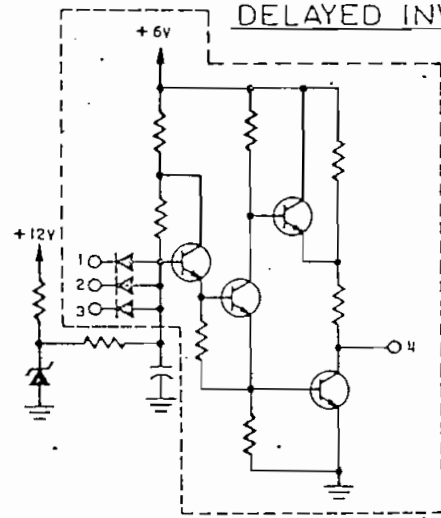
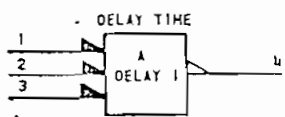


Figura 1.7.4.2.

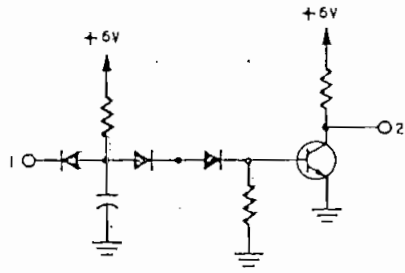
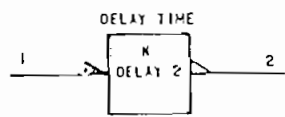
DELAYED INVERTERS



OUTPUT ④ IS LOW ONLY WHEN ALL INPUTS ARE HIGH. THE OUTPUT IS DELAYED BY THE SPECIFIED PERIOD WHEN GOING LOW.

INSIDE OF I.C. PACK

Figura 1.7.4.3.



INPUT SIGNAL IS INVERTED. THE OUTPUT IS DELAYED WHEN GOING LOW.

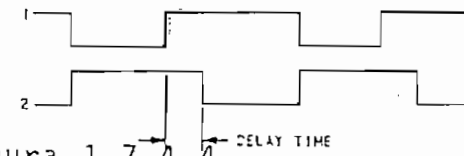


Figura 1.7.4.4.

1.7.5. FLIP - FLOPS

Como se aprecia en las figuras 1.7.5.1-4, los flip - flops son construídos a partir de los chips mencionados, con algunos diodos externos conectados a sus pines.

El flip-flop FF AI/OI de la figura 1.7.5.2., es utilizado para formar los Registros de Datos y de Entrada.

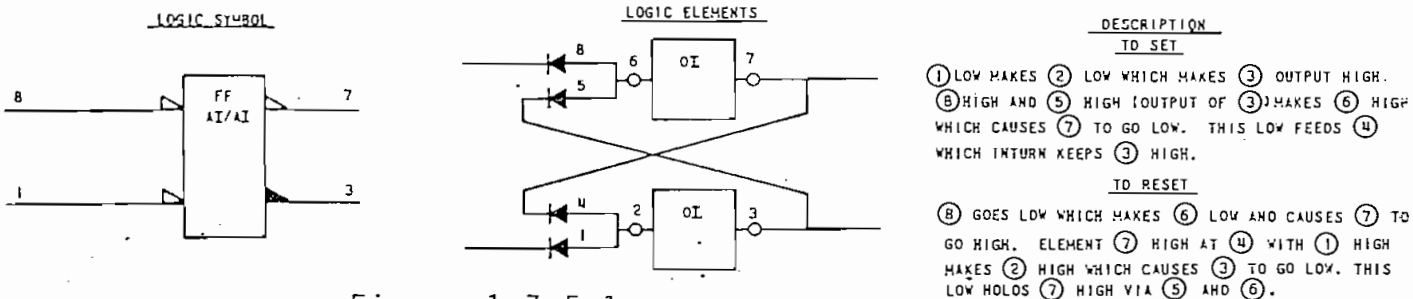


Figura 1.7.5.1.

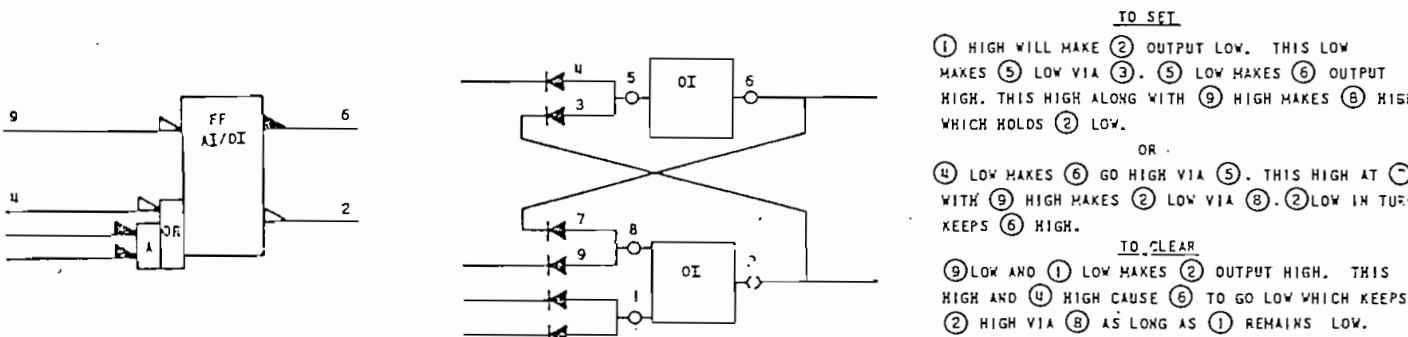


Figura 1.7.5.2.

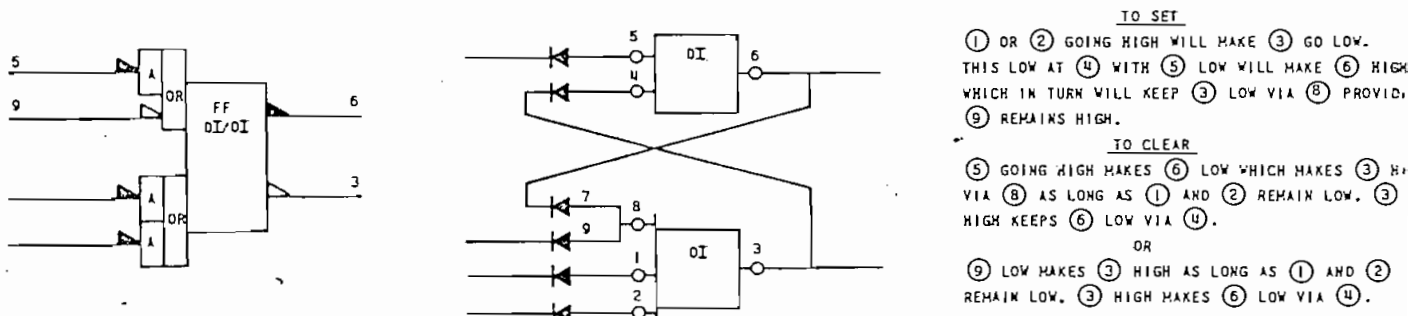
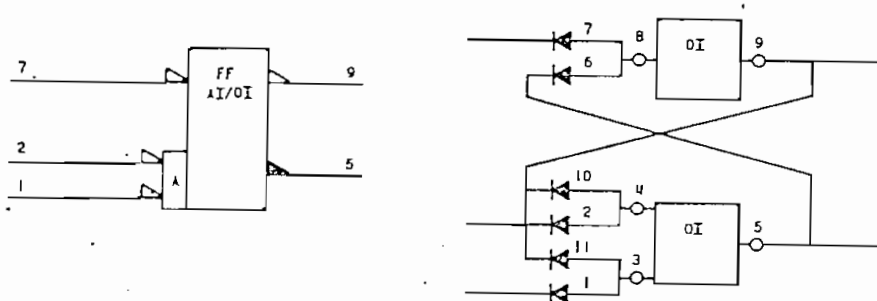


Figura 1.7.5.3.

FLIP-FLOPS de UNIVAC y su función lógica



TO SET

① AND ② GOING LOW MAKE ⑤ GO HIGH VIA ④ AND ⑩. THIS HIGH ON ⑥ WITH ⑦ ALSO HIGH MAKE ⑨ LOW VIA ⑧. ⑨ LOW FEEDS BOTH ⑩ AND ⑪. THESE IN TURN KEEP ⑤ HIGH VIA ③ AND ④.

TO CLEAR

⑦ GOING LOW MAKES ⑨ HIGH VIA ⑧. THIS HIGH FEEDS BOTH ⑩ AND ⑪. AS LONG AS EITHER ① OR ② REMAIN HIGH ⑤ WILL GO LOW AND KEEP ⑨ HIGH.

Figura 1.7.5.4. FLIP-FLOPS de UNIVAC y su función lógica

Se explica a continuación su funcionamiento con más detalle. Para ello se ha dibujado un circuito equivalente al flip-flop, que utiliza compuertas AND, NAND y NOR.

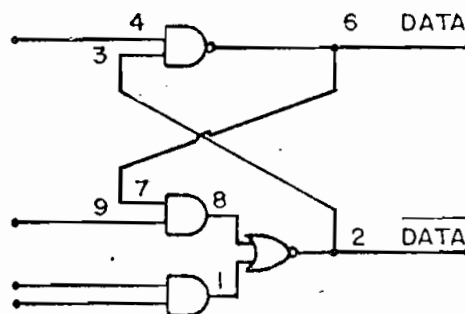


Figura 1.7.5.5. Circuito equivalente al flip-flop de la figura 1.7.5.2.

a) Para poner en DATA un 1 lógico:

- Poner en las líneas 1 y 9 un 1 lógico; cualquier transi



ción posterior no afectará el estado del flip-flop.

- Otra forma es poner en 9 un 1 lógico y efectuar una transición positiva en 4.

Si pasa 9 desde 1 lógico a 0 lógico cuando 1 está en bajo, puede cambiar el estado del flip-flop.

b) Para poner en DATA un 0 lógico:

- Poner la línea 1 en 0 lógico, 4 en 1 lógico y efectuar una transición positiva en 9.

Cambia el estado del flip-flop con el cambio de cualquiera de las líneas 1 o 4.

En resumen, para "*cargar artificialmente*" un dato en el flip-flop, debe observarse lo siguiente:

Si se desea poner en DATA un 1 lógico, puede conseguirse ésto simplemente haciendo que la línea 2 alcance un nivel de voltage correspondiente a 0 lógico; para cualquier estado previo del flip-flop.

En cambio, para poner en DATA un 0 lógico, deberán cumplirse ciertas condiciones previas en algunas líneas del flip-flop,

lo cual complica esta operación.

Por ello es conveniente poner s6lamente los datos correspondientes a 1 l6gico, borrando previamente el contenido del flip-flop, para evitar el caso de que 6ste posea un 1 l6gico en .DATA sin que se haya escrito ning6n dato desde el exterior.

#### 1.7.6. REPRESENTACION EN LAS LAMINAS DEL MANUAL

Todos los elementos existentes en UNIVAC son perfectamente identificados en las l6minas del Manual de Servicio. Esta identificaci6n incluye, como se encuentra indicado en las figuras 1.7.6.1-3, los siguientes aspectos:

- tipo de elemento; dado por el grupo de letras may6sculas - que est6n en la parte superior del bloque;
- n6mero de la l6mina en que se halla dibujado; que son los 3 n6meros de la izquierda ubicados bajo el grupo de letras mencionado anteriormente;
- n6mero con el que consta dentro de dicha l6mina; que son los 3 n6meros a continuaci6n de los anteriores;
- su ubicaci6n f6sica en la m6quina, dado por los siguientes items:

narlo con los niveles activos de las demás líneas que entran o sa  
len del elemento.

▷ Low

◀ Hi

- Los dos grupos de letras y números a la entrada de un blo  
que, señalan el pin correspondiente en la tarjeta y el pin  
del chip; de igual manera con las letras y números ubica-  
dos a la salida del bloque.

Para concluir este capítulo, cabe decir que se ha logrado a  
barcar en éste lo más útil en lo que concierne al sistema e  
léctrico y a la estructuración de UNIVAC; sin topar el aspec  
tó mecánico.

Lo escrito hasta aquí, permite proseguir con certeza en el  
logro del Interfase con UNIVAC. Toca seguidamente hacer lo  
propio con el microprocesador Z-80.

C A P I T U L O I IESTUDIO DEL MICROPROCESADOR Z-80 DESDE EL PUNTO DE VISTA DE INTERFASES

El microprocesador Z-80\* es en materia de microprocesadores, uno de los más avanzados que se han construido últimamente..

Su diseño parte del microprocesador INTEL 8080 poseyendo sin embargo sobre éste, muchas ventajas, así:

- 158 instrucciones básicas;
- frecuencia de reloj de hasta 4.5 MHz en el modelo Z-80A;
- 5 líneas dedicadas a la comunicación con sistemas periférico;
- alimentación simple de 5 V/90 mA;
- capacidad de realizar el "*refresh*" o refresco para memorias dinámicas;
- excelentes soportes en software como en hardware; etc.

---

\* En realidad se ha realizado en la práctica el Interfase con Z-80A que es la versión más veloz del Z-80 aunque en todo lo demás muy similar a este.

La capacidad que tiene Z80-CPU para aceptar interfaces es ampliada enormemente por la serie de chips que se han construído expresamente para tal fin y que se describen en el punto 2.4.

Un estudio a fondo de este microprocesador no viene al caso, mas bien se ha pretendido dar en este capítulo una somera visión de cómo se podría utilizar a Z80-CPU en el diseño y construcción de Interfaces.

En el anexo, punto A.2.3, se incluyen las 158 instrucciones básicas de Z80-CPU y otros datos de importancia. De esta forma, este capítulo se dedica únicamente al aspecto del hardware.

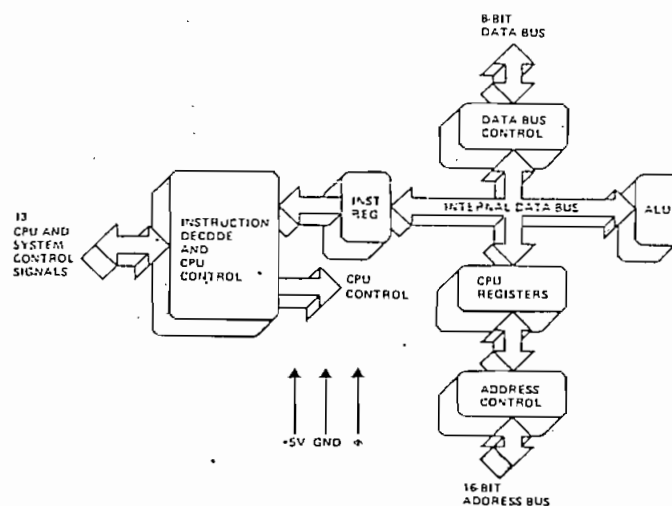


Figura 2.a. Diagrama en bloques de Z80-CPU

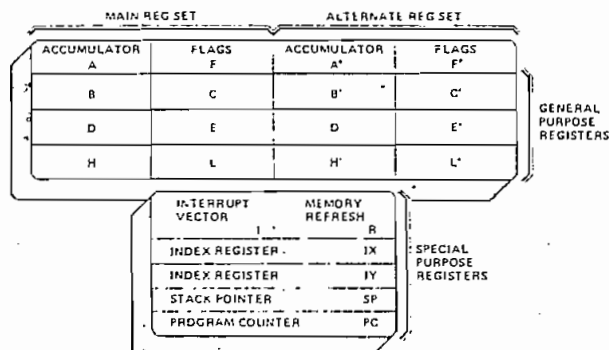


Figura 2.b. Los 22 registros de Z80-CPU

### 2.1. DESCRIPCION DE LOS PINES DE Z80-CPU

La descripción de los pines que tiene el chip Z80-CPU ofrece una idea panorámica de los dotes de este microprocesador al mismo tiempo que un conocimiento certero sobre la naturaleza de las señales que entran y salen de él, de manera que se tendría ya la información suficiente para juzgar cuáles señales podrían ser utilizadas en el Interfase y cuáles son sus características funcionales.

Los 40 pines o patas que posee el circuito integrado Z80-CPU se muestran en las figuras 2.1.1-2. En ellas se indica la notación de cada una de las 40 señales de entrada/salida de CPU así como también el número de pin que ocupan.

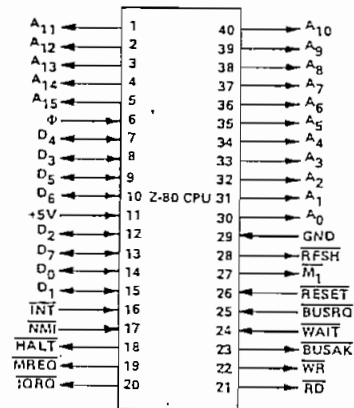


Figura 2.1.1. El circuito integrado Z80-CPU

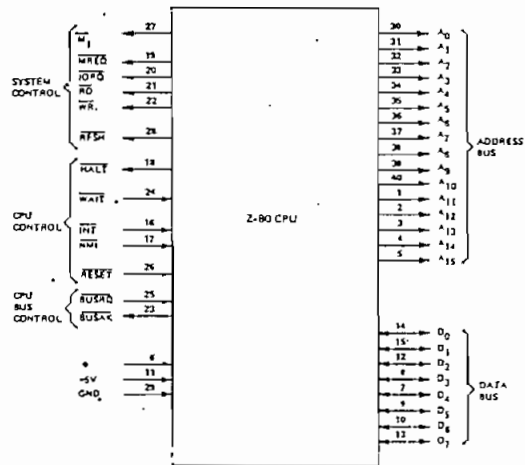


Figura 2.1.2. Pines de Z80-CPU agrupados según sus funciones

Seguidamente se explican las señales:

- A<sub>0</sub>-A<sub>15</sub> : (address Bus); son 16 líneas que constituyen la vara o bus de direccionamiento de 16 bits en paralelo con una capacidad de hasta 64Kbytes de direc

cionamiento directo de memoria.

Son líneas "tristate"\* de salida de CPU y lógica positiva - se activan en "High" (ver  $V_{OH}$  en la tabla 2.2.1.1) - .

- D0-D7 (Data Bus); son 8 líneas que constituyen la vara de datos. Sirven para el intercambio de palabras (bytes) de 8 bits con memoria y/o periféricos.

Son líneas tristate bidireccionales y de lógica positiva.

- $\overline{M1}$  (Machine Cycle one); indica que el ciclo de máquina que está tomando lugar cuando se activa -se pone en "Low" (ver  $V_{OL}$  en la tabla 2.2.1.1)- es el correspondiente al inicio de una nueva instrucción.

La señal sale de CPU y tiene lógica negativa.

- $\overline{MREQ}$  (Memory Request); indica que la vara de direccionamien

---

\* Las palabras "tristate" o "three-states" se usan para indicar que la línea o pin que sale del chip puede pasar a un estado de alta impedancia (*high Z*) que no es ni  $1_L$  ni  $0_L$ .



to posee una dirección válida para la lectura o escritura en memoria.

Es tristate, de salida de CPU y lógica negativa.

- $\overline{IORQ}$  (Input/Output Request); indica que la mitad inferior de la vara de direccionamiento contiene una dirección válida de entrada/salida para una operación de lectura o escritura en algún terminal periférico\*. Se activa también cuando una interrupción está siendo reconocida, para indicar que puede ser colocado en la vara de datos un vector de respuesta a interrupciones.

Es tristate, de salida de CPU y lógica negativa.

- $\overline{RD}$  (Memory Read); indica que CPU se dispone a leer un dato desde la memoria o desde algún periférico (esta línea debería ser utilizada para realizar la habilitación del elemento requerido).

Es tristate, de salida de CPU y lógica negativa.

- $\overline{WR}$  (Memory Write); indica que la vara de datos contiene

---

\* Pueden estar conectados hasta 256 ( $2^8$ ) terminales periféricos:

ne un dato válido para ser almacenado en memoria ó escrito en algún periférico.

Es tristate, de salida de CPU y lógica negativa...

- RFSH (Refresh); indica que los 7 bits inferiores de la vara de direccionamiento contienen una dirección de memoria que va a ser "refrescada"\* y que la señal MREQ debería ser usada para realizar una lectura de refresco a todas las memorias dinámicas.

La señal sale de CPU y tiene lógica negativa.

- HALT (Halt state); indica que CPU ha ejecutado una instrucción dada por software de "alto" ("Halt") y que se encuentra aguardando una interrupción "no enmascarada" (cuando la máscara está habilitada) antes que la operación pueda concluirse. Mientras está aguardando, CPU ejecuta ciclos de "no-operación" para mantener la actividad del *refresh* o refresco de memorias:

La señal sale de CPU y tiene lógica negativa.

---

\* esto es, los mismos datos son sobrescritos para evitar su pérdida (ocasionada en las memorias dinámicas por una degradación de sus niveles a lo largo del tiempo).

- $\overline{\text{WAIT}}$  (Wait); indica a Z80-CPU que la memoria o periférico direccionado no se encuentra listo para la transferencia de datos. CPU entonces entra a un estado de espera que durará tanto como dure la actividad de ésta y durante todo este tiempo no podrá ejecutar ninguna operación.

La señal entra en CPU y tiene lógica negativa.

- $\overline{\text{INT}}$  (Interrupt Request); esta señal es generada por algún periférico y es atendida al término de la instrucción que está siendo ejecutada si el "flip-flop de habilitación de interrupciones controlado por software" se encuentra habilitado y la señal de  $\overline{\text{BUSRQ}}$  no está activada.

La señal  $\overline{\text{INT}}$  entra en CPU y tiene lógica negativa.

- $\overline{\text{NMI}}$  (Non Maskable Interrupt); esta señal tiene mayor prioridad que  $\overline{\text{INT}}$  y es siempre reconocida al final de la instrucción en curso, independientemente del status del flip-flop de habilitación de interrupciones. Cuando esta señal ha ocurrido, Z80-CPU se ubica automáticamente en la localidad 0066H.

La señal entra en CPU y tiene lógica negativa.

- $\overline{\text{RESET}}$  (Reset); señal que inicializa a CPU de la siguiente manera:
  - vuelve a su estado original al flip-flop de habilitación de interrupciones;
  - encera al contador del programa (registro PC), así como a los registros I y R;-y
  - pone el modo de interrupción como en 8080A.

Durante el tiempo de "reset", las varas de datos y direccionamiento pasan a un estado de alta impedancia y todas las señales de control que salen de CPU pasan al estado inactivo, de suerte que otros periféricos puedan ganar el control de estas varas y señales.

La señal  $\overline{\text{RESET}}$  entra en CPU y tiene lógica negativa.

- $\overline{\text{BUSRQ}}$  (Bus Request); es usada cuando algún periférico necesita ganar el control de las varas de datos y direccionamiento así como también de las señales de control tristate de salida. Todas ellas pasan a un estado de alta impedancia cuando esta señal se ha activado.

La señal entra en CPU y tiene lógica negativa.

- BUSAK (Bus Acknowledge); es la señal de reconocimiento de CPU que sirve para indicar al periférico que ha activado la línea BUSRQ, que en ese momento se encuentran ya las varas en su estado de alta impedancia y que el periférico puede empezar a controlarlas.

La señal sale de CPU y tiene lógica negativa.

- $\phi$  (Clock); es la señal de reloj que debe ser introducida a CPU - hasta 4.5 MHz para Z-80A -.
- +5V, GND son las señales de alimentación del chip.

Hasta aquí la descripción de los 40 pines de Z80-CPU en su parte funcional. Empero, es necesario además saber sus características eléctricas y rangos máximos para sus conexiones. Esto se precisa en el siguiente punto.

## 2.2. CARACTERISTICAS ELECTRICAS

Para completar el estudio iniciado sobre los pines de Z80 - CPU, es indispensable, sobre todo si se lo enfoca desde el punto de vista de la construcción del Interfase, el poseer la necesaria seguridad en el plano eléctrico, es decir, analizar las características eléctricas, rangos mínimos, máxi

mos y típicos y, en lo posible, tolerancias y capacitancias en la carga.

Como se entenderá, este análisis hecho cabalmente podría llevar muchas páginas; pero puede también hacerse en forma concisa, agrupando a los principales parámetros con sus rangos, en cuadros que sean válidos para cualesquiera de los 40 pines que posee Z80-CPU.

En lo que se refiere a las características eléctricas que poseen los pines, éstas pueden ser agrupadas en:

- a) Características en DC o para el estado estable;
- b) características en AC o para el estado transitorio.

### 2.2.1. CARACTERISTICAS EN DC

Las características en DC para Z80-CPU se resumen en la tabla 2.2.1.1 en la que pueden observarse también los rangos mínimos, típicos y máximos.

Los diferentes parámetros voltamperimétricos de los pines de Z80-CPU resultan ser totalmente compatibles con los niveles TTL. Los pines, sin embargo, poseen un reducido "fan-out"

---

\* El "fan-out" es una característica de los IC's que da el número máximo de circuitos similares que pueden ser alimentados por una misma línea de salida.

## Absolute Maximum Ratings

Temperature Under Bias	0°C to 70°C
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

### \*Comment

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

<sup>11</sup> 250 mA para Z80A  
<sup>12</sup> 250 nseg. para Z-80A

## D.C. Characteristics

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 5\%$  unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
$V_{ILC}$	Clock Input Low Voltage	-0.3		0.45	V	
$V_{IHC}$	Clock Input High Voltage	$V_{CC}^{11}$		$V_{CC}$	V	
$V_{IL}$	Input Low Voltage	-0.3		0.8	V	
$V_{IH}$	Input High Voltage	2.0		$V_{CC}$	V	
$V_{OL}$	Output Low Voltage			0.4	V	$I_{OL} = 1.5\text{mA}$
$V_{OH}$	Output High Voltage	2.4			V	$I_{OH} = -100\mu\text{A}$ <sup>11</sup>
$I_{CC}$	Power Supply Current		90	200	mA	$t_c = 400\text{nsec}$ <sup>12</sup>
$I_{LI}$	Input Leakage Current			10	$\mu\text{A}$	$V_{IN} = 0$ to $V_{CC}$
$I_{LOH}$	Tri-State Output Leakage Current in Float			10	$\mu\text{A}$	$V_{OUT} = 2.4$ to $V_{CC}$
$I_{LOL}$	Tri-State Output Leakage Current in Float			-10	$\mu\text{A}$	$V_{OUT} = 0.4\text{V}$
$I_{LD}$	Data Bus Leakage Current in Input Mode			$\pm 10$	$\mu\text{A}$	$0 \leq V_{IN} \leq V_{CC}$

Tabla 2.2.1.1. Características en DC de Z80-CPU

siendo por ello conveniente el conectarlos a "buffers" de manera de poder conseguir el tener conectado a las varas un mayor número de cargas.

El valor máximo de capacitancia que tienen las líneas en DC se resume en la tabla 2.2.1.2.; y en la figura 2.2.1. se muestra la forma más conveniente de realizar la conexión para la señal de reloj  $\phi$  en Z80-CPU.

$T_A = 25^\circ\text{C}$ ,  $f = 1 \text{ MHz}$ ,  
unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
$C_\phi$	Clock Capacitance	20	pF
$C_{IN}$	Input Capacitance	5	pF
$C_{OUT}$	Output Capacitance	10	pF

Tabla 2.2.1.2. Capacitancia en los pines de Z80-CPU

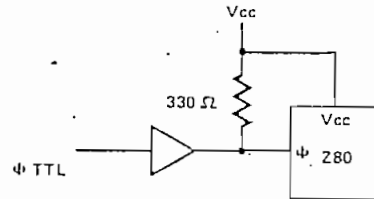


Figura 2.2.1. Circuito que cumple los requerimientos para la señal  $\phi$  en A.C. y D.C.

## 2.2.2. CARACTERÍSTICAS EN AC

En la tabla 2.2.2. constan las principales características eléctricas en AC para las 38 señales de Z80-CPU ( $V_{CC}$  y GND se incluyen en las características en DC).

La figura 2.2.2.2. complementa a la tabla, definiendo gráficamente los diversos tiempos que se hallan simbolizados.

Algunas notas adicionales sobre las características en AC:



1. Los datos deberían ser habilitados hacia el bus de datos de CPU el momento en que  $\overline{RD}$  se active. Durante el reconocimiento de una interrupción, los datos deberían ser habilitados cuando  $\overline{M1}$  y  $\overline{IORQ}$  se encuentren ambos activados.
2. Todas las señales de control se hallan internamente sincronizadas, así, pueden a su vez encontrarse totalmente a sincrónicas con respecto a la señal de reloj  $\phi$ .
3. La señal de  $\overline{RESET}$  deberá permanecer activa por lo menos durante 3 ciclos de reloj para lograr su fin.
4.  $t_w(\phi H)$  tiene garantizado un valor máximo de 200  $\mu$ seg. para Z-80A.
5. Las señales de control y de direccionamiento (A0-A15) tienen un retardo adicional de 30 nseg. para una carga capacitiva de 100 pf.

Con ésto se ha podido repasar prontamente sobre las características eléctricas y algo importante se ha concluído: todas esas características son totalmente compatibles con los niveles, velocidades de respuesta y lógica del sistema TTL.

T<sub>A</sub> = 0°C to 70°C, V<sub>cc</sub> = +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min.	Max.	Unit	Test Condiion
φ	t <sub>c</sub>	Clock Period	4	2	μsec	
	t <sub>w</sub> (φH)	Clock Pulse Width, Clock High	180		nsec	
	t <sub>w</sub> (φL)	Clock Pulse Width, Clock Low	180		nsec	
	t <sub>r,f</sub>	Clock Rise and Fall Time		30	nsec	
A <sub>0-15</sub>	t <sub>D</sub> (AD)	Address Output Delay		200	nsec	C <sub>L</sub> = 100pF
	t <sub>F</sub> (AD)	Delay to Float		100	nsec	
	t <sub>acm</sub>	Address Stable Prior to $\overline{\text{MREQ}}$ (Memory Cycle)	[1]		nsec	
	t <sub>aci</sub>	Address Stable Prior to $\overline{\text{TORQ}}$ , $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (I/O Cycle)	[2]		nsec	
	t <sub>ca</sub>	Address Stable From $\overline{\text{RD}}$ or $\overline{\text{WR}}$	[3]		nsec	
D <sub>0-7</sub>	t <sub>D</sub> (D)	Data Output Delay		350	nsec	C <sub>L</sub> = 100pF
	t <sub>F</sub> (D)	Delay to Float During Write Cycle		100	nsec	
	t <sub>SD</sub> (D)	Data Setup Time to Rising Edge of Clock During M1 Cycle	100		nsec	
	t <sub>SΦ</sub> (D)	Data Setup Time to Falling Edge of Clock During M2 to M5	100		nsec	
	t <sub>dcm</sub>	Data Stable Prior to $\overline{\text{WR}}$ (Memory Cycle)	[5]		nsec	
	t <sub>dci</sub>	Data Stable Prior to $\overline{\text{WR}}$ (I/O Cycle)	[6]		nsec	
	t <sub>dcf</sub>	Data Stable From $\overline{\text{WR}}$	[7]		nsec	
	t <sub>H</sub>	Any Hold Time for Setup Time	0		nsec	
$\overline{\text{MREQ}}$	t <sub>DLΦ</sub> (MR)	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ Low		130	nsec	C <sub>L</sub> = 50pF
	t <sub>DHΦ</sub> (MR)	$\overline{\text{MREQ}}$ Delay From Rising Edge of Clock, $\overline{\text{MREQ}}$ High		130	nsec	
	t <sub>DHΦ</sub> (MR)	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ High		150	nsec	
	t <sub>w</sub> (MRL)	Pulse Width, $\overline{\text{MREQ}}$ Low	[8]		nsec	
	t <sub>w</sub> (MRH)	Pulse Width, $\overline{\text{MREQ}}$ High	[9]		nsec	
$\overline{\text{TORQ}}$	t <sub>DLΦ</sub> (TR)	$\overline{\text{TORQ}}$ Delay From Rising Edge of Clock, $\overline{\text{TORQ}}$ Low		110	nsec	C <sub>L</sub> = 50pF
	t <sub>DHΦ</sub> (TR)	$\overline{\text{TORQ}}$ Delay From Falling Edge of Clock, $\overline{\text{TORQ}}$ Low		130	nsec	
	t <sub>DHΦ</sub> (TR)	$\overline{\text{TORQ}}$ Delay From Rising Edge of Clock, $\overline{\text{TORQ}}$ High		130	nsec	
	t <sub>DHΦ</sub> (TR)	$\overline{\text{TORQ}}$ Delay From Falling Edge of Clock, $\overline{\text{TORQ}}$ High		150	nsec	
	t <sub>w</sub> (TRL)	Pulse Width, $\overline{\text{TORQ}}$ Low				
$\overline{\text{RD}}$	t <sub>DLΦ</sub> (RD)	$\overline{\text{RD}}$ Delay From Rising Edge of Clock, $\overline{\text{RD}}$ Low		130	nsec	C <sub>L</sub> = 50pF
	t <sub>DHΦ</sub> (RD)	$\overline{\text{RD}}$ Delay From Falling Edge of Clock, $\overline{\text{RD}}$ Low		150	nsec	
	t <sub>DHΦ</sub> (RD)	$\overline{\text{RD}}$ Delay From Rising Edge of Clock, $\overline{\text{RD}}$ High		130	nsec	
	t <sub>DHΦ</sub> (RD)	$\overline{\text{RD}}$ Delay From Falling Edge of Clock, $\overline{\text{RD}}$ High		150	nsec	
	t <sub>w</sub> (RRL)	Pulse Width, $\overline{\text{RD}}$ Low				
$\overline{\text{WR}}$	t <sub>DLΦ</sub> (WR)	$\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ Low		110	nsec	C <sub>L</sub> = 50pF
	t <sub>DHΦ</sub> (WR)	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ Low		130	nsec	
	t <sub>DHΦ</sub> (WR)	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ High		150	nsec	
	t <sub>w</sub> (WRL)	Pulse Width, $\overline{\text{WR}}$ Low	[10]		nsec	
	t <sub>w</sub> (WRH)	Pulse Width, $\overline{\text{WR}}$ High				
$\overline{\text{MI}}$	t <sub>DL</sub> (M1)	$\overline{\text{MI}}$ Delay From Rising Edge of Clock, $\overline{\text{MI}}$ Low		160	nsec	C <sub>L</sub> = 30pF
	t <sub>DH</sub> (M1)	$\overline{\text{MI}}$ Delay From Rising Edge of Clock, $\overline{\text{MI}}$ High		220	nsec	
$\overline{\text{RFSH}}$	t <sub>DL</sub> (RF)	$\overline{\text{RFSH}}$ Delay From Rising Edge of Clock, $\overline{\text{RFSH}}$ Low		200	nsec	C <sub>L</sub> = 30pF
	t <sub>DH</sub> (RF)	$\overline{\text{RFSH}}$ Delay From Rising Edge of Clock, $\overline{\text{RFSH}}$ High		200	nsec	
$\overline{\text{WAIT}}$	t <sub>s</sub> (WT)	$\overline{\text{WAIT}}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{\text{HALT}}$	t <sub>D</sub> (HT)	$\overline{\text{HALT}}$ Delay Time From Falling Edge of Clock		240	nsec	C <sub>L</sub> = 50pF
$\overline{\text{INT}}$	t <sub>s</sub> (IT)	$\overline{\text{INT}}$ Setup Time to Rising Edge of Clock	70		nsec	
$\overline{\text{NMI}}$	t <sub>w</sub> (NML)	Pulse Width, $\overline{\text{NMI}}$ Low	60		nsec	
$\overline{\text{BUSRQ}}$	t <sub>s</sub> (BQ)	$\overline{\text{BUSRQ}}$ Setup Time to Rising Edge of Clock	70		nsec	
$\overline{\text{BUSAK}}$	t <sub>DL</sub> (BA)	$\overline{\text{BUSAK}}$ Delay From Rising Edge of Clock, $\overline{\text{BUSAK}}$ Low		150	nsec	C <sub>L</sub> = 50pF
	t <sub>DH</sub> (BA)	$\overline{\text{BUSAK}}$ Delay From Falling Edge of Clock, $\overline{\text{BUSAK}}$ High		150	nsec	
$\overline{\text{RESET}}$	t <sub>s</sub> (RS)	$\overline{\text{RESET}}$ Setup Time to Rising Edge of Clock	70		nsec	
	t <sub>F</sub> (C)	Delay to Float ( $\overline{\text{MREQ}}$ , $\overline{\text{TORQ}}$ , $\overline{\text{RD}}$ and $\overline{\text{WR}}$ )		100	nsec	

- [1] t<sub>acm</sub> = t<sub>w</sub>(φH) + t<sub>r</sub> - 120
- [2] t<sub>aci</sub> = t<sub>c</sub> - 40
- [3] t<sub>ca</sub> = t<sub>w</sub>(φL) + t<sub>r</sub> - 80
- [4] t<sub>caf</sub> = t<sub>w</sub>(φL) + t<sub>r</sub> - 100
- [5] t<sub>dcm</sub> = t<sub>c</sub> - 300
- [6] t<sub>dci</sub> = t<sub>w</sub>(φL) + t<sub>r</sub> - 320
- [7] t<sub>dcf</sub> = t<sub>w</sub>(φL) + t<sub>r</sub> - 80
- [8] t<sub>w</sub>(MRL) = t<sub>c</sub> - 80
- [9] t<sub>w</sub>(MRH) = t<sub>w</sub>(φH) - t<sub>r</sub>
- [10] t<sub>w</sub>(WRL) = t<sub>c</sub> - 80

Tabla 2.2.2. Características A.C. para Z80-CPU

Timing measurements are made at the following voltages, unless otherwise specified:

-1"	-0"
CLOCK	4.2 V    3 V
OUTPUT	2.0 V    3 V
INPUT	2.0 V    3 V
FLOAT	3 V      1.0.5 V

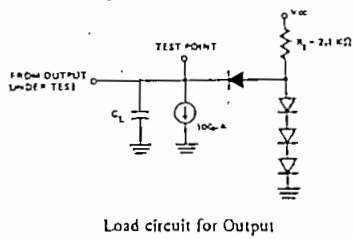


Figura 2.2.2.1 Condición de carga para las mediciones.

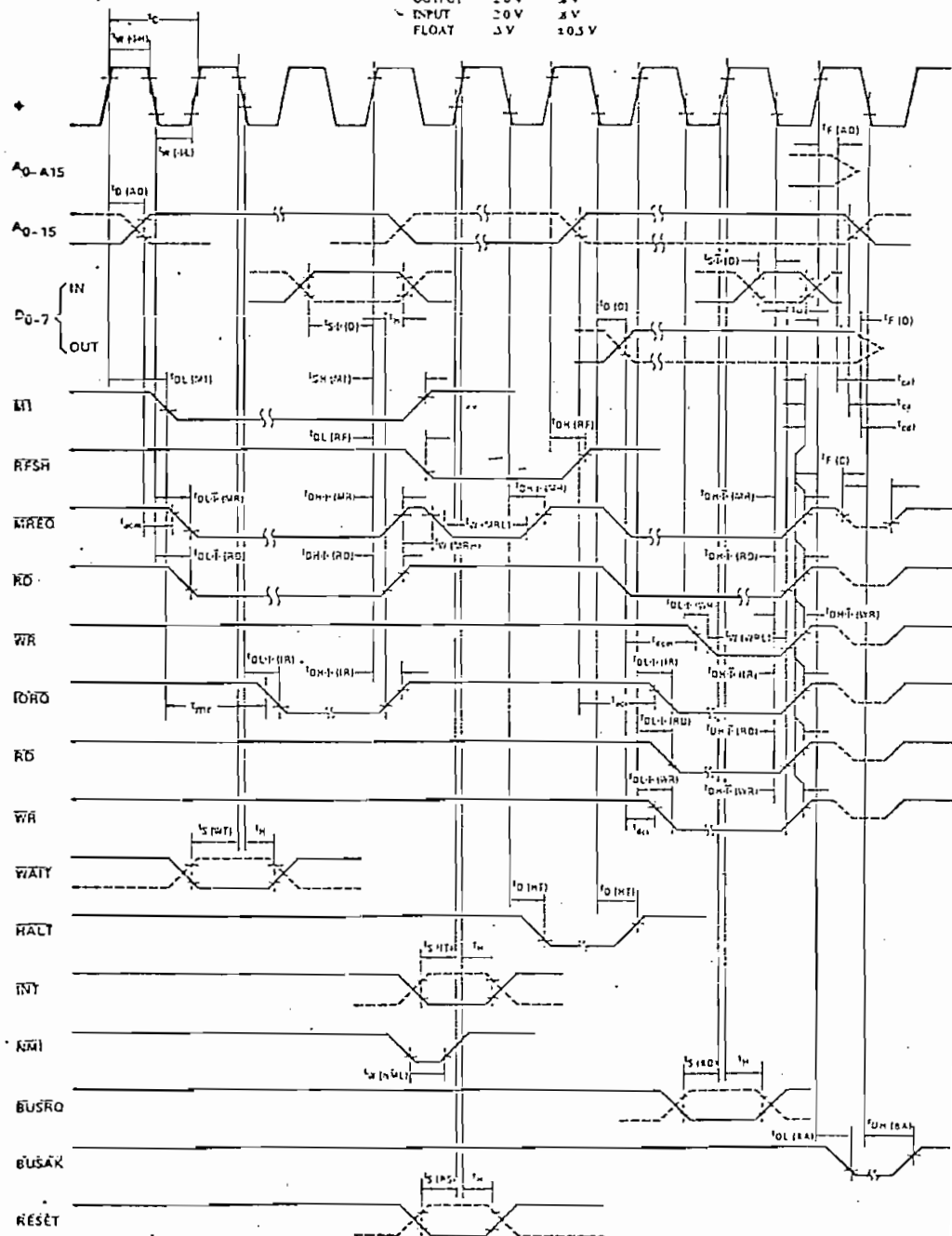


Figura 2.2.2.2 Diagrama de tiempos para A.C. que muestra gráficamente los tiempos a los que se refiere la tabla anterior.

### 2.3. DIAGRAMAS DE TIEMPO PARA LOS CICLOS DE ENTRADA Y SALIDA DE DATOS

Para saber como varían las señales durante las operaciones de entrada y salida de datos, se exponen los siguientes diagramas de tiempo:

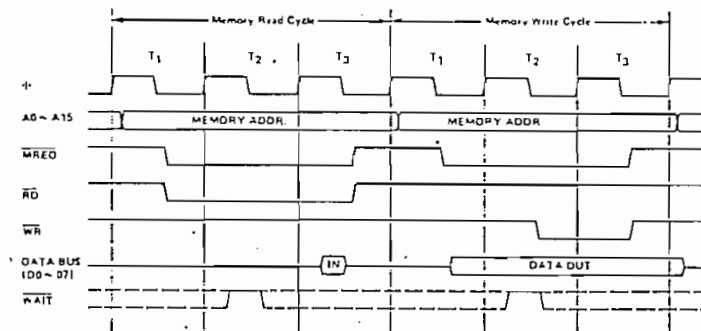


Figura 2.3.1. Diagrama de tiempos para los ciclos de lectura/escritura de datos de/en memoria.

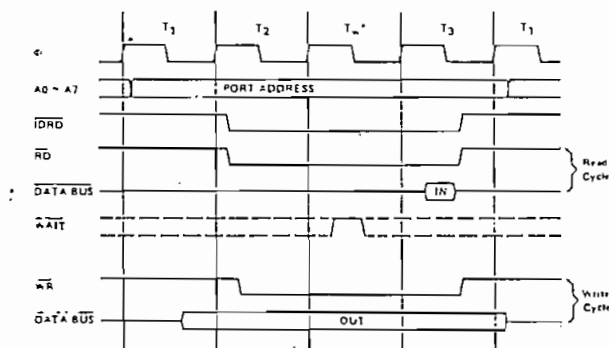


Figura 2.3.2. Diagrama de tiempos para los ciclos de entrada/salida de datos desde/hacia "puertas" o terminales periféricos.

