



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO DE UN SERVIDOR DE  
IMPRESORA PARA COMPUTADORAS  
PERSONALES, UTILIZANDO EL "DISEÑO  
ASISTIDO POR COMPUTADORA"

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

LEON BETANCOURT OMAR

Director de Tesis: ING. MARTIN PEREZ MONDRAGON



CD. UNIVERSITARIA

1993

TESIS CON  
FALLA DE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# INDICE

PROLOGO /iv

## CAPITULO I

Generalidades /1

I.1.- Condiciones de Diseño /1

I.1.1.-Método de Transferencia de Datos (Handshake) /3

- Strobe Básico /5
- Handshake Básico /5
- Handshake Doble /5

I.1.2.- Interfase Paralela de una Impresora /6

- Interfase Centronics /8

## CAPITULO II

**Análisis de Solución del Servidor de Impresora.**

II.1.- Planteamiento del Problema /11

II.1.1.-Etapa de Captura de Caracteres /12

- Las Señales Generadas por la Computadora /13
- Las Señales Generadas por el Servidor /14

II.1.2.-Etapa de Impresión de Caracteres /15

- Las Señales Generadas por el Servidor /16
- Las Señales Generadas por la Impresora /16

II.2.- Criterios de Selección para la Integración del Sistema /17

- Las Ventajas del 8088 /18
- Características Generales del 8088 /19

II.2.1.-Organización Interna /20

II.2.2.-Modos de Direccionamiento /23

- Direccionamiento Inmediato /23
- Direccionamiento de Registros /24
- Direccionamiento Directo /24
- Direccionamiento Indirecto de Registros /25
- Direccionamiento Indexado Directo /26
- Direccionamiento Base Indexado /26

- II.2.3.-Memoria /27
- II.2.4.-Conjunto de Instrucciones /27
- II.2.5.-Interfases e Interrupciones /27
  - Interfase de Modo Mínimo y Máximo /28
  - Interrupciones /29

## CAPITULO III

### Arquitectura del Sistema Digital.

- III.1.- Diagramas de Bloques del Sistema y Descripción de cada Bloque /30
  - El Microprocesador /32
  - Demultiplexación del Bus /34
  - Memoria /35
  - Puertos E/S para Comunicación con PCs /39
- III.1.1.-Transferencia de Caracteres entre Computadora y Servidor /40
  - Generación y Detección de Interrupción /41
  - La Detección de Carga de Caracter /43
  - Lectura del Caracter /44
- III.1.2.-Transferencia de Caracteres entre Servidor e Impresora /46
  - El 8255 (PPI) /47
  - Decodificación de Puertos /49
- III.2.- Diagramas de Flujo del Programa y descripción de cada Módulo /51
  - III.2.1.-Proceso de Inicialización del Sistema /51
    - Inicialización PPI 8255 /51
    - Estado Inicial Strobe /51
    - Estado Inicial Señal ACK /51
    - Estado Inicial de Enmascaramiento de Interrupciones /51
  - III.2.2.-Proceso de Petición de Caracter /52
  - III.2.3.-Proceso de Preimpresión /54
  - III.2.4.-Proceso de Impresión de Caracter /54
  - III.2.5.-Proceso de Actualización de Computadora en Servicio /57
    - III.2.1.-Proceso de Interrupción /59

## CAPITULO IV

### **Diseño del Prototipo y Herramientas de Diseño Asistido por Computadora.**

- AutoCAD /62
- Hi-WIRE /63

#### **IV.1.-Implementación del Prototipo, Basandose en una Computadora Personal /63**

IV.1.1.-Diseño de la Tarjeta Prototipo /64

IV.1.2.-Canal Entrada/Salida (slot) /66

IV.1.3.-Implementación del Prototipo /70

- Servicio de Interrupción /72

## CAPITULO V

### **Pruebas y Conclusiones.**

V.1.-Pruebas que Antecieron al Diseño /74

V.2.-Pruebas del Prototipo /75

V.3.-Conclusiones /80

### **APENDICE A.**

Guía de Operación del Sistema /83

### **APENDICE B.**

Listado del Programa /84

**BIBLIOGRAFIA /95**

## PROLOGO

La idea de diseñar un *Servidor de Impresora para Computadoras Personales*, surge de las "necesidades de impresión" observadas en el Instituto Mexicano del Petróleo, en donde se cuenta con gran número de Computadoras Personales y menor cantidad de impresoras, sobre todo del tipo laser.

Partiendo de esta necesidad y con la inquietud, de desarrollar el prototipo en una Computadora Personal, aprovechando las ventajas que ofrece ésta, para la optimización del diseño desde el punto de vista de su facilidad para realizar adecuaciones, que conduzcan a la reducción del tiempo de diseño y sobre todo, en el caso del autor, la oportunidad de adquirir conocimientos, aplicables en la vida profesional, se presenta en el presente trabajo el desarrollo de un prototipo de servidor de impresora para computadoras personales.

Para abordar el problema, se realizó una investigación de los equipos más comunes que existen en el mercado, resultando que la mayoría de ellos funcionan con base en multiplexión, es decir, abriendo una ruta de comunicación entre la computadora y la impresora, sin intervenir en el protocolo de transferencia de caracteres en paralelo.

Tomando en cuenta lo anterior, se decidió diseñar un verdadero "SERVIDOR", el cual funcionará con base en el protocolo de transferencia de caracteres, entre una computadora y una impresora. Además de crear una "Lista de Espera", logrando con ello dar servicio a más de una Computadora Personal.

La solución del problema se desarrolla, de modo que en el primer capítulo se plantea la estrategia con la cual se planea abordar el problema, además de explicar algunas generalidades y antecedentes básicos.

En el segundo capítulo, se plantea el problema a resolver, a través de la descripción de las señales que será necesario controlar y generar en cada etapa del sistema. En éste mismo capítulo, se describen los criterios de selección para la integración, observando las ventajas y desventajas que conlleva dicha elección.

En el tercer capítulo, se describe la solución del problema, mediante la

descripción de la arquitectura del sistema digital, dividiendo el hardware, en varios bloques para facilitar la descripción de cada bloque. Continuando con la descripción del diagrama de flujo de la programación y dividiendo el diagrama de flujo en atención a los procesos que integran el sistema.

En el cuarto capítulo, se describen las herramientas que contribuyeron para el desarrollo del prototipo en la Computadora Personal. Analizando algunos procesos que fueron de gran importancia en el diseño del prototipo.

Finalmente en el quinto capítulo, se dan a conocer los resultados obtenidos durante las pruebas de operación del sistema.

# CAPITULO I

## GENERALIDADES :

- **Qué es un "Servidor de Impresora para Computadoras Personales ?.**

Es un sistema electrónico conectado a dos o más Computadoras Personales que transfiere los caracteres enviados por éstas, a una impresora.

- **Objetivo.**

Diseñar un "Servidor de Impresora para PCs" que pueda inicialmente servir a dos o más "Computadoras Personales". El cual debe tener la capacidad de crecer, sin afectar su estructura básica, tanto en lo que respecta a "hardware" como en lo que corresponda a "software".

Cuando el sistema reciba un caracter de una Computadora diferente a la que se esté sirviendo en el sistema, deberá contarse con un proceso para crear una "lista de Espera".

## I.1 CONDICIONES DE DISEÑO.

Para llevar al cabo el diseño digital de cualquier sistema, es necesario tener una metodología que guíe y permita encontrar la solución idónea al problema.