Este último diagrama explica la forma como los datos provenientes de alguna "puerta"\* pueden ser leídos por CPU o escritos en ella. Ambas operaciones, "Read" y "Write", han sido graficadas en el mismo diagrama.

En la operación de entrada de datos desde una puerta, se tiene que apenas iniciado el primer período de este ciclo, CPU direcciona a la puerta. Luego, al empezar el segundo período, activa las señales  $\overline{IORQ}$  y  $\overline{RD}$  que indican a la puerta que debe escribir ya un byte en la vara de datos. Se tienen a continuación cinco semiperíodos para que los datos sean escritos y puedan estar ya estables en la vara.

Las señales  $\overline{IORQ}$  y  $\overline{RD}$  regresan a su estado inactivo en la mitad del último período (T3) escribiéndose en CPU el dato que se encontraba en la vara un instante antes.

Nótese que un estado simple de espera ( $T_w$ ) es insertado automáticamente (en algunos sistemas con Z80-CPU son opcionales 3, 2, 1 o ningún  $T_w$  adicional). Con esto se desea dar suficiente tiempo a la puerta para que decodifique su dirección y active la línea  $\overline{WAIT}$  si requiere todavía de un mayor tiempo para responder.

---

\* Como "puerta" o "PORT" se comprende cualquier sistema externo que puede conectarse directamente a CPU. En este sentido la palabra "periferico" es utilizada para referirse a cualquier sistema externo que para acoplarse a CPU requiere de un Interfase previo.

En la operación de salida de datos hacia una puerta, la puerta es direccionada apenas empezado el primer período de este ciclo. En el siguiente semiperíodo CPU escribe un dato en la vara y aproximadamente un semiperíodo más tarde se activan  $\overline{IORQ}$  y  $\overline{WR}$  permaneciendo así por unos cinco semiperíodos al final de los cuales se desactivan y desaparece el dato después de un semiperíodo.

La señal  $\overline{WAIT}$  también es utilizada para prolongar la duración de esta operación. Las demás señales de CPU no intervienen directamente en la realización de las dos operaciones anteriores.

#### 2.4. SOPORTES EN HARDWARE PARA Z80-CPU

Es oportuno tener ya un conocimiento, aunque sea general, de algunos circuitos integrados fabricados para trabajar con Z80-CPU y conseguir su interfase con sistemas periféricos.

A continuación se enumeran algunos de estos chips que son fáciles de encontrar:

- Z80(A)-PIO : utilizado para realizar el Interfase con casi cualquier terminal periférico standard, a base de dos "puertas" de 8 bits en paralelo.
- Z80(A)-SIO : para el Interfase con hasta dos canales de co

municación serial de datos en forma asincrónica o no.

- Z80(A)-CTC: temporizador-contador cuádruple, para eventos en los que se requiera de diversas señales de reloj.
- Z80(A)-DMA: para lograr el acceso directo a memoria por intermedio de dos puertas de 8 bits cada una.
- MK3886 : temporizador-RAM (memoria para entrada / salida de datos desde/hacia sistemas periféricos).
- Z80(A)-DART: chip programable con control de "modem" para dos canales seriales independientes.

De los seis chips anotados se estudia más detenidamente a PIO puesto que, como se habrá podido advertir, es el que más se adapta a los requerimientos del Interfase en curso.

## 2.5. EL CIRCUITO INTEGRADO PIO

Dada la necesidad que se tiene en este Interfase de transferir datos constituidos por varios bits en paralelo a una buena velocidad, se ha concluido que el elemento que serviría

mejor para ello es el circuito integrado Z80-PIO\*, por lo que se ha hecho en esta parte un estudio de este circuito.

### 2.5.1. FUNCIONAMIENTO DE PIO

PIO es un elemento que sirve de puente en la comunicación CPU-periférico.

Se tienen en PIO tres varas tristate de 8 bits, una de ellas comunica a PIO con CPU y las otras dos comunican a PIO con el periférico a través de las "puertas" o "PORTs" A y B.

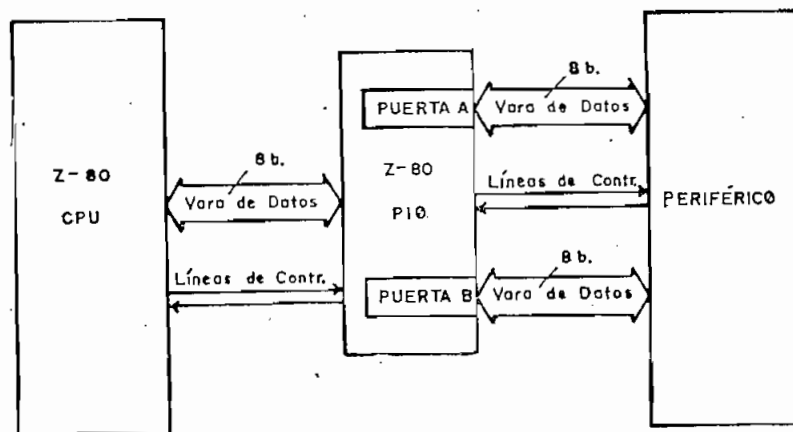


Figura 2.5.1. Esquema de la conexión CPU-PIO-Periférico

---

\* Similar a Z-80A PIO, con la única excepción de que este último puede operar a una mayor frecuencia.



Puede verse a PIO como si se tratara de cuatro localidades de memoria, dos de las cuales servirían para el intercambio de datos -puertas- y las dos restantes para determinar el modo de operación de las primeras.

Cuando se ha programado a alguna puerta como puerta de salida de datos, entonces CPU puede escribir un byte en ella como si lo escribiera en RAM. La puerta almacenará el dato y PIO indicará al periférico que se encuentra cargado. Así, el momento en que el periférico responda con una señal, saldrá el dato de la puerta hacia la vara que la une con el periférico; al mismo tiempo PIO activará una señal de interrupción dirigida hacia CPU, señal que puede ser atendida o no dependiendo de si se ha puesto o no la máscara a interrupciones.

CPU puede sobre-escribir datos en la puerta (propriadamente, en el "*Registro de Salida*" que hay dentro de ella).

Cuando se ha programado a alguna puerta como puerta de entrada de datos, entonces CPU puede leer el dato contenido en ella (propriadamente en su "*Registro de Entrada*"). Esta lectura puede efectuarse en cualquier momento y cualquier número de veces.

Por su parte, el periférico "*es informado*" de que ha sido leída la puerta por CPU y que puede introducir ya otro dato. Esto lo hará enviando una señal a PIO que habilita el Registro

de Entrada.

Estos dos modos de operación de las puertas son los principales. Estan explicados gráficamente en el punto 2.5.5.

Otra característica propia de PIO es la de poder ser programado para que envíe una señal de interrupción automáticamente a CPU en el momento en que se tenga en alguna puerta un byte particular -que ha sido previamente introducido en PIO para el efecto-. Así se consigue el Interfase con terminales sumamente rápidos.

#### 2.5.2. VENTAJAS DE PIO

La utilización de Z80-PIO implica varias ventajas, pues siendo construido expresamente para trabajar con Z80-CPU, su interconexión redundaría en la simplificación tanto del hardware como del software.

Simplificaría el software en cuanto que las instrucciones de entrada/salida de Z80-CPU son "leídas" y atendidas por PIO. Las señales de control que PIO posee permite una respuesta inmediata a estas operaciones.

Simplificaría el hardware en razón de que:

- no necesita de "drivers", "buffers" u otros elementos que hagan compatibles sus niveles eléctricos;
- los problemas de tiempos de retardo, de respuesta, etc. quedan de lado;
- el sincronismo entre los dos circuitos integrados resulta elemental al tener ambos la misma señal de reloj  $\phi$ ;
- las señales de control que requiere, corresponden justamente a algunas de las señales de control generadas en Z80-CPU\*; entonces no es necesario generar ninguna señal auxiliar por medio de circuitos combinatoriales externos.

Por otra parte, al ser PIO realmente un *microprocesador* en sí, viene a constituirse en el elemento fundamental para la realización de Interfases Interactuantes entre Z80-CPU y terminales periféricos sean éstos "torpes" o "inteligentes", sin necesidad virtualmente de lógica adicional; esto implica:

- menor costo para el Interfase;
- una casi completa adaptación de PIO al terminal deseado -

---

\* al trabajar con el bus S-100, algunas de ellas cambian de nomenclatura.

(PIO puede particularizar su modo de funcionamiento por medio de la programación previamente introducida en él cuando es inicializado);

- posibilidad de trabajar con terminales sumamente rápidos o lentos por medio del sistema de "handshake"\* que posee; y otras ventajas más.

Todas esas capacidades de PIO se comprenden mejor una vez que se conoce su estructura interna. Esta es tratada a continuación.

### 2.5.3 DESCRIPCION INTERNA

Un diagrama de bloques, que describe la arquitectura interna de PIO se muestra en la figura 2.5.3.1. Allí pueden observarse el "Bus de Datos", las 6 "Líneas de Control" de PIO que se interconectan con Z80-CPU, el "Control Lógico Interno", las "Puertas" A y B de entrada/salida, el "Control lógico para procesar las Interrupciones" y el "Bus Interno" que intercomunica a todos los anteriores.

---

\* el "handshake" o "darse la mano" es un protocolo sencillo de comunicación entre dos sistemas por el cual, el sistema más rápido espera hasta que el otro haya concluido su operación de recepción o envío de datos para reiniciar el envío o recepción de nuevos datos.

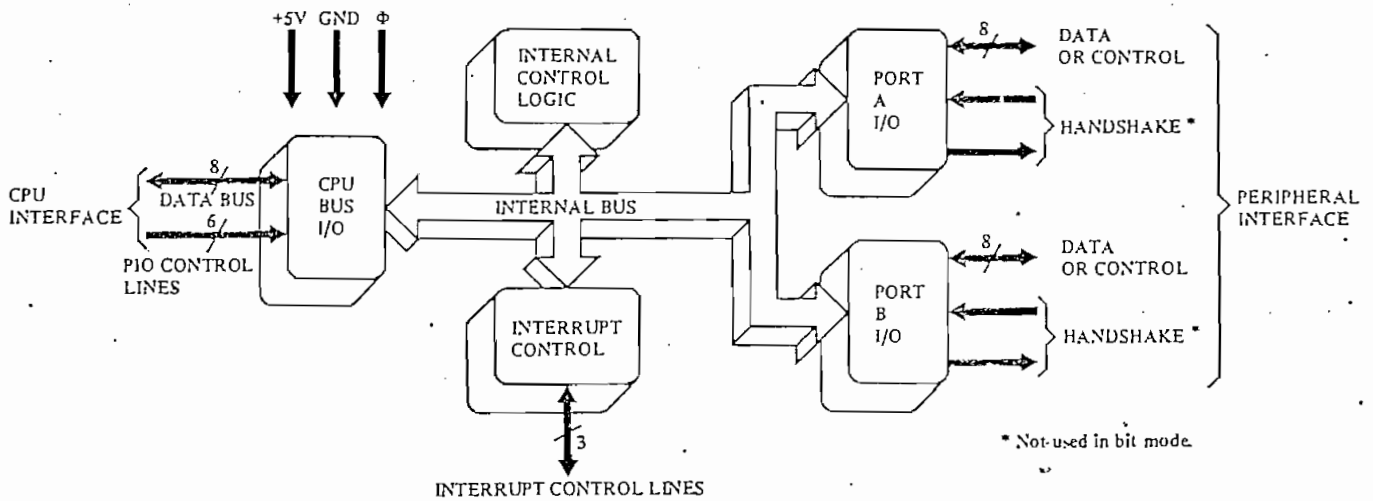


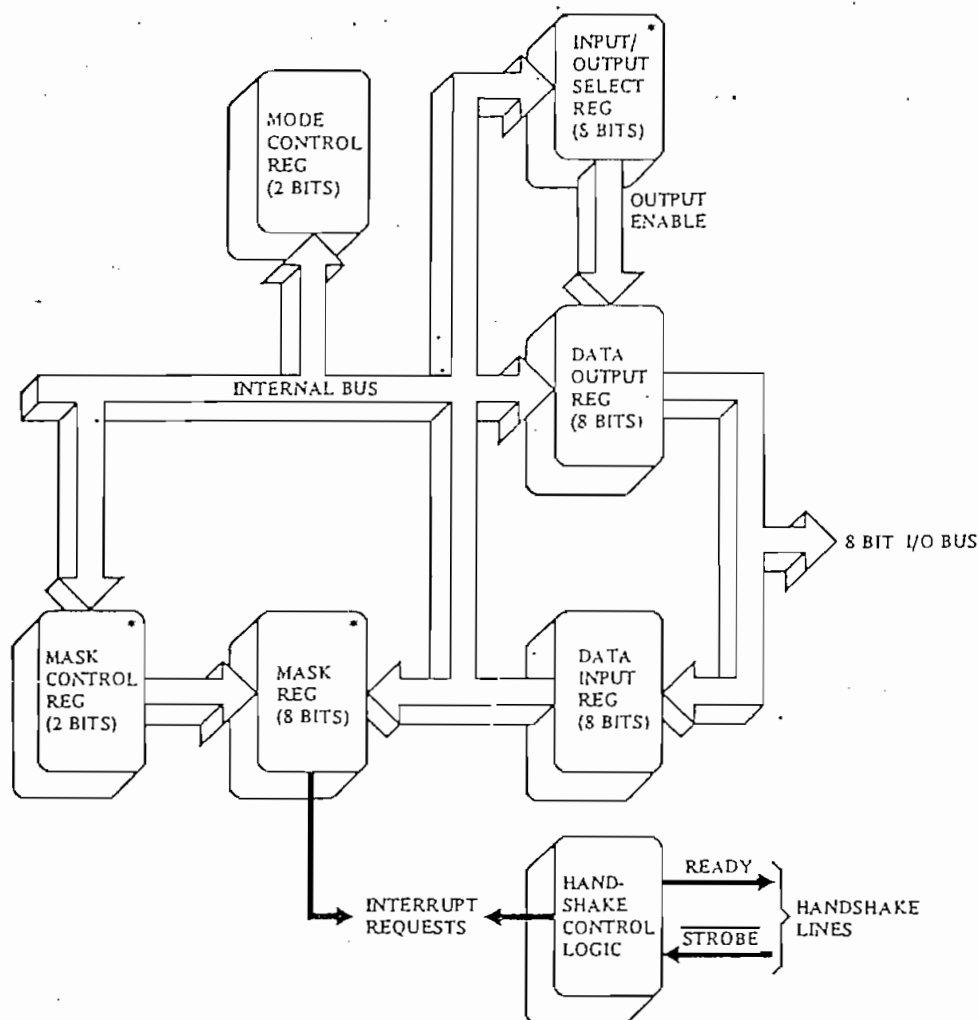
Figura 2.5.3.1. Diagrama de bloques de PIO

Cada una de las puertas se compone de seis registros con un "Control Lógico para el handshake", como se ve en la figura 2.5.3.2.

Los seis registros corresponden a:

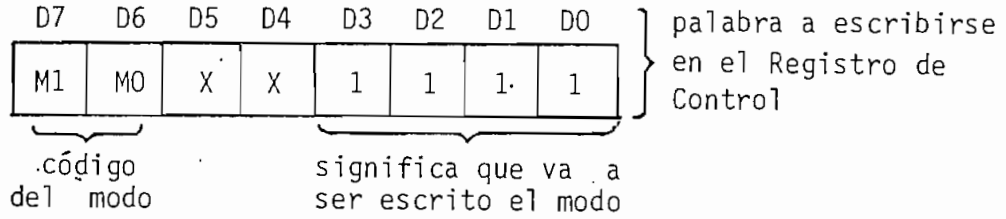
- 1) un registro de 8 bits para los datos provenientes de algún terminal y que entran por una de las puertas;
- 2) un registro de 8 bits para los datos que van a salir por la puerta hacia algún terminal periférico;
- 3) un registro de 2 bits para el control del modo de operación de la puerta; este registro es cargado desde CPU y de acuerdo a su estado lógico la puerta operará como:

- puerta de salida de datos;
- puerta de entrada de datos;
- puerta de entrada o salida de datos (bidireccionales);
- puerta de entrada y salida de datos (ciertos bits sirven para leer datos y otros para escribirlos en el periférico).



\* Used in the bit mode only to allow generation of an interrupt if the peripheral I/O pins go to the specified state.

Figura 2.5.3.2. Diagrama que muestra los registros y la configuración de una puerta.



X = no utilizado

Modo	M1	M0
de Salida	0	0
de Entrada	0	1
Bidireccional	1	0
de Bit	1	1

Tabla 2.5.3. Selección del modo de operación de la puerta.

#### 2.5.4. PINES

PIO posee, similarmente a Z80-CPU, 40 pines cuya descripción se encuentra resumida en las siguientes figuras.

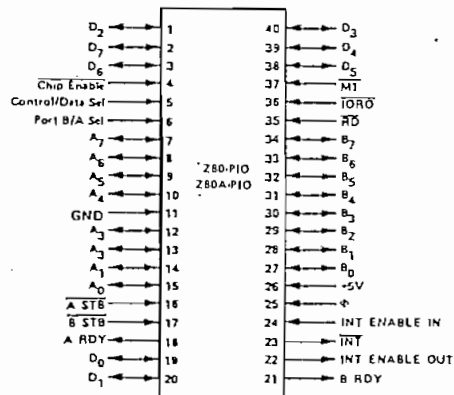
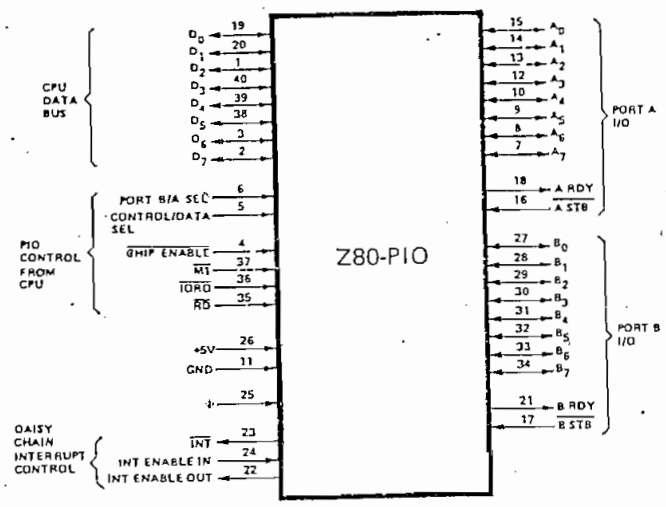


Figura 2.5.4.1. Los 40 pines de PIO

0111 1



- $D_7-D_0$  Z80-CPU Data Bus (bidirectional, tristate)
- B/A Sel Port B or A Select (input, active high)
- C/D Sel Control or Data Select (input, active high)
- $\overline{CE}$  Chip Enable (input, active low)
- $\Phi$  System Clock (input)

- $\overline{M1}$  Machine Cycle One Signal from CPU (input, active low)
- $\overline{IORQ}$  Input/Output Request from Z80-CPU (input, active low)
- $\overline{RD}$  Read Cycle Status from the Z80-CPU (input, active low)
- $\overline{IEI}$  Interrupt Enable In (input, active high)
- $\overline{IEO}$  Interrupt Enable Out (output, active high). and  $\overline{IEO}$  form a daisy chain connection for priority interrupt control.
- $\overline{INT}$  Interrupt Request (output, open drain, active low)
- $A_0-A_7$  Port A Bus (bidirectional, tristate)
- $A\ STB$  Port A Strobe Pulse from Peripheral Device (input, active low)
- $A\ RDY$  Register A Ready (output, active high)
- $B_0-B_7$  Port B Bus (bidirectional, tristate)
- $B\ STB$  Port B Strobe Pulse from Peripheral Device (input, active low)
- $B\ RDY$  Register B Ready (output, active high)

Figura 2.5.4.2. Descripción de las señales de PIO

Las señales "A RDY", "A STB", "B RDY" y "B STB" son utilizadas para el handshake.

Las señales "INT", "INT ENABLE IN" e "INT ENABLE OUT" sirven para el establecimiento de interrupciones -PORT A tiene una prioridad mayor que PORT B para la atención de alguna interrupción-.

Las señales "PORT B/A SEL.", "CONTROL/DATA SEL." y "CHIP ENA



"BLE" se utilizan para el direccionamiento de las cuatro localidades de PIO.

$\overline{\text{CHIP ENABLE}}$	PORT B/ $\overline{\text{A}}$ SEL.	CONTROL/ $\overline{\text{DATA SEL.}}$	LOCALIDAD
0	0	0	Puerta A.
0	0	1	Byte de control de la puerta A.
0	1	0	Puerta B
0	1	1	Byte de control de la puerta B.

Tabla 2.5.4. Direccionamiento de los registros de datos y de control de PIO.

### 2.5.5. DIAGRAMAS DE TIEMPO

Los diagramas de tiempo dependen del modo de operación de las puertas de PIO.

- Puerta de salida de datos:

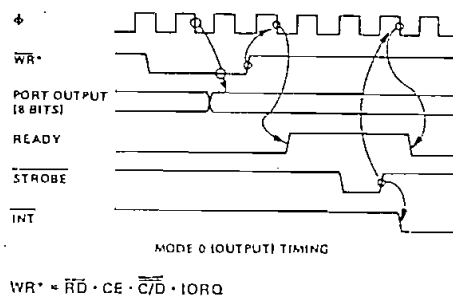


Figura 2.5.5.1. Puerta en el modo de salida

$\phi$  = reloj de CPU.

$\overline{WR}$  = señal proveniente de CPU que indica que va a ser escrito en el registro de salida de la puerta direccionada el dato contenido en el bus de datos (en realidad esta señal no entra en PIO sino que equivale a las condiciones de  $\overline{RD} = 1 \wedge \overline{CE} = 0 \wedge C/D = 0 \wedge \overline{TORQ} = 0$ ).

Una vez que  $\overline{WR}$  regresa a su estado inactivo, pueden variar los datos del bus, que no variarán los datos cargados ya en la puerta.

PORT OUTPUT = registro de salida de alguna de las puertas de PIO.

READY = señal generada por PIO que indica al periférico que la puerta se encuentra lista para enviar el dato que acaba de ser cargado.

$\overline{STROBE}$  = señal que envía el periférico para habilitar la puerta y así tener a su disposición los datos ahí contenidos (en conjunto con READY realiza el handshake).

$\overline{INT}$  = señal de interrupción que envía PIO a CPU y per

mite, si así está programado, entrar en alguna rutina de interrupción.

Las flechas y pequeños círculos en el diagrama indican la secuencia, la dependencia y el estado lógico necesario de las señales para efectivizarse las transiciones.

- Puerta de entrada de datos:

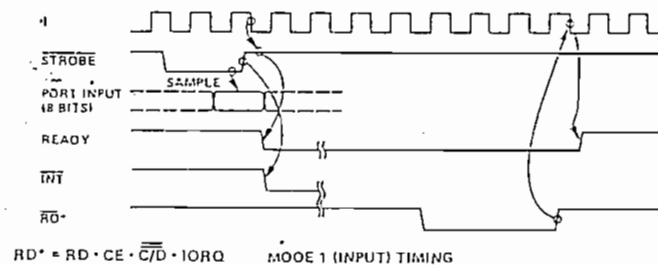


Figura 2.5.5.2. Puerta en el modo de entrada

Nótese que los datos retenidos por el registro de entrada de la puerta, corresponden a aquellos que envió el periférico - un instante antes de deshabilitarse  $\overline{STROBE}$ ; READY pasa entonces a su estado "stand-by" permaneciendo en él hasta cuando CPU ha efectuado la lectura de la puerta.

- Puerta bidireccional:

Es una combinación de los dos modos anteriores utilizando las 4 líneas para el handshake (de A y B).

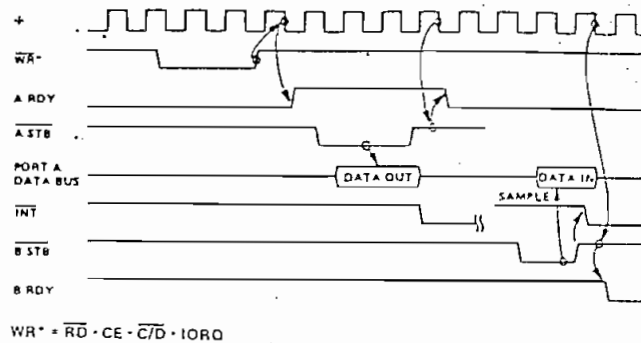


Figura 2.5.5.3. Puerta bidireccional

La puerta A es utilizada en el ejemplo como puerta bidireccional y la puerta B en el modo de Bit para efectuar el control de la operación.

- Puerta en el modo de Bit:

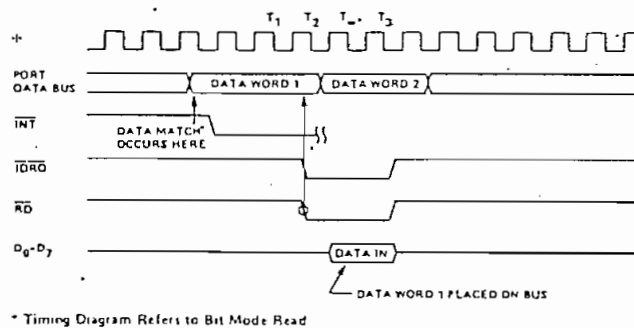


Figura 2.5.5.4. Puerta en el modo de "bit"

En este caso no se utiliza el handshake, pudiendo utilizarse las líneas de cualquier puerta para el control de la comunicación.

### 2.5.6. CARACTERÍSTICAS ELÉCTRICAS; RANGOS MÁXIMOS

Las características eléctricas de Z80-PIO con sus rangos máximos para DC y AC pueden verse en las tablas 2.5.6-1-2 y la figura 2.5.6.2.

TA = 0° C to 70° C, Vcc = 5 V ± 5% unless otherwise specified

Symbol	Parameter	Min.	Max.	Unit	Test Condition
V <sub>ILC</sub>	Clock Input Low Voltage	-0.3	.45	V	I <sub>OL</sub> = 2.0 mA I <sub>OH</sub> = -250 μA
V <sub>IHC</sub>	Clock Input High Voltage	V <sub>cc</sub> -.6	V <sub>cc</sub> +3	V	
V <sub>IL</sub>	Input Low Voltage	-0.3	0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0	V <sub>cc</sub>	V	
V <sub>OL</sub>	Output Low Voltage		0.4	V	
V <sub>OH</sub>	Output High Voltage	2.4		V	
I <sub>CC</sub>	Power Supply Current		70	mA	
I <sub>LI</sub>	Input Leakage Current		10	μA	
I <sub>LOH</sub>	Tri-State Output Leakage Current in Float		10	μA	
I <sub>LOL</sub>	Tri-State Output Leakage Current in Float		-10	μA	
I <sub>LD</sub>	Data Bus Leakage Current in Input Mode		±10	μA	V <sub>OUT</sub> = 2.4 to V <sub>cc</sub> V <sub>OUT</sub> = 0.4 V
I <sub>OH</sub>	Darlington Drive Current (PB0 - PB7)	-1.5	3.8	mA	0 ≤ V <sub>IN</sub> ≤ V <sub>cc</sub> V <sub>OH</sub> = 1.5 V R <sub>EXT</sub> = 390 Ω Port B Only

Tabla 2.5.6.1. Características D.C. de los pines de PIO

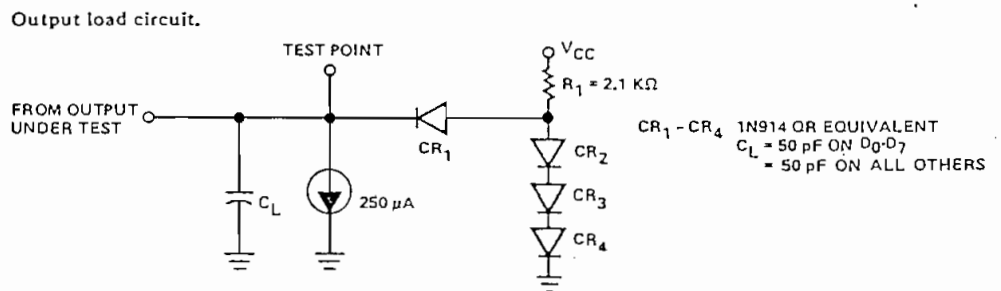


Figura 2.5.6.1. Circuito equivalente de carga a la salida.

TA = 0° C to 70° C, Vcc = +5 V ± 5%, unless otherwise noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
$\Phi$	$t_c$	Clock Period	400	[1]	nsec	
	$t_w(\Phi H)$	Clock Pulse Width, Clock High	170	2000	nsec	
	$t_w(\Phi L)$	Clock Pulse Width, Clock Low	170	2000	nsec	
	$t_r, t_f$	Clock Rise and Fall Times		30	nsec	
	$t_H$	Any Hold Time for Specified Set-Up Time	0		nsec	
CS, $\overline{CE}$ ETC.	$t_{S\Phi}(CS)$	Control Signal Set-Up Time to Rising Edge of $\Phi$ During Read or Write Cycle	280		nsec	
D <sub>0</sub> -D <sub>7</sub>	$t_{DR}(D)$	Data Output Delay from Falling Edge of $\overline{RD}$	50	430	nsec	[2]
	$t_{S\Phi}(D)$	Data Set-Up Time to Rising Edge of $\Phi$ During Write or $\overline{M1}$ Cycle			nsec	
	$t_{DI}(D)$	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle.		340	nsec	[3]
	$t_F(D)$	Delay to Floating Bus (Output Buffer Disable Time)		160	nsec	
IEI	$t_S(IEI)$	IEI Set-Up Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	$t_{DH}(IO)$	IEO Delay Time from Rising Edge of IEI		210	nsec	[5]
	$t_{DL}(IO)$	IEO Delay Time from Falling Edge of IEI		190	nsec	[5]
	$t_{DM}(IO)$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$ ) See Note A.		300	nsec	[5] C <sub>L</sub> = 50 pF
$\overline{IORQ}$	$t_{S\Phi}(IR)$	$\overline{IORQ}$ Set-Up Time to Rising Edge of $\Phi$ During Read or Write Cycle	250		nsec	
$\overline{M1}$	$t_{S\Phi}(M1)$	$\overline{M1}$ Set-Up Time to Rising Edge of $\Phi$ During INTA or $\overline{M1}$ Cycle. See Note B.	210		nsec	
$\overline{RD}$	$t_{S\Phi}(RD)$	$\overline{RD}$ Set-Up Time to Rising Edge of $\Phi$ During Read or $\overline{M1}$ Cycle	240		nsec	
A <sub>0</sub> -A <sub>7</sub> , B <sub>0</sub> -B <sub>7</sub>	$t_S(PD)$	Port Data Set-Up Time to Rising Edge of $\overline{STROBE}$ (Mode 1)	260	230	nsec	[5]
	$t_{DS}(PD)$	Port Data Output Delay from Falling Edge of $\overline{STROBE}$ (Mode 2)			nsec	
	$t_F(PD)$	Delay to Floating Port Data Bus from Rising Edge of $\overline{STROBE}$ (Mode 2)		200	nsec	C <sub>L</sub> = 50 pF
	$t_{DI}(PD)$	Port Data Stable from Rising Edge of $\overline{IORQ}$ During WR Cycle (Mode 0)		200	nsec	[5]
$\overline{ASTB}$ , $\overline{BSTB}$	$t_w(ST)$	Pulse Width, $\overline{STROBE}$	150		nsec	
			[4]		nsec	
$\overline{INT}$	$t_D(IT)$	$\overline{INT}$ Delay Time from Rising Edge of $\overline{STROBE}$		490	nsec	
	$t_D(IT3)$	$\overline{INT}$ Delay Time from Data Match During Mode 3 Operation		420	nsec	
ARDY, BRDY	$t_{DH}(RY)$	Ready Response Time from Rising Edge of $\overline{IORQ}$		$t_c+$ 460	nsec	[5]
	$t_{DL}(RY)$	Ready Response Time from Rising Edge of $\overline{STROBE}$		$t_c+$ 400	nsec	[5] C <sub>L</sub> = 50 pF

Tabla 2.5.6.2. Características A.C. de PIO

TA = 25° C, f = 1 MHz

Symbol	Parameter	Max.	Unit	Test Condition
C <sub>Φ</sub>	Clock Capacitance	10	pF	Unmeasured Pins Returned to Ground
C <sub>IN</sub>	Input Capacitance	5	pF	
C <sub>OUT</sub>	Output Capacitance	10	pF	

Tabla 2.5.6.3. Capacitancia en los pines de PIO

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	$V_{CC} - 6$	.45V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	$\Delta V$	$-0.5V$

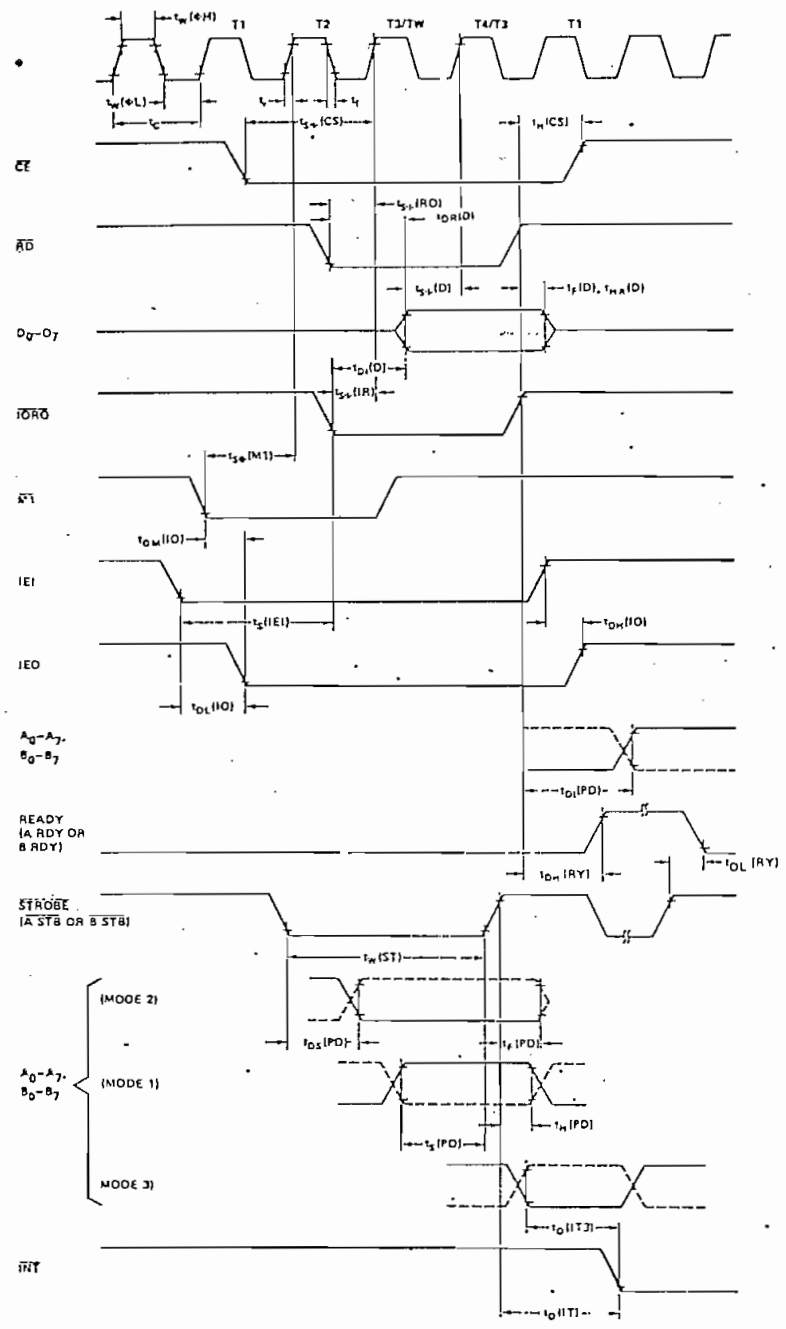


Figura 2.5.6.2. Diagrama de tiempos a los que se refiere la tabla anterior.

Se concluye del análisis de las tablas, que las características son similares a las dadas para Z80-CPU.

Con ésto se ha realizado el estudio de Z80-CPU y Z80-PIO en una forma breve. El continuar en él y profundizarlo requeriría el traer a colación ejemplos de aplicación práctica para un gran número de casos que puedan darse, ejemplos que, pese a ser muy valiosos para una comprensión mejor del asunto, no se los ha incluido porque irían a extender demasiado el capítulo; sin embargo, en la construcción del Interfase - Capítulo III -, se espera el saber aprovechar de las capacidades - vistas de ambos integrados.

El estudio (enminentemente bibliográfico) de este capítulo, ha estado apuntado a lo que concierne con la comunicación microprocesador - sistema periférico; y dentro de ello, al hardware.

El software toma importancia el momento en que se conocen ya la aplicación específica del Interfase y el hardware en el que se desarrollará. Estos dos aspectos, la configuración - del hardware y la dirección que requerirá el software, se podrán hallar en el capítulo siguiente.



C A P I T U L O    I I II N T E R F A S E3.1. DIVERSAS POSIBILIDADES PARA EL INTERFASE

El estudio llevado en los capítulos anteriores hace que se puedan ya visualizar algunas formas de realizar el Interfase entre los dos sistemas Z80-CPU y UNIVAC para lograr una comunicación bidireccional de datos.

Algunos factores que se toman en cuenta para el diseño son:

- Factor económico; que implica optimización de costos en su construcción.
- Velocidad de envío de datos; que para el presente caso es relativamente baja, lo cual permite obviar problemas propios de las líneas de transmisión como son irradiaciones, atenuaciones, ruido por reflexiones en guías de onda, etc.
- Tiempo efectivo de ocupación de CPU; el cual debe minimizar se, esto es, procurar no interferir en los procesos normales que realice CPU.
- Simplificación; procurando mantener sencillez y claridad en

el diseño, a sabiendas de que esto significa una mayor garantía de su funcionamiento y también mayor facilidad para la implementación de eventuales cambios y mejoras.

Así pues, tomando en cuenta estos factores y la índole del trabajo, se proponen algunas posibilidades:

a) La forma talvéz más obvia de realizar el Interfase, es teniendo un acceso directo a la memoria de UNIVAC y así poder escribir y leer datos de ella. Esta forma, sin embargo, presenta dos grandes dificultades:

1) La habilitación de las memorias de Entrada y de Salida se realiza automáticamente bajo determinadas condiciones de los Registros de Control y en sincronismo con la señal de reloj de UNIVAC, como se ha anotado en el punto 1.3; y,

2) Esta forma de acceso *asíncronico* puede alterar el funcionamiento secuencial electro-mecánico de UNIVAC.

Por estas razones ha sido relegada esta posibilidad.

b) Una posible forma de introducir datos a la Memoria de Entrada, puede ser "simulando" la digitación de teclas del "Keyboard" a base de switches comandados desde el micro-

computador. Esto requiere, sin embargo, de un bien elaborado circuito combinacional que pueda reproducir la "Matriz de Datos del Keyboard" - figura 3.1.b -, junto con algunas señales indicadoras dirigidas hacia flip-flops de Control, que son las que se indican en el período asíncrono de la operación de "Key-In" para el modo de Perforación en la tabla 3.1.

Aún más, esta forma de introducción de datos requeriría - por lo menos un tiempo de espera entre datos de más de 12.3 mseg., cosa que sí constituye una seria limitación.

- c) Otra forma de realizar el Interfase puede ser aprovechando la detección de datos desde la estación lectora de UNIVAC estando en el modo de Verificación o de Carga de Datos - figura 3.1.c-; y, para la perforación de datos, los Registros de Salida hacia las bobinas ("Punch Coils ODD/EVEN" en las figuras 1.2 y A.3.4), pero, el mayor inconveniente sería el mantener el sincronismo de Z80-CPU con UNIVAC durante todo el proceso de lectura ó perforación de tarjetas.
- d) En el punto 1.4 se han descrito algunas operaciones automáticas que realiza UNIVAC para la transferencia de datos entre sus diversos registros y memorias; de ellas se ha puesto especial énfasis en la Secuencia Automática de Salida de Datos, pues parece ser realmente la más adecuada

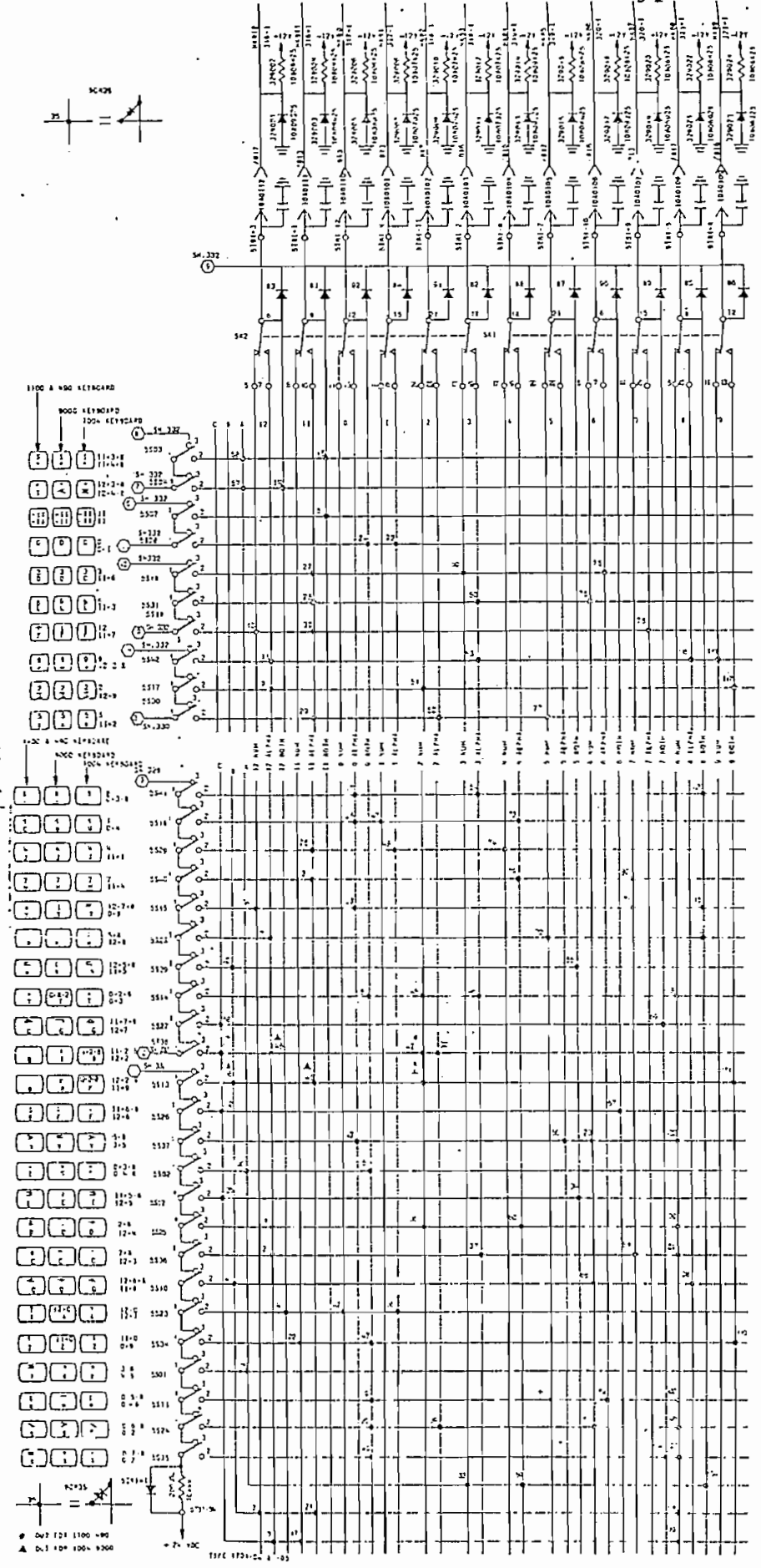
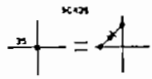


Figura 3.1.b. Matriz de conexiones que conforma el teclado.

● DUT 121 1100 490  
▲ D.1 104 1004 8300

## 1. PUNCH MODE KEY-IN DATA OPERATION

THE PUNCH MODE (327) KEY-IN DATA OPERATION CONSIST OF TWO BASIC MEMORY CYCLES THE FIRST CYCLE REPLACES THE DATA IN MEMORY WITH THE DATA JUST KEYED IN AND INCREMENTS THE COLUMN INDEX INDICATOR (310, 311) THE SECOND CYCLE CHECKS THE PROGRAM MEMORY FOR AUTOMATIC OPERATIONAL CONTROLS (SKIP, DUP, SHIFT, FEED, ETC) (THE AUTOMATIC OPERATIONAL CONTROLS ARE OMLT EFFECTIVE IN AUTOMATIC MODE.)

	1ST CYCLE	2ND CYCLE	3RD CYCLE
ASYN- CHRONOUS	GENERATE KE SPT (331-001) AB DATA GATED TO INPUT REG (316 THRU 321) GENERATE INPUT DATA ALERT (342-010) SET INPUT CYCLE ALERT FF (333-006)		
TP 0	SET INPUT CYCLE FF (333-011) (INITIATES NEW CYCLE (303-004))	CONTINUE IN NEW NEW CYCLE (303-004)	CLR INPUT CYCLE FF IF INPUT CYCLE ALERT FF IS CLEARED (333-010) OR PERFORM AUTO- MATIC OPERATION DETECTED DURING PRGM LOOK AHEAD CYCLE
TP 1	INPUT ADRS REG TO WAR (302-009) (ADDRESSES INPUT MEM)	INPUT ADRS REG TO WAR (302-009) CHECK PRGM SW (327) FOR PRGM LOCATION (302-003)	
TP 2	GENERATE READ MEM STROBE (303-011) INPUT MEM TO DATA REG (316 THRU 321)	GENERATE READ MEM STROBE (303-011) SELECTED PRGM MEM TO DATA REG (316 THRU 321) AND PRGM REG (335)	
TP 3			
TP 4	CLR DATA REG (324-003)		
TP 5	INPUT REG TO DATA REG (324-006, 316 THRU 321)	CLR PRGM OPERATIONAL CON- TROLS (326-001) WHEN NEW FIELD DETECTED	
TP 6	GENERATE WRITE MEM STROBE (303-015)	GENERATE WRITE MEM STROBE (303-015) SET PRGM OPERATIONAL CONTRL IF IN AUTO MODE (336)	
TP 7	OPERATE CLICK (331-014) INCREMENT INPUT ADRS REG (309-001) CLR INPUT REG (324-018) SET INPUT PRGM CYCLE FF AT T E OF TP 7 (333-012)	CLR INPUT CYCLE ALERT FF IF FIELD DEFINITION CTRL FF IS CLEARED (333-005) CLR INPUT PRGM CYCLE FF AT T E OF TP 7 (333-012)	

Tabla 3.1. Proceso de entrada de datos por teclado.

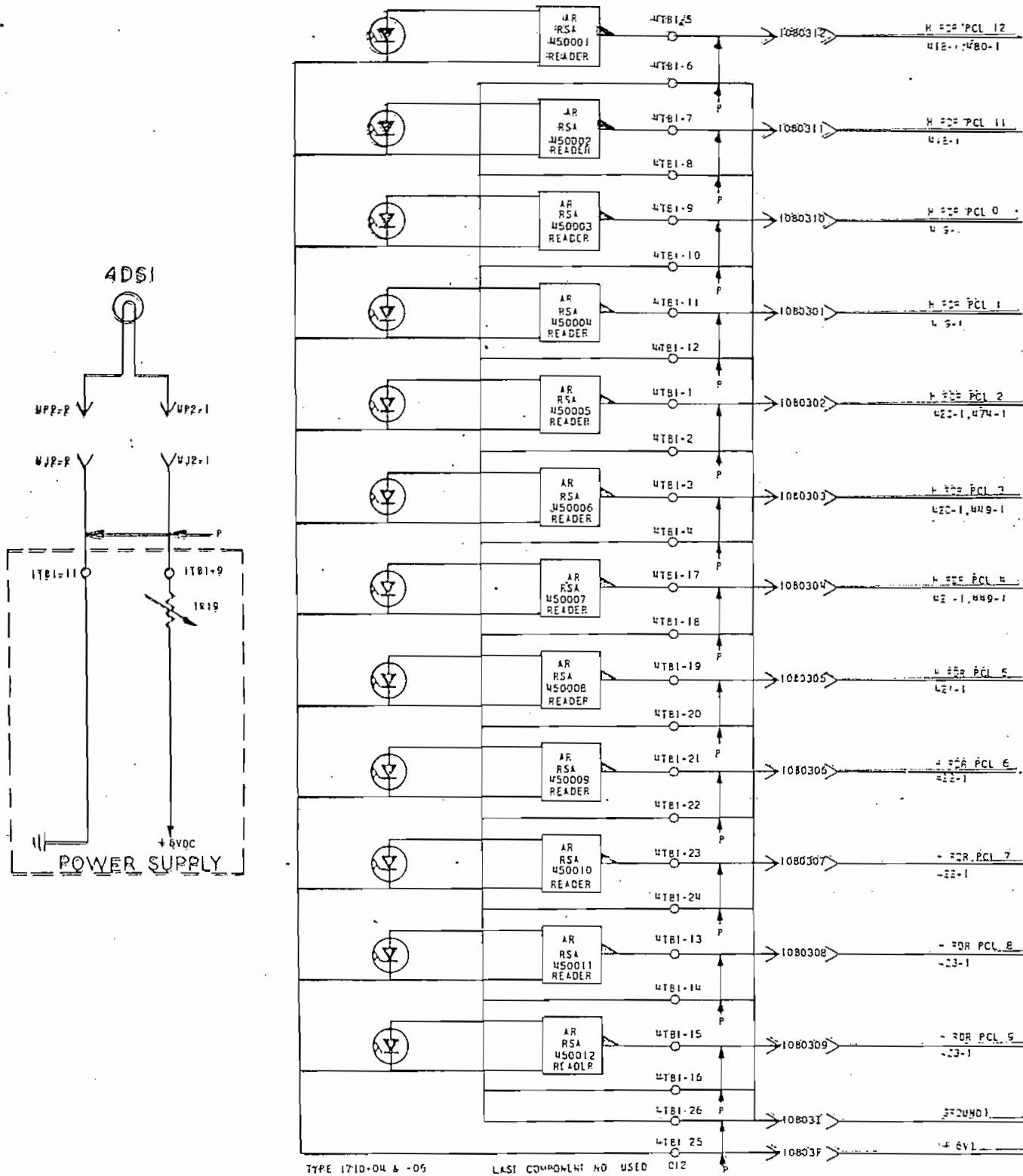


Figura 3.1.c. Estación de Lectura con sus amplificadores.

para propósitos del Interfase.

La primera fase de esta secuencia automática dá la posibilidad de manejar los datos desde el exterior y para ésto se han descrito a continuación por lo menos dos métodos; ambos tienen sus propias ventajas y desventajas.

- 1) El primero utilizaría una memoria adicional dedicada a almacenar los 80 datos provenientes de una tarjeta procesada en UNIVAC ó escrita por Z80-CPU. El almacenamiento y la lectura de los datos se realizaría en forma asincrónica, lográndose de esta manera un mínimo de inferencia en ambos sistemas y una alta independencia en sus funcionamientos individuales.

El diseño del circuito para este método requiere, en contraparte, cierta laboriosidad y atención en lo que respecta al software\*.

- 2) El segundo método, que es el desarrollado en el presente trabajo, realiza la transferencia de datos directamente entre los dos sistemas. No existe ninguna memoria interme

---

\* Se trabajó inicialmente en este diseño, pero no se prosiguió al ser necesario desarrollar un software algo sofisticado.

dia, requiriendo por tanto un funcionamiento más coordinado (*handshake*) entre ellos. Por otra parte, su construcción se torna más sencilla, segura y económica.

En este método es posible aprovechar las capacidades de interrupción de PIO; sin embargo, por asuntos de simplicidad en la programación y por la configuración del sistema en que se halla insertado el Interfase (existen otros periféricos con igual o mayor prioridad, que pueden, en determinado caso, requerir la atención de CPU en el preciso momento en que se esté llevando a cabo la Secuencia Automática de Salida), no se ha hecho uso de tales capacidades de interrupción.

Una vez escogido el método, se procede al desarrollo del diseño. Su diagrama de bloques se presenta a continuación.

### 3.2. DIAGRAMA DE BLOQUES PARA EL METODO ESCOGIDO

Se pueden observar en el diagrama de la figura 3.2 (pág. 128.b) los elementos principales de que va a constar el diseño, así como también el flujo de las señales de datos y de control que intervendrán en él.

Z80-CPU y UNIVAC están, en el diseño, enlazados por PIO, que tiene como principal función el coordinar la comunicación de datos en ambos sentidos; la puerta A está en el modo de Salid



da y la puerta B en el modo de Entrada de datos.

El bloque "PROM" sirve de "traductor de códigos"; así, un carácter codificado en ASCII a la entrada es convertido al código IBM-Hollerith a la salida, o también, un carácter codificado en IBM-Hollerith a la entrada es convertido al código ASCII a la salida, dependiendo de la dirección de los datos.

Los bloques de "Control de Flujo" permiten el paso o no de los datos. Estarán conformados en general por elementos tristate habilitados por determinadas señales de control.

El bloque de "ENCODIFICACION" reduce las doce líneas de datos del Código IBM-Hollerith, a siete líneas que direccionan a PROM.

El bloque de "DECODIFICACION" extiende las 8 líneas de salida de PROM, a doce líneas que corresponden al dato en el Código IBM-Hollerith.

Estos dos bloques últimos coadyuvan a PROM en su tarea de traducción de códigos.

El bloque de "Compatibilidad de Niveles Eléctricos" lleva los niveles lógicos de UNIVAC a los correspondientes TTL y viceversa, favoreciendo a la manipulación de señales de UNIVAC.

Los bloques "Feed - Clear" y "Clear Register" permiten operaciones como "alimentación de nueva ficha en UNIVAC", "terminación de una condición de error en UNIVAC" y "enceramiento del Registro de Entrada".

El bloque "Control Lógico" dirige el funcionamiento de los restantes bloques durante las diversas operaciones que se efectúan.

Las diversas señales que parten de este bloque, posibilitan mantener orden en el flujo de datos y en el funcionamiento individual y colectivo del sistema. (Gran parte de estas señales son generadas por los registros indicadores del proceso de la Primera Fase de la Secuencia Automática de Salida de Datos de UNIVAC).

### 3.3. FUNCIONAMIENTO DE CADA BLOQUE

En los siguientes puntos se explica el funcionamiento y el contenido de cada bloque, advirtiéndose que su desempeño dentro ya de todo el sistema en pleno trabajo, se ve más claro una vez realizada la interconexión de bloques, en el punto 3.4.

### 3.3.1. BLOQUE DE COMPATIBILIDAD DE NIVELES ELECTRICOS Y CONTROL DE FLUJO DE DATOS HACIA UNIVAC

Como ya se ha anotado, los niveles voltamperimétricos de UNIVAC y Z80-CPU (o TTL) resultan ser distintos. Es por ello conveniente en primer lugar hacer compatible los niveles de UNIVAC con los niveles más familiares TTL. Esto se ha conseguido con la ayuda de los siguientes circuitos:

a) Recepción de señales provenientes de UNIVAC

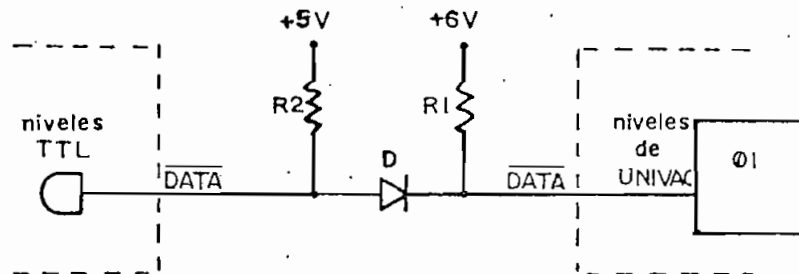


Figura 3.3.1.1 Circuito para la recepción de señales de UNIVAC.

$R_1$  y  $R_2$  aseguran la existencia de un  $1_L$  mientras no haya activación de la línea de dato,  $\overline{DATA}$ .

Para determinar  $R_1$ ,  $R_2$  y  $D$ , se consideran los circuitos equivalentes de entrada y salida LSTTL\* y UNIVAC respectivamente.

\* Se ha trabajado con la tecnología LSTTL por razones de menor consumo de potencia primordialmente, así como también por su aceptable fan-out. En el anexo, punto A.3.2, se tienen los circuitos equivalentes de entrada y salida de algunas familias.

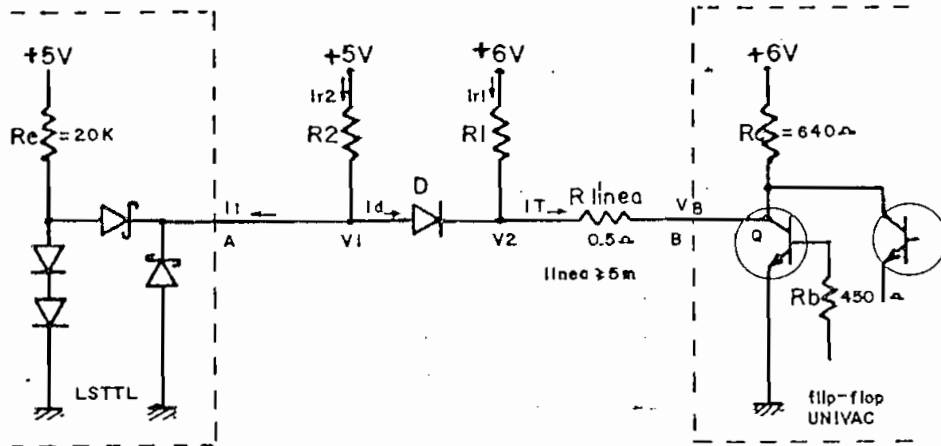


Figura 3.3.1.2. Circuito anterior incluyendo los circuitos e quivalentes conectados en los extremos.

Los cálculos siguientes mantienen cierta "holgura" en los valores numéricos.

-- Para  $B = 1_L$ :

$$V_B = 6 \text{ V}$$

$$V_2 = 6 \text{ V}$$

$$V_1 = 5 \text{ V}$$

$$I_D = 0 \text{ mA}$$

$R_1$ ,  $R_2$  y  $D$  prácticamente no influyen con su valor.

-- Para  $B = 0_L$ :

$$V_B \approx 0.2 \text{ V}$$

$$V_2 \approx 0.2 \text{ V}$$

es conveniente que D sea un diodo de Germanio para que el voltaje transmitido a  $V_1$  se encuentre dentro del margen aceptado como  $0_L$ ; entonces,

$$V_D \approx 0.3 \text{ V}$$

$$V_1 \approx 0.5 \text{ V}$$

Se trata de tener a  $I_T$  de un valor relativamente pequeño de suerte de no alterar mayormente la carga original de Q.

Para una velocidad de respuesta del circuito mediana y un consumo de potencia algo bajo, se toman

$$R_1 = R_2 = 10 \text{ K}\Omega$$

entonces,

$$I_{R_1} \approx 0.58 \text{ mA}$$

$$I_{R_2} \approx 0.45 \text{ mA}$$

Según la tabla A.3.3.:  $I_{IL} \leq -0.36 \text{ mA}$

con lo que,

$$I_D \approx 0.81 \text{ mA}$$

$$I_T \approx 1.39 \text{ mA}$$

El circuito anterior puede ser mejorado con el fin de disminuir el efecto del ruido inducido en la línea, ruido que es

producido entre otras causas, por las transiciones de estados en los circuitos de UNIVAC por donde atravieza.

En la siguiente figura se observa el "filtro R-C" dispuesto para el efecto.

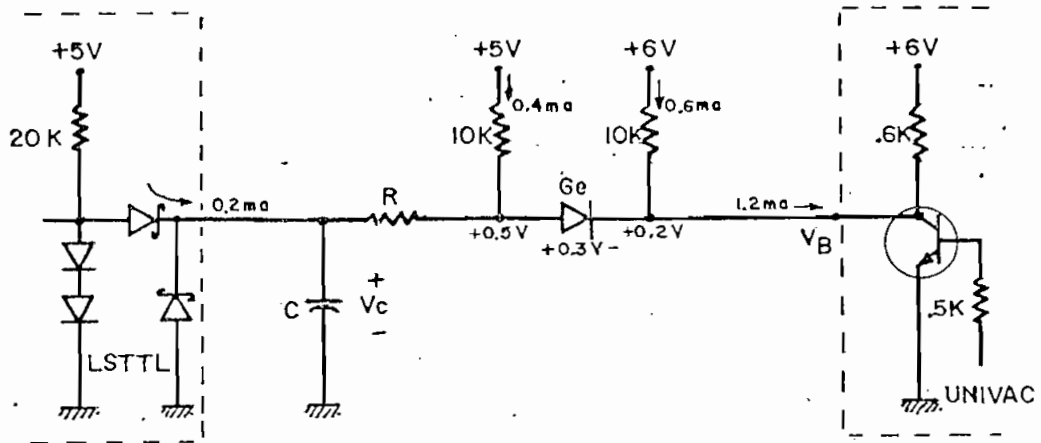


Figura 3.3.1.3. Circuito de la figura anterior en el que consta el filtro RC.

Hallando valores para R y C:

(Para las siguientes operaciones numéricas s lamente se considera el caso en que  $V_B = 0_L$  que es el que interesa. Cuando  $V_B = 1_L$  pr cticamente no hay cambio en los valores de voltajes y corrientes).

(Los valores num ricos son promedios; no por ello se resta validez a los resultados).

La constante de tiempo de la red  $\tau$  viene dada por la fórmula:

$$\tau \text{ [seg]} = R \text{ [\Omega]} \times C \text{ [f]}$$

es conveniente un valor de  $\tau$  desde unos 100 nseg, hasta unos 300 nseg. para una respuesta moderada a los pulsos de datos (cuyos anchos son de 2.5  $\mu$ seg.).

Según la tabla A.3.2. del anexo, para LSTTL:

$$V_{IL \text{ máx}} = 0.7 \text{ V}$$

asumiendo:  $V_C < 0.7 \text{ V}$

entonces,  $0.2 \text{ mA} \times R < 0.7 \text{ V} - 0.5 \text{ V}$

$$R < 1 \text{ K}$$

Se toman:  $R = 470 \text{ } \Omega$

$$C = 220 \text{ pf}$$

con lo que  $\tau = RC = 103 \text{ nseg. (aceptable)}$

$$V_{C \text{ low}} = 0.6 \text{ V} \quad (\text{aceptable})$$

El circuito de la figura 3.3.1.3. actúa en la detección del nivel lógico de ciertos flip-flops de UNIVAC que indican su

status y que son indispensables para el Bloque de Control; éstos son:  $\overline{SOSFF}$ ,  $\overline{LD DATA}$ ,  $\overline{PUSH FF}$ ,  $\overline{ERROR}$ ,  $\overline{WR2}$ ,  $\overline{FF1'2}$  e  $\overline{INH}$ .\*

b) Envío y recepción de datos a/de UNIVAC por una misma línea.-

Para este caso se ha construido, a partir del circuito anterior, el siguiente:

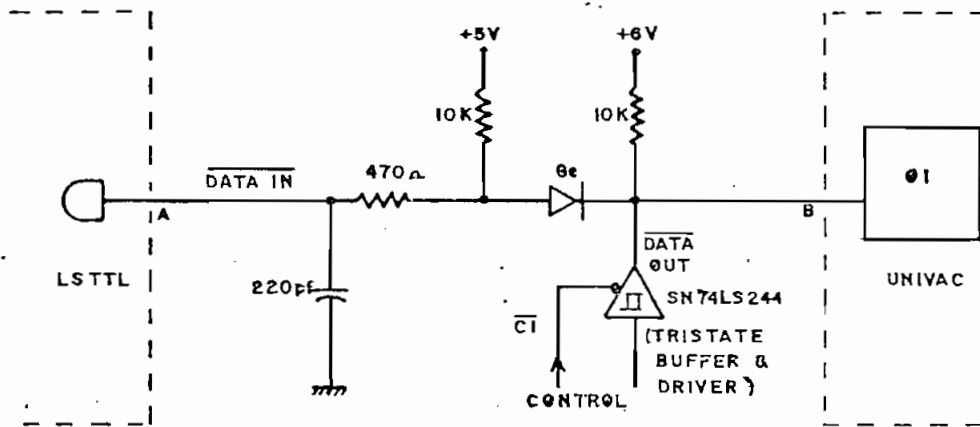


Figura 3.3.1.4. Circuito para envío/recepción de datos a/de UNIVAC.

Cuando se requiere la recepción de datos, los "Buffer-Drivers" \*\* pasan a su estado de alta impedancia, de manera que dicha operación se efectúa en la misma forma a la descrita en la parte (a).

\* Se estudian en los puntos 3.3.4.2 y 3.3.9.2.

\*\* En el anexo, punto A.3.7.7., son descritos.



Para el envío de datos, los Buffer-Drivers deberán imponer un  $0_L$  en el punto B (conforme se había anotado en el punto 1.7.5 el envío de datos se lo efectúa por el envío de un  $0_L$  en la línea  $\overline{DATA}$ , la misma que normalmente se encuentra en  $1_L$ ).

En el punto A.3.4. del anexo, se ha hecho un estudio para determinar la máxima corriente  $I_{OL}$  que demandaría una línea de dato para poder ser impuesta un  $0_L$ . Se ha encontrado que la línea "L DATA REG FF7" es la que tiene el mayor número de cargas y que para su activación, requiere de una corriente  $I_{OL \text{ máx}} = 26 \text{ mA}$  -durante  $2.5 \mu\text{seg.}$ - Este valor sí es factible de conseguir con los IC anotados.

El circuito de la figura 3.3.1.4 es utilizado para la detección y manejo de las doce líneas de datos provenientes del Registro de Datos de UNIVAC.

### c) Envío de señales hacia UNIVAC

Para el envío de señales hacia UNIVAC se han construido los siguientes circuitos:

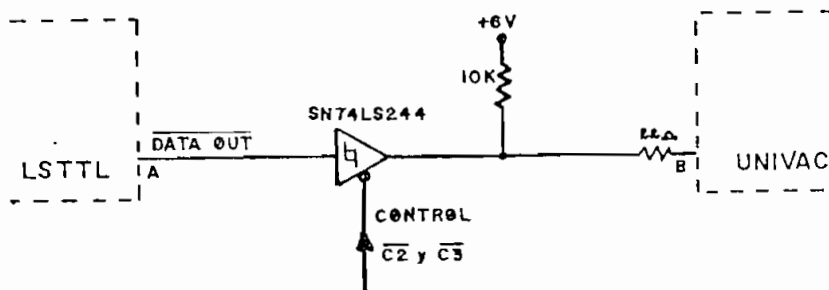


Figura 3.3.1.5. Circuito para el envío de señal hacia UNIVAC

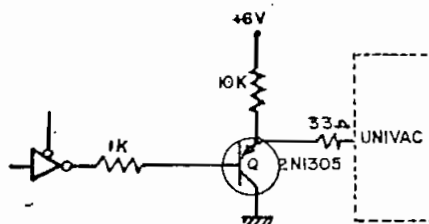


Figura 3.3.1.6. Circuito para el envío de señales con alto fan-out.

El circuito de la figura 3.3.1.5. viene a ser la sección de salida del circuito representado en la figura 3.3.1.4. Es utilizado para la alimentación automática de nueva ficha y para el borrado general de UNIVAC *por control remoto* desde el microcomputador.

El circuito de la figura 3.3.1.6. utiliza un transistor tipo PNP 2N1305 de buena respuesta de frecuencia y alta corriente de colector. Sirve para el borrado de los 12 bits del Registro de Entrada de UNIVAC, en donde se necesita un elemento con alto fan-out.

Así pues, el Bloque de Compatibilidad de Niveles Eléctricos y Control de Flujo de Datos hacia UNIVAC, lo conforman los circuitos de las figuras 3.3.1.3 - 4 - 5 y 6.

### 3.3.2. BLOQUE DE ENCODIFICACION\*

Las doce líneas de datos provenientes de UNIVAC, producen al combinarse, los 66 caracteres alfanuméricos diferentes\*\* que se indican en la tabla 3.3.2.1.

Un menor número de líneas puede brindar la misma información y resulta ser más manejable para objetos del Interfase. Este ha sido un motivo para realizar la encodificación.

Otro motivo es el de minimizar el costo, pues es más económico el uso de elementos como PROM, DRIVERS, etc., de menor número de líneas de entrada y salida (es decir, de menor capacidad).

Dado que la PROM utilizada en este Interfase\*\*\* sirve para realizar las recodificaciones Hollerith→ASCII y ASCII→Hollerith; y, siendo que el código ASCII se compone de 8 bit/carácter, con el bit más significativo fijo en 0<sub>L</sub>, entonces es deseable que PROM posea 8 bits de salida en paralelo y ocho pines de direccionamiento con uno de ellos actuando de indi

---

\* Se usa el término "encodificación", para significar una codificación que va de más a menos número de elementos de código.

\*\* No se toma en cuenta el caso de perforación múltiple o MULT PUNCH en el que se pueden sobrescribir uno o más datos.

\*\*\* Se la estudia en el punto 3.3.5.

... TABLA DE CODIGOS DE LA 1701 VP Y 1710 VIP  
PARA LA LINEA UNIVAC 9000

SIMBOLOS Y CARACTERES	CODIGOS (zonas de la tarjeta Hollerith)	SIMBOLOS Y CARACTERES	CODIGOS (zonas de la tarjeta Holl.)
A	12-1	8	8
B	12-2	9	9
C	12-3	0	0
D	12-4	#	3-8
E	12-5	, (coma)	0-3-8
F	12-6	\$	11-3-8
G	12-7	.	12-3-8
H	12-8	-11	11
I	12-9	+	12-6-8
J	11-1	—	0-5-8
K	11-2	)	11-5-8
L	11-3	ç	12-2-8
M	11-4	0-2-8	0-2-8
N	11-5		12-7-8
O	11-6	&-12	12
P	11-7	12-0	12-0
Q	11-8	>	0-6-8
R	11-9	:	2-8
S	0-2	;	11-6-8
T	0-3	⌋	11-7-8
U	0-4	, (apóstr.)	5-8
V	0-5	11-0	11-0
W	0-6	?	0-7-8
X	0-7	*	7-8
Y	0-8	=	6-8
Z	0-9	!	11-2-8
.1	1	{	12-5-8
2	2	@	4-8
3	3	%	0-4-8
4	4	*	11-4-8
5	5	<	12-4-8
6	6	/	0-1
7	7	SPACE	

Tabla 3.3.2.1

cador del sentido de la recodificación. Este ha sido el motivo por el que se ha hecho que la encodificación reduzca el número de las líneas de datos de doce a siete.

Un 8-Line-To-3 Line Priority ENCODER se constituye en el elemento básico de este bloque.

Poniendo ahora atención al hecho de que la capacidad potencial de información a la entrada del ENCODER es mayor que a su salida (teóricamente  $2^8$  combinaciones diferentes a la entrada pasan a ser  $2^3$  combinaciones diferentes a la salida - no se cuentan EO y GS -), entonces cabría decir que se va a producir una "disminución de la información" que atraviese por este Bloque.

Pues bien, analizando la tabla 3.3.2.1 se puede comprobar que los códigos -zonas- 1,2,3,4,5,6,7 y 9 nunca se dan simultáneamente. Esta característica del Código IBM para tarjetas Hollerith es aprovechada para salvar el "inconveniente" anotado cuando son éstas líneas las que se conectan a la entrada del ENCODER.

En la tabla de verdad del ENCODER se nota que existe un mismo status de salida para dos combinaciones diferentes a la entrada (cuando IO es  $0_L$  y cuando todas las entradas IO - 17 son  $1_L$ ). Por ello la entrada de IO a  $1_L$  en forma permanente

y se ha antepuesto al ENCODER un circuito combinacional a base de compuertas AND tipo *Open Collector*, con el fin de conseguir una nueva combinación para el caso de tener activa la zona 1; esta combinación no debe ser alterada ni sufrir equívoco para ningún código válido, incluyendo aquellos en que se active la zona 1 conjuntamente con otra zona.

Con esta salvedad se sigue manteniendo el número de siete líneas a la salida de este bloque.

En la siguiente figura se ha desarrollado el Bloque de Encodificación, poniendo atención a lo dicho.

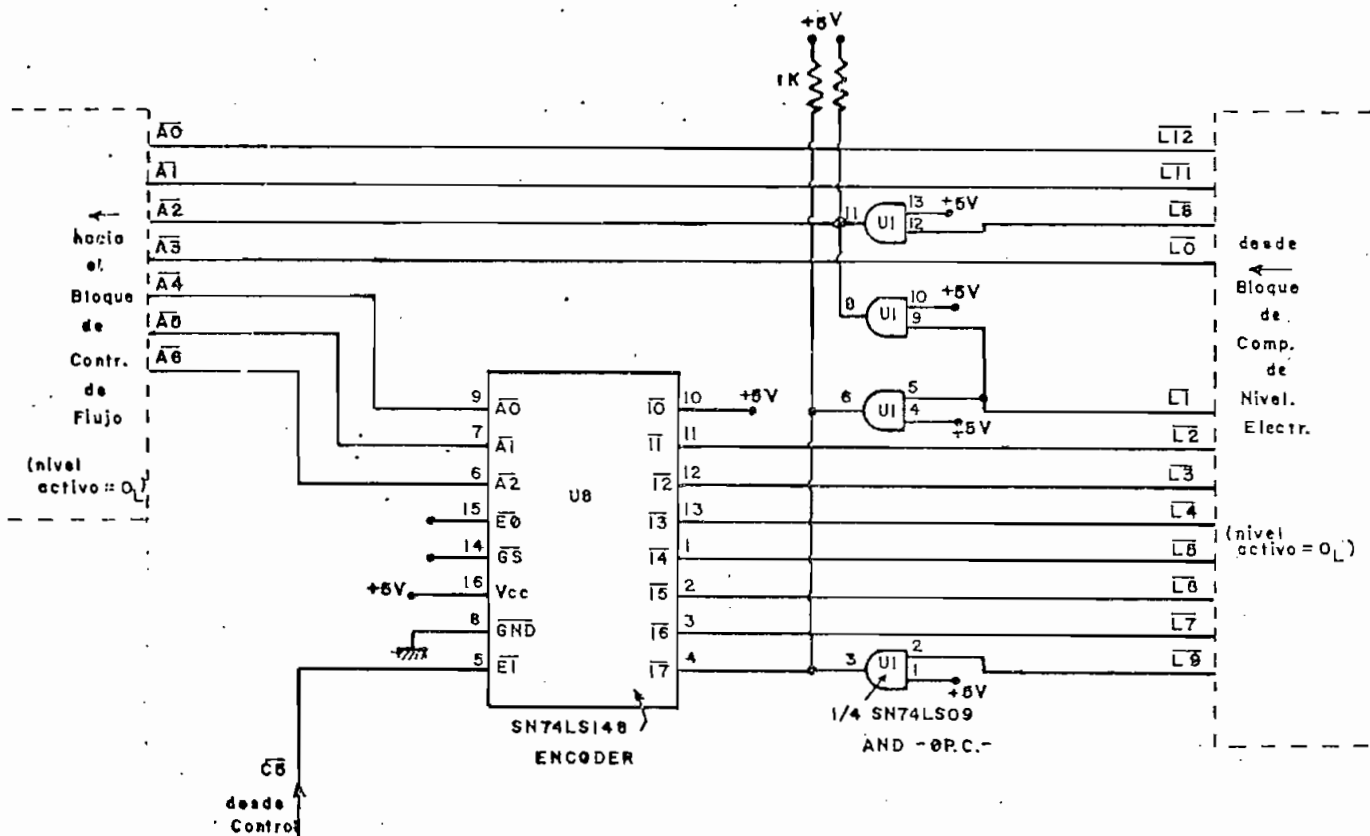


Figura 3.3.2.1 Bloque de ENCODIFICACION

La disposición de las líneas de la figura facilita mucho el trabajo de encodificación, así:

- las líneas L12, L11 y L0, las mismas que generan un buen número de combinaciones, atraviezan el bloque sin alteraciones;
- la línea L8 pasa también sin alteración, pero es activada junto con L9 el instante en que existe un dato en L1;
- La línea L1 es "reemplazada" por una combinación de L8 y L9; esto es, la activación simultánea de L8 y L9 (que nunca sucede conforme a la tabla 3.3.2) se "lee" como dato en L1;
- las líneas L2, L3, L4, L5, L6, L7 y L9, las cuales nunca se activan simultáneamente, son conectadas al ENCODER y reducidas a 3 líneas.

Las cuatro compuertas AND Open Collector intervienen en la formación de la combinación que reemplaza al dato L1 por medio del AND cableado (el valor de las resistencias se lo ha tomado de 1 K $\Omega$  como un valor aceptable).

Gracias a esta disposición, la codificación de los caracteres en los que intervienen 2 y 3 líneas, se efectúa sin problemas

con la activación de las respectivas líneas.

La tabla 3.3.2.2. resume la codificación de las 12 líneas que entran al Bloque, en 7 líneas que salen del mismo.

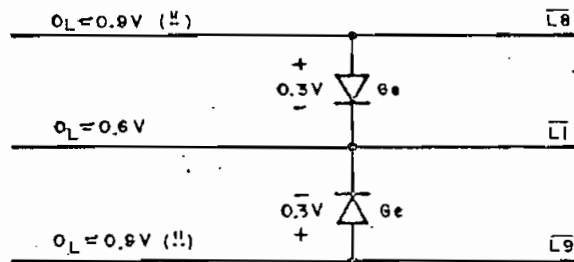


Figura 3.3.2.2. Conexión equivalente al circuito combinacional de las compuertas AND Open Collector, pero no realizable por los valores de  $O_L$ .

LINEAS ACTIVADAS A LA ENTRADA	LINEAS ACTIVADAS A LA SALIDA
L12	A0
L11	A1
L8	A2
L0	A3
L1	A2, A4, A5, A6
L2	A4
L3	A5
L4	A4, A5
L5	A6
L6	A4, A6
L7	A5, A6
L9	A4, A5, A6
ninguna	ninguna

Tabla 3.3.2.2 Encodificación de líneas



### 3.3.3. BLOQUE DE DECODIFICACION

Los datos que salen desde el Interfase hacia UNIVAC deben ir codificados en 12 zonas, conforme a la tabla 3.3.2.1., lo cual implica que los 8 bits de salida de PROM sean *extendidos* a 12. Esta función es justamente la que desempeña el presente Bloque.

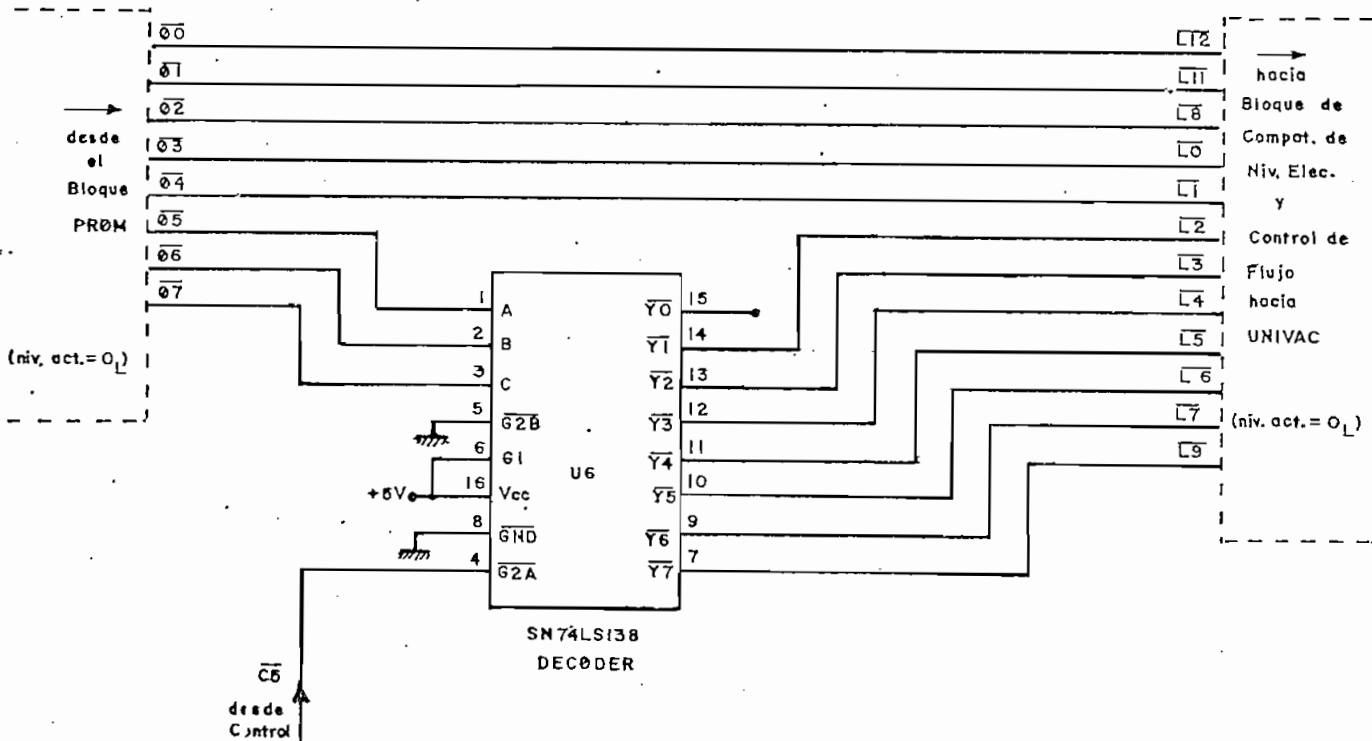


Figura 3.3.3. Bloque de Decodificación

Un 3-To-8 Line DECODER es el elemento básico de este Bloque.

Por iguales motivos a los expuestos en el punto 3.3.2., se han generado a las líneas L2, L3, L4, L5, L6, L7 y L9 desde

el DECODER.

La tabla 3.3.3 resume la decodificación, de 8 líneas que en tran al Bloque, en 12 que salen de él con la codificación de seada.

LINEAS ACTIVADAS A LA ENTRADA	LINEAS ACTIVADAS A LA SALIDA
$\phi 0$	L12
$\phi 1$	L11
$\phi 2$	L8
$\phi 3$	L0
$\phi 4$	L1
$\phi 6, \phi 7$	L2
$\phi 5, \phi 7$	L3
$\phi 7$	L4
$\phi 5, \phi 6$	L5
$\phi 6$	L6
$\phi 5$	L7
ninguna	L9
$\phi 5, \phi 6, \phi 7$	ninguna

Tabla 3.3.3 Decodificación de Líneas

Nótese que las líneas 05, 06 y 07 producen 7 combinaciones - diferentes que corresponden a 7 diferentes líneas a la salida

da del Bloque. En la programación de PROM se debe tomar en cuenta esta consideración.

#### 3.3.4. BLOQUE DE CONTROL DE FLUJO A/DE Z80-CPU Y DENTRO DEL INTERFASE

En la Figura 3.2 constan dos bloques de Control de Flujo entre Z80-CPU y el Interfase, así como también uno entre el bloque de Encodificación y PROM.

Seguidamente se detallan cada uno de tales bloques.

##### 3.3.4.1. BLOQUE DE CONTROL DE FLUJO DE DATOS DESDE EL BLOQUE DE ENCODIFICACION A PROM

Evita que suceda el manejo impropio de los pines de direccionamiento de PROM por los datos provenientes del Bloque de Encodificación y de PIO al mismo tiempo.

Se cuenta para ello con elemento cuya salida es tristate como son los IC SN74LS368 HEX BUS DRIVER y se cuenta también con el propio PIO cuya puerta A es tristate (se habilita o deshabilita por medio de la línea "A STB").

Vale anotar que las señales de habilitación  $\overline{C6}$  y  $\overline{A STB}$  deben ser no coincidentes.

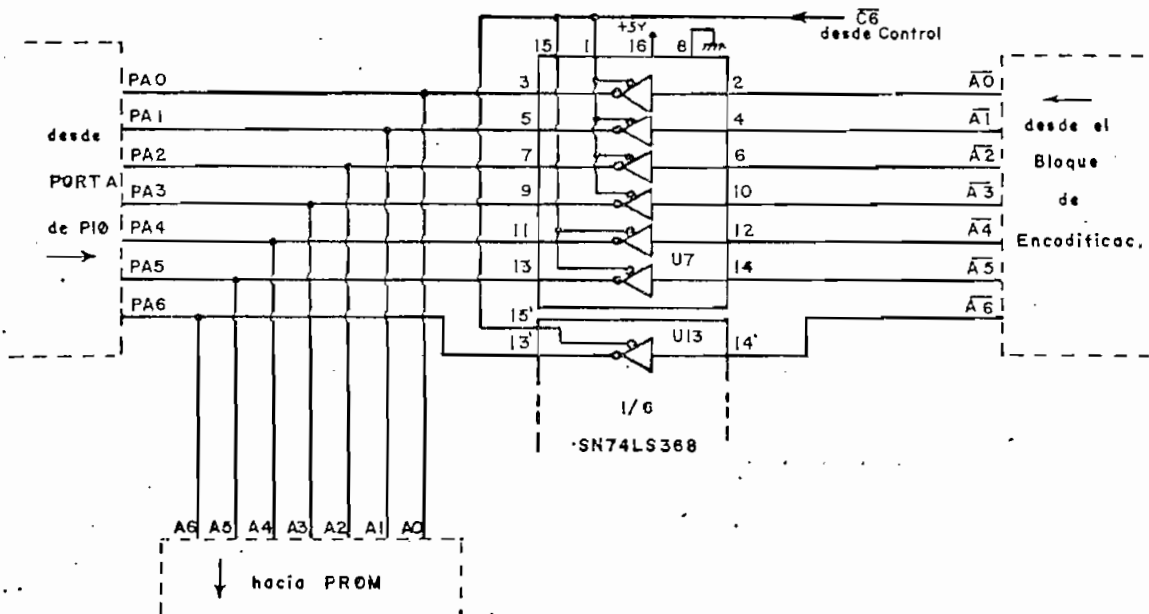


Figura 3.3.4.1. Bloque de Control de Flujo de Datos desde el Bloque de Encodificación a PROM.

### 3.3.4.2. BLOQUE DE CONTROL DE FLUJO DESDE EL BLOQUE DE CONTROL LOGICO A Z80-CPU

Las señales indicadoras del *status* de UNIVAC pueden tener acceso a la vara de datos del microcomputador a través de este bloque el momento en que así sea requerido.

La composición de este bloque es muy similar al anterior.

A continuación se indica la naturaleza de las cuatro señales de control de la figura.