La metodología con la que se desarrollará la solución del problema a plantear, parte del seccionamiento sistemático en subunidades, para su fácil comprensión. La necesidad de un método de trabajo resulta de mayor relevancia cuando el problema llegue a ser más grande. Por ejemplo, un programa en computadora que llegue a exceder 500,000 líneas de código, es un programa tan largo, que para una persona sería imposible poder abarcar toda la magnitud del programa sin la aplicación de una metodología de diseño. Estas técnicas o métodos tienen como nombre "TOP DOWN" y "ESTRUCTURADO".

El estudio del método para el diseño de sistemas digitales "Top Down" inicia con un análisis de las especificaciones del sistema completo, de manera ágil y sencilla a efecto de que una persona pueda rápidamente entenderlo.

Dicha metodología se apega a los principios de diseño, los cuales son :

1. **Modularidad.**- Es la subdivisión de un problema en problemas más pequeños (o módulos operativos ), con el objeto de enfrentarlos más fácilmente.
2. **Regularidad.**- Los diferentes módulos operativos deben tener gran semejanza entre sí, de manera que el tiempo de diseño se reduzca de módulo a módulo.
3. **Conectividad.**- Es la habilidad de poder transferir información entre subsistemas para comunicarse entre sí, sin que existan pérdidas de información.
4. **Localidad.**- Esta parte se puede ver en dos niveles:

**COMO SISTEMA .** - Es la relación que guarda el sistema con el medio ambiente en el cual va a operar.

**COMO MODULO .** - Establece la relación que guarda cada módulo del sistema y fija su lugar dentro del mismo.

Trabajando con el método de diseño "TOP DOWN", se hace imprescindible detallar la estrategia con la que se propone abordar el problema. Para este fin, se incluye un diagrama de flujo, en el cual se muestran los pasos a seguir para desarrollar un diseño con éste método (figura 1.1).

Basándonos en el diagrama anterior, que muestra una idea general del desarrollo de diseño, se plantea el siguiente diagrama (figura 1.2) que detalla los pasos a seguir en el diseño. Por este motivo, es importante detallar la secuencia del diagrama. Este se inicia exponiendo las especificaciones del sistema, que son base del mismo, porque de ellas depende su solidez teórica. En seguida se hace una formulación de la operación del sistema, de modo que la búsqueda de soluciones resulte más sencilla. Esta búsqueda se inicia proponiendo la utilización de una computadora, con el fin aprovechar el "hardware" de ésta. De no ser posible ésto, se tendrán que buscar

# TECNICA DE DISEÑO TOP-DOWN

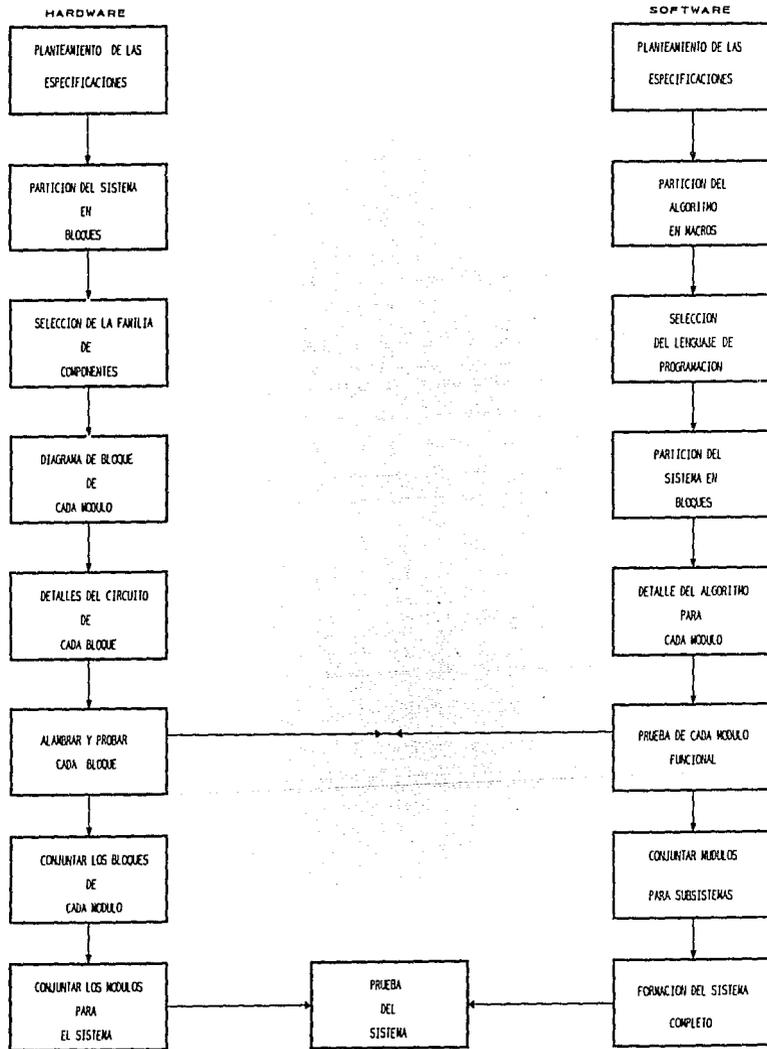


FIG. 1.1.- TECNICA DE DISEÑO "TOP DOWN"

DIAGRAMA DE FLUJO GENERAL DE DISEÑO

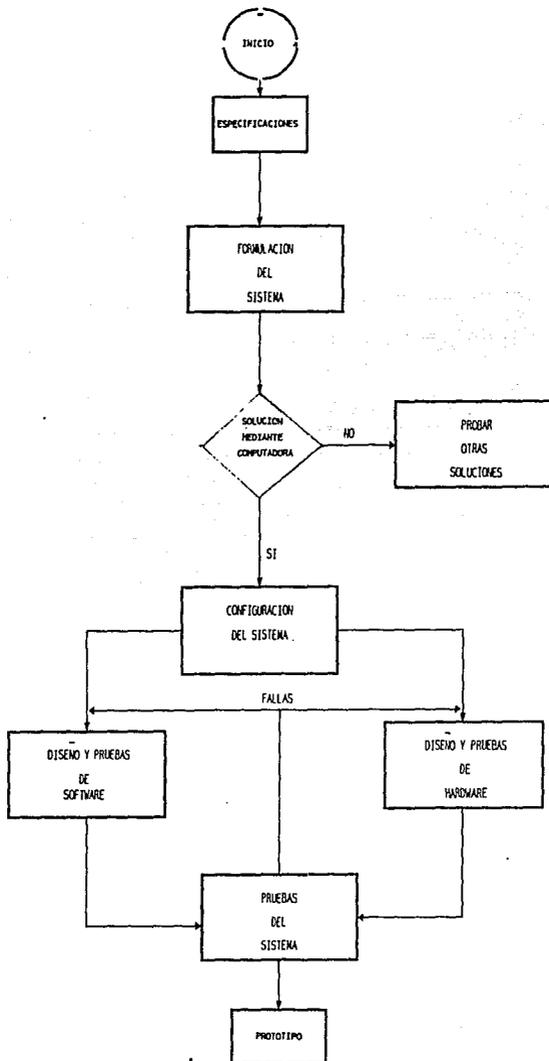


FIG. 1.2 -- TECNICA DE DISEÑO.

otras soluciones.

Una vez lograda esta solución, se establecerá la configuración del sistema desde el punto de vista "software" y " hardware". En esta parte, se habrá de tener cuidado de no cometer un error, que podría afectar a cualquiera de las dos partes del sistema.

A partir de este momento, el diseño de hardware y software se divide sin perder su interrelación, Con la finalidad de diseñar cada parte por separado.

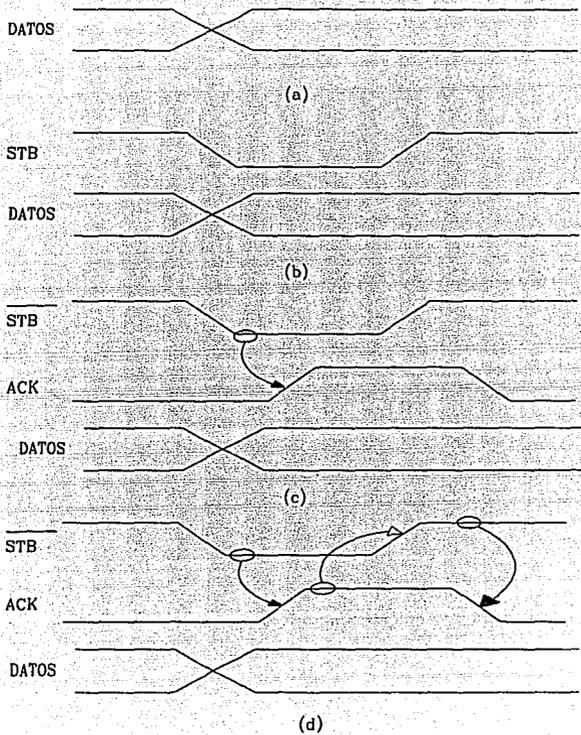
Una vez diseñado el hardware y software; hechas las pruebas de cada módulo por separado, se procede a juntarlos y a realizar las pruebas del sistema completo, con el fin de que si se llegará a presentar alguna falla, sea posible corregirla. Una vez que las pruebas sean satisfactorias, el sistema se convertirá en prototipo.

#### **1.1.1 .- METODO DE TRANSFERENCIA DE DATOS (Handshake).**

El "SERVIDOR DE IMPRESORA PARA PCs" se tiene que sujetar a normas y protocolos establecidos para transferencia paralela de datos, entre una computadora y una impresora.

En los casos en que se necesite obtener datos digitales de un simple "switch", como sería el de un termostato en un microprocesador, lo que se haría es conectar el switch a un puerto de entrada y leer este puerto. El dato del termostato, estará siempre presente y listo para ser leído en cualquier momento. En esto radica el problema, pues hay que saber en qué momento es válido el dato leído.

Estas formas o métodos de comunicación paralela para transferencia de datos se muestran en la fig. 1.3.



**fig.1.3.- Métodos de transferencia de caracteres**

En el diagrama de la fig. 1.3a el cruce de las líneas en las formas de onda, representa el tiempo en el cual un nuevo dato llega a ser válido. La ausencia de otras formas de onda indica que está operación no depende de otras señales.

### • • **Strobe Básico.**

En el diagrama 1.3b, se observan las formas de onda que representan este tipo de operación. La operación de un dispositivo como es un teclado, donde existen líneas de datos y la señal de salida STB (STROBE), permite conocer que el dato presente en el bus es válido.

Para bajas velocidades de transferencia de datos, como ocurre de un teclado a un microprocesador, un "STROBE BÁSICO" es más que suficiente. Para altas velocidades de transferencia de datos, este método no es recomendable porque no existe señal que haga saber cuándo debe enviarse el dato siguiente. En otras palabras, el sistema emisor enviaría datos más rápido, que la velocidad de lectura del sistema receptor.

### • • • **Handshake Básico.**

Las formas de onda que se muestran en la fig. 1.3c, representan transferencia de datos de un dispositivo periférico a un microprocesador. La salida periférica envía datos y "STB" (STROBE) a el microprocesador, éste detecta la señal "STB", que le indica la validez del dato en el bus y lee el byte de datos. El microprocesador transmite la señal "ACK" (ACKNOWLEDGE) es decir, reconocimiento de dato, que le indica al periférico que puede enviar el siguiente byte de datos.

Desde el punto de vista del esquema de protocolos, la señal "ACK", opera como un indicador de que el sistema está listo para recibir el siguiente byte, evitando la pérdida de datos.

### • **Handshake Doble.**

Para una transferencia de datos, aún más coordinada que las anteriores, se requiere un "handshake doble".

En el diagrama 1.3d, se muestra un ejemplo de "handshake doble", de un periférico a un microprocesador. Por ejemplo. Pensando en una conversación entre dos personas, en donde cada flanco de la señal tiene un significado, el dispositivo enviará la señal "STB" en estado "bajo", para preguntar si está listo el receptor, quién después de haber recibido esta señal, responderá con "ACK" en estado "alto" para indicar que está listo. El dispositivo periférico enviará entonces el byte de datos, levantando la señal "STB" e indicándole que hay un dato válido en el bus de datos. El

receptor después de haber recibido y leído el dato, bajará la señal "ACK" indicando al periférico que ha recibido el dato y está listo para el siguiente.

Para un "Handshake" de salida, de este tipo, de un microprocesador a un periférico, las formas de onda son las mismas pero el microprocesador envía el "STB" , los datos y el periférico la señal "ACK". Un ejemplo es la transferencia de fonemas de un microprocesador a un dispositivo sintetizador de voz.

Una vez vistos los tipos de transferencia de datos en paralelo, se puede decir que, el que se ocupará para el desarrollo de este sistema, será el "Handshake Básico", el cual describe la comunicación de un dispositivo periférico a un microprocesador.

## **I.2.- INTERFASE PARALELA DE UNA IMPRESORA.**

### **ESPECIFICACIONES:**

Sincronización: Señal externa STROBE

Niveles lógicos: TTL (Transistor-Transistor Lógico)

Handshake: Señales BUSY y ACK

En la mayoría de las impresoras, como la IBM PC, la Epson FX-80, y la NEC 8023, el dato impreso, es enviado a la impresora como caracter ASCII, en 8 líneas paralelas.

La impresora recibe los caracteres para ser almacenados en una memoria RAM interna y cuando la impresora detecta una carga de un caracter de retorno (ODH), se imprime el primer renglón del buffer. Cuando la impresora detecta un segundo caracter de retorno, éste imprime el segundo renglón de caracteres . El proceso continúa hasta que todos los caracteres han sido impresos.

La transferencia de caracteres ASCII, de una computadora a una impresora debe ser hecho en "handshake básico", porque la computadora puede enviar caracteres mucho más rápido que la impresora los pueda imprimir. La impresora debe, de alguna forma, permitir a la computadora conocer que su buffer está lleno, y que no podrá aceptar algún caracter más hasta que la impresión desaloje caracteres suficientes. Todo esto a través de la señal BUSY y ACK. El estandard de la interfase

con impresoras paralelas, es el "Standard Paralel Centronics", llamado así por la compañía que lo desarrolló.

En la figura 1.4, se muestran los tiempos y formas de ondas para transferir caracteres a una impresora IBM usando las señales de "handshake básico".

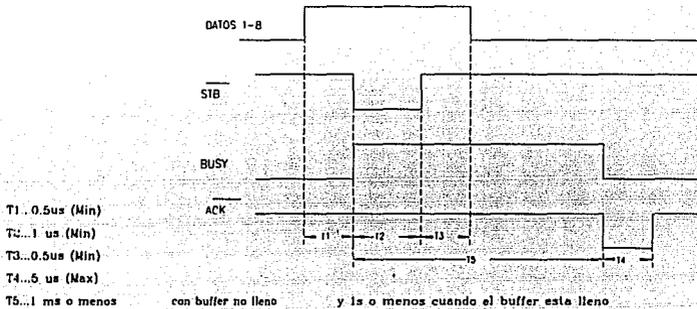


fig.1.4.- Formas de onda para transferencia de Caracteres.