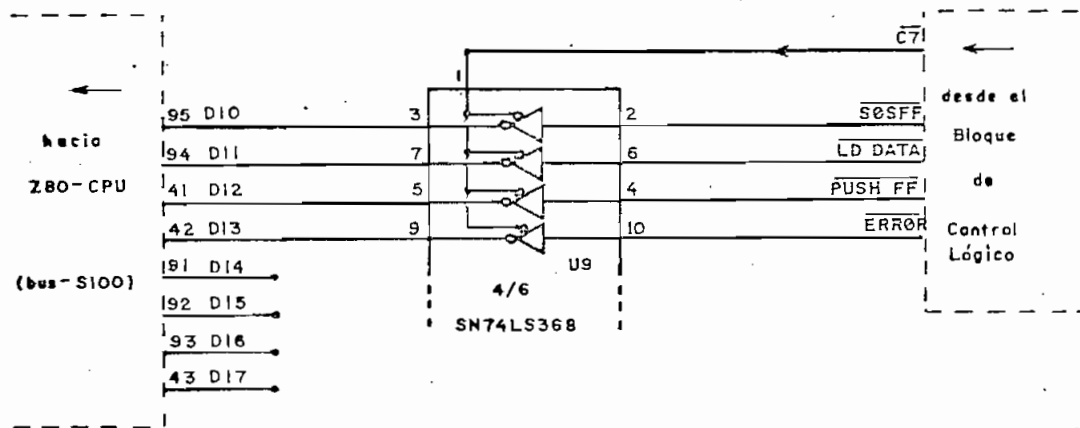


Figura 3.3.4.2. Bloque de Control de Flujo desde el Bloque de Control Lógico a Z80-CPU.

- $\overline{\text{SOSFF}}$  : (Start Output Sequence Flip-Flop) es la señal que se activa al iniciarse la Primera Fase de la Secuencia Automática de Salida de Datos de UNIVAC, permaneciendo así hasta el término de dicha fase.
- $\overline{\text{LD DATA}}$  : (Load Data) se activa el momento en que UNIVAC se pone en el modo de Carga de Datos. En este modo, los datos leídos de tarjetas deberán transferirse al microcomputador.
- $\overline{\text{PUSH FF}}$  : (Push Flip-Flop) es la señal que señala la movilización de la tarjeta a través de las estaciones de lectura y perforación. El momento en que deja de estar activa, UNIVAC estará en condiciones de ser alimentada con una nueva tarjeta.

- $\overline{\text{ERROR}}$  : (Error) es la señal que se activa el momento en que se tiene en UNIVAC cualquier condición de error. De ésta deberá ser informado el microcomputador en el menor tiempo posible.

### 3.3.4.3. BLOQUE DE CONTROL DE FLUJO DE DATOS ENTRE Z80 - CPU Y PIO

Permite el paso de datos del microcomputador al Interfase y viceversa. Su salida -como la de cualquier sistema periférico que se conecte a la vara de datos- es tristate, y su habilitación, tanto como la dirección de los datos; dependen - de las señales  $\overline{\text{CS}}$ ,  $\overline{\text{WR}}$  y  $\overline{\text{RD}}$  provenientes del Bloque de Control.

La vara de datos es del tipo S-100 standard\*, con 8 líneas de dicadas a los bits de salida desde CPU (D00-D07) y 8 líneas para los bits de entrada (DI0-DI7). Por su parte, PIO posee 8 líneas para una comunicación bidireccional de datos; en tal virtud, el presente bloque se lo ha construído en la forma - como se observa en las siguientes figuras:

---

\* Se detalla en el punto A.3.5. del anexo.

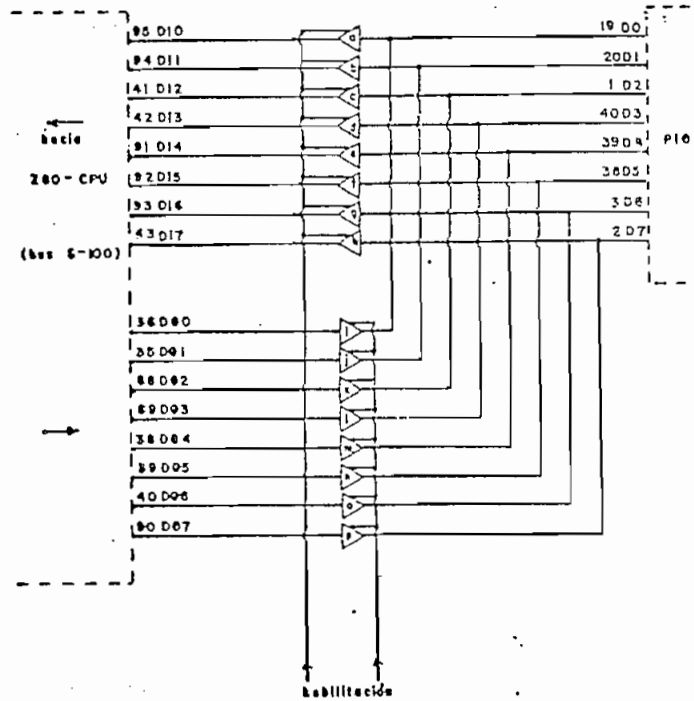


Figura 3.3.4.3.1 Bloque de Control de Flujo de Datos entre Z80-CPU y PIO

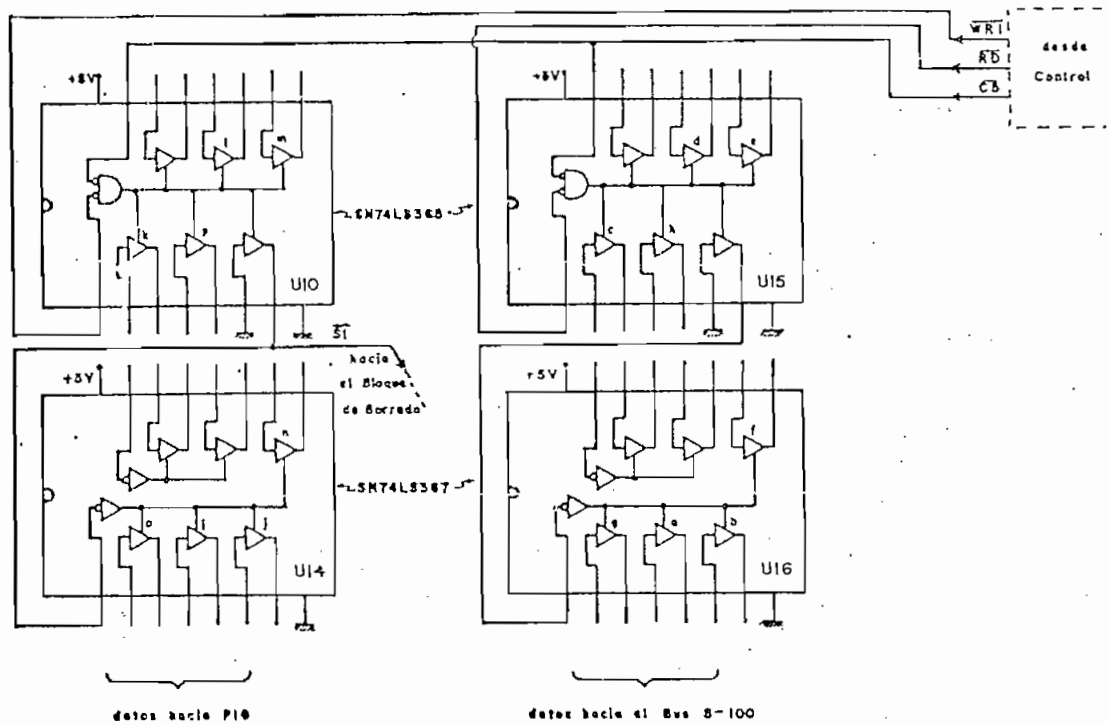


Figura 3.3.4.3.2 Detalle de las conexiones del bloque anterior.

### 3.3.5. BLOQUE PROM

Está constituido por una INTEL B2758\* UV ERASABLE PROM, cuya función será la de realizar la conversión de códigos:

IBM Hollerith → ASCII

ASCII → IBM Hollerith

Esta conversión se realiza en concordancia con los bloques de Encodificación y Decodificación.

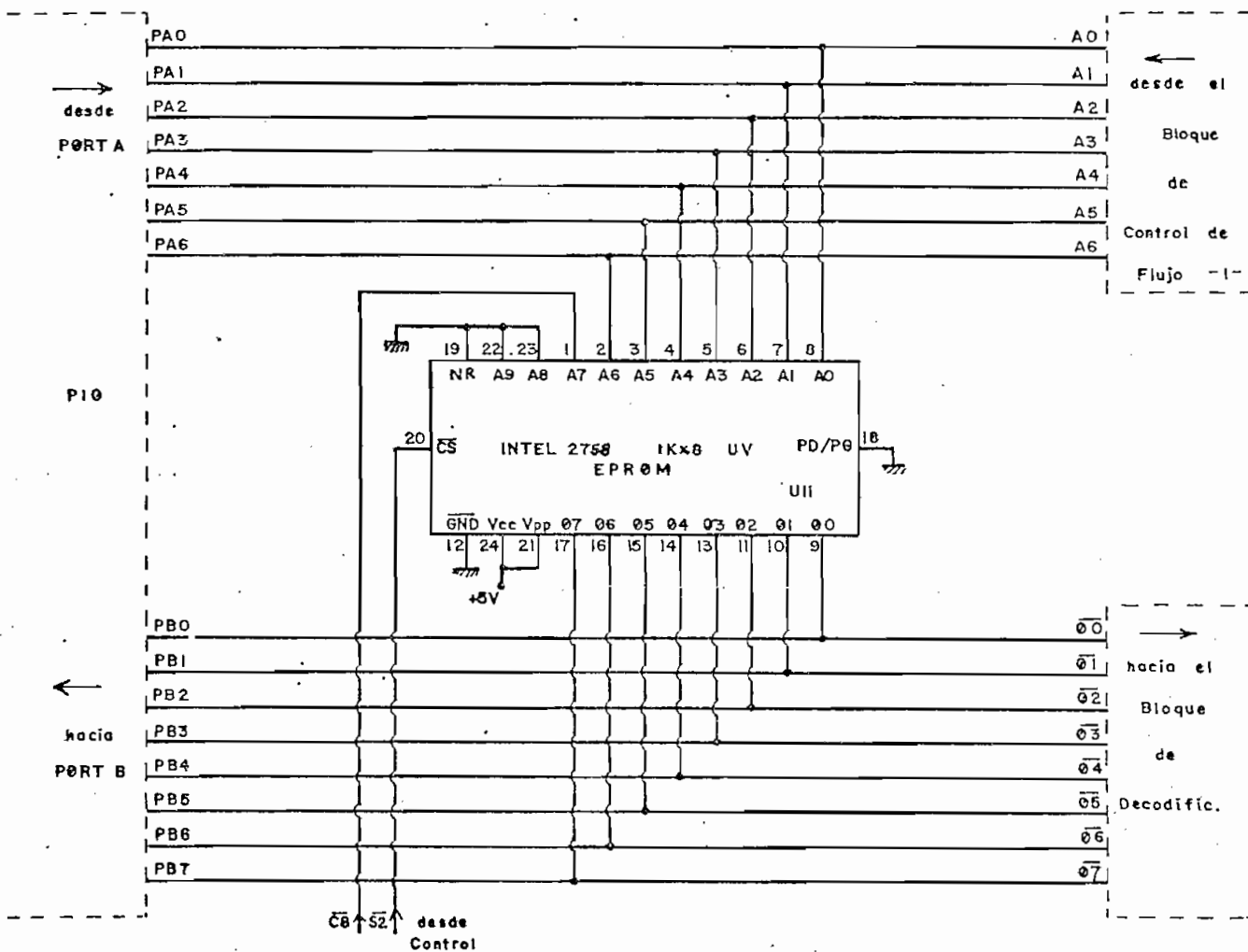


Figura 3.3.5. Bloque PROM

\* Aunque el diseño está hecho para una PROM de menor capacidad - de 8 adreses -, se ha utilizado la PROM 2758 de 10 adreses por causas ajenas a éste.



De las 8 líneas de direccionamiento, la más significativa, A7, determina el sentido de la conversión de códigos y es manejada desde Control.

La programación de PROM se ha basado en las tablas 3.3.2.1 y 3.3.5.1 y se la ha efectuado cuidadosamente en conformidad - con las tablas 3.3.2.2 y 3.3.3.

Chart E-1. ASCII Character Code

		LEAST SIGNIFICANT HEX DIGIT															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
MOST SIGNIFICANT HEX DIGIT	0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
	1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
	2	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	←
	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Tabla 3.3.5.1. Código ASCII de los 128 caracteres alfanuméricos que se utilizan.

PROM			
ADD		DATA	
H	L	H	L
2	0	1	F
2	1	3	9
2	2	D	B
2	3	5	B
2	4	5	9
2	5	7	3
2	6	1	E
2	7	9	B
2	8	9	A
2	9	9	9
2	A	7	9
2	B	B	A
2	C	5	3
2	D	1	D
2	E	5	A
2	F	0	7
3	0	1	7
3	1	0	F
3	2	3	F
3	3	5	F
3	4	7	F
3	5	9	F
3	6	B	F
3	7	D	F
3	8	1	B
3	9	F	F
3	A	3	B
3	B	B	9
3	C	7	A
3	D	B	B
3	E	B	3
3	F	D	3
4	0	7	B
4	1	0	E
4	2	3	E
4	3	5	E
4	4	7	E
4	5	9	E
4	6	B	E
4	7	D	E
4	8	1	A
4	9	F	E
4	A	0	D
4	B	3	D
4	C	5	D
4	D	7	D
4	E	9	D
4	F	B	D

PROM			
ADD		DATA	
H	L	H	L
5	0	D	D
5	1	1	9
5	2	F	D
5	3	3	7
5	4	5	7
5	5	7	7
5	6	9	7
5	7	B	7
5	8	D	7
5	9	1	3
5	A	F	7
5	B	9	A
5	C	D	A
5	D	9	9
5	E	1	6
5	F	1	5
6	0	9	B
6	1	0	E
6	2	3	E
6	3	5	E
6	4	7	E
6	5	9	E
6	6	B	E
6	7	D	E
6	8	1	A
6	9	F	E
6	A	0	D
6	B	3	D
6	C	5	D
6	D	7	D
6	E	9	D
6	F	B	D
7	0	D	D
7	1	1	9
7	2	F	D
7	3	3	7
7	4	5	7
7	5	7	7
7	6	9	7
7	7	B	7
7	8	D	7
7	9	1	3
7	A	F	7
7	B	3	A
7	C	D	A
7	D	D	9
7	E	3	3
7	F	9	3

PROM			
ADD		DATA	
H	L	H	L
F	C	2	F
F	8	5	A
F	6	4	A
F	5	4	1
F	4	3	1
F	2	5	2
F	1	4	9
F	0	3	9
E	C	3	F
E	8	5	8
E	6	7	D
E	5	7	C
E	4	2	2
E	2	5	0
E	1	4	7
E	0	3	7
D	C	3	E
D	8	5	7
D	6	3	B
D	5	2	B
D	4	3	D
D	2	4	F
D	1	4	6
D	0	3	6
C	C	7	F
C	8	5	6
C	6	2	9
C	5	2	8
C	4	2	7
C	2	4	E
C	1	4	5
C	0	3	5
B	C	2	5
B	2	5	5
B	6	2	A
B	5	3	C
B	4	4	0
B	2	4	D
B	1	4	4
B	2	3	4
A	C	2	C
A	8	5	4
A	6	2	4
A	5	2	E
A	4	2	3
A	2	4	C
A	1	4	3
A	0	3	3

PROM			
ADD		DATA	
H	L	H	L
9	C	7	E
9	8	5	3
9	6	2	1
9	5	7	B
9	4	3	A
9	2	4	B
9	1	4	2
9	0	3	2
8	C	5	9
8	A	5	F
8	9	5	E
8	8	3	0
8	6	5	1
8	5	4	8
8	4	3	8
8	2	2	D
8	1	2	6
8	0	2	0

H = bits más signi-  
ficativos.

L = bits menos sig-  
nificativos.

ADD  $\equiv$  address

DATA  $\equiv$  data

Tabla 3.3.5.2. Programa de PROM (en Hexadecimal)

(En la tabla A.3.6 del anexo se muestra la programación en forma desarrollada).

### 3.3.6. BLOQUE PARA EL BORRADO DEL REGISTRO DE ENTRADA DE UNIVAC

En el punto 1.7.5. se ha analizado la forma como podrían ser introducidos los datos desde el exterior en los flip - flops que constituyen el Registro de Datos de UNIVAC, viéndose la necesidad de efectuar el borrado del Registro de Entrada previamente a esta operación.

De acuerdo a lo escrito en el punto 1.5.2. este borrado debería realizarse durante el período T6 de la Primera Fase de la Secuencia Automática de Salida de Datos, período particularmente favorable para el efecto.

Esto implica el que se deba mantener un sincronismo con esa Secuencia.

La sincronización de esta operación de borrado se la ha hecho con la ayuda de algunas señales; así es como se han tomado desde Z80-CPU la señal  $\overline{WR1}$  y desde el bloque de Control las señales  $\overline{C5}$ ,  $\overline{SOSFF}$  y  $\overline{WR2}$  (estas dos últimas procedentes, en definitiva, de UNIVAC).

Para cumplir su función, el Bloque deberá enviar un  $0_L$  por la línea Cl.I.R. el momento en que se cumplan simultáneamente las siguientes condiciones:

- 1) el microcomputador haya direccionado al Interfase;
- 2) el microcomputador se encuentre ejecutando una operación de salida de datos;
- 3) UNIVAC se encuentre operando la Primera Fase de la Secuencia Automática de Salida de Datos.
- 4) dentro de un ciclo de memoria de dicha Secuencia, se suceda el período T6.

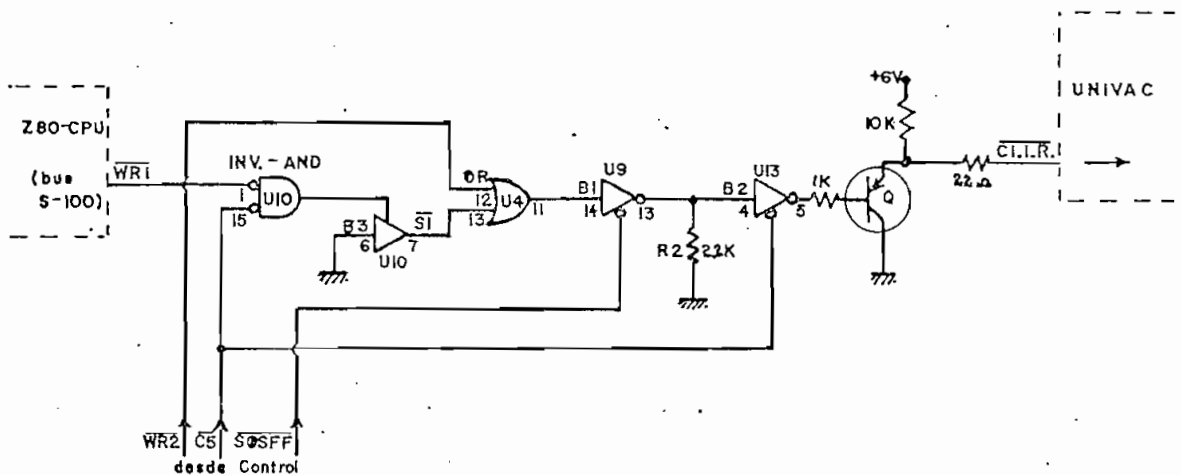


Figura 3.3.6.1. Circuito que constituye el Bloque de Borrado del Registro de Entrada de UNIVAC.

A continuación se explican las señales que intervienen:

- $\overline{WR1}$  : Es la señal proveniente de Z80-CPU que señala la existencia de un dato válido en la vara de datos,

el mismo que deberá ser escrito en el Interfase.

Corresponde a la señal  $\overline{WR}$  del diagrama 2.3.2.

- $\overline{C5}$  : Es una señal que proviene del Bloque de Control y que se activa cuando CPU ha direccionado el Interfase.

En la figura 3.3.9.2 se observa su generación.

- $\overline{S1}$  : Es la señal producida en el Bloque de Control de Flujo entre Z80-CPU y PIO de la figura 3.3.4.3.2.
- $\overline{WR2}$  : Es la misma señal que se muestra en la figura 1.6 como "L WRITE MEM" o " $\overline{TP6}$ ", pero una vez que ha atravesado el Bloque de Compatibilidad de Niveles Eléctricos.
- $\overline{SOSFF}$  : Ya ha sido explicada en el punto 3.3.4.2.
- $\overline{Cl.I.R.}$ : (Clear Input Register) es la señal que sale de este Bloque y se introduce en UNIVAC para realizar el "borrado artificial" del Registro de Entrada.

La figura 3.3.6.3 indica el sitio más propicio para la introducción de la señal en UNIVAC.

La señal  $\overline{Cl.I.R.}$  debe mantenerse en alto mientras no sea el momento de realizarse el borrado; de no ser así, se podría alterar el normal desenvolvimiento de las diferentes operaciones automáticas de UNIVAC.

La sección de salida del Bloque corresponde al "Circuito de Envío de Señal con alto fan-out hacia UNIVAC" de la figura 3.3.1.6.

R2 posibilita mantener un  $0_L$  a la entrada de B2 en el caso de que B1 no se encuentre habilitado; ésto asegura que al encontrarse habilitado sólo B2, el transistor Q no se active.

Un circuito más simple que el grafizado podría cumplir la misma función, pero el motivo de su constitución estriba en la optimización del uso de los chips existentes en el Interfase.

La figura siguiente muestra las diferentes conexiones de los IC's.

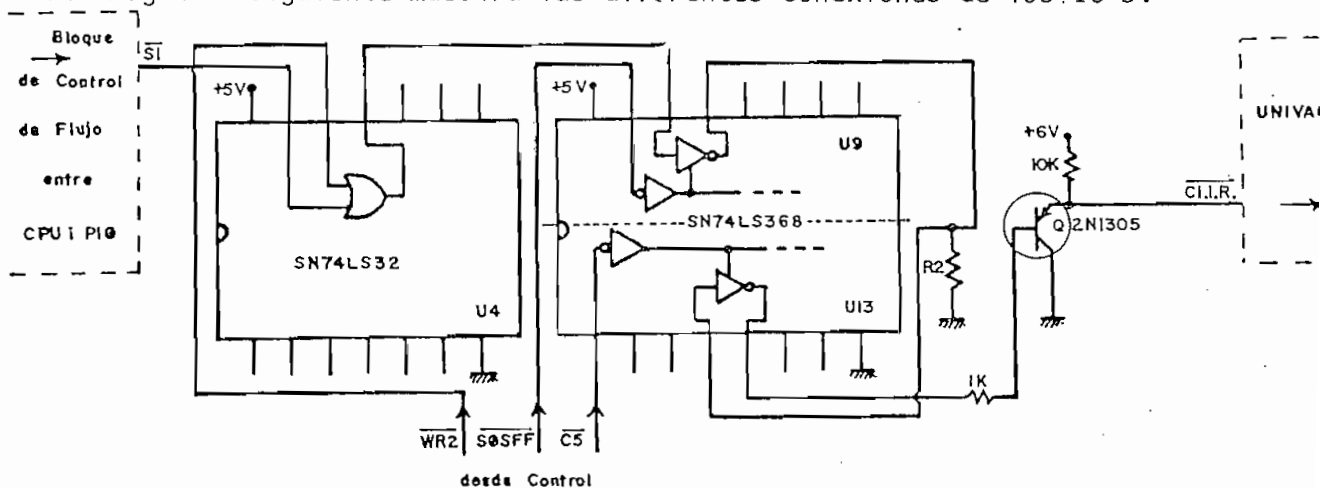


Figura 3.3.6.2. Detalle de las conexiones del circuito anterior

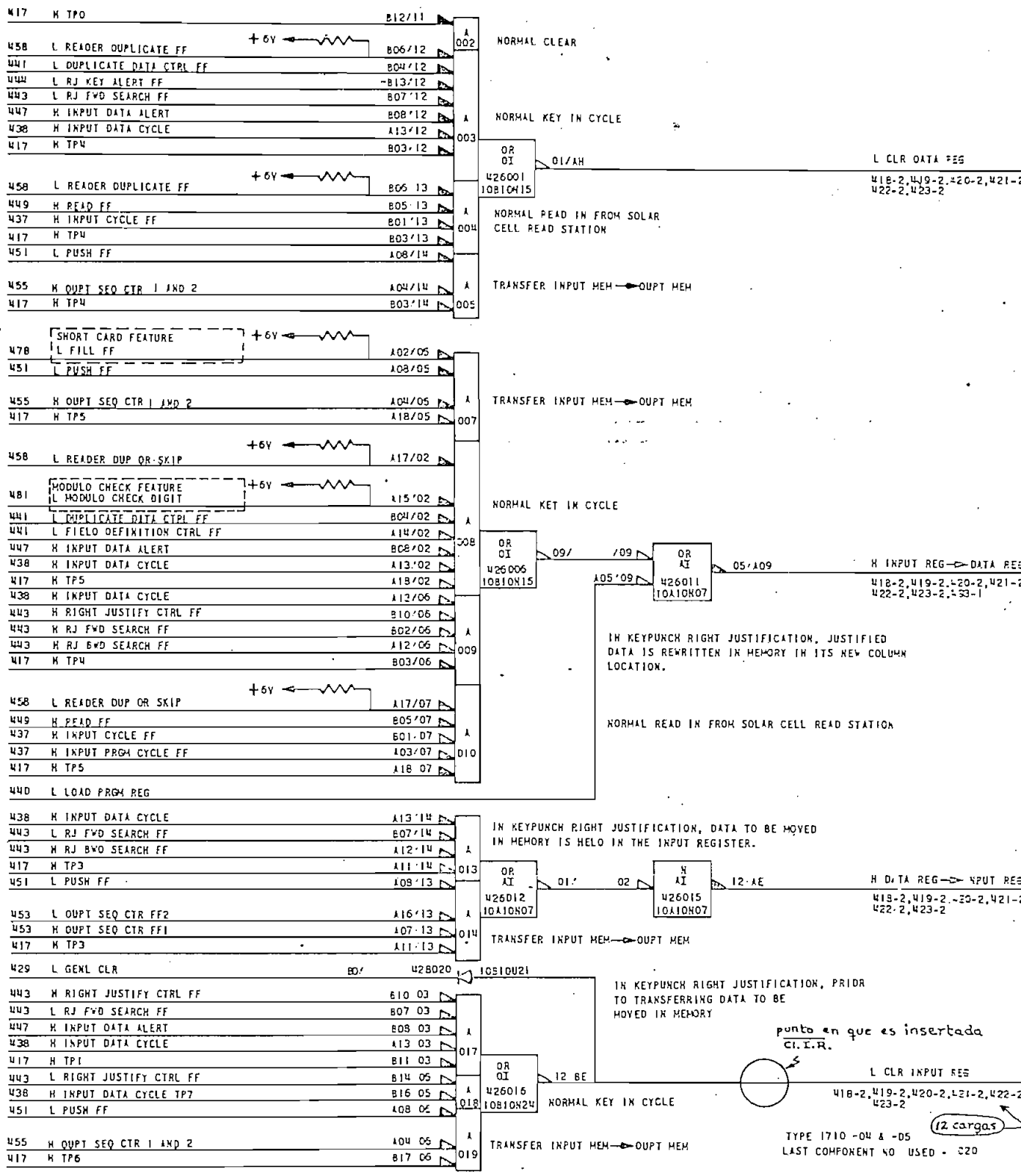


Figura 3.3.6.3. Señales de control para el encaminamiento de datos y borrado en UNIVAC.

### 3.3.7. BLOQUE DE ALIMENTACION DE NUEVA FICHA Y DE BORRADO GENERAL

Este bloque permite al Interfase realizar dos funciones que son:

- la alimentación de una nueva ficha en UNIVAC una vez que los datos de la anterior ficha han sido ya procesados debidamente; y,
- el borrado general de UNIVAC cuando se tenga una condición de error.

Ambas funciones equivalen a la digitación de las teclas *FEED* y *CLR* desde el teclado.

En la siguiente figura se muestra la configuración de este bloque.

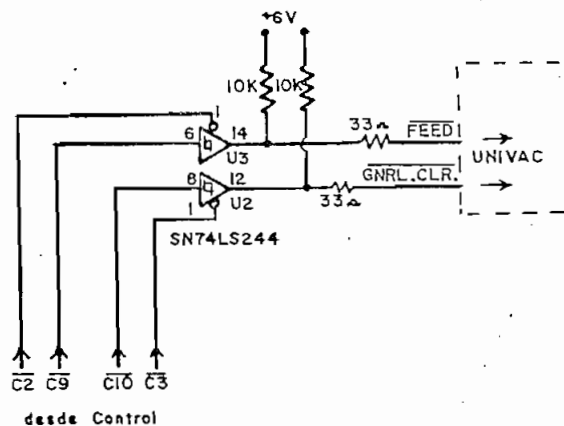
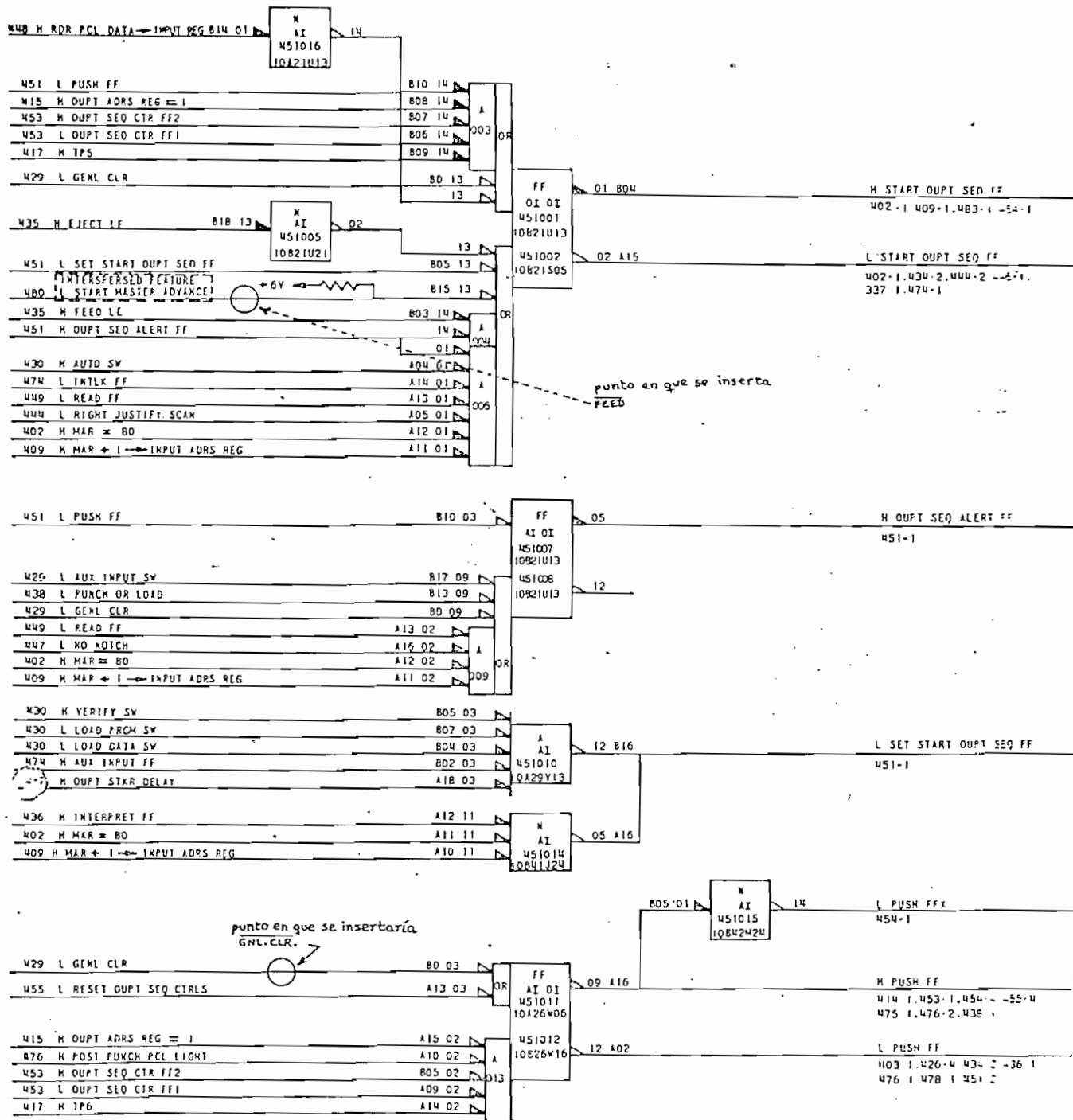


Figura 3.3.7.1. Bloque de Alimentación de Nueva Ficha y Borrado General





TYPE 1710 04 & 05  
LAST COMPONENT NO USED 016

Figura 3.3.7.2. Circuitos para el inicio de la secuencia de salida de datos.

ininterrumpidamente la transferencia de datos en la Primera Fase de UNIVAC; esto hace inservibles a las líneas A RDY y B RDY de PIO para el handshake con UNIVAC y por ello se les ha sido asignada otra función particular en el Interfase. Estas líneas van a conseguir el handshake pero con Z80-CPU.

Repasando, A RDY se activa luego de que Z80-CPU escribe en la puerta A un dato; B RDY se activa luego de ser leída la puerta B por Z80-CPU. Ambas señales regresan a su estado inactivo con una transición positiva de  $\overline{A\ STB}$  y  $\overline{B\ STB}$  respectivamente.

Si se toman a las líneas A RDY y B RDY para generar una condición de espera ( $\overline{WAIT}$ ) mientras se mantengan activadas y se aplica en  $\overline{A\ STB}$  y  $\overline{B\ STB}$  el pulso  $\overline{WR2}$  -fig. 1.6.2-, resultará:

- que el dato es escrito en PIO o en UNIVAC en el instante preciso; y
- que Z80-CPU podrá tener acceso a las puertas de PIO, solo después de transferido un nuevo dato de UNIVAC.

Por lo tanto se habrá creado el handshake para el Interfase, que, en adición, atendería a alguna eventual interrupción únicamente después de laborado el último dato, sin interferir

en el proceso global.

### 3.3.9. BLOQUE DE CONTROL LOGICO

En todos los bloques estudiados se habrán observado ciertas líneas que partiendo del Bloque de Control, se introducen en los mismos.

El estudio de estas líneas se lo hace en gran parte por medio de gráficos; sea los que se presentan en este punto, los vistos en los bloques anteriores, o el Diagrama General Semi-extendido de la fig. 3.4. en donde este "Bloque" se encuentra en forma distribuída.

#### 3.3.9.1. SEÑALES PROVENIENTES DE Z80-CPU

Como se ve en la figura siguiente, algunas señales son alteradas ligeramente para sustituir a sus correspondientes del Bus S-100 estándar (ver Anexo A.3.5 y A.4.2).

Todas estas señales han sido ya estudiadas en la parte 2.1.

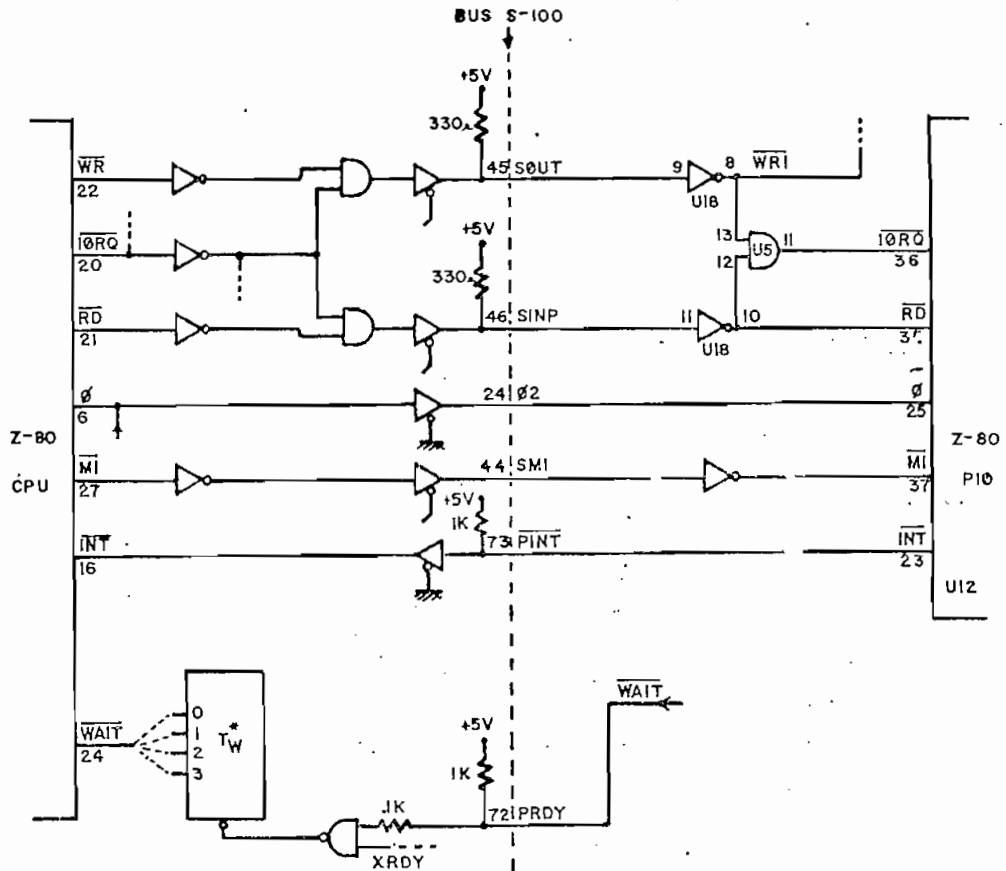


Figura 3.3.9.1. Señales del Bus S-100 "adaptadas" para el Interfase.

### 3.3.9.2. DIRECCIONES DEL INTERFASE

Las diferentes funciones del Interfase son "vistas" por Z80-CPU como localidades periféricas (puertas).

Debe tenerse en cuenta que las líneas que se miran en el gráfico son en realidad sus equivalentes.

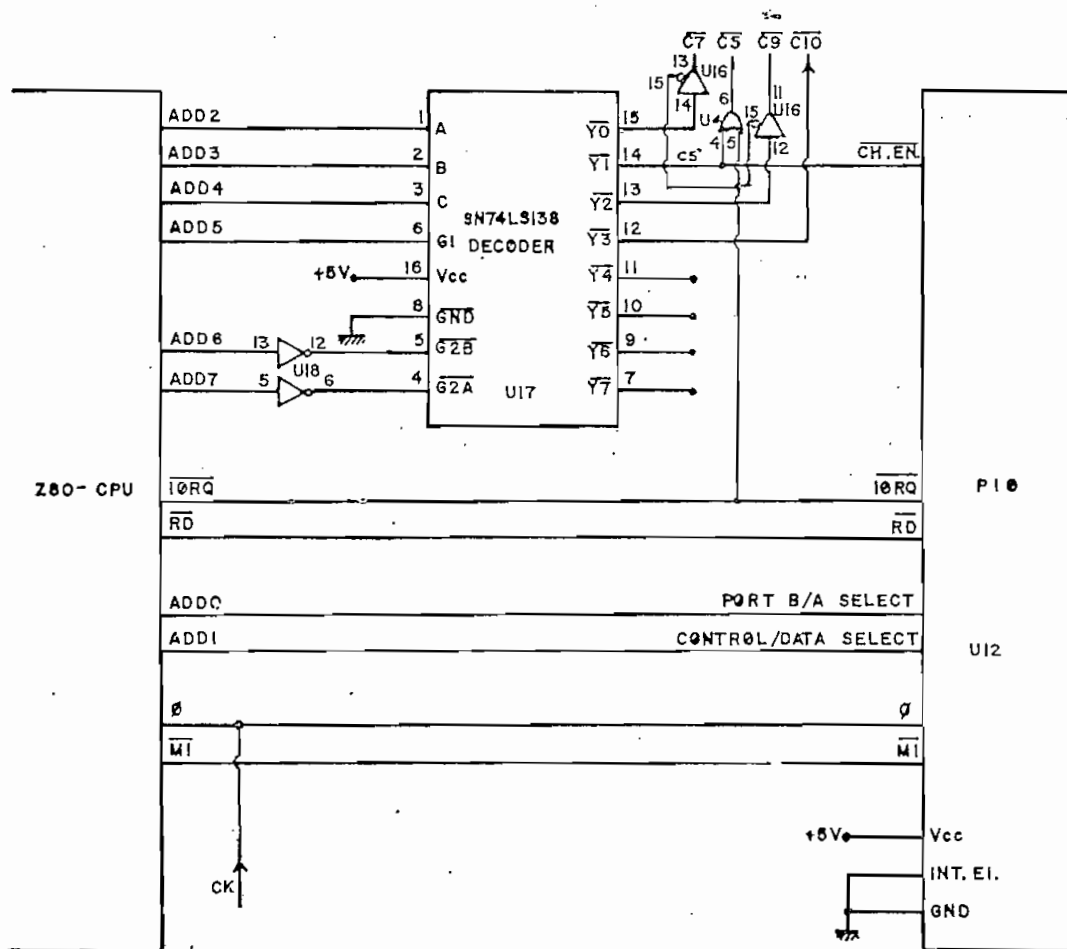


Figura 3.3.9.2. Señales de direccionamiento del Interfase.

$\overline{IORQ}$	BITS DE DIRECCION A7                      AO	CORRES- PONDEN (H.)	SE TOMA	FUNCION	LINEA DE CONTROL
0	1 1 1 0 0 0 X X	E0 → E3	E3	Status de UNIVAC	$\overline{C7}$
0	1 1 1 0 0 1 0 0	E4	E4	Dato puerta A	$\overline{C5}$
0	1 1 1 0 0 1 0 1	E5	E5	Dato puerta B	$\overline{C5}$
0	1 1 1 0 0 1 1 0	E6	E6	Control puerta A	$\overline{C5}$
0	1 1 1 0 0 1 1 1	E7	E7	Control puerta B	$\overline{C5}$
0	1 1 1 0 1 0 X X	E8 → EB	E8	"FEED"	$\overline{C9}$
X	1 1 1 0 1 1 X X	EC → EF	--	"GNRL.CLR:"	$\overline{C10}$

Tabla 3.3.9.2. Direcciones en el Interfase

Como se advierte, las direcciones van de E0 a EF (16 puertas) aunque en la práctica se ocupan 6 ( $\overline{GNRL.CLR.}$  no ha sido tomada).

La línea  $\overline{C5}$  es la más importante de las líneas de control, pues ella "habilita" a los bloques de Decodificación, Encodificación, Control de Flujo hacia el Bus S-100 y Borrado de Registro de UNIVAC.

### 3.3.9.3. IMPOSICION DEL ESTADO DE ESPERA

Concretando lo dicho ya en el punto 3.3.8.1., el circuito que se encarga de imponer el estado de espera es el siguiente:

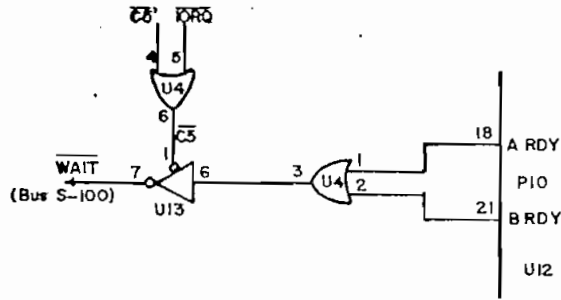


Figura 3.3.9.3. Circuito que impone el estado de espera.