Asumiendo que la impresora a sido inicializada, lo primero que se debe verificar es la señal "BUSY"(ocupado), para darse cuenta si la impresora está lista para recibir carcteres. Si la señal se encuentra en estado "bajo" ésto indicará que la impresora se encuentra lista (no ocupada). Se envía entonces un caracter ASCII en ocho líneas de datos. Después de por lo menos 0.5 us, se enviará la señal STROBE que se coloca en estado "bajo", para comunicar a la impresora que un caracter a sido enviado. Esta señal dará validez al dato en el bus, colocando la señal "BUSY" en estado "alto". Después de un mínimo de 0.5 us la señal "STROBE" podrá ser llevada a estado "alto", otra vez. Observando que el dato debe ser válido en el bus, por lo menos 0.5 us después que la señal "STROBE" haya sido levantada.

Cuando la impresora esté lista para recibir el siguiente caracter, la señal "ACK" baja por cerca de 5 us. El flanco de subida de esta señal, indicará a la computadora

(sistema) que puede enviar el siguiente caracter. El flanco de subida de la señal "ACK", también restablecerá la señal "BUSY"(baja) de la impresora, la cual indicará que la impresora está lista aceptar el siguiente caracter.

Algunos sistemas utilizan la señal "ACK"(activa baja) para el intercambio y otros sistemas usan con la misma finalidad, la señal "BUSY"(activa alta).

#### • Interfase Centronics.

Una vez que se ha descrito en forma general, la comunicación entre computadora e impresora, es necesario hacer una descripción más detallada, de lo que son las señales que se manejan. Para ello se inicia describiendo la interfase "Centronics" y con ello se describen algunos parámetros, los cuales se seguirán al pie de la letra.

La interfase centronics, generalmente tiene 36 pines (señales) en su conector. Algunos fabricantes usan uno o dos pines en forma diferente, así que es necesario consultar el manual de la impresora antes de hacer cualquier conexión.

El número de pines(36), puede parecer excesivo sólo para enviar caracteres ASCII a una impresora. El número tan grande de líneas es causado por el hecho de que cada dato y señal, tienen una "tierra individual". Por ejemplo: el pin 2 es el menos significativo de los datos enviados a la impresora y el pin 20 es la "tierra" de esta señal.

La utilidad que tienen las "tierras individuales", es reducir el ruido eléctrico que pudiera generarse en las líneas.

En el caso de hacer una interfase paralela para impresora, estas tierras deberán ir unidas en el extremo de conexión de la computadora y a "tierra" de la misma.

Hablando acerca de tierras observamos que el pin 16 está como tierra lógica y el pin 17 como tierra de chasis. Esto es para prevenir una circulación de ruido, que fluya en la tierra de las líneas.

El resto de los pines estan divididos en dos categorias: señales que entran a la computadora, para decirle la operación que se va realizar y señales de la impresora que informan el estado en que se encuentra.

La más importante de las señales de control para la impresora, es la señal "INIT", en el pin 31. Esta señal tiene la función de inicializar la secuencia internamente. La señal "STROBE" en el pin 1, tiene la función de decir a la impresora "Aquí está un caracter para tí". Hay dos pines adicionales de entrada, el pin 14 y el 16 que son generalmente tomados del lado de la impresora.

Las señales de estado de la impresora son :

- 1.- La señal "ACK" en el pin 10. En estado "bajo" indica que el caracter ha sido aceptado y la impresora está lista para el siguiente.
- 2.- La señal "BUSY" en el pin 11. Llega a estar en estado "alto", cuando por alguna razón el papel de la impresora está fuera de posición o no está lista para recibir ningún caracter.
- 3.- La señal "PE" en el pin 12. En estado "alto", indica que el switch, de papel fuera de posición ha sido activado.
- 4.- La señal "SLCT" en pin 13. En estado "alto", indica que la impresora está dispuesta a recibir datos.
- 5.- La señal "ERROR" en pin 32. En estado "bajo" se coloca en problemas diversos con las condiciones de la impresora.

---

\* Señal que indica la presencia de dato en el bus de datos

\*\* Protocolo de comunicación, para transferencia de datos en paralelo

SEÑAL PIN No.	PIN DE RETORNO	SEÑAL	DIRECCION	DESCRIPCION
1	19	-STROBE	IN	INDICA PRESENCIA DE UN CARACTER EN EL BUS DE DATOS"BAJA": NORMALMENTE "ALTA"
2	20	DATO1	IN	ESTAS LINEAS REPRESENTAN EN CONJUNTO, LOS DATOS RECIBIDOS EN PARALELO, EL ESTADO DE LA SEÑAL INDICA EN CONJUNTO UN CARACTER.
3	21	DATO2	IN	
4	22	DATO3	IN	
5	23	DATO4	IN	
6	24	DATOS	IN	
7	25	DATO6	IN	
8	26	DATO7	IN	
9	27	DATO8	IN	
10	28	-ACK	OUT	
11	29	BUSY	OUT	LA SEÑAL EN ESTADO "ALTO" INDICA QUE LA IMPRESORA NO PUEDE RECIBIR DATOS. ESTE ESTADO SE DEBE: 1.-DURANTE ENTRADA DE DATOS                      3.-CUANDO LA IMPRESORA ESTA FUERA DE LINEA 2.-DURANTE LA IMPRESION.                         4.-DURANTE EL ESTADO DE EROR DE LA IMPRESORA
12	30	PE	OUT	EN ESTADO "ALTO" INDICA QUE EL PAPEL ESTA FUERA DE LA IMPRESORA.
13	--	SLCT	OUT	ESTA INDICA QUE LA IMPRESORA ESTA EN LINEA.
14	--	-AUTO FEED XT	IN	ESTA SE REFIERE A LA ALIMENTACION DEL PAPEL DE LA IMPRESORA.
15	--	NC		NO USADO
16	--	OV		NIVEL LOGICO GND (TIERRA).
17	--	GND CHASIS	--	TIERRA DEL CHASIS DE LA IMPRESORA.
18	--	NC	--	NO USADO.
19-30	--	GND	--	TIERRA "GND".
31	--	-INIT	IN	CUANDO ES "BAJO" LA IMPRESORA ES INICIALIZADA Y SU BUFFER LIMPIADO.
32	--	-ERROR	OUT	EL ESTADO ES "BAJO" CUANDO LA IMPRESORA NO TIENE PAPEL, ESTA FUERA DE LINEA Y EN ESTADO DE ERROR.
33	--	TIERRA(GND)	--	IGUAL COMO EN LOS PINES 19 A 30.
34	--	NC	--	NO USADO.
35	--			+SV DIRECTO A RESISTENCIAS DE PULLED UP 4.7 k-ohms.
36	--	- SLCT	IN	SELECCION PARA HABILITAR ENTRADA DE DATOS ES POSIBLE QUE ESTE LA SEÑAL "BAJA".

\* LA "DIRECCION" SE REFIERE A LA DIRECCION DEL FLUJO DESDE EL PUNTO DE VISTA DE LA IMPRESORA.  
IN (ENTRADA), OUT (SALIDA).

FIG. 1.E.- CONEXION DE PINES Y DESCRIPCION DE LA INTERFASE TIPO-CENTRONIX, A UNA COMPUTADORA PERSONAL.

## CAPITULO II

### ANALISIS DE SOLUCION DEL SERVIDOR DE IMPRESORA

#### II.1.-PLANTEAMIENTO DEL PROBLEMA.

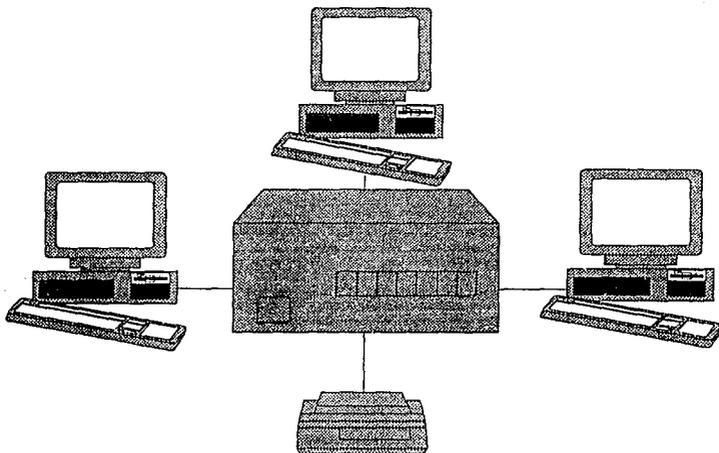


fig. 2.1.- Representación física del "Servidor de Impresora para Computadoras Personales"

Para iniciar la descripción del proceso que realizará el sistema, se dirá que el sistema inicialmente se entrará en una rutina de petición de caracteres a todas las PCs, continuará este proceso hasta que cualquiera de las Computadoras envíe el primer caracter, provocando una interrupción al procesador. Esta rutina, ocurrirá únicamente cuando la Computadora Personal (PC) envíe el primer caracter, creando una "Lista de Espera", que será incrementada tantas veces como PCs interrumpen al sistema. Este será el primer contacto de intercambio de caracteres, entre la Computadora y el sistema. Una vez que una computadora se encuentra en "Lista de Espera", el sistema se encargará en darla de "Alta"; con ésto se quiere decir que, el sistema dentro del

proceso normal, al no estar sirviendo o cargando caracteres de cualquier PC, podrá dar de "Alta" a la que se encuentra en "Lista de Espera", iniciando la carga de caracteres para ser almacenados y enviados a la impresora para ser impresos.

Una vez terminado el proceso de carga de caracteres provenientes de la Computadora en "Servicio", se procederá a darla de "BAJA", con la finalidad de que puede volver a enviar otro archivo. Además de verificar la existencia de PC en "Lista de Espera", en caso de existir, se procederá a dar de "Alta" y continuar con el proceso.

El proceso de impresión continuará, hasta que se haya impreso el último caracter del buffer, no importando que se haya dado de "Baja la PC" de donde provenían dichos caracteres.

La "Lista de Espera" se formará, asignando un número al registro correspondiente de la PC que interrumpa al procesador y que será incrementado tantas veces, como PCs interrumpan al procesador del sistema.

Para lograr todo lo anterior, el sistema tendrá que generar algunas señales como parte del protocolo para transferencia de caracteres en paralelo. Todo ésto con la finalidad de sincronizar la transferencia de caracteres, tanto del proceso de captura o carga de caracteres, como del proceso de impresión. Con el objeto de conocer mejor estas señales, se hará una descripción de las manejadas en cada proceso. Para ésto se dividirá en dos etapas la operación del sistema :

- 1.-Etapas de captura de caracteres.
- 2.-Etapas de impresión de caracteres.

### **II.1.1. ETAPA DE CAPTURA DE CARACTERES.**

En ambas etapas, es importante la sincronización de la transferencia de caracteres; pero en este caso es necesaria, debido a que cualquier pérdida de sincronización provocaría también pérdida de caracteres proveniente de alguna Computadora.

Para tener una mejor idea, de cuáles señales serán manejadas por las computadoras y cuáles por el sistema, en esta etapa la Computadora funciona como la transmisora de caracteres y el sistema (Servidor), vendría siendo el receptor de éstos.

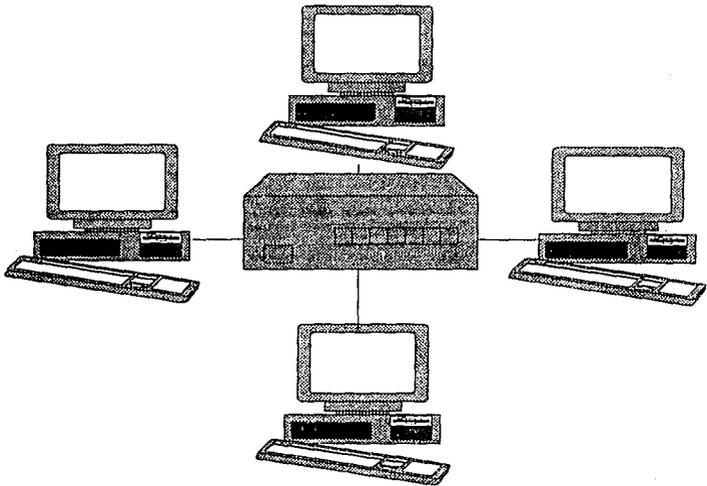


fig. 2.1.- Transferencia de Caracteres Computadora-Servidor.

• **Las Señales Generadas por la Computadora. ( Transferencia de Computadora -- Servidor).**

**STROBE.-** Esta señal le indicará al sistema, el momento en que el caracter estará presente en el bus de datos. La señal será activa "BAJA". La primera señal de activación, provocará una interrupción que dará por resultado la integración de la Computadora a la "Lista de Espera".

**D0-D7.-** En estas 8 líneas se presentará el caracter, que será cargado y almacenado por el sistema.

Este conjunto de señales serán generadas por las Computadoras Personales y con ellas el Servidor reconocerá la existencia de un caracter válido.

• **Señales Generadas por el Servidor (Receptor de caracteres).**

Las señales que generará el sistema (Servidor), tendrán la finalidad de complementar la sincronización de transferencia de caracteres, para evitar la pérdida de alguno de estos.

**ACK .-** Por medio de esta señal, el Servidor indicará a la Computadora el momento en que a terminado de procesar el caracter anterior y dará aviso de enviar el siguiente caracter. Está señal será activa "BAJA".

**BUSY .-** Con esta señal el Servidor indicará a las PCs, que se encuentra ocupado o desocupado para realizar cualquier operación, según sea el estado de esta señal. El estado de la señal dependerá de las señales "ACK" y "STROBE". La señal busy será activa "ALTA".

Las señales que indican "Estado de la Impresora", en esta etapa serán generadas por el sistema, las cuales permanecerán fijas todo el tiempo indicándole a las PCs, que no existe ningún problema con el sistema que se encuentra conectado en su puerto paralelo de impresión. Los estados en que permanecerán estas señales, serán los siguientes :

**SLCT .-** Esta señal se permanecerá en estado "ALTO", indicando a la computadora que el sistema se encuentra listo para recibir caracteres (ON LINE).

**PO .-** Está señal permanecerá en estado "BAJO", indicándole a la computadora que no existe error en la posición del papel en la impresora (Servidor).

**ERROR.-** Con señal se comunicará a las PCs, que no existe ningún tipo de error en el Servidor. Esta señal es activa "Baja", por lo tanto, permanecerá en estado "ALTO".

En esta etapa de captura de caracteres, el sistema podrá capturar caracteres a gran velocidad, ya que la computadora podrá enviar caracteres, tan rápido como el sistema se los pida.

En la figura 2.2 se muestran dichas señales.