La activación de  $\overline{\text{WAIT}}$  ocurre si:

- i)  $\overline{\text{C5}} = 0$  (Z80-CPU está trabajando con el Interfase).
- ii)  $\left. \begin{array}{l} \text{A RDY} = 1 \\ \text{B RDY} = 1 \end{array} \right\} \text{(Ha sido ya escrito o leído un dato de PIO)}$

Solamente la desactivación de A RDY o B RDY, según el caso, sacará a Z80-CPU del estado de espera y le permitirá la operación de salida o entrada de un dato nuevo.

#### 3.3.9.4. HABILITACIONES SEÑALADAS POR UNIVAC

Es importante aquí volver a revisar el diagrama de tiempos de UNIVAC de la figura 1.6.2.

El instante mismo en que se escribe el dato en la Memoria de

Entrada de UNIVAC se tiene cuando se cumplen simultáneamente las siguientes condiciones:

$\overline{\text{SOSFF}} = 0$  (se realiza la Primera Fase de la Sec. Automática)

$\text{FF1} \cdot 2 = 1$  (es el "ciclo de escritura en memoria"; el tercero de cuatro ciclos).

$\overline{\text{WR2}} = 0$  (semiperíodo de habilitación de la memoria).

La señal  $\overline{\text{INH}}$  acompaña a  $\overline{\text{WR2}}$  con un período algo mayor. Resulta por esto ser muy útil.

El siguiente gráfico considera los tiempos necesarios para la respuesta de EPROM y PIO; también los tiempos de retardo que sufren los datos tanto en lectura como en escritura.

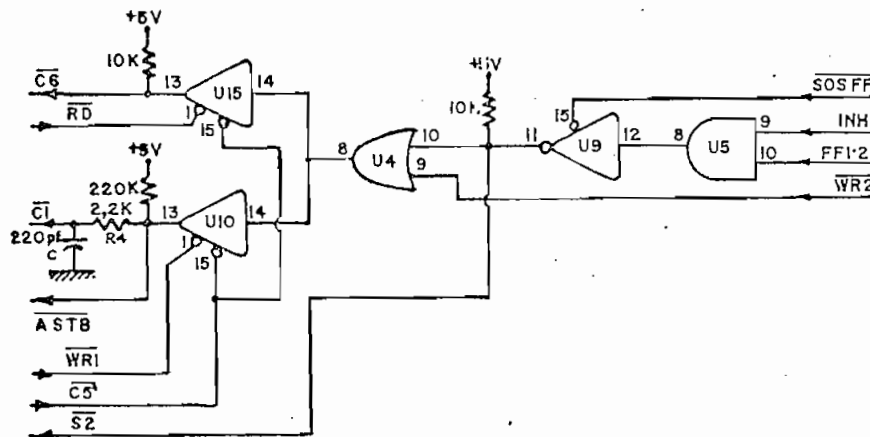


Figura 3.3.9.4.1. Señales de Habilidad de EPROM, puertas de PIO, Flujo de Datos hacia EPROM y Flujo de Datos hacia UNIVAC.



$\overline{S2}$  se anticipa a  $\overline{WR2}$  con  $2.2 \mu s$ . Habilita a EPROM.

$\overline{A STB}$  y  $\overline{C6}$  son excluyentes.  $\overline{C6}$  está conectado a  $\overline{B STB}$  y al Control de Flujo de Datos hacia EPROM. Una transición positiva en ellas desactiva a A RDY o B RDY y levanta el estado de espera.

$\overline{C1}$  habilita a los buffer-drivers de salida y se activa luego de un tiempo de retardo adicional dado por R4 y C (aproximadamente  $0,6 \mu s$ .) con el fin de que los datos a su entrada se hallen ya estabilizados para ser enviados a los flip-flops del Registro de Entrada de UNIVAC.

Para el estado de alta impedancia de los drivers se ha asegurado un  $1_L$  por medio de las resistencias conectadas a + 5V.

En la siguiente figura se observa la línea  $\overline{C8}$  que controla el sentido de la conversión de códigos (A7 de EPROM).

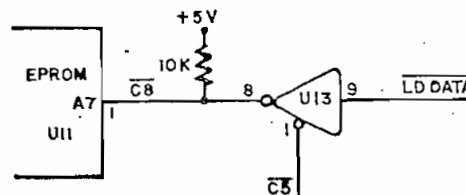


Figura 3.3.9.4.2. Control del sentido de conversión de los caracteres.

Si:

$\overline{LD DATA} = 0, A7 = 1; \Rightarrow$  IBM Hollerith  $\rightarrow$  ASCII

$\overline{\text{LD DATA}} = 1, \quad A7 = 0; \Rightarrow \text{ASCII} \rightarrow \text{IBM Hollerith}$

Se ha dispuesto además de un par de led's que dan a conocer en forma visual de algunas condiciones críticas que pueden suceder en el funcionamiento del Interfase.

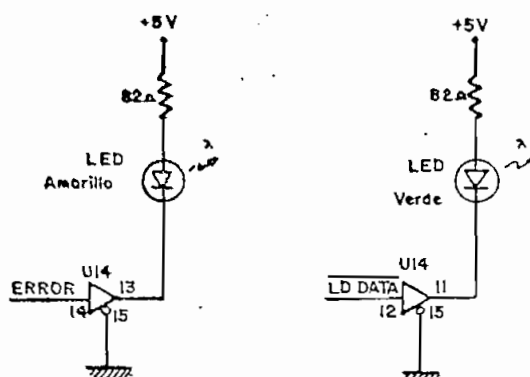


Figura 3.3.9.4.3. Leds para control visual

- Led amarillo encendido implica:
  - cables conectores en orden,
  - UNIVAC prendida,
  - no hay condición de error en UNIVAC.
- Led verde encendido implica:
  - UNIVAC en modo de lectura de datos.

El resto de las líneas del Bloque de Control que no han sido

estudiadas en este punto pueden ser observadas y analizadas en el diagrama del siguiente punto.

### 3.4. INTERCONEXION DE BLOQUES

Puede decirse que el Diagrama General Semi-extendido de la figura 3.4 es el resúmen de todo el circuito electrónico del Interfase y por lo mismo, es el diagrama más importante.

### 3.5. PROGRAMACION

Para la programación se habrán de considerar los siguientes aspectos:

- a) inicialización de PIO,
- b) alistamiento de la zona de memoria a/de la cual se transfieren los datos,
- c) inicialización de los registros de Z80-CPU,
- d) instrucciones de Z80-CPU
- e) control del proceso.

a) La inicialización de PIO consiste en:

- poner la puerta A en el modo 0,
- poner la puerta B en el modo 1,
- leer y desechar el primer dato de la puerta B, con lo que se prepara además a PIO para el handshake del sistema.

No hace falta poner la máscara para interrupciones en PIO.

b) La zona de memoria con la que se trabaja debe ser tratada según el modo de operación del Interfase y el desarrollo del proceso. Puede componerse de 80 localidades como de un número mayor.

c) La inicialización de los registros de Z80-CPU tiene que ver con las instrucciones utilizadas.

d) Las instrucciones INIR y OTIR de entrada y salida automática en bloque, sirven muy bien para el objeto.

En estas instrucciones, el contenido de los registros de Z80-CPU es el siguiente:

Regs. HL → dirección de memoria de/a donde se transfieren los datos.

C → dato que sale o entra.

B → número de datos a transferirse -en hexadecimal-.

Automáticamente:

HL aumentan en 1 cada vez que se transfiere un dato.

B disminuye en 1 cada vez que se transfiere un dato. Si  $B \neq 0$  entonces un siguiente dato es transferido; si  $B = 0$  se rompe el lazo.

Los siguientes son programas en lenguaje mnemotécnico para la lectura y escritura de los 80 datos de un registro.

<u>Lectura de 80 datos</u>		<u>Escritura de 80 datos</u>
Ld HL, 1601 <sub>H</sub>	(dirección memoria = 5.633 <sub>10</sub> )	Ld HL, 1601
Ld B, 50 <sub>H</sub>	(contador = 80 <sub>10</sub> )	Ld B, 50
Ld C, E5 <i>← DPMO</i>	(dirección de la puerta)	Ld C, E4
Out E8, A <i>← DPMO</i>	("FEED")	Out E8, A
INIR		OTIR
RETURN		RETURN

Tabla 3.5.a. Programas en lenguaje mnemotécnico para lectura y escritura de una tarjeta.

e) Por último, al tratarse de una serie de tarjetas, Z80-CPU mantendrá un control a lo largo del proceso y para esto se sirve del status de UNIVAC. La información leída antes de la transferencia de una tarjeta es fundamental para el buen desarrollo de las operaciones del software.

### 3.5.1. EL STATUS DE UNIVAC

Sirve de referencia para la programación de los diferentes modos de operación del interfase.

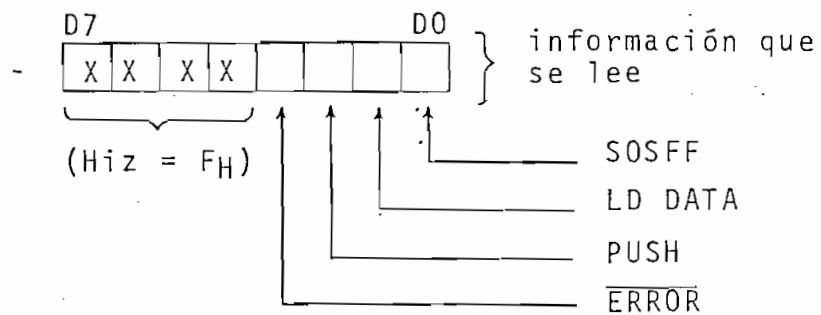


Figura 3.5.1. El status de UNIVAC

Concluída la transferencia de un registro o tarjeta,

- SOSFF regresa a su estado inactivo; pasa de 1 a 0,
- PUSH FF se activa (pasa de 0 a 1) y queda así hasta que la tarjeta termine de pasar por la estación visible. Si se transfiere desde teclado, permanece activado menos tiempo

-aparecerá luego una condición de Error-.

-  $\overline{\text{ERROR}}$  al activarse (0) "avisa" de cualquier anomalía en el proceso como ser: "agotadas las tarjetas", "tarjeta obstruída", "lleno el cajetín de salida", etc.

En las operaciones de lectura y perforación, el regreso de PUSH FF a 0 señala el momento en que se puede alimentar ya una siguiente tarjeta.

En el modo de terminal de teclado, la señal  $\overline{\text{ERROR}}$  permanecerá activada hasta que el operador haya digitado los datos y levantado la condición de error; solo entonces Z80-CPU procederá a la lectura del ítem de datos digitados y almacenados en memoria.

Por último, LD DATA será revisada antes de iniciar las operaciones para asegurar el modo correcto en que se encuentra UNIVAC:

LD DATA = 0 → perforadora de tarjetas o terminal de teclado

LD DATA = 1 → lectora de tarjetas.

C A P I T U L O IVCONSTRUCCION PRACTICA

El llevar a la práctica todas las proposiciones teóricas desarrolladas hasta aquí en los tres capítulos anteriores, siempre lleva consigo un atender a diversos otros factores que no deben descuidarse. Un repaso de las más importantes se da en los puntos que contiene el presente capítulo.

4.1. VENTAJAS DEL WIRE-WRAP

El "Wire-wrap" (*alambre enrollado*) es un método nuevo que se está utilizando cada vez más en la construcción de tableros que contienen circuitos electrónicos. Es ciertamente muy recomendable sobre todo para diseños experimentales.

Entre las ventajas que presenta el Wire-Wrap están:

- Puede acomodar prácticamente a cualquier número de interconexiones dentro de una placa, evitando la complicada construcción de *tableros impresos* con varios niveles de estratos.
- Facilita grandemente la realización de cambios en las conexiones.



- Tiene una "duración esperada de vida" de unos 40 años - mucho más de lo que tendría la misma unión hecha con suelda. Esto se debe a que las uniones, mejor dicho, los cortes - del alambre en las esquinas del pin (aproximadamente  $7 \times 4 = 28$  cortes para 7 vueltas) evitan la oxidación de las juntas y además se ha probado que con el uso, ambas se van fusionando.
- No presenta problemas en diseños que utilizan frecuencias de hasta aproximadamente unos 10 MHz.
- El "*Just-Wrap*", que es una variante del Wire-Wrap y que ha sido utilizado en el presente trabajo, permite la conexión "en cadena", de manera que con un mismo alambre se pueden conectar muchos pines sin necesidad de hacer cortes. En el Just-Wrap el alambre se envuelve forrado en el pin, pelándose solamente en la zona de contacto con éste.

Para utilizar el Wire-Wrap es necesario que los ICs se encuentren insertados en zócalos apropiados. Aquí se utilizaron zócalos de 3 niveles (teóricamente para 3 conexiones distintas de 7 vueltas cada una). También se utilizó un buen número de pines individuales para las resistencias, diodos, etc., tanto como para tomar las señales provenientes del Bus S-100 que terminan en orificios en la placa *Wrap-Board* utilizada.

Antes de realizar el Wire-Wrap, se pegaron sobre los zócalos pedazos de papel señalando la naturaleza del chip y sus pines, luego se iba comprobando cada una de las conexiones introduciendo para ellos pines sobre los zócalos según el circuito y midiendo con un multímetro la resistencia; si esta difería de  $0\Omega$ , entonces debía revisarse la conexión (algún pin conectado equivocadamente).

Resumiendo, la construcción del circuito en esta forma fue relativamente sencilla y rápida.

#### 4.2. DISTRIBUCION DE LOS ELEMENTOS EN LA "PLACA-INTFASE"

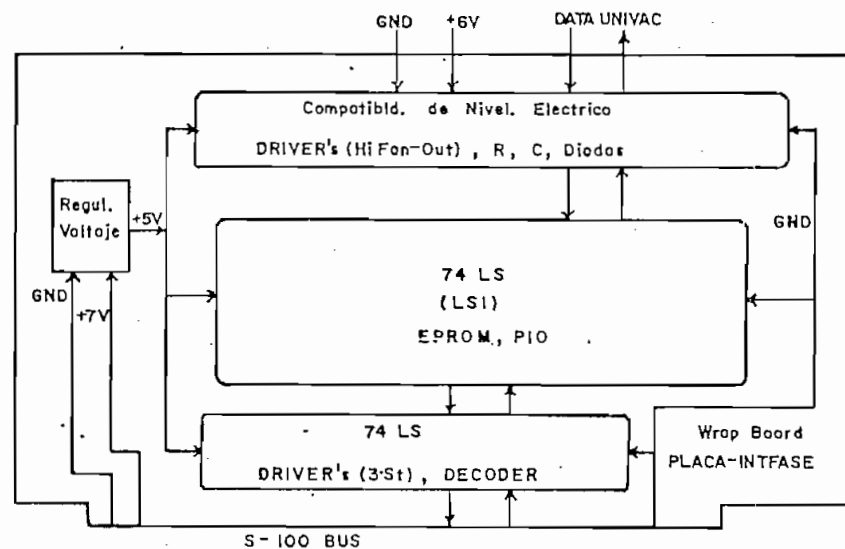
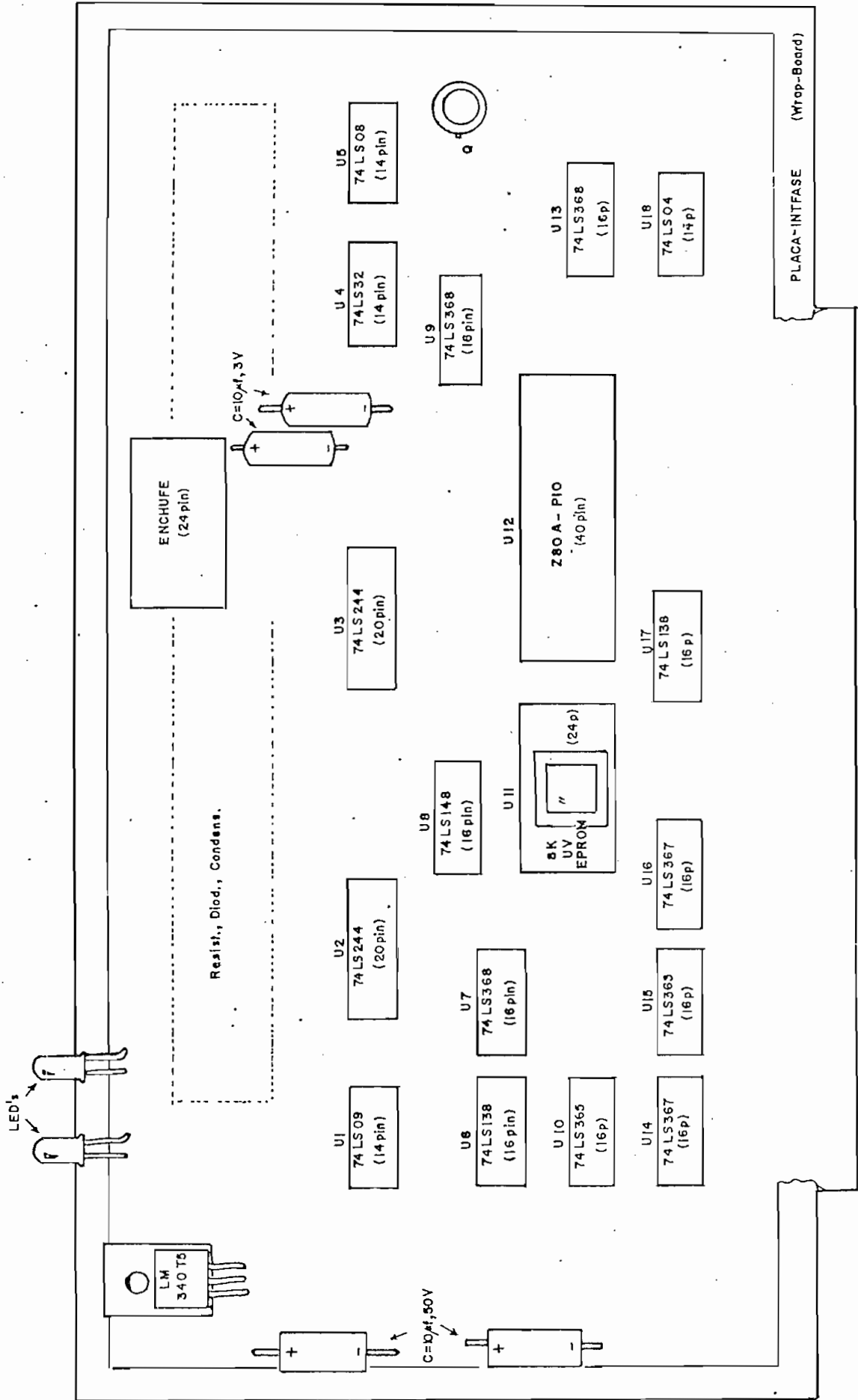


Figura 4.2.1. Diagrama general de la distribución de los elementos en la PLACA-INTFASE.



#### 4.3. EXTENDER<sup>2</sup> CARD Y PROBADOR LOGICO

El "Extender Card" o "tablero de extensión" es una placa de mica simple (no contiene más elementos que tres fusibles) cuya función es la de prolongar las 100 Líneas del Bus S - 100 y posibilitar de esta manera el tomar mediciones y efectuar pruebas en otra tarjeta conectada a su extremo superior, mientras el computador se encuentra trabajando.

Con el fin de hacer más útil al Extender Card, se ha diseñado e implementado luego en él un circuito al que se lo ha denominado "Probador Lógico" - diagramado en la figura 4.3-.

La función del Probador lógico es determinar el nivel lógico (1, 0 ó Hiz) que posee determinado punto de un circuito; así, de acuerdo al led que se encienda, el nivel lógico será:

LED AMARILLO	LED ROJO	NIVEL LOGICO
Encendido	Apagado	0
Apagado	Encendido	1
Encendido	Encendido	HIZ
Apagado	Apagado	? (desenchufada la punta de prueba)

Tabla 4.3. Nivel indicado por el Probador Lógico.

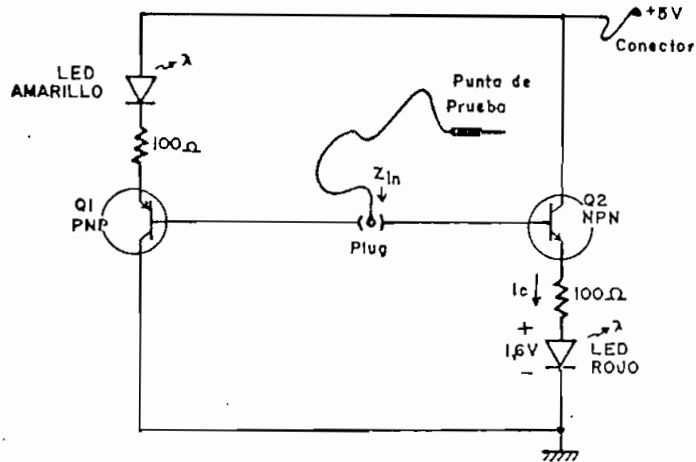


Figura 4.3. Circuito electrónico del Probador Lógico.

Los transistores -complementarios- están conectados como *seguidores de emisor* con el fin de incrementar la impedancia de la punta de prueba y así no introducir mayor carga en el circuito del que se va a detectar su nivel lógico. Seguidamente se muestra un par de cálculos sobre el circuito de la figura 4.3.

$Q_1, Q_2$  complementarios

$$\beta \approx 100$$

$$Z_{in} \approx R_E$$

$$\approx 100 (100)$$

$$\approx 10 \text{ K}$$

$$I_C \approx \frac{5V - 0.2V - 1.6V}{100 \text{ V}}$$

$$I_C \approx 32 \text{ mA}$$

En las pruebas experimentales ha resultado este Probador de

bastante utilidad. Su construcción es muy rápida y su costo, módico.

#### 4.4. INSTALACION

En esta sección se analizan algunos factores inherentes a la construcción física y puesta en funcionamiento de un sistema que tiene uno o más periféricos. Los siguientes puntos han sido repasados:

- ruido e interferencia,
- alimentación y puesta a tierra,
- conectores y enchufes.

##### 4.4.1. RUIDO E INTERFERENCIA

Los aspectos de ruido e interferencia deben ser tomados muy en cuenta en el diseño. La serie de factores involucrados en la inducción de ruido en el cable y la interferencia de radiofrecuencias en el envío de las señales son bastante complejos y pueden no ser entendidos en primera instancia. Es así como en muchas ocasiones puede suceder que el sistema funcione correctamente en un sitio, pero al ser trasladado a otro sitio ya no lo haga.

En realidad tales factores pueden explicarse por medio de los fenómenos electro-magnéticos y ecuaciones particularizadas al caso tratado. Esto, no obstante, resulta siempre bastante laborioso.

En las primeras pruebas experimentales hechas con todo el sistema interconectado, se indujo externamente (cuando se activó casualmente un timbre cercano al cable conector), un tren de pulsos que hizo que se alimentaran imprevistamente las tarjetas. Esto fue solucionado incrementando la resistencia del hilo "FEED" como se ve en la figura 4.4.1.

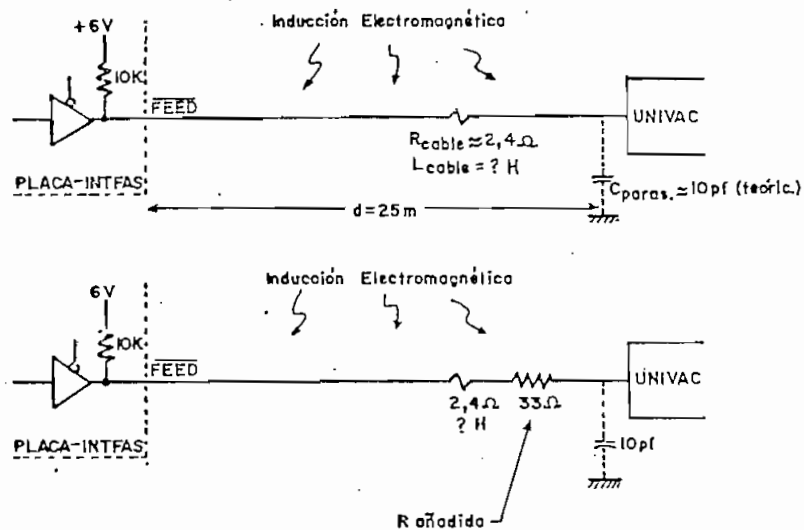


Figura 4.4.1. Aumentando la resistencia del hilo "FEED", para hacerlo menos "sensitivo" a inducciones electromagnéticas del medio.

Es bueno en este punto recordar algunas formas de reducir el ruido, como son:

- blindaje de la línea de transmisión; y
- puesta a tierra del sistema.

Otras técnicas de reducción del ruido son:

- balance de la línea,
- desacoplamiento,
- filtraje.

El cable utilizado en este Interfase es blindado y sus hilos entorchados ("twisted"). Los hilos terminan en un filtro R-C (figuras 3.3.1.3 y 3.3.1.4) bastante eficiente para la eliminación de frecuencias altas (HF, VHF, etc.)

La frecuencia de los pulsos no es muy crítica (0.4 MHz). Las capacitancias parásitas debidas a ésta no tienen mayor influjo y tampoco requieren de un apreciable número de hilos de tierra internamente en el cable (mientras mayor es la frecuencia, mayor número de puntos y conductores de tierra son necesarios).



#### 4.4.2. ALIMENTACION Y PUESTA A TIERRA.

Toda la atención que se ponga en lograr una adecuada *alimentación y tierra* es por demás justificable. Esto evitará futuros problemas relacionados con cambios inexplicables en los datos que se procesan, daños en los ICs, ruidos, etc., ahorrando a la larga, tiempo y dinero.

La alimentación y puesta a tierra se las ha visto desde dos puntos:

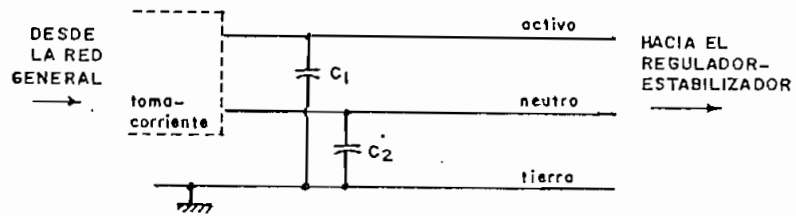
- a) Para el sistema en general; y,
- b) en el Interfase en particular.

a) Para la alimentación del sistema se utilizó un Regulador Estabilizador de Voltaje por *Saturación Magnética*, que precave los problemas de:

- microcortes de energía en la red; y,
- caídas o subidas de tensión.

Además se implementó el circuito de la figura 4.4.2.1. para enfrentar el problema de los sobrepicos transitorios de voltaje.

Para conseguir tierra -masa- en el sistema, tuvo que acu-



$$C_1 = C_2 = 0.01 \mu\text{f} / 2000 \text{ V}_{\text{DC}} \text{ } 1.500 \text{ V}_{\text{AC}}$$

(Condens. de aceite)

Figura 4.4.2.1. Circuito protector a sobrepicos en la fuente.

dirse a la entrada principal del edificio (donde se encuentran las 3 fases y tierra real); desde allí se tendió (hasta el segundo piso) un alambre # 12.

En esta forma se asegura una "buena tierra" ya que el tomar de referencia para ella tuberías de agua, etc., es nada aconsejable (en mediciones hechas en este trabajo, el nivel de voltaje de la tubería variaba entre +0.8 V hasta + 1.3V con respecto al neutro).

Una vez instalada tierra, se detectaron todavía ligeras variaciones de voltaje (ya no críticas) respecto a neutro, causadas por las lámparas fluorescentes del piso, así:

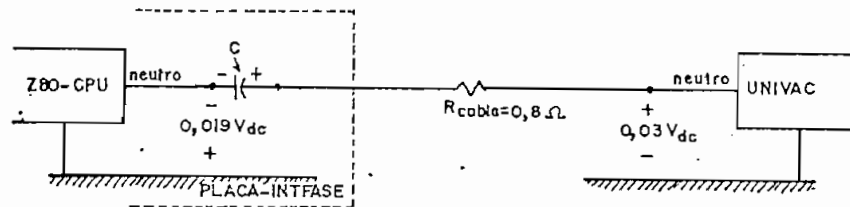
Lámparas apagadas
+ 0.02 V

Lámparas encendidas
+ 0.08 V

- b) En el Interfase, el cable conductor que une a la PLACA - INTFASE con UNIVAC lleva las referencias de alimentación (+ 6 V<sub>DC</sub>) y neutro.

Originalmente, cuando UNIVAC no se encontraba con referencia de tierra (tomacorriente de 2 patas), la diferencia que existía entre los neutros de Z80-CPU y UNIVAC era de aproximadamente + 1.3 V, con una alta componente alterna.

Una vez conectada con tierra, se obtuvieron valores como los de la siguiente figura.



$$C = 10 \mu\text{f}, 3\text{V}, \text{electrolítico}$$

Figura 4.4.2.2: Circuito neutro-tierra del Interfase que muestra el condensador de acoplamiento de neutros.

La línea de tierra consta en realidad de 3 hilos de los 26 que posee el cable conector.

La referencia de + 6 V<sub>DC</sub> (necesaria en la PLACA-INTFASE), fue tomada por medio del siguiente circuito.

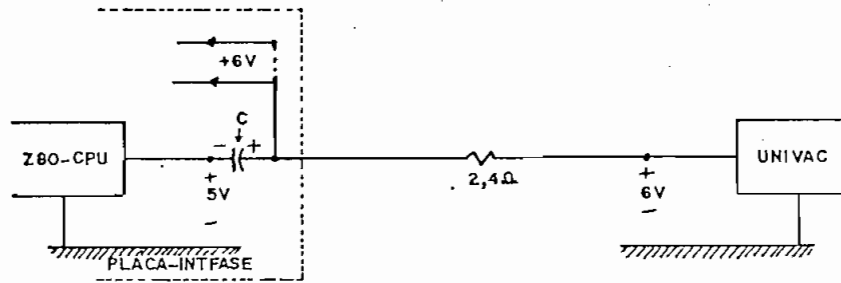


Figura 4.4.2.3. Circuito que muestra la forma como se ha tomado la referencia de +6V desde UNIVAC,

En el momento del encendido va a existir siempre un transitorio que no afectará mayormente, pues los valores de las corrientes involucradas en ambos casos son pequeños.

En lo que respecta a los pines de entrada de señales que no se usan de los ICs, es recomendable fijarlos a tierra o a +5V según se precise; y así se lo ha hecho.

#### 4.4.3. CONECTORES Y ENCHUFES

Las señales tomadas desde UNIVAC han sido conducidas hacia el computador por medio de dos cables, uno de 20 hilos y otro de 6\*. La longitud de éstos es de !25 metros! (bastante más de la recomendada para estos casos, como es 5 m).

\*Aunque es más práctico un solo cable; por causas ajenas a éste han sido utilizados dos.

La longitud del cable hace que se aumente, entre otras cosas, la atenuación de la señal y el tiempo de retardo. Teóricamente, el tiempo total de retardo resulta ser:

$$t_{\text{TOT.RET.}} = t_{\text{RET FILTRO R-C}} + t_{\text{RET LONG CABLE}} + t_{\text{RET. ADICIONAL}}$$

$$\approx 470 \Omega \times 220 \text{ pf} + \frac{25 \text{ m}}{3 \times 10^8 \text{ m/seg}} + 3 \Omega \times 20 \text{ pf}$$

$\uparrow$   $R_{\text{prom. del cable}}$        $\uparrow$   $C_{\text{parás. teórica}}$

$$\approx 0.103 \mu\text{s} + 0.083 \mu\text{s} + 0.0006 \mu\text{s}$$

$$\approx 0.187 \mu\text{s}$$

Los hilos del cable terminan en el un extremo, en 24 "lagartos" (*alligator clips*) que se conectan en la parte posterior del Módulo de Placas de UNIVAC. Para agilizar su conexión durante las pruebas, éstos fueron marcados con etiquetas.

La forma en que entran las 24 señales a la PLACA-INTFASE, es en base a un "Dip Jumper" como se indica en la siguiente figura.\*

---

\* Anexo, fig A.4.1.

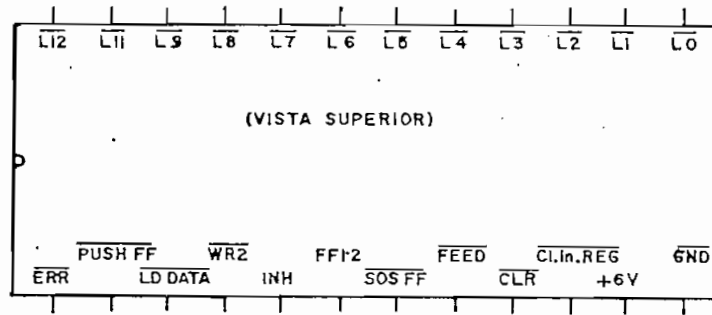


Figura 4.4.3.1. Dip Jumper que se inserta en un zócalo de 24 pines (se muestra la ubicación de las 24 señales).

La resistencia medida en los 24 hilos, de extremo a extremo -incluyendo todas las uniones- fue la siguiente:

HILO	RESISTENCIA	CONSIDERACIONES
$\overline{\text{GND}}$	$R = 0.8\Omega$	3 hilos en paralelo
$\overline{\text{FEED}}$	$R = 36\Omega$	con $R_{\text{adic}} = 33\Omega$
$\overline{\text{CL.IN.REG.}}$	$R = 36\Omega$	con $R_{\text{adic}} = 33\Omega$
los 21 restantes	$R = 2.4\Omega$	

#### 4.4.4. LAS 24 SEÑALES EN UNIVAC

La determinación de los pines de los que se toman las 24 señales de UNIVAC para el Interfase, fue concienzudamente analiza

da. Una lista de ellos que indica su designación en los manuales, su nivel de activación, el número de la lámina en que se encuentra detallado y su localización física, se da a continuación.

DESIGNACION (Nivel lógico de activación)	LAMINA EN EL MANUAL	LOCALIZACION: Placa/Zona A o B y # de PIN
$\overline{\text{DATA REG FF12}}$	5031418	09/A11
$\overline{\text{DATA REG FF11}}$	18	09/A10
$\overline{\text{DATA REG FF0}}$	19	08/A10
$\overline{\text{DATA REG FF1}}$	19	07/A10
$\overline{\text{DATA REG FF2}}$	20	07/A11
$\overline{\text{DATA REG FF3}}$	20	07/A14
$\overline{\text{DATA REG FF4}}$	21	09/A14
$\overline{\text{DATA REG FF5}}$	21	08/A11
$\overline{\text{DATA REG FF6}}$	22	08/A14
$\overline{\text{DATA REG FF7}}$	22	06/A10
$\overline{\text{DATA REG FF8}}$	23	06/A11
$\overline{\text{DATA REG FF9}}$	23	06/A14
$\overline{\text{START OUTP. SEQ. FF}}$	51	21/A15
$\overline{\text{LD DATA}}$	51	29/B04
$\overline{\text{PUSH FF}}$	51	26/A02
ERROR	31	28/B13
$\overline{\text{TP6 (WR2)}}$	03	23/B12
FF1'2	55	26/A18
INH (LONG.T6)	03	11/A06
$\overline{\text{CLR. IN. REG.}}$	26	06/BE
$\overline{\text{START MASTER ADV.}}$ (FEED)	51	21/B15
$\overline{\text{GEN. CLR.}}$	29	-----
+ 6V		-/AF
GND		-/BI

Figura 4.4.4. Las 24 señales de UNIVAC tomadas para el Interfase.

Todas han sido ya explicadas en los capítulos anteriores.



#### 4.5. PRUEBAS

Construida la PLACA-INTFASE, se procedió con las pruebas respectivas. Estas pueden dividirse así:

1. Pruebas con UNIVAC.
2. Pruebas con Z80-CPU y la PLACA-INTFASE.
3. Pruebas con todo el sistema interconetado.

Por cierto, el desarrollo estricto de las pruebas requeriría de ciertos instrumentos de medición como: Osciloscopio de buen ancho de banda, Generador de Señales, Analizador Lógico, ... no obstante, los instrumentos con que se contó - y sirvieron bien - fueron:

PRUEBAS CON: INSTRUMENTOS	UNIVAC	Z80-CPU Y PLACA INTERFASE	Todo el sistema in terconecta do.
multímetro	V		
Extender Card y Probador Lógico.		V	V
el propio microcomputador		V	V
resistencias	V		
cables conectores	V	V	V

Tabla 4.5. Instrumentos utilizados en las pruebas.

#### 4.5.1. PRUEBAS CON UNIVAC

Se resumen en mediciones de voltajes y corrientes en las líneas tomadas para el Interfase; así como también, en la activación artificial (por alambres conectores y resistencias) de ciertas líneas como "FEED" y Datos.

Fueron en general satisfactorias. Permitieron confirmar la factibilidad del uso de los 24 pines de UNIVAC.

#### 4.5.2. PRUEBAS CON Z80-CPU Y LA PLACA-INTFASE

Insertada la PLACA-INTFASE en el Bus del microcomputador, se dió paso a una serie de pruebas. En éstas se detectaron algunos inconvenientes:

- Deficiencia en la suelda de los contactos de los fusibles del Extender Card.
- Falla en la suelda de un pin del Regulador de Voltaje.
- Falla en la programación de EPROM.
- Falla en PIO; la línea "B RDY", indispensable para generar el "WAIT" y con ello el handshake del sistema, no se activó en ningún momento. Realmente éste fue el más gran

de inconveniente.

Se planteó entonces el uso de un circuito alternativo que pudiera reemplazar a B RDY en la imposición del estado de espera.

Los cambios hechos en el circuito de Control Lógico pueden apreciarse en la figura 4.5.2. (compararlo con la figura 3.3.9.4.1).

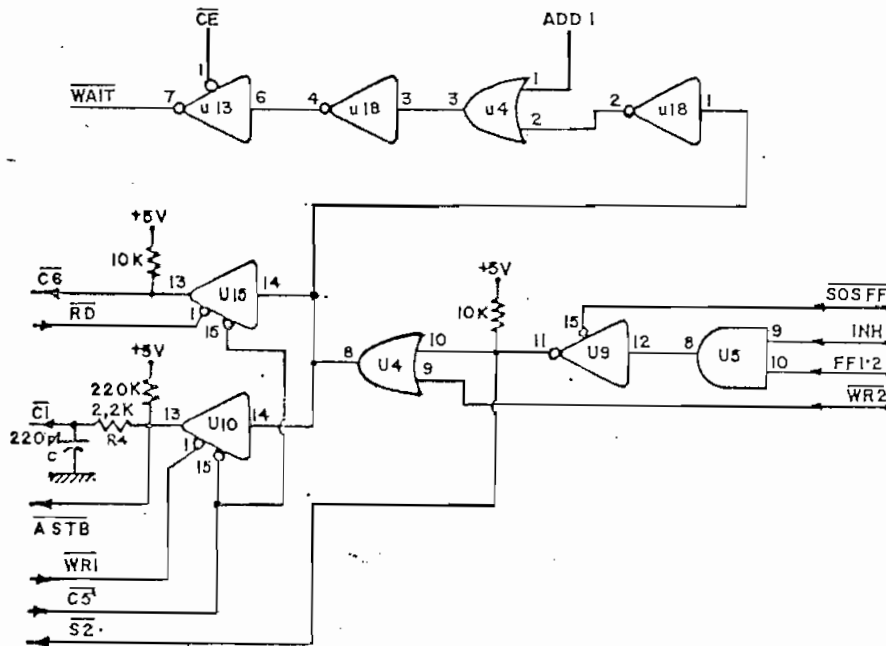


Figura 4.5.2. Circuito alternativo para el control lógico.

- A RDY y B RDY fueron desconectados.
- Los bloques de Encodificación y Decodificación, habilitados permanentemente.

- ADD 1 es la misma línea que entra a C/D SELECT de PIO.

#### 4.5.3. PRUEBAS CON TODO EL SISTEMA INTERCONECTADO

El momento de energizar el sistema ya interconectado, debe cuidarse de seguir la siguiente secuencia: 1° encender el microcomputador y 2° encender UNIVAC. Lo contrario puede producir pulsos transitorios que lleguen a afectar eventualmente en forma nociva los circuitos de UNIVAC.

Hubo el problema (ya descrito en el numeral 4.4.1) de inducción en el cable -línea  $\overline{FEED}$ - que pronto fue salvado.

Con el circuito de control de la Figura 4.5.2. y con Z80-CPU trabajando a  $f = 2$  MHz, se logró exitosamente la lectura de datos desde UNIVAC (de tarjetas y desde teclado). A  $f=4$  MHz se presentó el problema (esperado por cierto) de tiempos de retardo, pues realmente, por la forma como está el circuito de control, Z80-CPU copia directamente el dato que llega a la puerta B de PIO en el instante mismo de la lectura.

El programa que se usó para las pruebas de lectura fue el mismo descrito en la tabla 3.5.a.

Reemplazado PIO y restituido el circuito original, las prue

bas confirmarían el funcionamiento completo del Interfase.

#### 4.6. FUNCIONAMIENTO

Ahora se darán algunos lineamientos a seguir para el manejo del Interfase.

- 1) Revisar la conexión del cable, de la cinta conectora interna del microcomputador y de los lagartos en el Módulo de Placas.
- 2) Para energizar el sistema, no olvidar de encender primero el microcomputador, también los demás periféricos que van a utilizarse y luego UNIVAC.
- 3) Cargar el programa para la operación automática del Interfase.
- 4) Revisar los switches de UNIVAC (ver Tabla 4.6, siguiente página).
- 5) Si el modo es "lectora de tarjetas", alistar las mismas en el cajetín de entrada y alimentar manualmente la primera. Quitar entonces la condición de error que se produce.
- 6) Si el modo es "entrada desde el teclado", no hace falta

MODO SWITCH	LECTORA DE FICHAS	ENTRADA DESDE TECLADO	PERFORADORA DE FICHAS
ZERO/BLANK FILL	Indiferente	Opcional	Indiferente
PRINT/OFF	Opcional	Indiferente	Opcional
PUNCH/VERIFY	Indiferente	PUNCH	PUNCH
LOAD PROG/OFF/LOAD DATA	LOAD DATA	OFF	OFF
AUTO/MANUAL	Indiferente	Opcional	Indiferente
PROG1/PROG2	Indiferente	Opcional	Indiferente

Tabla 4.6. Posición de los Switches de UNIVAC según el modo de operación requerido

cargar tarjetas. Una vez digitados los 80 datos de una tarjeta o registro, debe quitarse la condición de error causada por el anterior ítem de datos enviados (no hace falta digitar FEED pues esto es dado por el microcomputador).

- 7) Si el modo es "perforadora de fichas", ubicar las tarjetas en blanco en el cajetín de entrada de UNIVAC, alimentar manualmente la primera y quitar la condición de error que se produce.
- 8) Lo que sigue del funcionamiento en todos los casos, es tá dado por el software. Este debe incluir la producción de mensajes o indicaciones durante el proceso según lo requieran las circunstancias.
- 9) Para apagar el sistema es preferible apagar primero UNIVAC y luego el microcomputador -mismas razones del encendido-.

#### 4.7. SOTFWARE

La programación que se indicó en el punto 3.5 puede servir de partida para el desarrollo de programas más sofisticados que incluyan una amplia gama de opciones y facilidades para el usuario.

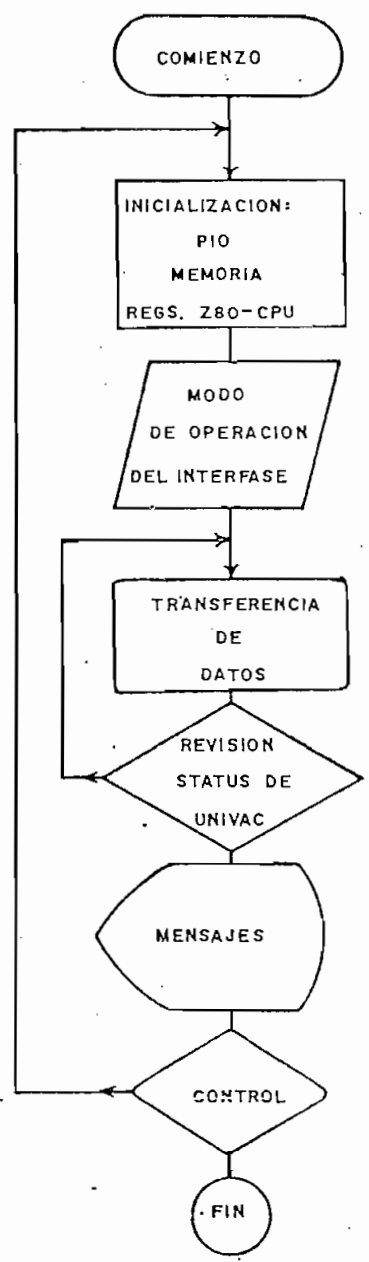


Figura 4.7. Diagrama general de flujo para el Interfase



Esto se ve facilitado en cuanto que el microcomputador puede *ensamblar* (traducir) programas escritos en lenguajes superiores como FORTRAN, BASIC, etc. De hecho, tales programas una vez probados, serán grabados por ejemplo en Disketts, para hacer más fácil su manejo.

La intención de este numeral, más que tratar de la elaboración detallada de los programas, es el de indicar lo útil y simplificado que puede volverse el manejo del Interfase una vez automatizado.

El diagrama de flujo general de la figura 4.7 puede dar una idea de los aspectos, acaso los más importantes, que se toman en cuenta en la creación de los programas.

#### 4.8. MANTENIMIENTO GENERAL

El Interfase en realidad no precisa de cuidados especiales más que los normales, es decir, prevención del polvo en la PLACA-INTFASE y conectores, alejar del cable toda fuente de señales radioeléctricas, guardar el orden del encendido y a pagado del sistema, asegurar los contactos sobre todo en los pines de UNIVAC -lagartos-.

Es mejor, una vez instalado y probado el sistema, soldar los

hilos a los pines de UNIVAC y tener un sólo enchufe para la desconexión o conexión del cable.

Es aconsejable al insertar la PLACA-INTFASE (o cualquier otra) en el Bus S-100 o desenchufarla de él, esperar unos 10 segundos luego de apagado el microcomputador. Nunca hacerlo mientras se encuentre prendido.

#### 4.9. COSTOS

El resumen de lo que significa el costo "teórico" del Interfase (no se toma en cuenta repuestos, transportes, impuestos, depreciaciones, etc.) se ha dado a continuación.

CIRCUITOS INTEGRADOS			
NOMINA CION	TIPO	DESCRIPCION	PRECIO \$
U 1	SN74LS09	QUAD 2 AND GATE O.C.	0,35
U 2	SN74LS244	OCTAL 3-ST BUFFER	1,95
U 3	SN74LS244	OCTAL 3-ST BUFFER	1,95
U 4	SN74LS32	QUAD 2 IN.OR GATE	0,35
U 5	SN74LS08	QUAD 2 IN. AND GATE	0,29
U 6	SN74LS138	3-TO-8 DECODER	0,89
U 7	SN74LS368	HEX 3-ST DRIVER INV	0,75
U 8	SN74LS148	OCTAL PRIOR. ENCODER	2.49
U 9	SN74LS368	HEX 3-ST DRIVER INV	0,75
U10	SN74LS365	HEX 3-ST DRIVER	0,75
U11	2758 EPROM	8K UV INTEL EPROM	7,95
U12	PIO Z-80A	PIO	14,95
U13	SN74LS368	HEX 3-ST DRIVER INV	0,75
U14	SN74LS367	HEX 3-ST DRIVER	0,75
U15	SN74LS365	HEX 3-ST DRIVER	0,75
U16	SN74LS367	HEX 3-ST DRIVER	0,75
U17	SN74LS138	3-TO-8 DECODER	0,89
U18	SN74LS04	HEX INVERTER	0,35
TOTAL			\$ 37,66

\$ 37,66 = S/. 1.318,00\*

\*NOTA: \$ 1 = S/. 35

RESISTENCIAS (1/4 W; 5 % Tol.)			
CANTIDAD	VALOR	PRECIO	
		UNIT.	TOTAL
45	10 K	0,06	2,70
20	470 $\Omega$	0,06	1,20
4	82 $\Omega$	0,06	0,24
3	1 K	0,06	0,18
2	33 $\Omega$	0,06	0,12
1	220 K		0,06
1	2.2 K		0,06
TOTAL \$			4,44 (S/. 155,40)

CONDENSADORES			
CANTIDAD	VALOR, TIPO	PRECIO	
		UNIT.	TOTAL
24	220pf ceramic. 50V	0,06	1,44
2	10 $\mu$ f electrol. 3V	0,40	0,80
TOTAL \$			2,24 (S/. 78,40)
(incluidos en el WRAP-BOARD)			
14	473pf electrol. 50V	0,39	5,46
2	10 $\mu$ f electrol. 50V	0,40	0,80

ZOCALOS (wire-wrap 3 level golded)			
CANTIDAD	# PINES	PRECIO \$	
		UNIT.	TOTAL
1	40		2,29
2	24	1,39	2,78
2	20	1,19	2,38
10	16	0,85	8,50
4	14	0,79	3,16
		TOTAL \$	19,11

(S/.668,85)

DIODOS / LEDS			
CANTIDAD	TIPO	PRECIO \$	
		UNIT.	TOTAL
19	Diodos de Germa- nio 10 mA	0.22	3,18
1	led rojo		0,22
1	led verde		0,22
		TOTAL \$	3,62

(S/.126,70)

1	transistor	\$	0,50
---	------------	----	------

(S/.17,50)

CONECTORES			
CANTIDAD	ESPECIFICACIONES	PRECIO \$	
		UNIT.	TOTAL
1	cable conector blindado, 24 hilos, long. 25 m.		38,61
2	enchufes, macho y hembra, 25 pin con cubierta plástica.		8,00
12	"lagartos"	0,27	3,24
1	cinta de 24 hilos, long. 0,5 m		2,00
1	enchufe para zócalos de 24 pin		1,53
TOTAL \$			53,38

OTRAS HERRAMIENTAS		
CANTIDAD	ESPECIFICACIONES	PRECIO \$
		TOTAL
1	Extender card.	24,95
1	Probador lógico	7,40
1	Extractor de ICs	1,49
1	Insertador de ICs	3,49
8	"lagartos"	3,24
TOTAL \$		40,57

ELEMENTOS PARA EL "JUST-WRAP"		
CANTIDAD	DESCRIPCION	PRECIO \$ TOTAL
1	WRAP-BOARD (tablero para el wire-wrap) incluyendo 1 regulador de voltaje LM340T5 disipador de calor, 14 condensadores distribuidos de 473pf, 50 V; 2 condensadores electrolíticos de 10 $\mu$ f, 50 V.	68,90
1	Enrollador-pelador para el "JUST-WRAP"	25,80
4	Rollos de alambre de 50 pies no térmico para el JUST-WRAP (colores: rojo y verde).	11,92
1	Herramienta para desenvolver y remover el wire-wrap.	3,49
1	Cortador de alambre.	1,00
200	pinos individuales de 3 niveles, dorados.	9,80
		TOTAL \$120,91

GRAN. TOTAL: \$ 282,43 (S/. 9.885,05)
---------------------------------------

# C A P I T U L O V

## COMENTARIOS, RECOMENDACIONES Y CONCLUSIONES

### 5.1. COMENTARIOS

Algunos puntos que merecen ser comentados son:

- El tiempo requerido entre la habilitación de  $\overline{PIO}$ ,  $t_{\overline{CE}}$ , y la activación de su señal  $\overline{IORQ}$ , no consta en la mayoría de folletos y libros utilizados. La no activación de  $\overline{BRDY}$  (parte 4.5.2) hizo necesario el ahondar en detalles de tiempos requeridos entre transiciones. Por último, en ZILOG 1981 DATA BOOK se pudo determinar que tal tiempo debe ser mínimo de 50 nseg.
- La descripción completa y detallada de los diagramas de tiempo para las operaciones del Interfase en su conjunto, no se la ha incluido en razón de que resultaría muy tedioso y complicado para el lector; han sido, sin embargo, estudiados en forma particular.
- La línea  $\overline{M1}$  de  $\overline{PIO}$  no ha sido conectada por razones experimentales.
- En la construcción se han utilizado todas las compuertas



de todos los ICs.

- El hecho de haber diseñado e implementado los circuitos del Interfase en una placa que puede insertarse en el Bus S-100 de cualquier microcomputador, es una ventaja. Es cierto que podía habérselo hecho en la propia UNIVAC pero con muchas complicaciones: construcción especial de la placa, dificultad para su ubicación en el Módulo de Placas, no existencia de una fuente de alimentación de +5V, etc.
- Es de recalcar la eficacia del circuito de envío/recepción de datos, pues tratándose de niveles de entre 0V y 6V y de una distancia periférico-computador no muy usual (long. cable = 25 .m), sin embargo no se han tenido lecturas o escrituras de datos alterados por causa de la transmisión. Por supuesto, el uso de un cable más corto, debidamente en torchado y blindado, asegurará definitivamente la no ocurrencia de errores por transmisión.
- Con el desarrollo del software pueden tenerse muchas ventajas del Interfase; así, poder invertir el contenido de las tarjetas, clasificarlo, arreglarlo, etc. y luego devolverlo en la perforación de nuevas tarjetas. Lamentablemente, la tardansa en la remisión del circuito integrado PIO desde los EE.UU. obligó a efectuar las pruebas con el circuito alternativo de la Fig. 4.5.2. y por tanto, éstas no fue

ron completas, como se habría deseado.

## 5.2. RECOMENDACIONES

Hay formas de mejorar el Interfase, y en esta parte se dan algunas sugerencias al respecto:

- La línea GEN.CLR., no utilizada, puede ser aprovechada con otros fines, por ejemplo, para indicar en la consola de UNIVAC que ésta se encuentra en el modo de terminal periférico
- Siendo que el tiempo efectivo de ocupación de Z80-CPU por parte del Interfase es mínimo (1% del tiempo que le toma a UNIVAC la alimentación de una nueva tarjeta), podría hacerse sin más reparo que el ocupar un poco de memoria de Z80-CPU, la conversión de caracteres en base a software.
- Algunos procesos necesitan almacenar datos en tarjetas multiperforadas y también leer tales datos para guardarlos en cintas, etc. (así "ahorran" espacio y memoria). Sería interesante el diseñar un circuito adicional dentro del Interfase (que puede ocupar alguna dirección libre) que almacene los 12 bits originales de UNIVAC tanto en lectura como en perforación.

Aquí, por supuesto, el software deberá cambiar apreciable-

mente y en lugar de EPROM se utilizaría un par de Registros de Desplazamiento con salida Tristate como lo son los ICs 74LS367 OCTAL D-TYPE LATCHES.

- La transmisión en serie de los datos facilitaría la instalación del Interfase; reduciría las 12 líneas de datos a una sóloa con la posibilidad de transmisión a mayores distancias (y aún la utilización de MODEM's para la transmisión digital por líneas telefónicas).

Para esto se debería diseñar un circuito que irá a residir en UNIVAC, circuito que generará las señales lógicas para el handshake y que en definitiva "acomodaría" a UNIVAC a la forma standard de un terminal periférico que puede enchufarse a la entrada/salida serial de cualquier microcomputador, o aún, computador.

### 5.3. CONCLUSIONES

El Interfase para Microcomputador y máquina VIP de tarjetas UNIVAC realizado en forma teórico-práctica es un tratado detallado que muestra como se pueden utilizar los circuitos integrados y algunos conocimientos de sistemas digitales, para la construcción de un Interfase.

El microcomputador se sustenta en el sofisticado microprocesa

dor Z-80A, cuyos soportes en software le capacitan para correr programas escritos en lenguajes superiores de programación.

VIP UNIVAC 1710 es una máquina perforadora, verificadora e interpretadora de tarjetas Hollerith, fuerte y compacta; su tecnología electrónica (RTL) es diferente a la actual.

Con la implementación del interfase, UNIVAC se convierte en un terminal periférico más del microcomputador, siendo sus funciones las de:

- lectora de datos introducidos desde tarjetas,
- terminal de teclado; y
- perforadora de datos en tarjetas.

El procedimiento seguido en el diseño es lógico y natural. Se empieza por estudiar, en base a bibliografía, cada uno de los sistemas electrónicos que toman parte en el Interfase; se amplían aquellos aspectos que pueden ser aprovechados para el diseño, dándose oportunidad para encauzarlo en diferentes sentidos. En la parte práctica se presenta un material que reúne puntos fundamentales en la construcción de Interfases. La presencia de gráficos y tablas, así como de un buen anexo

complementa y hace del presente, un trabajo que puede ser ex  
tendido y convertido en herramienta para diseños de este ti  
po.

A N E X OA.1. OTRAS CARACTERISTICAS DE UNIVAC VIP 1710

## A.1.1. ASPECTO MECANICO

- Velocidad de perforación: 35 tarjetas por minuto (80 cols).
- Velocidad de verificación: 60 tarjetas por minuto (80 col.).
- Velocidad de interpretación-impresión: 40 tarjetas por minuto.
- Velocidad de salto-duplicación: 80  $\mu$ seg. por columna.

## A.1.2. OPERACION

## CARGA DE FICHAS DE PROGRAMA O DE DATOS

- Poner el interruptor LOAD PROGRAM/LOAD DATA en una de estas dos posiciones, según la función que se quiera realizar.
- Colocar la ficha programa o la ficha de datos en la Entrada Auxiliar y pulsar FEED.
- Poner el interruptor LOAD PROGRAM/LOAD DATA en OFF.

## PERFORACION

- Poner la máquina en posición de PUNCH.
- Si se desea, poner PRINT.
- Seleccionar la posición adecuada de los interruptores AUTO/MANUAL y PROG 1/PROG 2, según el funcionamiento que se desee.
- Pulsar FEED para que la ficha pase a la Estación Visible.
- Pulsar CLEAR.
- Teclear los datos variables.
- Si se usa el modo MANUAL, pulsar la tecla SKIP y luego FEED para expulsar la ficha.

Tabla A.1.2. Resumen de la operación de UNIVAC 1710

### VERIFICACION

- Poner la máquina en posición de VERIFY.
- Si se desea, poner PRINT.
- Colocar el interruptor de modo de funcionamiento en posición AUTO.
- Pulsar la tecla FEED y a continuación la tecla CLEAR.
- Teclar los datos variables.
- Inmediatamente después de la verificación de cada ficha, se produce la alimentación automática.
- Corrección:  
Siempre que se encuentre un error, se puede observar lo siguiente:
  - luz amarilla: volver a teclar los datos.
  - luz amarilla: teclar los datos de nuevo.
  - luz amarilla y luz roja: teclar los datos de nuevo.

En este momento, la memoria ha quedado corregida y puede continuar la verificación. Al terminar de verificar la ficha, no entra ninguna otra ficha nueva hasta que se introduzca una en blanco en la Entrada Auxiliar.

### INTERPRETACION

- Poner la máquina en posición de PUNCH.
- Poner después PRINT.
- Colocar el interruptor de Selección de Programas en PROG 1 o PROG 2.
- Pulsar la tecla de INTERPRET-START para comenzar la alimentación automática.
- Pulsar la tecla INTERPRET-STOP para detener la alimentación automática.

Tabla A.1.2. Resumen de la operación de UNIVAC 1710

A.1.3. CORRECCION DE FICHAS

- Método 1: A utilizar cuando se Interrumpe el trabajo normal de perforación para realizar UNA O MUY POCAS CORRECCIONES.
- Método 2: Para realizar CORRECCIONES DE CAMPO FIJO en un lote de fichas.

METODO 1 - EN MODO PERFORACION, MANUAL Y CARGA DE DATOS.

No es necesario programa de corrección.

- Modo LOAD DATA para introducir la imagen de la ficha incorrecta alimentada desde la Entrada Auxiliar.
- En modo MANUAL se corrige la imagen.
- Las fichas blancas para la corrección se alimentan desde la Entrada Auxiliar, habiendo puesto previamente la máquina en OFF.

METODO 2 - EN MODO VERIFICACION.

- 1) Introducir el programa de corrección. En él se programa un «skip» hasta la primera columna del campo a corregir; esta columna queda en blanco y el resto del campo lleva definición de campo.

FECHA	NUM	CT	DESCRIPCION	DESTINO	TOTAL
000000	0000	00			
111111	1111	11			
222222	2222	22			
333333	3333	33			
444444	4444	44			
555555	5555	55			
666666	6666	66			

- 2) Poner las fichas que se van a corregir en el cajetín de entrada.
- 3) Poner el interruptor en VERIFY (Modo AUTOMATICO).
- 4) Pulsar FEED y CLEAR. Con esto la máquina efectúa el salto hasta la primera columna del campo que se va a corregir.
- 5) Pulsar CORR y teclear los datos correctos.
- 6) Introducir una ficha blanca en la Entrada Auxiliar y pulsar FEED. Con esto se hace la perforación de la ficha correcta.



## A.1.4. PROGRAMACION

## CODIGOS

FUNCION	1ª COLUMNA DEL CAMPO		RESTANTES COLUMNAS	
	Prog. 1	Prog. 2	Progs. 1	2
Cambio alfabético (ALPHA)	1	7	A	<del>4</del>
Cambio numérico (NUMERIC)	b (blanco)	b	12	4
Duplicación (DUPLICATE)	0	6	preferible (0) 12	<del>4</del> (6)
Salto (SKIP) (no usa en verificación)	11	5	12	4
Comienzo de RJ	11-1	5-7	12	4
Fin de RJ	12-11-1 (Prog. 1) ó 4-5-7 (Prog. 2) en la última columna del campo			

Tabla A.1.4.a. Caracteres alfanuméricos y blancos que deben escribirse en la tarjeta-programa.

CODIGOS		MODO AUTO O MANUAL	LUGAR DE PERFORACION	FUNCION		
PROG 1	PROG 2			PERFORACION	VERIFICACION	INTERPRETACION
12	4	A/M	En todas las columnas del campo menos en la primera.	Definición de campo.	Definición de campo.	Definición de campo.
11	5	A	Solamente en la primera columna.	Comienza Salto y No Impresión.	Comienza No Impresión y Salto sin verificación (Bypass).	Comienza No Impresión.
0	6	A	Solamente en la primera columna.	Comienza Duplicación.	Comienza Duplicación.	Comienza Duplicación.
1	7	A	En todas las columnas en que sea necesario.	Cambio alfabético (ALPHA).	Cambio alfabético (ALPHA).	
2	8	A	Solamente en la primera columna.	Comienza la Supresión de Impresión de ceros a la izquierda.	Comienza la Supresión de Impresión de ceros a la izquierda.	Comienza la Supresión de Impresión de ceros a la izquierda.
2	8	A	En todas las columnas en que sea necesario, menos en la primera.	Eliminación de la Impresión de 11/12.	Eliminación de la Impresión de 11/12.	Eliminación de la Impresión de 11/12.
3	9	A	En la primera columna que sigue a la última que se va a utilizar.	Comienza la Alimentación anticipada.	Comienza Bypass y No Impresión.	Comienza la Alimentación anticipada.
11-2	5-3	A	Solamente en la primera columna.	Comienza Salto y No Impresión.	Comienza Bypass y Supresión de ceros a la izquierda.	Comienza No Impresión.
11-1	5-7	A/M	Solamente en la primera columna.	Comienza Justificación a la derecha (RJ).	Comienza RJ.	
12-11-1	4-5-7	A/M	Solamente en la última columna.	Fin de RJ.	Fin de RJ.	Definición de campo.

Tabla A.1.4.b. Códigos de la tarjeta-programa

## A.1.5. PROCESO DE VERIFICACION, CARGA DE DATOS Y LECTURA

VERIFY AND LOAD DATA READ OPERATION

THE READ OPERATION USES FOUR CLOCK CYCLES. THE FIRST MEMORY CYCLE COMPARES AND REPLACES THE DATA IN MEMORY WITH THE DATA JUST READ (IF NOT A LOAD DATA SEQUENCE THE COMPARISON CHECK IS NOT PERFORMED AND THE NEW DATA IS TRANSFERRED DIRECTLY INTO MEMORY.) THE SECOND MEMORY CYCLE IS USED TO PLACE OF MEMORY A MARKER BIT (12, IN THE DATA MEMORY WHICH REFLECTS THE RESULT OF THE COMPARISON MADE IN THE FIRST CYCLE. THE SECOND MEMORY CYCLE ALSO INCREMENTS THE MEMORY ADDRESS REG AND ES:16, 5-15 THE START OF THE POST READ CONTROL OPERATION FOR THE VERIFY AND LOAD DATA OPERATION. THE MEMORY CYCLE FOR THE FIRST COLUMN OCCURS WHEN THE READER SPROCKET COUNTER EQUALS 48 AND THE MEMORY CYCLES FOR SUBSEQUENT CARD COLUMNS OCCUR EVERYTIME THE COUNTER REACHES A COUNT OF 16.

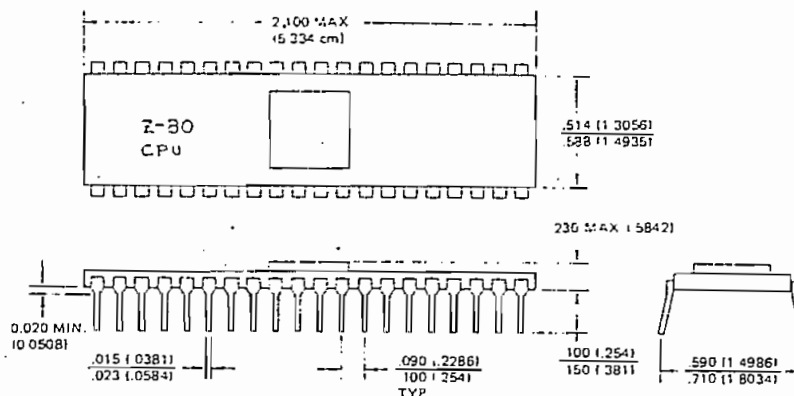
AFTER THE 80 COLUMNS ARE READ, THE TWO ADDITIONAL CLOCK CYCLES ARE USED TO GENERATE 25 ADDITIONAL COLUMN SPROCKETS WITH THE INPUT DATA CIRCUITS DISABLED. THIS COUNT IS USED TO INITIATE THE CARD PATTERN OPERATION IN THE ADVANCE STATION AFTER THE CARD IS PASSED BEYOND THE READ STATION-OUTPUT SPROCKET ROLLS.

NOTE: LOAD DATA READ OPERATES IDENTICALLY FOR KEYPUNCH OR VERIFY OPERATIONS. THE MISMATCH FF REMAINS CLEARED AND THE PRGM LOOK AHEAD CYCLE IS NOT USED AFTER THE READ OPERATION.

	1ST MEMORY CYCLE (PRGM CYCLE)	2ND MEMORY CYCLE (DATA CYCLE)	3RD CLOCK CYCLE	4TH CLOCK CYCLE
ASYN- CHRONOUS	L.E. OF CARD SETS READ FIRST 80 COLUMNS FF (445-014) ACTIVATE READER SPT CLR (446-023) (STARTS SPT COUNT OF 2) SET READ FF (445-020) SET RDR INTERRUPT ALERT FF (445-031) CLR DATA TO INPUT REG (418 THRU 423) SET INPUT PRGM CYCLE FF (437-024) AND INPUT CYCLE ALERT FF (437-007)			
TP 0	SET INPUT CYCLE FF (437-011) INITIATES NEW CYCLE (403-004)	INPUT CYCLE FF REMAINS SET (437)	SET RDR INTERRUPT FF (445-035) WHEN INPUT CYCLE ALERT FF IS CLEARED. CLR INPUT CYCLE AND INPUT PRGM CYCLE FFs IF WAR = 80. PSEK WAR = 80, INITIATE PRGM LOOK AHEAD CYCLE TO SEARCH FOR AUTOMATIC OPERATIONAL CONTROLS WITH POST READ FF SET CLR READ FIRST 80 COLUMNS FF (445-013) WHICH CLEARS READ FF (445-027)	CLR RDR INTERRUPT FF (445-035)
TP 1	INPUT ADRS REG TO WAR (402-019) ADDRESSES INPUT NEW	INPUT ADRS REG TO WAR (402-019) ADDRESSES DUPT NEW (402-004)		
TP 2	GENERATE READ NEW STROBE (403-011) INPUT NEW TO DATA REG (418 THRU 423)	GENERATE READ NEW STROBE (403-011) DUPT NEW TO DATA REG (418 THRU 423)		
TP 3	SET MISMATCH FF ON NON-COMPARE IF NOT IN KEYPUNCH OR LOAD DATA OPERATION (446-017)		INCREMENT DUPT ADRS REG (415-007) (COUNT READER COL SPT)	
TP 4	CLR DATA REG (426-004)	CLR DATA REG (426-004)		CLR POST READ FF (445-012) WHEN DUPT ADRS REG IS CLEARED AT COUNT OF 25 (414-010)
TP 5	INPUT REG TO DATA REG (426-010)	IF MISMATCH FF IS SET (446) GENERATE MISMATCH MARKER — DATA REG FF 12 (445-018) AND SET DATA REG FF 12 (418-007) CLR MISMATCH FF (446-015)		
TP 6	GENERATE WRITE NEW STROBE (403-015)	GENERATE WRITE NEW STROBE (403-015) MARKER BIT TO DUPT NEW (VER. FF 220E)) SET POST READ FF WHEN WAR = 80 (445-018) (ALLOWS 25 ADDITIONAL SPT COUNTS AFTER COMPLETE CARD IS READ) CLR INPUT CYCLE ALERT FF IF WAR ≠ 80 (437-004)	CLR RDR INTERRUPT ALERT FF (445-032)	
TP 7	INPUT ADRS REG NOT INCREMENTED (409-001) CLR INPUT PRGM CYCLE FF AT T.E. OF TP 7 (437-012)	INCREMENT INPUT ADRS REG (409-001) CLR INPUT REG (426-018) SET INPUT PRGM CYCLE FF AT T.E. OF TP 7 (437-012)		

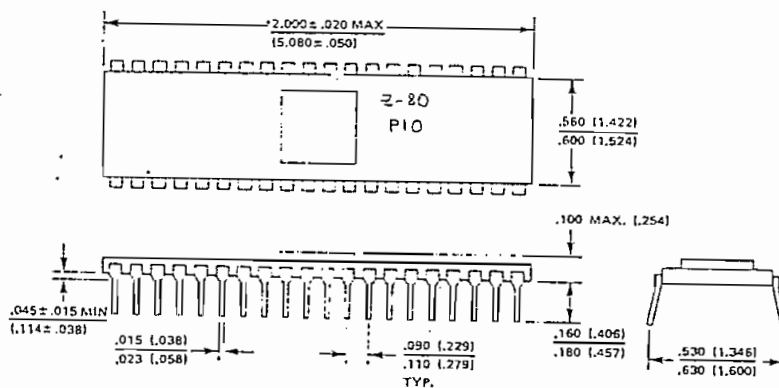
## A.2. OTRAS CARACTERISTICAS DE Z80-CPU y Z80-PIO

### A.2.1. DIMENSIONES FISICAS DE Z80-CPU



\*Dimensions for metric system are in parenthesis

### A.2.2. DIMENSIONES FISICAS DE Z80-PIO

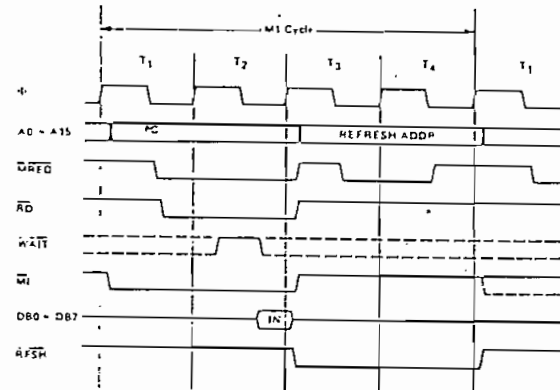


NOTE: Dimensions in parentheses are for metric system (cm).

## A.2.3. OTROS DIAGRAMAS DE TIEMPO DE Z80-CPU

## INSTRUCTION OP CODE FETCH

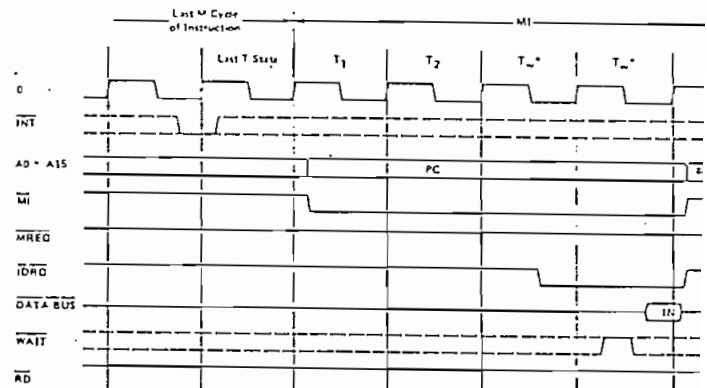
The program counter content (PC) is placed on the address bus immediately at the start of the cycle. One half clock time later  $\overline{MREQ}$  goes active. The falling edge of  $\overline{MREQ}$  can be used directly as a chip enable to dynamic memories.  $\overline{RD}$  when active indicates that the memory data should be enabled onto the CPU data bus. The CPU samples data with the rising edge of the clock state  $T_3$ . Clock states  $T_3$  and  $T_4$  of a fetch cycle are used to refresh dynamic memories while the CPU is internally decoding and executing the instruction. The refresh control signal  $\overline{RFSH}$  indicates that a refresh read of all dynamic memories should be accomplished.



## A.2.3.a. CICLO DE DECODIFICACION DE NUEVA INSTRUCCION

## INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

The interrupt signal is sampled by the CPU with the rising edge of the last clock at the end of any instruction. When an interrupt is accepted, a special  $M_1$  cycle is generated. During this  $M_1$  cycle, the  $\overline{IORQ}$  signal becomes active (instead of  $\overline{MREQ}$ ) to indicate that the interrupting device can place an 8-bit vector on the data bus. Two wait states ( $T_w^*$ ) are automatically added to this cycle so that a ripple priority interrupt scheme, such as the one used in the Z80 peripheral controllers, can be easily implemented.



## A.2.3.b. CICLO DE REQUERIMIENTO/RECONOCIMIENTO DE NUEVA INSTRUCCION

**A.2.4. INSTRUCCIONES (SOFTWARE)**

The following is a summary of the Z80 instruction set showing the assembly language mnemonic and the symbolic operation performed by the instruction. A more detailed listing appears in the Z80-CPU technical manual. The instructions are divided into the following categories:

8-bit loads	Miscellaneous Group
16-bit loads	Rotates and Shifts
Exchanges	Bit Set, Reset and Test
Memory Block Moves	Input and Output
Memory Block Searches	Jumps
8-bit arithmetic and logic	Calls
16-bit arithmetic	Restarts
General purpose Accumulator & Flag Operations	Returns

In the table the following terminology is used.

- b  $\equiv$  a bit number in any 8-bit register or memory location
- cc  $\equiv$  flag condition code
  - NZ  $\equiv$  non zero
  - Z  $\equiv$  zero
  - NC  $\equiv$  non carry
  - C  $\equiv$  carry
  - PO  $\equiv$  Parity odd or no over flow
  - PE  $\equiv$  Parity even or over flow
  - P  $\equiv$  Positive
  - M  $\equiv$  Negative (minus)

- d  $\equiv$  any 8-bit destination register or memory location
  - dd  $\equiv$  any 16-bit destination register or memory location
  - e  $\equiv$  8-bit signed 2's complement displacement used relative jumps and indexed addressing
  - L  $\equiv$  8 special call locations in page zero. In decimal notation these are 0, 8, 16, 24, 32, 40, 48 and 56
  - n  $\equiv$  any 8-bit binary number
  - nn  $\equiv$  any 16-bit binary number
  - r  $\equiv$  any 8-bit general purpose register (A, B, C, D, E, H, or L)
  - s  $\equiv$  any 8-bit source register or memory location
  - sb  $\equiv$  a bit in a specific 8-bit register or memory location
  - ss  $\equiv$  any 16-bit source register or memory location
  - subscript "L"  $\equiv$  the low order 8 bits of a 16-bit register
  - subscript "H"  $\equiv$  the high order 8 bits of a 16-bit register
  - ( )  $\equiv$  the contents within the ( ) are to be used as a pointer to a memory location or I/O port number
- 8-bit registers are A, B, C, D, E, H, L, I and R  
 16-bit register pairs are AF, BC, DE and HL  
 16-bit registers are SP, PC, IX and IY

Addressing Modes implemented include combinations of the following:

Immediate	Indexed
Immediate extended	Register
Modified Page Zero	Implied
Relative	Register Indirect
Extended	Bit

	Mnemonic	Symbolic Operation	Comments
8-BIT LOADS	LD r, s	r ← s	s $\equiv$ r, n, (HL), (IX+e), (IY+e)
	LD d, r	d ← r	d $\equiv$ (HL), r (IX+e), (IY+e)
	LD d, n	d ← n	d $\equiv$ (HL), (IX+e), (IY+e)
	LD A, s	A ← s	s $\equiv$ (BC), (DE), (nn), I, R
	LD d, A	d ← A	d $\equiv$ (BC), (DE), (nn), I, R
16-BIT LOADS	LD dd, nn	dd ← nn	dd $\equiv$ BC, DE, HL, SP, IX, IY
	LD dd, (nn)	dd ← (nn)	dd $\equiv$ BC, DE, HL, SP, IX, IY
	LD (nn), ss	(nn) ← ss	ss $\equiv$ BC, DE, HL, SP, IX, IY
	LD SP, ss	SP ← ss	ss = HL, IX, IY
	PUSH ss	(SP-1) ← ss <sub>H</sub> ; (SP-2) ← ss <sub>L</sub>	ss = BC, DE, HL, AF, IX, IY
POP dd	dd <sub>L</sub> ← (SP); dd <sub>H</sub> ← (SP+1)	dd = BC, DE, HL, AF, IX, IY	
EXCHANGES	EX DE, HL	DE ↔ HL	
	EX AF, AF'	AF ↔ AF'	
	EXX	$\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	

	Mnemonic	Symbolic Operation	Comments
MEMORY BLOCK MOVES	LDI	(DE) ← (HL); DE ← DE+1 HL ← HL+1, BC ← BC-1	
	LDIR	(DE) ← (HL); DE ← DE+1 HL ← HL+1, BC ← BC-1 Repeat until BC = 0	
	LDD	(DE) ← (HL); DE ← DE-1 HL ← HL-1, BC ← BC-1	
	LDDR	(DE) ← (HL); DE ← DE-1 HL ← HL-1, BC ← BC-1 Repeat until BC = 0	
MEMORY BLOCK SEARCHES	CPI	A ← (HL); HL ← HL+1 BC ← BC-1	
	CPIR	A ← (HL); HL ← HL+1 BC ← BC-1. Repeat until BC = 0 or A = (HL)	A ← (HL) sets the flags on A is not affected
	CPD	A ← (HL); HL ← HL-1 BC ← BC-1	
	CPDR	A ← (HL); HL ← HL-1 BC ← BC-1. Repeat until BC = 0 or A = (HL)	
8-BIT ALU	ADD s	A ← A + s	
	ADC s	A ← A + s + CY	CY is the carry flag
	SUB s	A ← A - s	
	SBC s	A ← A - s - CY	s $\equiv$ r, n, (HL), (IX+e), (IY+e)
	AND s	A ← A ∧ s	
OR s	A ← A ∨ s		

8-BIT ALU	CPI s	A ← s	s = r, n (HL) (IX+e), (IY+e)
	INC d	d ← d + 1	d = r, (HL) (IX+e), (IY+e)
	DEC d	d ← d - 1	
16-BIT ARITHMETIC	ADD HL, ss	HL ← HL + ss	$\left. \begin{array}{l} ss \equiv BC, DE \\ \cdot HL, SP \end{array} \right\}$ $ss \equiv BC, DE, IX, SP$ $ss \equiv BC, DE, IY, SP$ $dd \equiv BC, DE, HL, SP, IX, IY$ $dd \equiv BC, DE, HL, SP, IX, IY$
	ADC HL, ss	HL ← HL + ss + CY	
	SBC HL, ss	HL ← HL - ss - CY	
	ADD IX, ss	IX ← IX + ss	
	ADD IY, ss	IY ← IY + ss	
	INC dd	dd ← dd + 1	
DEC dd	dd ← dd - 1		
CPU ACC. & FLAG	DAA	Converts A contents into packed BCD following add. or subtract.	Operands must be in packed BCD format.
	CPL	A ← $\overline{A}$	
	NEG	A ← $\overline{A} + 1$	
	CCF	CY ← $\overline{CY}$	
	SCF	CY ← 1	
MISCELLANEOUS	NOP	No operation	
	HALT	Halt CPU	
	DI	Disable Interrupts	
	EI	Enable Interrupts	
	IM 0	Set interrupt mode 0	8080A mode
	IM 1	Set interrupt mode 1	Call to 0035 <sub>H</sub>
IM 2	Set interrupt mode 2	Indirect Call	
ROTATES AND SHIFTS	RLC s		s = r, (HL) (IX+e), (IY+e)
	RL s		
	RRC s		
	RR s		
	SLA s		
	SRA s		
	SRL s		
	RLD		
	RRD		

INPUT AND OUTPUT	BIT s, r, & 1	BIT b, s SET b, s RES b, s	Z ← s <sub>b</sub> s <sub>b</sub> ← 1 s <sub>b</sub> ← 0	Z is zero flag s = r, (HL) (IX+e), (IY+e)		
	JUMPS	IN A (n)	IN r, (C)	A ← (n) r ← (C)	Set flags	
		INI	INIR	(HL) ← (C), HL ← HL + 1 B ← B - 1 (HL) ← (C), HL ← HL + 1 B ← B - 1 Repeat until B = 0		
		IND	INDR	(HL) ← (C), HL ← HL - 1 B ← B - 1 (HL) ← (C), HL ← HL - 1 B ← B - 1 Repeat until B = 0		
		OUT(n), A	OUT(C), r	(n) ← A (C) ← r		
		OUTI	OTIR	(C) ← (HL), HL ← HL + 1 B ← B - 1 (C) ← (HL), HL ← HL + 1 B ← B - 1 Repeat until B = 0		
		OUTD	OTDR	(C) ← (HL), HL ← HL - 1 B ← B - 1 (C) ← (HL), HL ← HL - 1 B ← B - 1 Repeat until B = 0		
		JP nn	JP cc, nn	PC ← nn If condition cc is true PC ← nn, else continue		cc $\left\{ \begin{array}{l} NZ \\ Z \\ NC \\ C \end{array} \right.$
		JR e	JR kk, e	PC ← PC + e If condition kk is true PC ← PC + e, else continue		
		JP (ss)	DJNZ e	PC ← ss B ← B - 1, if B = 0 continue, else PC ← PC + e		ss = HL, IX, IY
CALLS		CALL nn	CALL cc, nn	(SP-1) ← PC <sub>H</sub> (SP-2) ← PC <sub>L</sub> , PC ← nn If condition cc is false continue, else same as CALL nn		cc $\left\{ \begin{array}{l} NZ \\ Z \\ NC \\ C \end{array} \right.$
	RST L	RET	(SP-1) ← PC <sub>H</sub> (SP-2) ← PC <sub>L</sub> , PC <sub>H</sub> ← 0 PC <sub>L</sub> ← L			
RETURNS	RET	RET cc	PC <sub>L</sub> ← (SP), PC <sub>H</sub> ← (SP+1) If condition cc is false continue, else same as RET			
	RETI	RETN	Return from interrupt, same as RET Return from non- maskable interrupt			

## A.2.5. DIAGRAMAS DE TIEMPOS DE PIO PARA INTERRUPCIONES

During  $\overline{M1}$  time, peripheral controllers are inhibited from changing their interrupt enable status, permitting the  $\overline{INT}$  Enable signal to ripple through the daisy chain. The peripheral with IEI high and IEO low during  $\overline{INTA}$  will place a preprogrammed 8-bit interrupt vector on the data bus at this time. IEO is held low until a return from interrupt (RETI) instruction is executed by the CPU while IEI is high. The 2-byte RETI instruction is decoded internally by the PIO for this purpose.

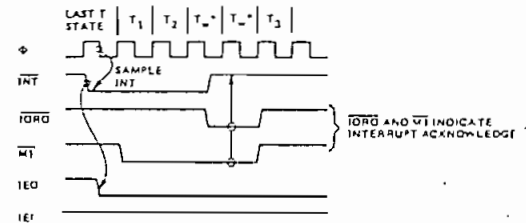


Figura A.2.5.a. Reconocimiento de una interrupción

If a Z80 peripheral device has no interrupt pending and is not under service, then its  $\overline{IEO}=\overline{IEI}$ . If it has an interrupt under service (i.e., it has already interrupted and received an interrupt acknowledge) then its  $\overline{IEO}$  is always low, inhibiting lower priority chips from interrupting. If it has an interrupt pending which has not yet been acknowledged,  $\overline{IEO}$  will be low unless an "ED" is decoded as the first byte of a two byte opcode. In this case,  $\overline{IEO}$  will go high until the next opcode byte is decoded, whereupon it will again go low. If the second byte of the opcode was a "4D" then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service will have its  $\overline{IEI}$  high and its  $\overline{IEO}$  low. This device is the *highest* priority device in the daisy chain which has received an interrupt acknowledge. All other peripherals have  $\overline{IEI}=\overline{IEO}$ . If the next opcode byte decoded is "4D", this peripheral device will reset its "interrupt under service" condition.