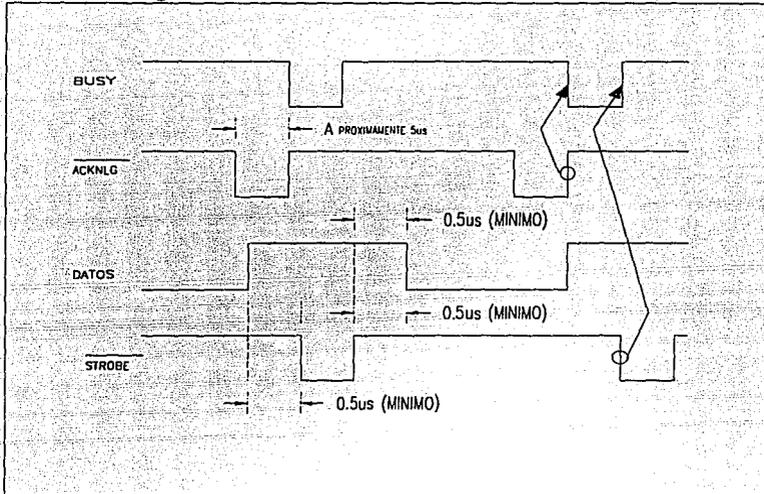


fig.2.2 .- Señales para la transferencia de Caracteres.

### II.1.2.- ETAPA DE IMPRESION DE CARACTERES.

Esta etapa del proceso se caracteriza por la transferencia de caracteres del servidor a la impresora. Ahora el sistema funcionará como transmisor de caracteres y la impresora como receptor de caracteres; el proceso se representa en la figura 2.3 .

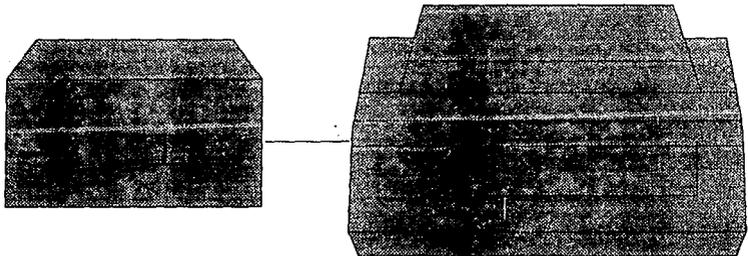


fig.2.3 .- Transferencia de Caracteres: Servidor - Impresora

• **Señales Generadas por el Servidor (Transmisor de caracteres).**

Las señales que manejará el sistema para indicar a la impresora la presencia de un caracter, serán :

**STROBE** .- Con esta señal el sistema (Servidor) indicará a la impresora, el momento en que se encontrará un caracter en el bus. Esta señal permanecerá activa en estado "Bajo", mínimo 0.5 us como se indicó anteriormente. Señal activa "Baja".

**D0-D7** .- Con este conjunto de ocho señales, el Servidor enviará el caracter para ser tomado por la impresora. En conjunto estas señales representan un caracter ASCII.

• **Señales Generadas por la Impresora (Receptor de caracteres).**

Las señales que complementarán el protocolo para la transferencia de caracteres entre el Servidor y la impresora, serán :

**ACK** .- Con esta señal, la impresora indicará a la Computadora el reconocimiento del caracter recibido y además, solicitará al Servidor un nuevo caracter. La señal deberá permanecer en estado "bajo", por un tiempo máximo de 5 us, con la finalidad de estandarizar el servidor. Esta señal es activa "Baja".

**BUSY** .- Esta señal al encontrarse en estado "Alto" evita que el Servidor envíe algún caracter a la impresora, hasta que se encuentre en estado "Bajo". Indicándole al sistema que está desocupado.

En esta etapa las señales de "estado de la impresora", serán de gran importancia para el sistema. Ya que serán realmente generadas por la impresora, de ellas dependerá la transferencia de caracteres hacia la impresora e inclusive del sistema completo.

Las distintas combinaciones de estas señales, indicarán distintas situaciones de operación de la impresora; como se observa en la tabla 2.5.

	BUSY	SLCT	PO	ERROR
EN LINEA	BAJO	ALTO	BAJO	ALTO
FUERA DE LINEA	ALTO	BAJO	BAJO	BAJO
PAPEL FUERA	ALTO	BAJO	ALTO	BAJO

Fig. 2.5.- Niveles de las señales de "Estado de Impresora".

Para lograr la correcta transferencia de caracteres, es necesario que se cumplan los límites en parámetros de tiempo, además de que la impresora es demasiado lenta en comparación a la etapa anterior. Por ejemplo: en una impresora de matriz de puntos, el espacio que existe entre una activación de la señal STROBE (Presencia de caracter) y una activación de la señal ACK (Petición de caracter), es aproximadamente de 1ms, esto en el caso de que no estuviese lleno el buffer de la impresora. Porque en situación distinta, sería aproximadamente de 1 seg.

Es necesario considerar lo anterior, para que el sistema pueda transferir caracteres sin ningún problema. En base a lo expresado, se hará una selección de componentes electrónicos, para integrar el sistema.

## II.2.- CRITERIOS DE SELECCION PARA LA INTEGRACION DEL SISTEMA

Los criterios que ayudarán a hacer una buena selección de los componentes; serán los siguientes :

**a) ¿Qué Velocidad y resolución requieren las variables propias del problema planteado?**

Como se puede deducir con base a los planteamientos anteriores, el sistema no requiere tener gran velocidad de procesamiento, principalmente por la lentitud de la impresora y por lo tanto no requiere de una respuesta rápida del sistema. Por otra parte las Computadoras, aunque serán atendidas lo más rápido posible gracias a la

memoria con la que cuenta el sistema, la velocidad seleccionada deberá ser tal que no afecte la operación general del Servidor.

El sistema manejará caracteres ASCII, por lo tanto el bus de datos que requerirá el sistema, tendrá una amplitud de ocho bits

**b) ¿Qué cantidad de datos se requiere almacenar, para atender adecuadamente la solución del problema planteado?**

El sistema no requiere de memoria para operar, pero considerando la lentitud de la impresora, sobre todo la de matriz de puntos( caso crítico ) que es base del diseño del sistema, se considera que el Servidor deberá contar con un buffer. En este caso, la cantidad de memoria que se tenga disponible, aumentará o disminuirá el tiempo de liberación de la Computadora Personal.

Tomándose en cuenta lo anterior se utilizará un microprocesador, con una capacidad de direccionamiento de memoria que permita aumentarla considerablemente. Observando las características de algunos microprocesadores y los elementos con que se cuenta para realizar el diseño y tomando en cuenta que se plantea utilizar una Computadora para desarrollar y optimar el diseño del Servidor, se decidió utilizar al microprocesador "8088" de Intel , debido también al precio y disponibilidad en el mercado.

A continuación se verán algunas de las principales características del 8088, que dieron origen a esta elección.

**• Las Ventajas Del 8088.**

El 8088 tiene un bus externo reducido a 8 bits. La razón de crearlo con este bus de datos es prever la continuidad entre el 8086 y los antiguos procesadores de 8 bits de Intel, el 8080 y el 8085. El 8088 es internamente un procesador de 16 bits, puede reemplazar a uno de los primeros procesadores de 8 bits en un sistema ya existente. Esto mejorará el rendimiento del sistema a un costo muy bajo.

Otra ventaja del 8088, es que los nuevos sistemas, pequeños y baratos aunque muy potentes, pueden diseñarse basándose en este chip. La buena razón precio y rendimiento de los sistemas basados en éste chip.