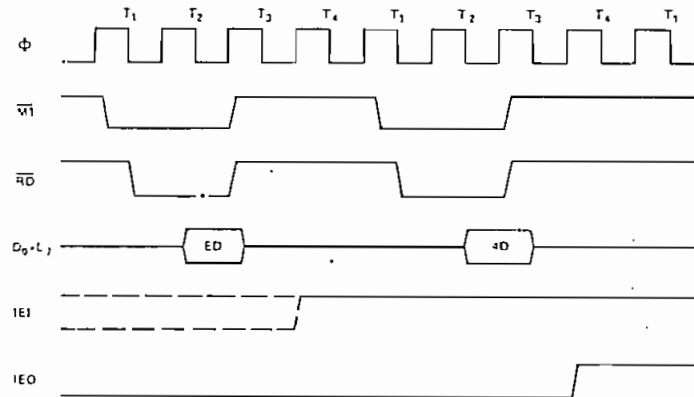


Figura A.2.5.b. Regreso de un ciclo de interrupción



## A.2.6. CARACTERISTICAS EN A.C. PARA Z80A-PIO

TA = 0° C to 70° C, Vcc = +5 V ± 5%, unless otherwise noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
$\Phi$	$t_c$	Clock Period	250	[1]	nsec	
	$t_W (\Phi H)$	Clock Pulse Width, Clock High	105	2000	nsec	
	$t_W (\Phi L)$	Clock Pulse Width, Clock Low	105	2000	nsec	
	$t_r, t_f$	Clock Rise and Fall Times		30	nsec	
	$t_h$	Any Hold Time for Specified Set Up Time	0		nsec	
CS, $\overline{CE}$ ETC.	$t_{S\Phi} (CS)$	Control Signal Set-Up Time to Rising Edge of $\Phi$ During Read or Write Cycle	145		nsec	
$D_0-D_7$	$t_{DR} (D)$	Data Output Delay From Falling Edge of $\overline{RD}$	50	380	nsec	[2] $C_L = 50 \text{ pF}$ [3]
	$t_{S\Phi} (D)$	Data Set-Up Time to Rising Edge of $\Phi$ During Write or $\overline{M1}$ Cycle			nsec	
	$t_{DI} (D)$	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle		250	nsec	
	$t_F (D)$	Delay to Floating Bus (Output Buffer Disable Time)		110	nsec	
IEI	$t_S (IEI)$	IEI Set-Up Time to Falling edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
$\overline{IEO}$	$t_{OH} (IO)$	IEO Delay Time from Rising Edge of IEI		160	nsec	[5] [5] $C_L = 50 \text{ pF}$ [5]
	$t_{OL} (IO)$	IEO Delay Time from Falling Edge of IEI		130	nsec	
	$t_{DM} (IO)$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$ ) See Note A.		190	nsec	
$\overline{IORQ}$	$t_{S\Phi} (IR)$	$\overline{IORQ}$ Set-Up Time to Rising Edge of $\Phi$ During Read or Write Cycle.	115		nsec	
$\overline{M1}$	$t_{S\Phi} (M1)$	$\overline{M1}$ Set-Up Time to Rising Edge of $\Phi$ During INTA or $\overline{M1}$ Cycle See Note B	90		nsec	
$\overline{RD}$	$t_{S\Phi} (RD)$	$\overline{RD}$ Set-Up Time to Rising Edge of $\Phi$ During Read or $\overline{M1}$ Cycle	115		nsec	
$A_0-A_7,$ $B_0-B_7$	$t_S (PD)$	Port Data Set-Up Time to Rising Edge of $\overline{STROBE}$ (Mode 1)	230	210	nsec	[5] $C_L = 50 \text{ pF}$ [5]
	$t_{DS} (PD)$	Port Data Output Delay from Falling Edge of $\overline{STROBE}$ (Mode 2)			nsec	
	$t_F (PD)$	Delay to Floating Port Data Bus from Rising Edge of $\overline{STROBE}$ (Mode 2)		180	nsec	
	$t_{DI} (PD)$	Port Data Stable from Rising Edge of $\overline{IORQ}$ During WR Cycle (Mode 0)		180	nsec	
$\overline{ASTB},$ $\overline{BSTB}$	$t_W (ST)$	Pulse Width, $\overline{STROBE}$	150	[4]	nsec	
INT	$t_D (I')$	INT Delay time from Rising Edge of $\overline{STROBE}$		440	nsec	
	$t_D (I'')$	INT Delay Time from Data Match During Mode 3 Operation		380	nsec	
ARDY, BRDY	$t_{DH} (RY)$	Ready Response Time from Rising Edge of $\overline{IORQ}$		$t_c +$ 410	nsec	[5] $C_L = 50 \text{ pF}$ [5]
	$t_{DL} (RY)$	Ready Response Time from Rising Edge of $\overline{STROBE}$		$t_c +$ 360	nsec	

## NOTES:

- A.  $2.5 t_c > (N-2) t_{DL} (IO) + t_{DM} (IO) + t_S (IEI) + \text{TTL Buffer Delay, if any}$   
 B.  $\overline{M1}$  must be active for a minimum of 2 clock periods to reset the PIO.

[1]  $t_c = t_W (\Phi H) + t_W (\Phi L) + t_r + t_f$

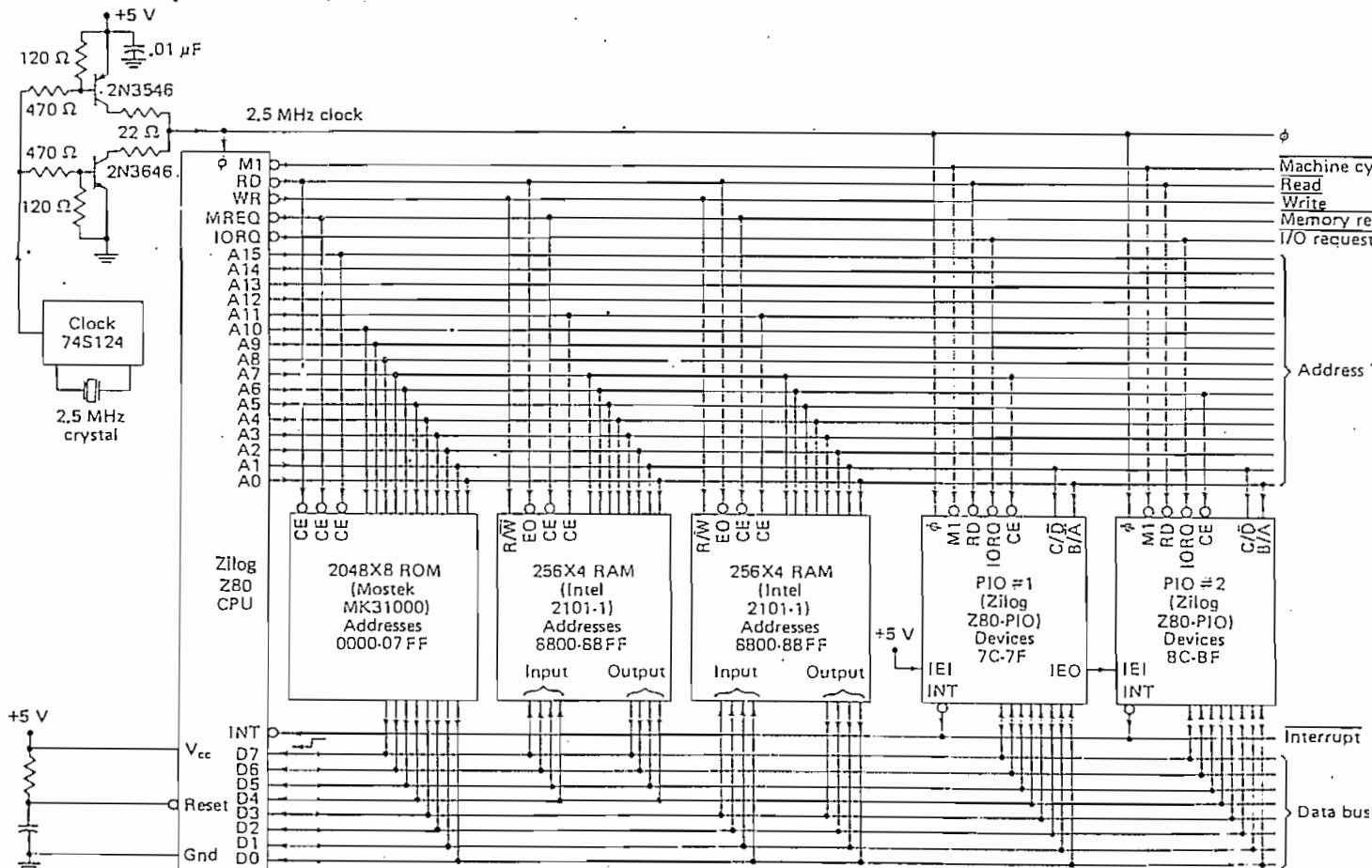
[2] Increase  $t_{DR} (D)$  by 10 nsec for each 50 pF increase in loading up to 200 pF max.[3] Increase  $t_{DI} (D)$  by 10 nsec for each 50 pF increase in loading up to 200 pF max.[4] For Mode 2:  $t_W (ST) > t_S (PD)$ 

[5] Increase these values by 2 nsec for each 10 pF increase in loading up to 100 pF max.

### A.2.7. CONDICIONES INICIALES DE PIO AL MOMENTO DEL ENCENDIDO

- Registros de salida encendados;
  - puesta la máscara a interrupciones;
  - seleccionado el modo 1 en ambas puertas;
  - líneas de entrada/salida de las puertas en alta impedancia;
- y,
- las señales para el handshake inactivas.

### A.3.1. UN EJEMPLO DE CONFIGURACION FUNDAMENTAL DE UN MICROCOMPUTADOR BASADO EN Z80-CPU CON ALGUNOS SOPORTES EN HARDWARE



## A.3.2. FAMILIAS LOGICAS

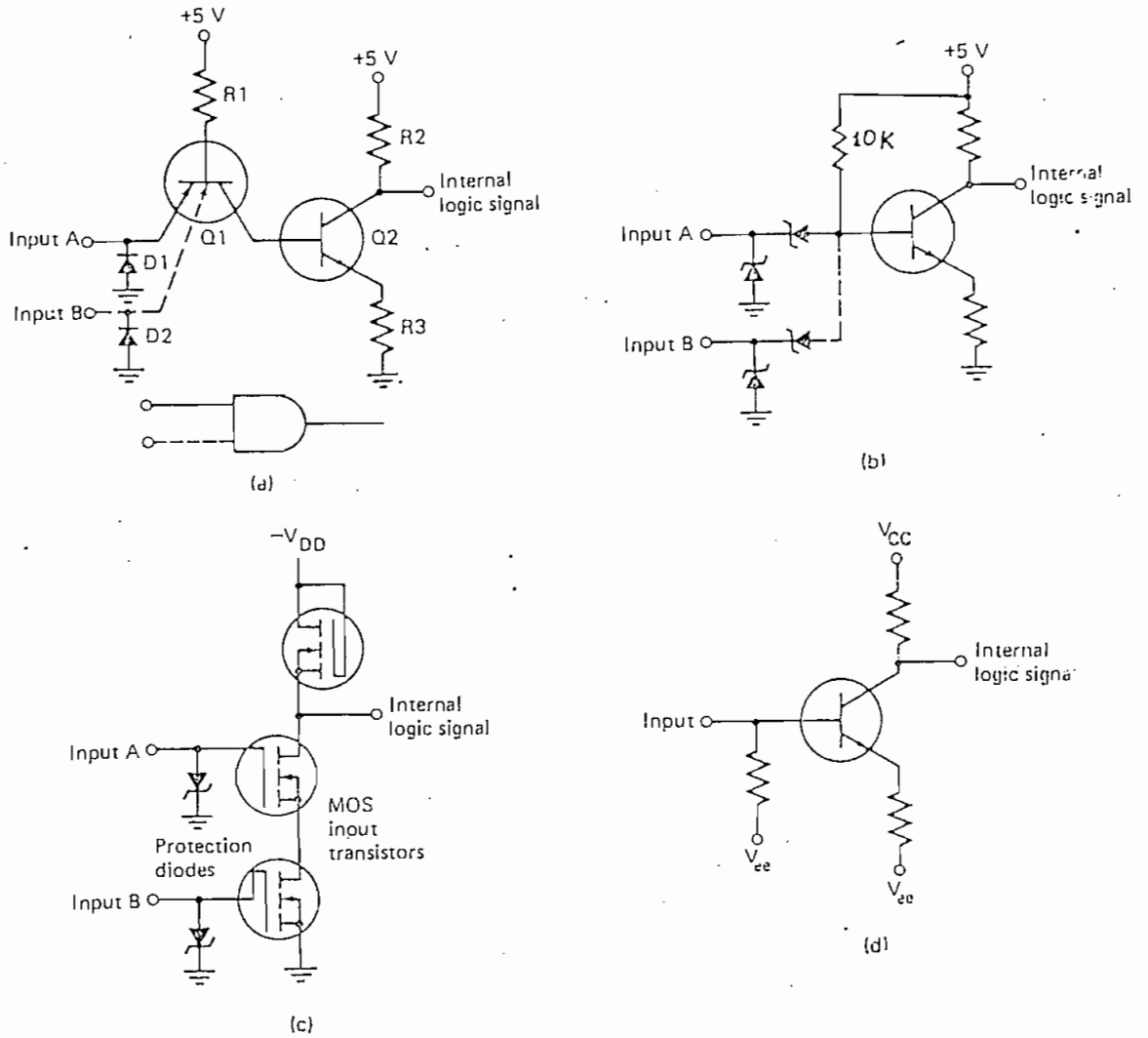


Figura A.3.2.a. Circuitos equivalentes de entrada típicos:

- a) Familia TTL,
- b) LSTTL,
- c) MOS,
- d) ECL

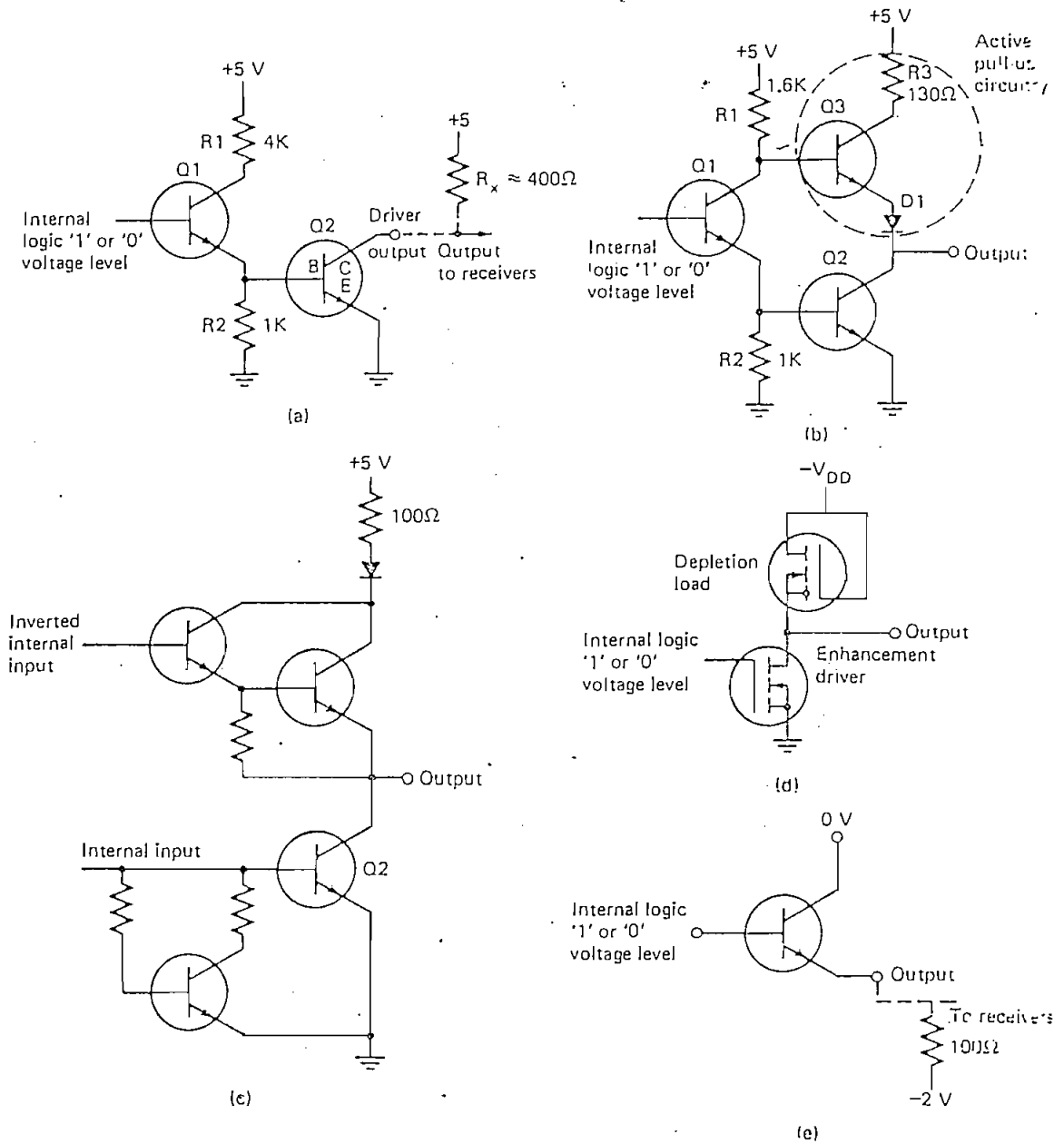


Figura A.3.2.b. Circuitos equivalentes de salida típicos:

- a) familia TTL Open Collector,
- b) TTL Totem Pole,
- c) TTL Tristate,
- d) MOS,
- e) ECL

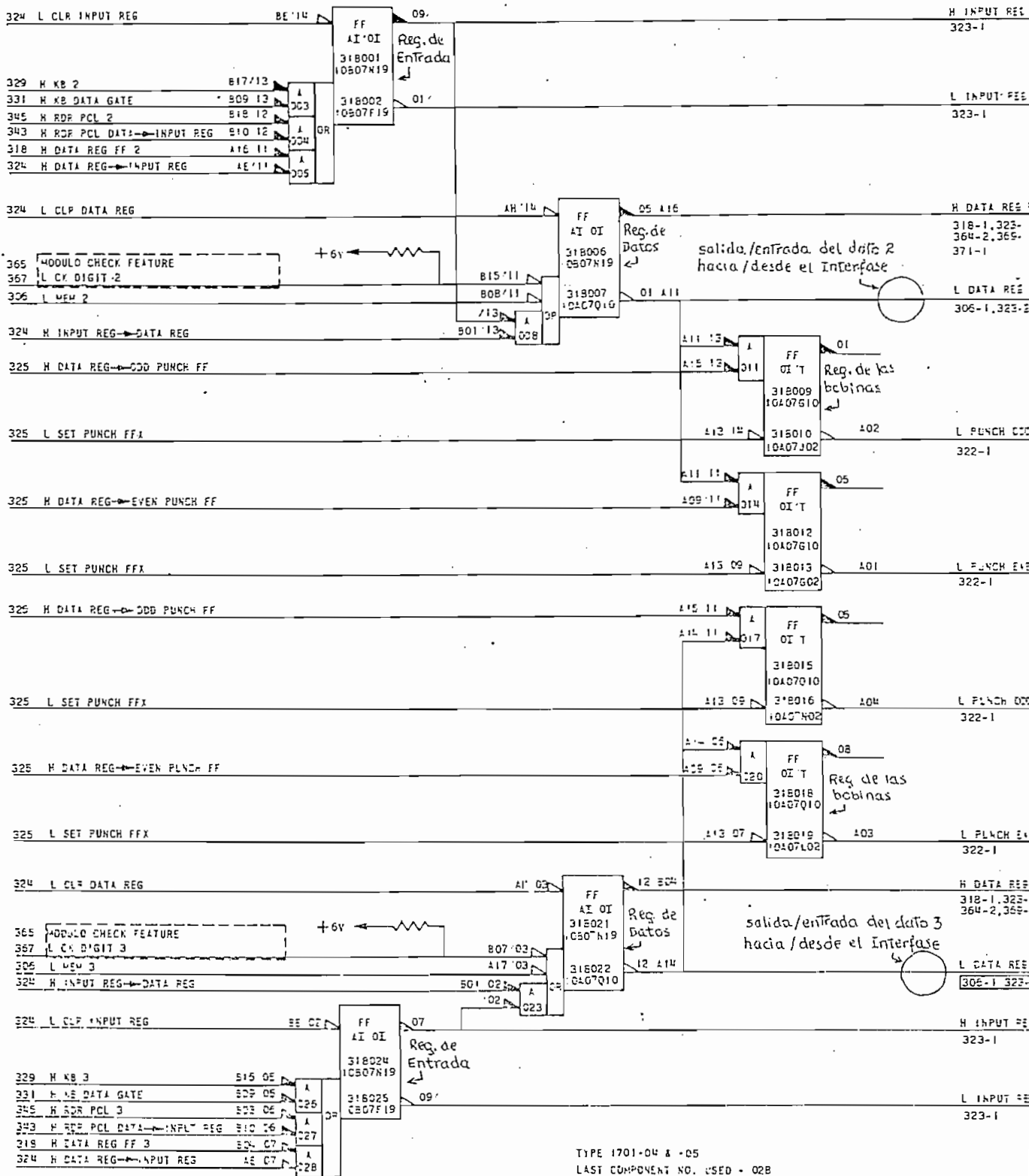
## A.3.3. CARACTERISTICAS VOLTAMPERIMETRICAS DE LA FAMILIA LSTTL

(Driver) 74LS00			
Voltage	Logic 1	High level input V	2 V min
	Logic 0	Low level input V	0.7 V max
	Logic 1	High level output V	2.5 V min
	Logic 0	Low level output V	0.5 V max
Current	Logic 1	High input I	20 $\mu$ A max
	Logic 0	Low input I	-36 mA max
	Logic 1	High output I	-400 $\mu$ A max
	Logic 0	Low output I	8 $\mu$ A max

## A.3.4. INTERFASE EN UNIVAC -DATOS-

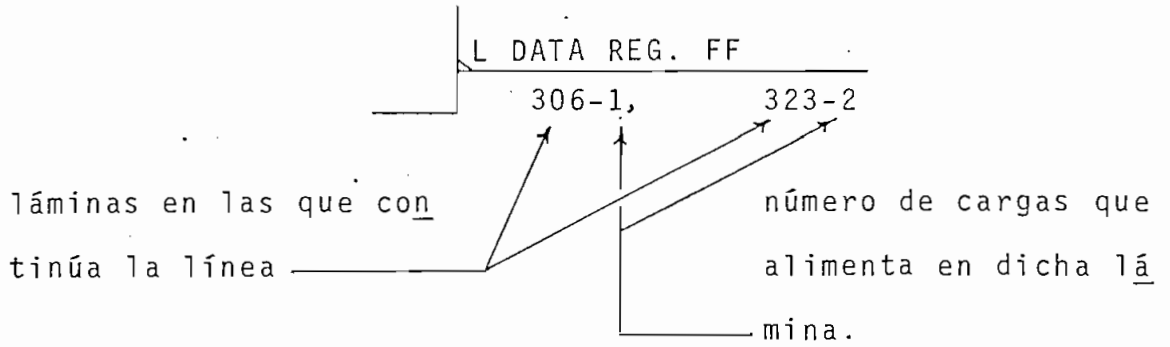
En el diagrama siguiente se encuentran representadas las líneas que corresponden a los datos 2 y 3. Las demás líneas de datos (0, 1, 4, 5, 6, 7, 8, 9, 11 y 12) tienen diagramas muy similares.

Los grupos de números que se observan en la parte inferior de las líneas de salida de los flip-flops significan, por ejemplo:



TYPE 1701-04 & -05  
LAST COMPONENT NO. USED - 028

Figura A.3.4.  
REGISTROS DE DATOS, ENTRADA Y PERFORACION 2 Y 3 DE UNIVAC



Es necesario hacer el análisis del *máximo requerimiento de corriente* que pueda tener cualquiera de las 12 líneas de datos. En tal virtud, se ha realizado este análisis basándose en las láminas del Manual de Servicio de UNIVAC y se han sacado los siguientes resultados:

#### CARGAS EN LAS 12 LINEAS DE DATOS

.LINEA DE DATOS "L DATA REG FF"	CARGAS
0	C1, C2, C3, C4, C5
11, 12	C1, C2, C3, C4, C5, C6
1, 2, 4, 8	C1, C2, C3, C4, C5, C6, C7
3, 5, 6, 9	C1, C2, C3, C4, C5, C6, C7, C8
7	C1, C2, C3, C4, C5, C6, C7, C8, C9

Tabla A.3.4.

En donde C1, C2, ..., C9 son nueve tipos de *cargas* que se explican a continuación.

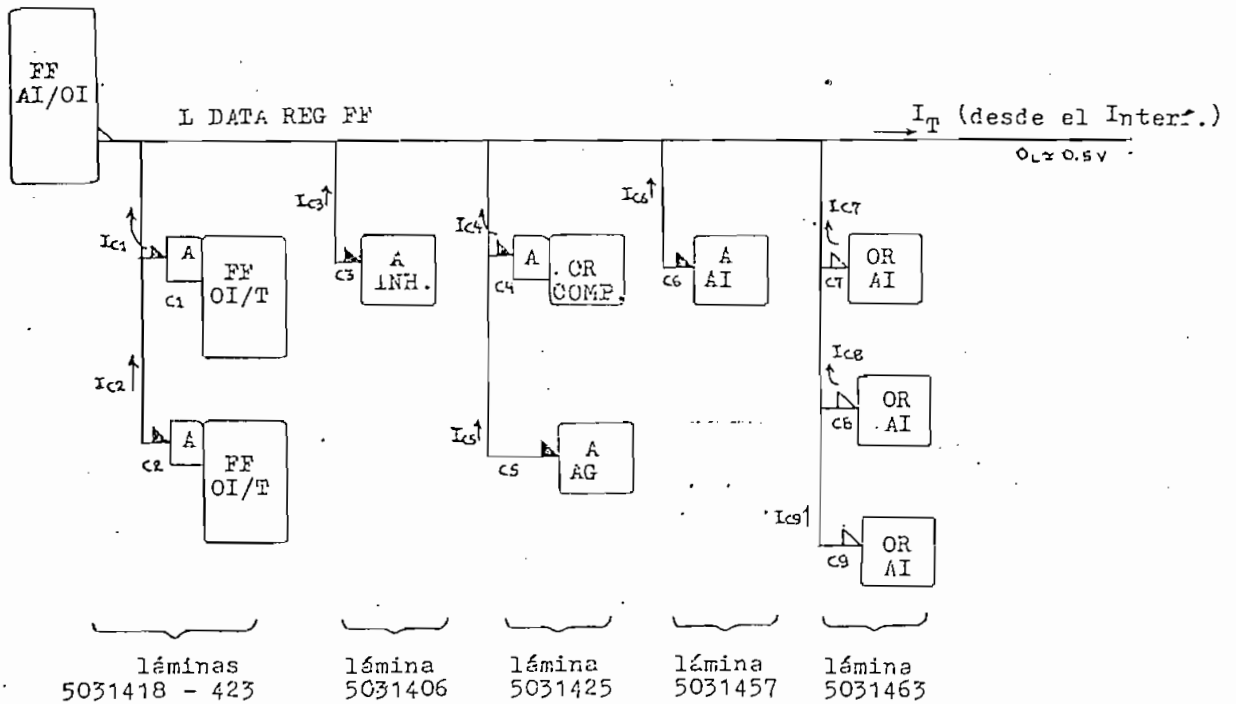
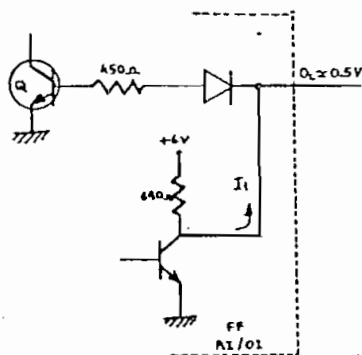


Figura A.3.4.1. Esquema general de las cargas a la salida de los flip-flops que conforman el Registro de datos.

Interesa solamente el análisis para L DATA REG FF = 0

- Flip-flop de Dato:



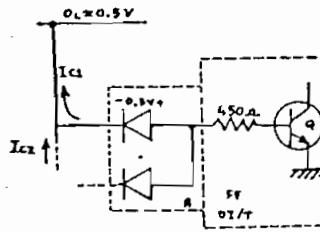
$$I \approx \frac{6V - 0,5V}{640\Omega}$$

$$I \approx 8.6 \text{ mA}$$

Figura A.3.4.2. Circuito equivalente del bloque "FF AI/OI"



- Cargas C1 y C2:

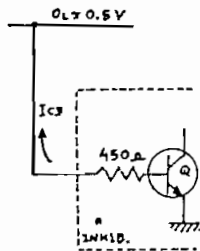


$$I_{C1} \approx 0 \text{ mA}$$

$$I_{C2} \approx 0 \text{ mA}$$

Figura A.3.4.3. Circuito equivalente del bloque "A + FF 01/T"

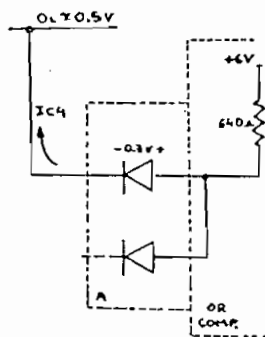
- Carga C3:



$$I_{C3} \approx 0 \text{ mA}$$

Figura A.2.4.4. Circuito equivalente del bloque "A INHIBIT"

- Carga C4:

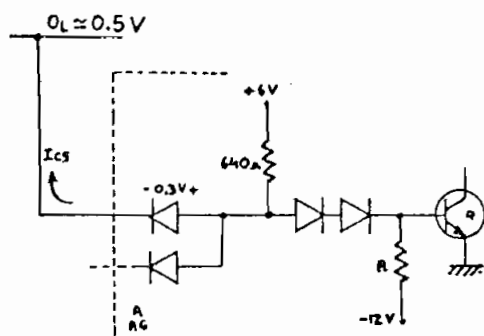


$$I_{C4} \approx \frac{6V - 0.5V - 0.3V}{640\Omega}$$

$$I_{C4} \approx 8.1 \text{ mA}$$

Figura A.3.4.5. Circuito equivalente del bloque "A + OR COMP"

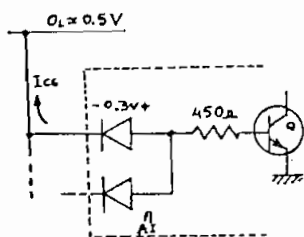
- Carga C5



$$I_{C5} \approx 8.1 \text{ mA}$$

Figura A.3.4.6. Circuito equivalente del bloque "A AG"

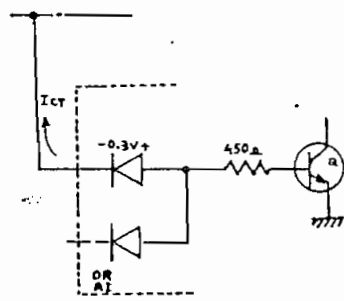
- Carga C6



$$I_{C6} \approx 0 \text{ mA}$$

Figura A.3.4.7. Circuito equivalente del bloque "A AI"

- Cargas C7, C8, C9



$$I_{C7} = I_{C8} = I_{C9} \approx 0 \text{ mA}$$

Figura A.3.4.8. Circuito equivalente del bloque "OR AI"

Así pues, tomando la línea de mayor número de cargas (7), la corriente requerida sería:

$$I_T = I_{C1} + I_{C2} + I_{C3} + I_{C4} + I_{C5} + I_{C6} + I_{C7} + I_{C8} + I_{C9} + I_{FF}$$

$$I_T \approx 16.2 \text{ mA} + 8.6 \text{ mA}$$

$$I_T \approx 24.8 \text{ mA}$$

a tal corriente se sumará la producida en el mismo Bloque de Compatibilidad de la PLACA-INTFASE (figuras 3.3.1.3-4):

$$I_{R1} \approx 0,4 \text{ mA}$$

$$I_{R2} \approx 0,6 \text{ mA}$$

$$i_{OL} \approx 0,2 \text{ mA}$$

por tanto, la corriente total que debería "chupar" cada Buffer-Driver SN74LS244, es:

$$I_{IL} \approx I_T + I_{R1} + I_{R2} + i_{OL}$$

$$I_{IL} \approx 24,8 \text{ mA} + 0,4 \text{ mA} + 0,6 \text{ mA} + 0,2 \text{ mA}$$

$$I_{IL} \approx 26 \text{ mA}$$

## A.3.5. BUS S-100

Pin #	Signal Name and Type	Polarity	
1	+8 volts (B) <sup>1</sup>		Instantaneous minimum greater than 7 volts, instantaneous maximum less than volts, average maximum less than volts.
2	+16 volts (B)		Instantaneous minimum greater than 14 volts, instantaneous maximum less than 35 volts, average maximum less than 20 volts.
3	XRDY (S) <sup>1-10</sup>	positive	One of two ready inputs to the current Bus Master. The bus is ready when both these ready inputs are true.
4	$\overline{VI}_0$ (S) <sup>10</sup>	negative	Vectored interrupt line $\phi$ .
5	$\overline{VI}_1$ (S) <sup>10</sup>	"	" " 1.
6	$\overline{VI}_2$ (S) <sup>10</sup>	"	" " 2.
7	$\overline{VI}_3$ (S) <sup>10</sup>	"	" " 3.
8	$\overline{VI}_4$ (S) <sup>10</sup>	"	" " 4.
9	$\overline{VI}_5$ (S) <sup>10</sup>	"	" " 5.
10	$\overline{VI}_6$ (S) <sup>10</sup>	"	" " 6.
11	$\overline{VI}_7$ (S) <sup>10</sup>	"	" " 7.
12	—	—	Not specified.
13	—	—	"
14	—	—	"
15	—	—	"
16	—	—	"
17	—	—	"
18	$\overline{STAT DSB}$ (M) <sup>1-10</sup>	negative	The control signal to disable the 8 status signals <sup>2</sup> .
19	$\overline{C/C DSB}$ (M) <sup>10</sup>	"	The control signal to disable the 6 command/control signal <sup>3</sup> .
20	UNPROT	—	Not specified.
21	SS	—	Not specified.
22	$\overline{ADD DSB}$ (M) <sup>10</sup>	negative	The control signal to disable the 16 address signals <sup>4</sup> .
23	$\overline{DO DSB}$ (M) <sup>10</sup>	"	The control signal to disable the 8 data output <sup>5</sup> signals.
24	$\phi_2$ (B)	positive	The master timing signal for the bus.
25	$\phi_1$		Not specified.
26	PHLDA (M)	positive	A command/control signal used in conjunction with PHOLD to coordinate Bus Master transfer operations.
27	PWAIT (M)	"	The acknowledge signal to either of the bus ready signals XRDY, PRDY or to a HLT instruction.
28	PINTE	"	Not specified.
29	A5 (M)	"	Address bit 5.
30	A4 (M)	"	Address bit 4.

Tabla A.3.5.a. Las 100 líneas del BUS S-100

Pin #	Signal Name and Type	Polarity	
31	A3 (M)	"	Address bit 3.
32	A15 (M)	"	Address bit 15 (most significant).
33	A12 (M)	"	Address bit 12.
34	A9 (M)	"	Address bit 9.
35	D01 (M)/EA1 (M)/DATA1 (M/S)	"	Data out bit 1. Extended address bit 1, Bidirectional data bit 1.
36	D00 (M)/EA0 (M)/DATA0 (M/S)	"	Data out bit $\beta$ . Extended address with bit $\beta$ . Bidirectional data bit $\beta$ (least significant)
37	A10 (M)	"	Address bit 10.
38	D04 (M)	positive	Data out bit 4.
39	D05 (M)	"	" 5.
40	D06 (M)	"	" 6.
41	D12 (M)	"	Data in <sup>6</sup> bit 2.
42	D13 (M)	"	Data in bit 3.
43	D17 (M)	"	Data in bit 7 (most significant).
44	SMI (M)	"	The status signal which indicates that the current bus cycle <sup>7</sup> is an op-code fetch.
45	SOUT (M)	"	The status signal identifying the data transfer bus cycle of an OUT instruction.
46	SINP (M)	"	The status signal identifying the data transfer bus cycle of an IN instruction.
47	SMEMR (M)	"	The status signal identifying bus cycles which transfer data from memory to a Bus Master which are not interrupt acknowledge instruction fetch cycle(s).
48	SHLTA (M)	"	The status signals which acknowledges that a HLT instruction has been executed. that a HLT instruction has been executed.
49	$\overline{\text{CLOCK}}$	—	Not specified.
50	GND		Signal and power ground.
51	+8 volts (B)		See comments above for pin #1.
52	-16 volts (B)		Instantaneous maximum less than -14 volts. Instantaneous maximum greater than -35 volts. average minimum greater than -20 volts.
53	$\overline{\text{SSWI}}$	—	Not specified.
54	EXT CLR	negative	A reset signal to reset Bus Slaves. When this signal goes low, it must stay low for at least 3 bus states.
55	—	—	Not specified.
56	—	—	"
57	—	—	"

Tabla A.3.5.b. Las 100 líneas del Bus S-100

Pin #	Signal Name and Type	Polarity	
58	—	—	Not specified.
59	SXTRQ (M)	negative	Master signal which requests 16 bit wide slaves to respond by asserting SXTN.
60	—	—	Not specified.
61	SXTN (S)	negative	The signal generated by 16 bit slaves in response to the 16 bit request signal SXTRQ.
62	—	—	Not specified.
63	—	—	..
64	—	—	..
65	—	—	..
66	—	—	..
67	PHANTOM (B)	negative	A bus signal which disables normal slave devices and enables phantom slaves — primarily used for bootstrapping systems without hardware front panels.
68	MWRITE (B)	positive	The logical negation of PWR and SOUT; this signal must follow PWR by not more than 30 ns.
69	PS	—	Not specified.
70	PORT	—	..
71	RUN	—	..
72	PRDY (S) <sup>10</sup>	positive	See comments above for pin #3.
73	PINT (S) <sup>10</sup>	negative	The primary interrupt request bus signal.
74	PHOLD (M) <sup>10</sup>	..	The command-control signal used in conjunction with PHLDA to coordinate Bus Master transfer operations.
75	PRESET (B) <sup>10</sup>	..	The reset signal to reset Bus Master devices. When this signal goes low, it must stay low for at least 3 bus states.
76	PSYNC (M)	positive	The command/control signal identifying BS <sub>1</sub> . (See bus states comments.)
77	PWR (M)	negative	The command control signal signifying the presence of valid data on the DO bus <sup>9</sup> .
78	PDBIN (M)	positive	The command control signal that requests data on the DI bus <sup>9</sup> from the currently addressed slave.
79	A <sub>β</sub> (M)	..	Address bit β (least significant).
80	A1 (M)	..	Address bit 1.
81	A2 (M)	..	.. 2.
82	A6 (M)	..	.. 6.
83	A7 (M)	..	.. 7.
84	A8 (M)	..	.. 8.

Tabla A.3.5.c. Las 100 líneas del Bus S-100

Pin #	Signal Name and Type	Polarity	
85	A13 (M)	"	" 13.
86	A14 (M)	"	" 14.
87	A11 (M)	"	" 11.
88	D02 (M)/A8 (M)/DATA2 (M/S)	positive	Data out bit 2, extended address bit 2, & bidirectional data bit 2.
89	D03 (M)/A19 (M)/DATA3 (M/S)	"	Data out bit 3, extended address bit 3, & bidirectional data bit 3.
90	D07 (M)/A23 (M)/DATA7 (M/S)	"	Data out bit 7, extended address bit 7, & bidirectional data bit 7 (most significant).
91	D14 (S)/DATA12 (M/S)	positive	Data in bit 4 & bidirectional data bit 12.
92	D15 (S)/DATA13 (M/S)	"	Data in bit 5 & bidirectional data bit 13.
93	D16 (S)/DATA14 (M/S)	positive	Data in bit 6 & bidirectional data bit 14.
94	D11 (S)/DATA9 (M/S)	positive	Data in bit 1 & bidirectional data bit 9.
95	D10 (S)/DATA8 (M/S)	positive	Data in bit 0 & bidirectional data bit 8.
96	SINTA (M)	"	The status signal identifying the instruction fetch cycle(s) that immediately follow and accepted interrupt request presented on PINT.
97	$\overline{\text{SWO}}$ (M)	negative	The status signal identifying a bus cycle which transfers data from a Bus Master to a slave.
98	SSTACK		Not specified.
99	$\overline{\text{POC}}$ (B)	negative	The power-on clear signal for all bus devices; when this signal goes low, it must stay low for at least 3 bus states.
100	GND		Signal and power ground.

Adapted from table by the IEEE Computer Society Microprocessor Standards Committee.

Tabla A.3.5.d. Las 100 líneas del Bus S-100

Denom.	Línea	Denom.	Línea
DO 0	36	A 0	79
DO 1	35	A 1	80
DO 2	88	A 2	81
DO 3	89	A 3	31
DO 4	38	A 4	30
DO 5	39	A 5	29
DO 6	40	A 6	82
DO 7	90	A 7	83
DI 0	95		
DI 1	94	2	24
DI 2	41	SOUT	45
DI 3	42	SINP	46
DI 4	91		
DI 5	92	PRDY	72
DI 6	93		
DI 7	43		
		+8V	1, 51
		$\overline{\text{GND}}$	50, 100

Tabla A.3.5.e. Líneas del Bus S-100 utilizadas en el Interfa  
se

Del bus S-100 cabe indicar que el microcomputador utiliza



adicionalmente las siguientes líneas:

No. Línea	Bus S-100 stándard	Este caso particular
65	no especifica	MEM.REQ.
66	no especifica	REFRESH
70	no especifica	PROTECT MEM.

(Referirse al diagrama de la figura A.4.1).

#### A.3.6. DESARROLLO DE LA PROGRAMACION DE EPROM

La tabla A.3.6 muestra la forma cómo son transformados los caracteres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, J, R, Y, >, ?, /, y SPACE desde el Código IBM Hollerith al Código ASCII. De manera similar se ha procedido con el resto de caracteres.

El uso de las figuras indicadas ahí, así como de la tabla A.3.7.6, facilitará mucho la comprensión del desarrollo (en binario) de esta parte del programa de EPROM.

Por otra parte, la programación requerida para la codificación de ASCII a Hollerith sigue la misma tónica, aunque aquí interviene el "DECODER" (ya no el "ENCODER") y no existe inversión de los datos. No se ha hecho constar su desarrollo para no hacer más extenso este punto.

CARACTER	CÓDIGO IBM-HOLLERITH (Viene desde UNIVAC) -fig. 3.3.2.1-												PASANDO "ENCODER" -figs. 3.3.2.1 y 3.3.4.1-						PROGRAMADO EN EPROM					DATA						
																			ADDRESS -figs. 3.3.4.1 y 3.3.5.-					en Hex.	Salida de EPROM					
	L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	A6	A5	A4	A3	A2	A1	A0	A7	A6	A5	A4	A3	A2	A1	A0	A7-A0	07-0C
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	1	0	0	0	8 8	3 0	
1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	F 4	3 1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	0	1	0	0	0	9 0	3 2
3	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	A 0	3 3
4	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	1	0	0	0	B 0	3 4
5	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	C 0	3 5
6	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	D 0	3 6
7	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	E 0	3 7
8	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	1	0	8 4	3 8
9	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	F 0	3 9
A	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1	0	1	1	1	1	0	1	0	1	F 5	4 1
B	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	0	1	0	1	0	0	0	1	9 1	4 2	
C	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	0	1	0	0	0	1	A 1	4 3
J	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	0	0	1	1	1	1	1	0	1	1	0	F 6	4 A
R	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	1	1	0	1	1	1	1	1	0	0	1	0	F 2	5 2
Y	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1	0	0	8 C	5 9	
>	0	1	1	1	1	1	0	1	0	1	1	1	1	0	1	0	0	0	1	1	1	1	0	1	1	1	0	0	D C	3 E
?	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	1	1	1	1	0	1	1	0	0	E C	3 F	
/	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	F C	2 F
SPACE	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	8 0	2 C	

Tabla A.3.6.a. Camino que siguen los bits de datos codificados en IBM Hollerith, para ser cambiados al código ASCII.

Se ha creído conveniente que algunos símbolos se hagan equivalentes por la programación; éstos son:

UNIVAC		ASCII
Zonas	Impresión	
0,5,8	—	DEL.
11,3	§	{
0,3	┘	}
0,2,8	^	~
12,0	¢	^
11,0	!	—

Tabla A.3.6.b. Símbolos que se han hecho equivalentes.

En el punto A.5.3. se ha mencionado que existió un error en el programa inicial. En efecto, EPROM tuvo que ser borrada y reprogramada. El borrado se lo hizo por la exposición de 0,8 horas a luz ultravioleta de 4 Watt de potencia.

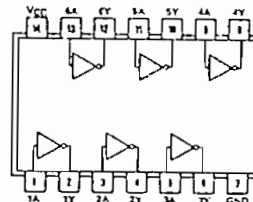
### A.3.7. CIRCUITOS INTEGRADOS UTILIZADOS

#### A.3.7.1. SN74LS04 Hex Inverter

HEX INVERTERS

04

positive logic:  
Y =  $\bar{A}$ .