El primer microprocesador de 8 bits barato, fue el 8080, que llegó a ser uno de los favoritos de los diseñadores, por encima de otros como el 6500 y 6800 que salieron al mercado casi al mismo tiempo. Se produjo una gran cantidad de hardware y software para sistemas basados en 8080. Sin embargo, el 8080 tenía ciertas desventajas, como la de requerir tres tensiones de alimentación y dos señales de reloj diferentes. Mas tarde se diseño el 8085 para solucionar estos problemas. El 8085 requería únicamente una tensión y producía sus propias señales de reloj (hay terminales que permiten adaptar el cristal oscilador). Además tenía una velocidad mayor que el 8080. Debido a estas mejoras, el 8085 no es directamente compatible en hardware, pero sí es, en un 99 porciento de lo casos, compatible en software.

#### • Características Generales del 8088

En esta sección se verá las principales características del circuito integrado 8088. Para después extenderse un poco. El 8088 como el 8086 tienen un bus de direccionamiento de 20 bits de amplitud, lo que provee la capacidad de direccionar un megabyte de memoria.

$$2^{20} = 1.048,576 = 1 \text{ megabyte}$$

Sin embargo el registro de direccionamiento del 8088 tiene únicamente una amplitud de 16 bits. Esto equivale a 64K octetos. Para esto el procesador emplea un método, llamado "segmentación" para permitir el direccionamiento a todo un megaocteto de memoria. Zilog tiene dos versiones del Z8000, una con segmentación y otra sin ella. El Motorola MC68000 tiene 32 bits con sus registros de direccionamiento, pero solo en vía 24 bits al mundo exterior. Obviamente, esta diseñado para futuras expansiones y al igual que el Z8000, el 68000 está diseñado para trabajar con una unidad de gestión de memoria (MMU o Unidad Manejadora o controladora de Memoria). El 8088 no necesita de ningún chip adicional para realizar la segmentación.

Esta selección se hizo tomando varios criterios, además de los mencionados anteriormente.

Estos criterios son :

- 1.- Organización interna.
- 2.- Conjunto de instrucciones.
- 3.- Modos de direccionamiento.
- 4.- Memoria.
- 5.- Interfases e interrupciones.

Con los avances de los microprocesadores durante la segunda generación de microprocesadores, fueron llegando innovaciones de Intel, con el 8088 y Motorola, con el 6809. Estos dispositivos ofrecieron buses externos de ocho bits, pero internamente los datos se procesan en palabras de 16 bits, permitiendo compatividad con hardware de ocho bits, mientras también proveían de rapidez en procesamiento y una agradable transición al procesador de 16 bits.

### II.2.1.-ORGANIZACION INTERNA.

El 8088 es un microprocesador con bus de datos de 8 bits externos e internamente de 16 bits como se observa en la figura 2.6 . Los registros están agrupados en 5 categorías.

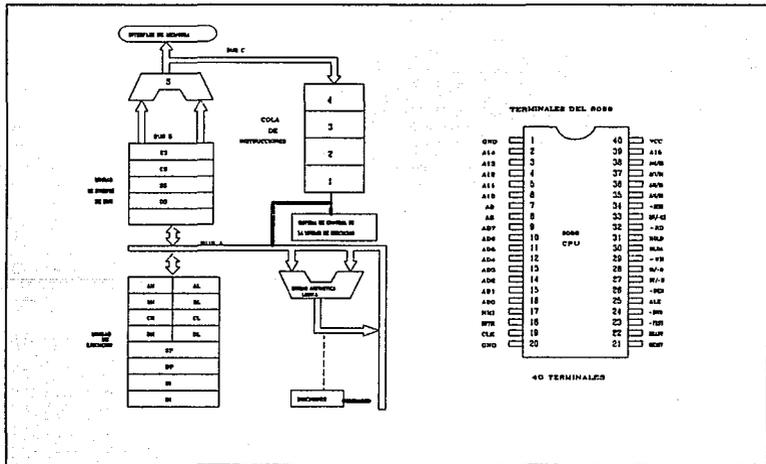


fig. 2-6.- Procesador de Propósito General 8088.

- Registro de propósito general (ax, bx, cx, dx).
- Registro apuntador e indexado (sp, bp, si, di).
- Registros Segmentados (cs, ds, ss, es).
- Apuntador de Instrucción (ip).
- Banderas (sf, df, tf, if, of, zf, af, pf, cf).

Todos los registros del 8088 tienen un ancho de 16 bits.

Los cuatro registros de propósito general ax, bx, cx, y dx, son subdivididos en parte alta y baja, cada uno de 8 bits. El registro ax de 16 bits, por ejemplo, está compuesto de dos partes de ocho bits, ah y al. Igualmente los registros bx, cx, dx. Esto permite operar directamente un registro de 16 bits o trabajar separadamente con registros de ocho bits. Teniendo en cuenta que el cambio en el valor de ax de 16 bits, también cambiará los valores de al y ah.

El acumulador es generalmente utilizado para guardar el resultado de sumas, subtracciones, etc. El registro base frecuentemente especifica el número de veces que alguna operación es repetida. Y el registro de datos dx, más frecuentemente congela datos, tal vez pasando a una subrutina para procesamiento. Estas definiciones no son fijas, y muchas veces el programador decide como usar estos registros de propósito general.

El "stack pointer" sp siempre punto de cima de la pila del procesador. El "base pointer" bp generalmente direcciona variables almacenadas dentro de la pila. El "Fuente indexado" (si) y el "Index destino" (di) son conocidos como registros encadenados.

Los registros segmentados -cs, ds, ss y es- localizan el inicio de cuatro segmentos de 64K en memoria. Como se ilustra en la figura 2.7. Un programador es libre para colocar más de cuatro segmentos, pero en este caso tiene que cambiar el correcto valor, para direccionar los segmentos adicionales.

CS:	Segmento de Código
DS:	Segmento de Datos
ES:	Segmento Extra
SS:	Segmento de Stack

fig. 2.7.- Segmentación del Microprocesador.

El "Instruction pointer" (Puntero de instrucción), especifica la siguiente instrucción de código máquina que será ejecutado, relativo a la localización en el segmento cs.

Las banderas de estado, tienen un ancho de 16 bits, solo 9 son usadas, como lo muestra la figura 2.8 . Las otras 7 no son usadas para programas.

SÍMBOLO	NOMBRE COMPLETO
o ó of	Bandera de sobreflujo
d ó df	Bandera de dirección
i ó if	Hab. de bandera de interrupción.
t ó tf	Bandera de error.
s ó sf	Bandera de signo.
z ó zf	Bandera de Cero.
a ó af	Bandera auxiliar.
p ó pf	Bandera de paridad.
c ó cf	Bandera de carga.

fig.2.8.- Banderas del 8088.

Muchas veces las banderas reflejan el resultado de varias instrucciones y operaciones.

En la figura 2.6, se proporciona la conexión externa del 8088. Del cual podemos decir que es un chip de 40 pin con 8 líneas de datos, 20 líneas de direccionamiento, 1 línea de alimentación y una de tierra.

Examinando este diagrama se muestra que:

- 1.- Direcciones y datos están colocados juntos y esto hace necesario que sean demultiplexadas estas señales.
- 2.- Un byte de datos .
- 3.- El espacio de direccionamiento es de 1Mbyte.
- 4.- Un reloj externo es requerido (8284).
- 5.- Solo es necesario un voltaje de alimentación de 5V.
- 6.- Existen líneas de control que deben ser entendidas antes de conectarse.