SN5404 (J)            SN7404 (J, N)  
 SN54H04 (J)        SN74H04 (J, N)  
 SN54L04 (J)        SN74L04 (J, N)  
 SN54LS04 (J, W)    SN74LS04 (J, N)  
 SN54S04 (J, W)    SN74S04 (J, N)

Figura A.3.7.1.

recommended operating conditions

	54 FAMILY	SERIES 54			SERIES 54H			SERIES 54L			SERIES 54LS			SERIES 54S			UNIT
	74 FAMILY	SERIES 74			SERIES 74H			SERIES 74L			SERIES 74LS			SERIES 74S			
		'00, '04, '10, '20, '30	'H00, 'H04, 'H10, 'H20, 'H30			'L00, 'L04, 'L10, 'L20, 'L30			'LS00, 'LS04, 'LS10, 'LS20, 'LS30			'S00, 'S04, 'S10, 'S20, 'S30, 'S133					
		MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX	MIN NOM MAX				
Supply voltage, V <sub>CC</sub>	54 Family	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	4.5	5	5.5	V
	74 Family	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	4.75	5	5.25	V
High-level output current, I <sub>OH</sub>	54 Family			-400			-500			-100			-400			-1000	mA
	74 Family			-400			-500			-200			-400			-1000	mA
Low-level output current, I <sub>OL</sub>	54 Family			16			20			2			4			20	mA
	74 Family			16			20			3.6			8			20	mA
Operating free-air temperature, T <sub>A</sub>	54 Family	-55		125	-55		125	-55		125	-55		125	-55		125	°C
	74 Family	0		70	0		70	0		70	0		70	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS†	SERIES 54	SERIES 54H	SERIES 54L	SERIES 54LS	SERIES 54S	UNIT						
			SERIES 74	SERIES 74H	SERIES 74L	SERIES 74LS	SERIES 74S							
			'00, '04, '10, '20, '30	'H00, 'H04, 'H10, 'H20, 'H30	'L00, 'L04, 'L10, 'L20, 'L30	'LS00, 'LS04, 'LS10, 'LS20, 'LS30	'S00, 'S04, 'S10, 'S20, 'S30, 'S133							
			MIN TYP‡ MAX	MIN TYP‡ MAX	MIN TYP‡ MAX	MIN TYP‡ MAX	MIN TYP‡ MAX							
V <sub>IH</sub> High-level input voltage	1, 2		2	2	2	2	2	V						
V <sub>IL</sub> Low-level input voltage	1, 2		0.8	0.8	0.7	0.7	0.7	V						
V <sub>IK</sub> Input clamp voltage	3	V <sub>CC</sub> = MIN, I <sub>I</sub> = 5	-1.5	-1.5			-1.5	V						
V <sub>OH</sub> High-level output voltage	1	V <sub>CC</sub> = MIN, V <sub>IL</sub> = V <sub>IL</sub> max, I <sub>OH</sub> = MAX	2.4 3.4	2.4 3.5	2.4 3.3	2.5 3.4	2.5 3.4	V						
V <sub>OL</sub> Low-level output voltage	2	V <sub>CC</sub> = MIN, V <sub>IH</sub> = 2 V, I <sub>OL</sub> = MAX	0.2 0.4	0.2 0.4	0.15 0.2	0.25 0.4	0.25 0.5	V						
I <sub>I</sub> Input current at maximum input voltage	4	V <sub>CC</sub> = MAX, V <sub>I</sub> = 5.5 V		1	1	0.1		mA						
I <sub>IH</sub> High-level input current	4	V <sub>CC</sub> = MAX, V <sub>IH</sub> = 2.4 V		40				mA						
		V <sub>IH</sub> = 2.7 V			10		20	mA						
I <sub>IL</sub> Low-level input current	5	V <sub>CC</sub> = MAX, V <sub>IL</sub> = 0.3 V				-0.18		mA						
		V <sub>IL</sub> = 0.4 V		-1.5	-2		-0.4	mA						
		V <sub>IL</sub> = 0.5 V						mA						
I <sub>OS</sub> Short-circuit output current*	6	V <sub>CC</sub> = MAX	-20	-55	-40	-100	-3	-15	-20	-100	-40	-100	-40	mA
I <sub>CC</sub> Supply current	7	V <sub>CC</sub> = MAX	-16	-55	-40	-100	-3	-15	-20	-100	-40	-100	-40	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.  
 ‡ All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.  
 \* Not more than one output should be shorted at a time, and for SN54H/SN74H, SN54LS/SN74LS, and SN54S/SN74S, duration of short-circuit should not exceed 100 ns.

TYPE	I <sub>CC</sub> (mA)		I <sub>CC</sub> L (mA)		I <sub>CC</sub> (mA)
	Total with outputs high	Total with outputs low	Average per gate (50% duty cycle)	Average per gate (50% duty cycle)	
	TYP	MAX	TYP	MAX	TYP
'00	4	8	12	22	2
'04	6	12	18	33	2
'10	3	6	9	18.5	2
'20	2	4	6	11	2
'30	1	2	3	6	2
'H00	10	16.8	26	40	4.5
'H04	16	26	40	58	4.5
'H10	7.5	12.6	19.5	30	4.5
'H20	5	8.4	13	20	4.5
'H30	2.5	4.2	6.5	10	4.5
'L00	0.44	0.8	1.16	2.04	0.20
'L04	0.66	1.2	1.74	3.06	0.20
'L10	0.33	0.6	0.87	1.53	0.20
'L20	0.22	0.4	0.58	1.02	0.20
SN54L30	0.11	0.33	0.29	0.51	0.20
SN74L30	0.11	0.2	0.29	0.51	0.20
'LS00	0.8	1.6	2.4	4.4	0.4
'LS04	1.2	2.4	3.6	6.6	0.4
'LS10	0.8	1.2	1.8	3.3	0.4
'LS20	0.4	0.8	1.2	2.2	0.4
'LS30	0.35	0.5	0.8	1.1	0.48
'S00	10	16	20	36	3.75
'S04	15	24	30	54	3.75
'S10	7.5	12	15	27	3.75
'S20	5	8	10	18	3.75
'S30	3	5	5.5	10	4.25
'S133	3	5	5.5	10	4.25

TYPE	TEST CONDITIONS#	t <sub>PLH</sub> (ns)			t <sub>PLL</sub> (ns)		
		Propagation delay time, low-to-high-level output			Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
'00, '10	C <sub>L</sub> = 15 pF, R <sub>L</sub> = 400 Ω	11	22	7	11	11	
'04, '20		12	22	8	11	11	
'30		13	22	8	11	11	
'H00	C <sub>L</sub> = 25 pF, R <sub>L</sub> = 280 Ω	5.9	10	6.7	10	10	
'H04		6	10	6.5	10	10	
'H10		5.9	10	6.3	10	10	
'H20		6	10	7	10	10	
'H30	6.8	10	8.9	10	10	10	
'L00, 'L04, 'L10, 'L20, 'L30	C <sub>L</sub> = 50 pF, R <sub>L</sub> = 4 kΩ	35	60	31	60	60	
'LS00, 'LS04, 'LS10, 'LS20, 'LS30		C <sub>L</sub> = 15 pF, R <sub>L</sub> = 2 kΩ	9	15	10	15	15
'S00, 'S04			3	4.5	3	5	5
'S10, 'S20	4.5			5		5	
'S30, 'S133	C <sub>L</sub> = 50 pF, R <sub>L</sub> = 280 Ω	4	6	4.5	6	6	
		5.5		6.5		6.5	

# Load circuits and voltage waveforms are shown on pages 3-10 and 3-11.

† Maximum values of I<sub>CC</sub> are over the recommended operating ranges of V<sub>CC</sub> and T<sub>A</sub>; typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

Tabla A.3.7.1. Características Eléctricas

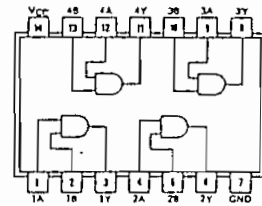
POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

A.3.7.2. SN74LS08 QUAD 2 INPUT AND GATE

QUADRUPLE 2-INPUT  
POSITIVE-AND GATES

**08**

positive logic:  
 $Y = AB$



SN5408 (J, W)    SN7408 (J, N)  
SN54LS08 (J, W)    SN74LS08 (J, N)  
SN54S08 (J, W)    SN74S08 (J, N)

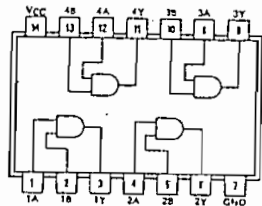
Figura A.3.7.2.

A.3.7.3. SN74LS09 QUAD 2 INPUT AND GATE OPEN COLLECTOR

QUADRUPLE 2-INPUT  
POSITIVE-AND GATES  
WITH OPEN-COLLECTOR OUTPUTS

**09**

positive logic:  
 $Y = AB$



SN5409 (J, W)    SN7409 (J, N)  
SN54LS09 (J, W)    SN74LS09 (J, N)  
SN54S09 (J, W)    SN74S09 (J, N)

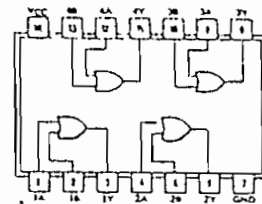
Figura A.3.7.3.

A.3.7.4. SN74LS32 QUAD 2 INPUT OR GATE

QUADRUPLE 2-INPUT  
POSITIVE-OR GATES

**32**

positive logic:  
 $Y = A+B$



SN5432 (J, W)    SN7432 (J, N)  
SN54LS32 (J, W)    SN74LS32 (J, N)  
SN54S32 (J, W)    SN74S32 (J, N)

Fig. A.3.7.4.

### A.3.7.5. SN74LS138 3-TO-8 LINE DECODER

- Designed Specifically for High-Speed: Memory Decoders Data Transmission Systems
- 'S138 and 'LS138 3-to-8-Line Decoders Incorporate 3 Enable Inputs to Simplify Cascading and/or Data Reception
- 'S139 and 'LS139 Contain Two Fully Independent 2-to-4-Line Decoders/ Demultiplexers
- Schottky Clamped for High Performance

TYPE	TYPICAL PROPAGATION DELAY (3 LEVELS OF LOGIC)	TYPICAL POWER DISSIPATION
'LS138	22 ns	32 mW
'S138	8 ns	245 mW
'LS139	22 ns	34 mW
'S139	7.5 ns	300 mW

#### description

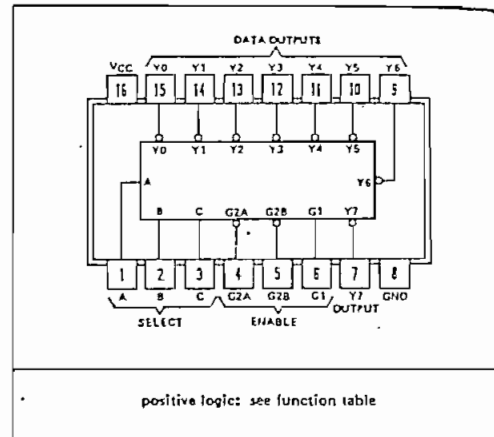
These Schottky-clamped TTL MSI circuits are designed to be used in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast-enable circuit the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the Schottky-clamped system decoder is negligible.

The 'LS138 and 'S138 decode one-of-eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

The 'LS139 and 'S139 comprise two individual two-line-to-four-line decoders in a single package. The active-low enable input can be used as a data line in demultiplexing applications.

All of these decoders/demultiplexers feature fully buffered inputs each of which represents only one normalized Series 54LS/74LS load ('LS138, 'LS139) or one normalized Series 54S/74S load ('S138, 'S139) to its driving circuit. All inputs are clamped with high-performance Schottky diodes to suppress line-ringing and simplify system design. Series 54LS and 54S devices are characterized for operation over the full military temperature range of  $-55^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ . Series 74LS and 74S devices are characterized for  $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$  industrial systems.

SN54LS138, SN54S138 ... J OR W PACKAGE  
SN74LS138, SN74S138 ... J OR N PACKAGE  
(TOP VIEW)



#### schematics of inputs and outputs

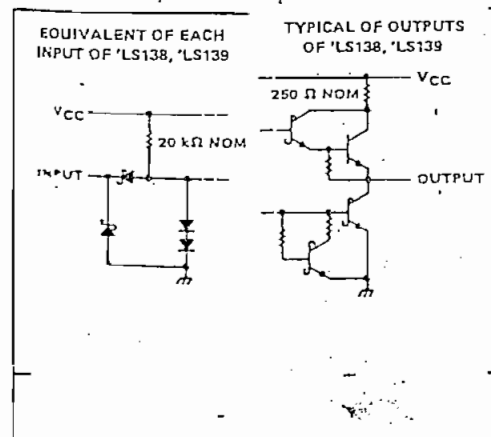
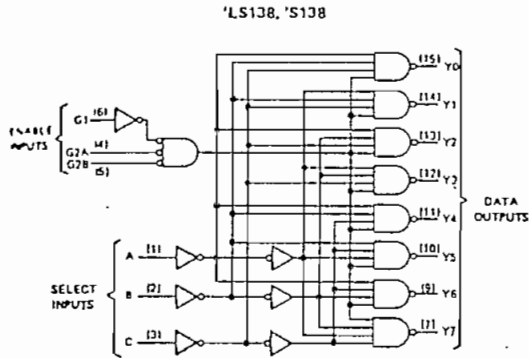


Figura A.3.7.5.

block diagrams and logic  
FUNCTIONAL



'LS138, 'S138  
FUNCTION TABLE

ENABLE			SELECT			OUTPUTS							
G1	G2*	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	X	X	X	X	L	H	H	H	H	H	H	H
H	L	L	X	X	X	L	L	H	H	H	H	H	H
H	L	L	L	X	X	L	L	L	H	H	H	H	H
H	L	L	L	L	X	L	L	L	L	H	H	H	H
H	L	L	L	L	L	L	L	L	L	L	H	H	H
H	L	L	L	L	L	L	L	L	L	L	L	L	H
H	L	L	L	L	L	L	L	L	L	L	L	L	L

\*G2 = G2A + G2B  
H = high level, L = low level, X = irrelevant

Tabla A.3.7.5. Tabla de verdad del DECODER

A.3.7.6. SN74LS148 8-TO-3 LINE PRIORITY ENCODER

- Encodes 8 Data Lines to 3-Line Binary (Octal)
- Applications Include:

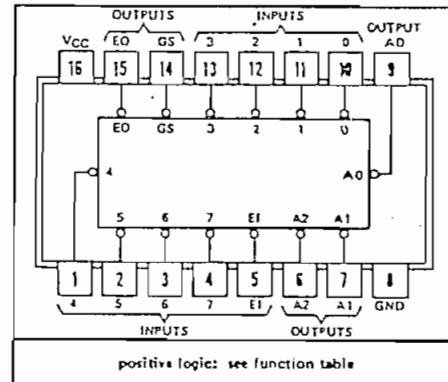
N-Bit Encoding  
Code Converters and Generators

TYPE	TYPICAL DATA DELAY	TYPICAL POWER DISSIPATION
'147	10 ns	225 mW
'148	10 ns	190 mW
'LS147	15 ns	60 mW
'LS148	15 ns	60 mW

'148, 'LS148  
FUNCTION TABLE

EI	INPUTS								OUTPUTS				
	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	L	L
L	X	X	X	X	X	X	L	L	L	L	L	L	L
L	X	X	X	X	X	L	H	H	L	L	L	L	L
L	X	X	X	X	L	H	H	H	L	L	L	L	L
L	X	X	X	L	H	H	H	H	L	L	L	L	L
L	X	X	L	H	H	H	H	H	L	L	L	L	L
L	X	L	H	H	H	H	H	H	L	L	L	L	L
L	L	H	H	H	H	H	H	H	H	H	L	L	L

SN54148, SN54LS148 ... J OR W PACKAGE  
SN74148, SN74LS148 ... J OR N PACKAGE  
[TOP VIEW]



description

These TTL encoders feature priority decoding of the inputs to ensure that only the highest-order data line is encoded. The '147 and 'LS147 encode nine data lines to four-line (8-4-2-1) BCD. The implied decimal zero condition requires no input condition as zero is encoded when all nine data lines are at a high logic level. The '148 and 'LS148 encode eight data lines to three-line (4-2-1) binary (octal). Cascading circuitry (enable input EI and enable output EO) has been provided to allow octal expansion without the need for external circuitry. For all types, data inputs and outputs are active at the low logic level. All inputs are buffered to represent one normalized Series 54/74 or 54LS/74LS load, respectively.

Figura A.3.7.6.

### A.3.7.7. SN74LS244 OCTAL BUFFER-DRIVER 3-STATE

	Typical	Typical	Typical Propagation		Typical	Typical Power	
	IOL (Sink Current)	IOH (Source Current)	Delay Times	Delay Times		Enable/ Disable Times	Dissipation (Enabled)
			Inverting	Noninverting		Inverting	Noninverting
SN74ALS*	12 mA	-12 mA	10,5 ns	12 ns	18 ns	130 mW	135 mW
SN74LS*	24 mA	-15 mA	10,5 ns	12 ns	18 ns	130 mW	135 mW
SN54S*	48 mA	-12 mA	4,5 ns	6 ns	9 ns	450 mW	538 mW
SN74S*	64 mA	-15 mA	4,5 ns	6 ns	9 ns	450 mW	538 mW

- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

description

These octal buffers and line drivers are designed specifically to improve both the performance and density of three-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. The designer has a choice of selected combinations of inverting and noninverting outputs, symmetrical  $\bar{G}$  (active-low output control) inputs, and complementary G and  $\bar{G}$  inputs. These devices feature high fan-out, improved fan-in, and 400-mV noise-margin. The SN74LS\* and SN74S\* can be used to drive terminated lines down to 133 ohms.

schematics of inputs and outputs

\*LS240, \*LS241, \*LS244

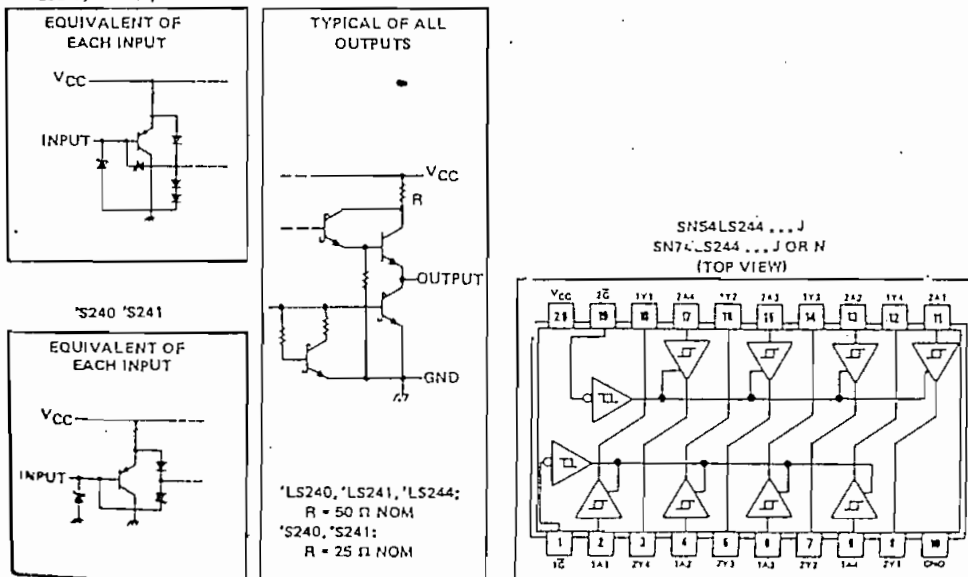


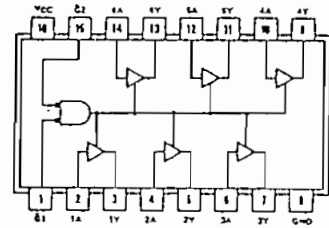
Figura A.3.7.7.



A.3.7.8. SN74LS365 HEX DRIVERS 3-STATE

HEX BUS DRIVERS

**365** 3-STATE OUTPUTS  
NONINVERTED DATA OUTPUTS  
GATED ENABLE INPUTS



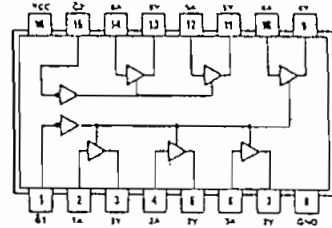
SN54365A (J, W) SN74365A (J, N)  
SN54LS365A (J, W) SN74LS365A (J, N)

Figura A.3.7.8.

A.3.7.9. SN74LS367 HEX (2-4) DRIVER 3-STATE

HEX BUS DRIVERS

**367** NONINVERTED DATA OUTPUTS  
4-LINE AND 2-LINE ENABLE INPUTS  
3-STATE OUTPUTS



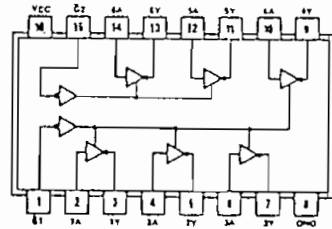
SN54367A (J, W) SN74367A (J, N)  
SN54LS367A (J, W) SN74LS367A (J, N)

Figura A.3.7.9.

A.3.7.10. SN74LS368 HEX (2-4) INVERTER DRIVER 3-STATE

HEX BUS DRIVERS

**368** INVERTED DATA OUTPUTS  
4-LINE AND 2-LINE ENABLE INPUTS  
3-STATE OUTPUTS



SN54368A (J, W) SN74368A (J, N)  
SN54LS368A (J, W) SN74LS368A (J, N)

Figura A.3.7.10

A.3.7.11

## 2758\* 8K (1K x 8) UV ERASABLE LOW POWER PROM

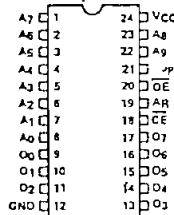
- Single +5V Power Supply
- Simple Programming Requirements  
Single Location Programming  
Programs with One 50 ms Pulse
- Low Power Dissipation  
525 mW Max. Active Power  
132 mW Max. Standby Power
- Fast Access Time: 450 ns Max. in  
Active and Standby Power Modes
- Inputs and Outputs TTL Compatible  
during Read and Program
- Completely Static
- Three-State Outputs for OR-Ties

The Intel® 2758 is a 8192-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2758 operates from a single 5-volt power supply, has a static standby mode, and features fast single address location programming. It makes designing with EPROMs faster, easier and more economical. The total programming time for all 8192 bits is 50 seconds.

The 2758 has a static standby mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW, while the maximum standby power dissipation is only 132 mW, a 75% savings. Power-down is achieved by applying a TTL-high signal to the  $\overline{CE}$  input.

A 2758 system may be designed for total upwards compatibility with Intel's 16K 2716 EPROM (see Applications Note 30). The 2758 maintains the simplest and fastest method yet devised for programming EPROMs — single pulse TTL-level programming. There is no need for high voltage pulsing because all programming controls are handled by TTL signals. Now it is possible to program on-board, in the system, in the field. Program any location at any time — either individually, sequentially, or at random, with the single address location programming.

**PIN CONFIGURATION\***



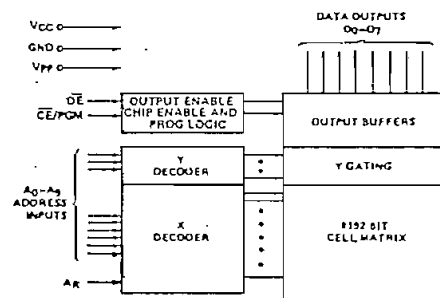
**MODE SELECTION**

MODE	PINS					
	$\overline{CE}/\overline{PGM}$ (18)	$A_R$ (19)	$\overline{OE}$ (20)	$V_{PP}$ (21)	$V_{CC}$ (24)	OUTPUTS (9-11, 13-17)
Read	$V_{IL}$	$V_{IL}$	$V_{IL}$	+5	+5	$D_{OUT}$
Standby	$V_{IH}$	$V_{IL}$	Don't Care	+5	+5	High Z
Program	Pulsed $V_{IL}$ to $V_{IH}$	$V_{IL}$	$V_{IH}$	+25	-5	$D_{IH}$
Program Verify	$V_{IL}$	$V_{IL}$	$V_{IL}$	+25	-5	$D_{OUT}$
Program Inhibit	$V_{IL}$	$V_{IL}$	$V_{IH}$	+25	+5	High Z

**PIN NAMES**

$A_0-A_9$	ADDRESSES
$\overline{CE}/\overline{PGM}$	CHIP ENABLE/PROGRAM
$\overline{OE}$	OUTPUT ENABLE
$O_0-O_7$	OUTPUTS
$A_R$	SELECT REFERENCE INPUT LEVEL

**BLOCK DIAGRAM**



\*Pin 18 and pin 20 have been renamed to conform with the entire family of 16K, 32K, and 64K EPROMs and ROMs. The die, fabrication process, and specifications remain the same and are totally unaffected by this change.

Tablă A.3.7.11.a

**PROGRAMMING**

The programming specifications are described in the Data Catalog PROM/ROM Programming Instructions on page 4-83.

**Absolute Maximum Ratings\***

Temperature Under Bias . . . . . -10°C to +80°C  
 Storage Temperature . . . . . -65°C to +125°C  
 All Input or Output Voltages with Respect to Ground . . . . . +6V to -0.3V  
 V<sub>PP</sub> Supply Voltage with Respect to Ground During Programming . . . . . +26.5V to -0.3V

*\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

**READ OPERATION**

**D.C. and Operating Characteristics**

T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub><sup>[1,2]</sup> = +5V ±5%, V<sub>PP</sub><sup>[2]</sup> = V<sub>CC</sub> ±0.6V<sup>[3]</sup>

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ. <sup>[4]</sup>	Max.		
I <sub>LI</sub>	Input Load Current			10	μA	V <sub>IN</sub> = 5.25V
I <sub>LO</sub>	Output Leakage Current			10	μA	V <sub>OUT</sub> = 5.25V
I <sub>PP1</sub> <sup>[2]</sup>	V <sub>PP</sub> Current			5	mA	V <sub>PP</sub> = 5.85V
I <sub>CC1</sub> <sup>[2]</sup>	V <sub>CC</sub> Current (Standby)		10	25	mA	$\overline{CE} = V_{IH}, \overline{OE} = V_{IL}$
I <sub>CC2</sub> <sup>[2]</sup>	V <sub>CC</sub> Current (Active)		57	100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
A <sub>R</sub> <sup>[5]</sup>	Select Reference Input Level	-0.1		0.8	V	I <sub>IN</sub> = 10 μA
V <sub>IL</sub>	Input Low Voltage	-0.1		0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0		V <sub>CC</sub> + 1	V	
V <sub>OL</sub>	Output Low Voltage			0.45	V	I <sub>OL</sub> = 2.1 mA
V <sub>OH</sub>	Output High Voltage	2.4			V	I <sub>OH</sub> = -400 μA

- NOTES: 1. V<sub>CC</sub> must be applied simultaneously or before V<sub>PP</sub> and removed simultaneously or after V<sub>PP</sub>.  
 2. V<sub>PP</sub> may be connected directly to V<sub>CC</sub> except during programming. The supply current would then be the sum of I<sub>CC</sub> and I<sub>PP1</sub>.  
 3. The tolerance of 0.6V allows the use of a driver circuit for switching the V<sub>PP</sub> supply pin from V<sub>CC</sub> in read to 25V for programming.  
 4. Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.  
 5. A<sub>R</sub> is a reference voltage level which requires an input current of only 10 μA. The 2758 S1865 is also available which has a reference voltage level of V<sub>IH</sub> instead of V<sub>IL</sub>.

**Typical Characteristics**

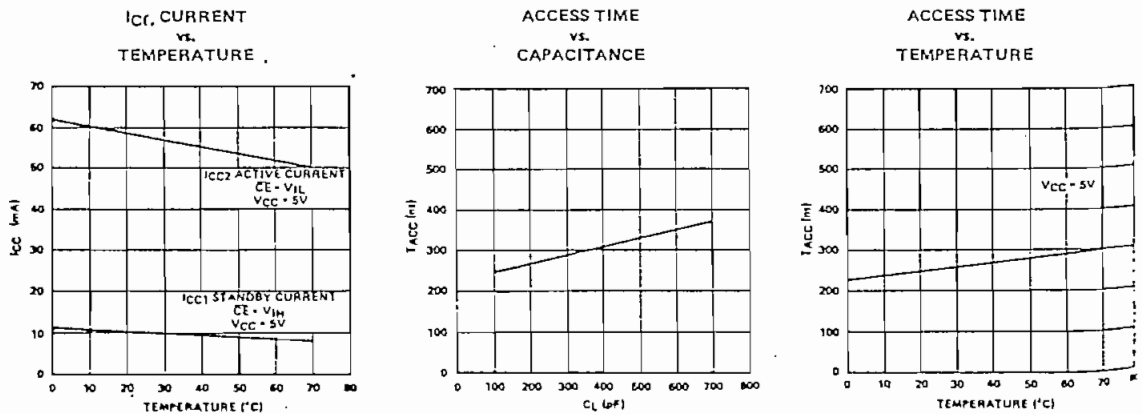


Tabla A.3.7.11.b.

## A.C. Characteristics

 $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC}^{[1]} = +5\text{V} \pm 5\%$ ,  $V_{PP}^{[2]} = V_{CC} \pm 0.6\text{V}^{[3]}$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>[4]</sup>	Max.		
$t_{ACC}$	Address to Output Delay		250	450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
$t_{CE}$	$\overline{CE}$ to Output Delay		280	450	ns	$\overline{OE} = V_{IL}$
$t_{OE}$	Output Enable to Output Delay			120	ns	$\overline{CE} = V_{IL}$
$t_{OF}$	Output Enable High to Output Float	0		100	ns	$\overline{CE} = V_{IL}$
$t_{OH}$	Address to Output Hold	0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

Capacitance<sup>[5]</sup>  $T_A = 25^\circ\text{C}$ ,  $f = 1\text{ MHz}$ 

Symbol	Parameter	Typ.	Max.	Unit	Conditions
$C_{IN}$	Input Capacitance	4	6	pF	$V_{IN} = 0\text{V}$
$C_{OUT}$	Output Capacitance	8	12	pF	$V_{OUT} = 0\text{V}$

NOTE: Please refer to page 2 for notes.

## A.C. Test Conditions:

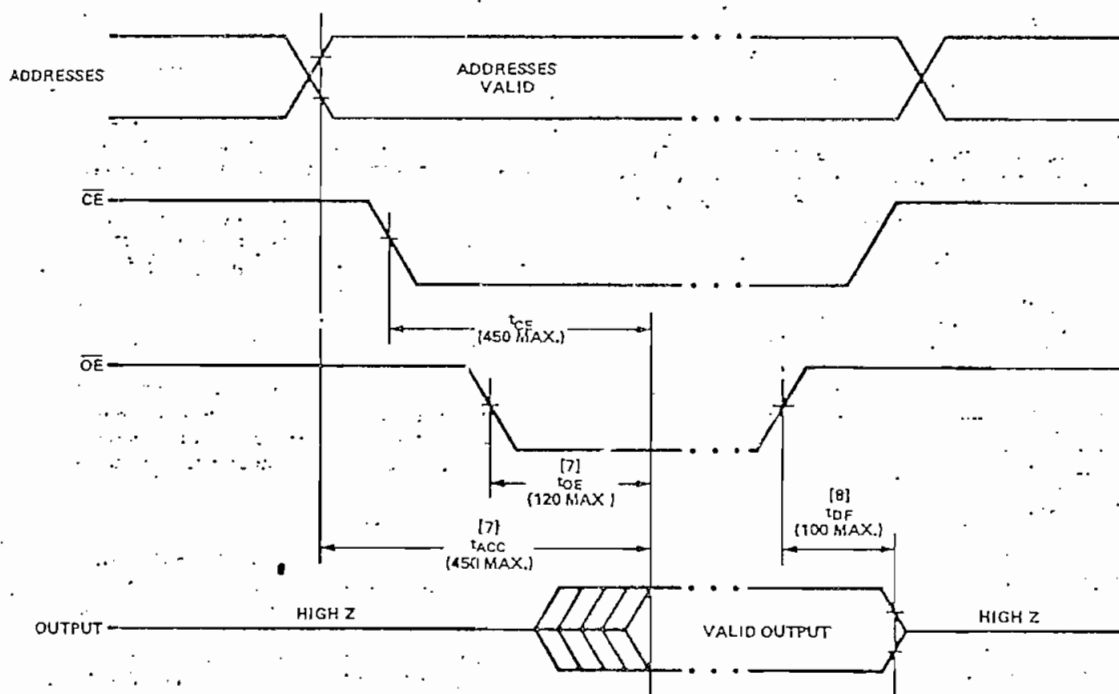
Output Load: 1 TTL gate and  $C_L = 100\text{ pF}$ Input Rise and Fall Times:  $\leq 20\text{ ns}$ 

Input Pulse Levels: 0.8V to 2.2V

Timing Measurement Reference Level:

Inputs 1V and 2V

Outputs 0.8V and 2V

A.C. Waveforms<sup>[6]</sup>

- NOTES:
- $V_{CC}$  must be applied simultaneously or before  $V_{PE}$  and removed simultaneously or after  $V_{PP}$ .
  - $V_{PE}$  may be connected directly to  $V_{CC}$  except during programming. The supply current would then be the sum of  $I_{CC}$  and  $I_{PE}$ .
  - The tolerance of 0.6V allows the use of a driver circuit for switching the  $V_{PE}$  supply pin from  $V_{CC}$  in read to 2.6V for programming.
  - Typical values are for  $T_A = 25^\circ\text{C}$  and nominal supply voltages.
  - This parameter is only sampled and is not 100% tested.
  - All times shown in parentheses are minimum times and are used unless otherwise specified.
  - $\overline{OE}$  may be delayed up to 330ns after the falling edge of  $\overline{CE}$  without impact on  $t_{ACC}$ .
  - $t_{OF}$  is specified from  $\overline{OE}$  or  $\overline{CE}$ , whichever occurs first.

Tabla A.3.7.11.c.

### ERASURE CHARACTERISTICS

The erasure characteristics of the 2758 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms ( $\text{\AA}$ ). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000 $\text{\AA}$  range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2758 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2758 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 2758 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog page 4-83) for the 2758 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms ( $\text{\AA}$ ). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm<sup>2</sup>. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12,000  $\mu\text{W}/\text{cm}^2$  power rating. The 2758 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

### DEVICE OPERATION

The five modes of operation of the 2758 are listed in Table 1. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are a +5V  $V_{CC}$  and a  $V_{PP}$ . The  $V_{PP}$  power supply must be at 25V during the two programming modes, and must be at 5V in the other three modes. In all operational modes,  $A_R$  must be at  $V_{IL}$  (except for the 2758.S1865 which has  $A_R$  at  $V_{IH}$ ).

TABLE 1. MODE SELECTION

MODE	PINS					
	$\overline{CE}/\text{PGM}$ (18)	$A_R$ (19)	$\overline{OE}$ (20)	$V_{PP}$ (21)	$V_{CC}$ (24)	OUTPUTS (9-11, 13-17)
Read	$V_{IL}$	$V_{IH}$	$V_{IL}$	+5	+5	$O_{OUT}$
Standby	$V_{IH}$	$V_{IL}$	Don't Care	+5	+5	High Z
Program	Pulsed $V_{IL}$ to $V_{IH}$	$V_{IL}$	$V_{IH}$	+25	+5	$O_{IH}$
Program Verify	$V_{IL}$	$V_{IL}$	$V_{IL}$	+25	+5	$O_{OUT}$
Program Inhibit	$V_{IL}$	$V_{IL}$	$V_{IH}$	+25	+5	High Z

### READ MODE

The 2758 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ( $\overline{CE}$ ) is the power control and should be used for device selection. Output Enable ( $\overline{OE}$ ) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time ( $t_{ACC}$ ) is equal to the delay from  $\overline{CE}$  to output ( $t_{CE}$ ). Data is available at the outputs 120 ns ( $t_{OE}$ ) after the falling edge of  $\overline{OE}$ , assuming that  $\overline{CE}$  has been low and addresses have been stable for at least  $t_{ACC} - t_{OE}$ .

### STANDBY MODE

The 2758 has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The 2758 is placed in the standby mode by applying a TTL high signal to  $\overline{CE}$  input. When in standby mode, the outputs are in a high impedance state, independent of the  $\overline{OE}$  input.

### OUTPUT DESELECTION

The outputs of two or more 2758s may be OR-tied together on the same data bus. Only one 2758 should have its output selected ( $\overline{OE}$  low) to prevent data bus contention between 2758s in this configuration. The outputs of the other 2758s should be deselected by raising the  $\overline{OE}$  input to a TTL high level.

### PROGRAMMING

Initially, and after each erasure, all bits of the 2758 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2758 is in the programming mode when the  $V_{PP}$  power supply is at 25V and  $\overline{OE}$  is at  $V_{IH}$ . The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec active high, TTL program pulse is applied to the  $\overline{CE}/\text{PGM}$  input. A program pulse must be applied at each address location to be programmed. You can program any location at any time — either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2758 must be programmed with a DC signal applied to the  $\overline{CE}/\text{PGM}$  input.

Programming of multiple 2758s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2758s may be connected together when they are programmed with the same data. A high level TTL pulse applied to the  $\overline{CE}/\text{PGM}$  input programs the paralleled 2758s.

### PROGRAM INHIBIT

Programming of multiple 2758s in parallel with different data is also easily accomplished. Except for  $\overline{CE}/\text{PGM}$ , all like inputs (including  $\overline{OE}$ ) of the parallel 2758s may be common. A TTL level program pulse applied to a 2758's  $\overline{CE}/\text{PGM}$  input with  $V_{PP}$  at 25V will program that 2758. A low level  $\overline{CE}/\text{PGM}$  input inhibits the other 2758 from being programmed.

### PROGRAM VERIFY

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with  $V_{PP}$  at 25V. Except during programming and program verify,  $V_{PP}$  must be at 5V.

Tabla A.3.7.11.d.

## A.4.1. CINTA CONECTORA DENTRO DEL MICROCOMPUTADOR

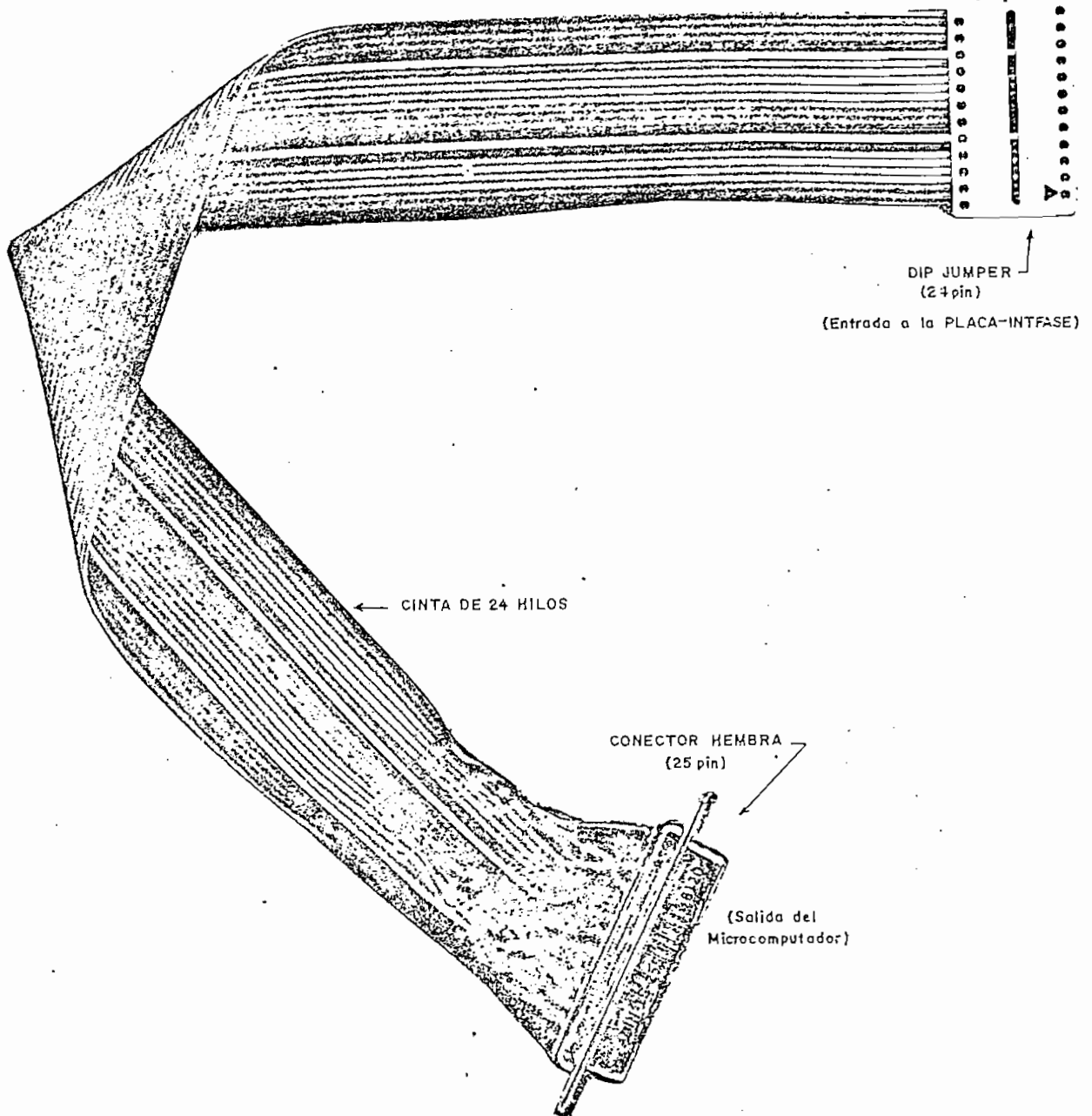


Foto A.4.1. Cinta que transporta las señales desde el cable conector a la PLACA-INTFASE

B I B L I O G R A F I A

- UNIVAC Servicing Documents. DPD Utica-Ilion, N.Y. 1968.
- The Z-80 MICROCOMPUTER HANDBOOK. William Barden, Indiana USA 1980.
- MICROCOMPUTER INTERFACING. Bruce A. Artwick, Prentice Hall Inc. N.J. USA 1980.
- INTRODUCCION A LAS MICROCOMPUTADORAS. Alfonso Espinosa. EPN 1981.
- UNIVAC Manual del Operador. Utica-Ilion, N.Y. 1970.
- ZILOG 1981 DATA BOOK. Zilog Inc. California EE.UU.
- MICROCOMPUTER-BASED DESIGN. John B. Peatman. Mc Graw-Hill USA 1977.
- THE GIANT HANDBOOK OF COMPUTER PROJECTS. By the Editors of 73 Magazine, Mc. Graw-Hill USA 1981.
- Digital INTEGRATED ELECTRONICS. H. Taub and D. Schilling . Mc Graw Hill USA 1977.

- NOISE REDUCTION TECHNIQUES IN ELECTRONIC SYSTEMS. Henry W. Ott. Wiley-Interscience Publication. N.J. USA 1976.
- Z80-CPU ZILOG PRODUCT SPECIFICATION. Zilog Inc. Calif. USA 1977.
- Z80-PIO ZILOG PRODUCT SPECIFICATION. Zilog Inc. Calif. USA 1977.
- INTEL DATA CATALOG, 1977-78.
- THE TTL DATA BOOK for Design Engineers. Texas Instrument Inc. 1978.
- BYTE. Dec. 1977, Vol. 2, No. 12.
- INTERFACE AGE. May 1977, Vol. 2, Issue 6.
- IEEE. Designing with Microprocessors. USA. 1976.
- JAMECO ELECTRONICS 1981 CATALOG. Belmont, CA, USA.