## **II.2.2.-MODOS DE DIRECCIONAMIENTO.**

El 8088 tiene 25 modalidades de direccionamiento diferentes. Una instrucción del 8088 no solamente contiene información sobre la operación particular a realizar, sino que también incluye las especificaciones para el tipo de operandos a manipular, así como también la posición de estos operandos. Existen siete modos importantes de direccionamiento.

1. Direccionamiento inmediato.
2. Direccionamiento de registros.
3. Direccionamiento directo.
4. Direccionamiento indirecto de registros.
5. Direccionamiento basado.
6. Direccionamiento indexado directo.
7. Direccionamiento indexado base, con o sin desplazamiento.

### **• Direccionamiento Inmediato.**

El microprocesador decodifica el modo de direccionamiento que está siendo referenciado por la sintaxis de la operación.

Por ejemplo, si la instrucción se escribe :

```
MOV AH,00
```

```
MOV AL,04H
```

entonces el valor del operando está contenido en la instrucción. Aquí en el

registro AH se pone (00) y el registro AL se carga con 0000 0100 en binario.

- **Direccionamiento de Registros.**

Con el direccionamiento de registros, el valor del operando fuente ya ha sido almacenado en uno de los registros de memoria interna del 8088. este puede ser un valor de 8 bits, de 16 bits. El microprocesador interpreta la longitud del operando por el nombre del registro. Por ejemplo:

MOV DS,AX

Este modo de direccionamiento de registros instruye al microprocesador para que tome el contenido de los 16 bits del operando fuente (registro AX) y lo desplace al registro DS de 16 bits.

De los modos principales de direccionamiento, el direccionamiento inmediato y el direccionamiento de registros tienen los ciclos de ejecución más pequeños de la máquina. Como el operando puede ser incluido en la misma instrucción y como el dato operando está ya almacenado internamente, se evita el tiempo de acceso a dispositivos externos o en memoria externa.

Los otros, requieren más tiempo de ejecución, ya que el microprocesador debe calcular la dirección del operando en la base de la dirección del segmento, desplazamiento del segmento y posiblemente contenidos del registro base o registro índice. esta dirección derivada del operando es referenciada como la dirección efectiva del operando, o EA.

- **Direccionamiento Directo.**

Con el direccionamiento directo, el desplazamiento del segmento del operando está contenido en la instrucción como una cantidad de 16 bits. Este desplazamiento se suma al contenido desplazado del registro del segmento de datos (DS) y se devuelve a la EA de 20 bits, o dirección física real.

Habitualmente, el operando de direccionamiento directo es un rólulo.

Por ejemplo, al cargar el registro AX con el contenido de la posición de memoria señalada por la dirección de memoria asociada con el rólulo MYDATA. Observar como el microprocesador almacena el byte de orden inferior en la dirección de memoria inferior y el byte de orden superior en la dirección de memoria superior.

- **Direccionamiento Indirecto de Registros.**

Con el direccionamiento indirecto de registros, en lugar de referenciar por un rótulo la dirección del operando fuente, el valor del operando es señalado por una dirección de desplazamiento almacenada en uno de los registros siguientes: SI (índice fuente), DI (índice destino), BX (registro base) o, bajo algunas circunstancias, el BP (puntero base).

El microprocesador reconoce el direccionamiento indirecto de registros por la sintaxis de la instrucción. El designador del operando fuente es reconocido por corchetes [].

- **Direccionamiento Relativo de Base.**

La dirección efectiva de un operando al utilizar direccionamiento relativo de base se obtiene de la suma del desplazamiento y contenido de un registro base ( el BX o el BP), relativo al segmento seleccionado. El modo relativo de base es más usado frecuentemente para acceder a estructuras de datos complejas, como registros. El registro base apunta a la base de la estructura, y un campo particular se selecciona el desplazamiento se introduce a diferentes campos en el registro. Al introducirse al mismo campo en diferentes registros es tan simple como cambiar el contenido del registro base.

El ensamblador reconoce los tres métodos siguientes de indicar direccionamiento relativo base:

LEA [BX]+4

LEA 4[BX]

LEA [BX+4]

El primer método es la sintaxis más frecuentemente utilizada, pero el desplazamiento puede preceder al registro base o ser incluido entre corchetes []. El movimiento a través de MESGE1 se realizaría incrementando el desplazamiento, y cambiar las referencias de mensajes se realizaría cambiando la dirección base (BX), posiblemente con la siguiente instrucción:

LEA BX,MESGE2

• **Direccionamiento Indexado Directo.**

En el desplazamiento indexado directo, la dirección de desplazamiento del operando se calcula sumando el desplazamiento a un registro índice (SI o DI) en el segmento seleccionado. (frecuentemente, el direccionamiento indexado se utiliza para introducirse a los elementos de un arreglo estático. El valor del desplazamiento localiza el comienzo del arreglo el valor almacenado en el registro índice selecciona un elemento simple en la estructura. De manera distinta a los registros, cuyas longitudes del campo individual pueden variar en tamaño y tipo de datos, los elementos del arreglo son homogéneos. Como lo elementos son del tipo de datos y tamaño, para moverse a través del arreglo basta con incrementar o decrementar sistemáticamente el desplazamiento. Por ejemplo :

```
MOV SL,4
```

```
MOV AL,ARAT1[SI]
```

Se debe tener cuidado para seleccionar un valor de desplazamiento apropiado relativo al tipo de datos de los elementos del arreglo. El ejemplo dado cargaría el registro AL con el quinto valor de ARRAY1.

```
MOV AX,ARRAY1[SI]
```

Como se muestra en la figura 2.d, la misma instrucción podría cargar el registro AX con el valor de 16 bits, dependiendo del tipo de datos del arreglo.

• **Direccionamiento Base Indexado.**

Con el direccionamiento base indexado, el operando se localiza en el segmento seleccionado en un desplazamiento determinado por la suma de los contenidos del registro base, registro índice y opcionalmente, un desplazamiento. Si no se incluye un desplazamiento, entonces el direccionamiento base indexado se utiliza con más frecuencia para tener acceso a los elementos de un arreglo dinámico (esto es, un arreglo cuya dirección base puede cambiar durante la ejecución de un programa). El incluir desplazamiento permite tener acceso a un elemento individual de un arreglo, siendo el arreglo un campo en una estructura, como por ejemplo un registro.

En este último caso, el registro base apuntaría a la base de la estructura del registro, el desplazamiento (almacenado en DI) contendría la distancia desde el comienzo del registro al comienzo del campo del arreglo, y el desplazamiento del elemento estará siempre contenido en la variable ELEMENT. Supongamos que

ELEMENT contiene 03.

### **II.2.3. MEMORIA.**

El bus de direcciones representa en número de localidades de memoria que puede ser accedido. El 8088 cuenta con un bus de direcciones de 20-bits y puede direccionar un máximo de  $2^{20}$  (cerca de un millón ) localidades diferentes de memoria. El problema es que los registros el 8088, tienen un ancho de 16 bits. Para acceder por completo a la memoria, se utiliza la segmentación de memoria, la memoria se divide en bloques de 64K lógicos. Con este método, la dirección puede ser expresado en dos valores : la dirección del bloque o segmento, el cual contiene el inicio del segmento y el offset respectivamente, los dos de 16 bits. En conjunto son llamados dirección lógica. Cada una de estas direcciones, tiene su dirección física.

Con segmentos, el procesador 8088 puede direccionar eficientemente 1 megabyte de memoria mientras emplea registros de 16 bits. Un beneficio adicional de la segmentación, es que hace más fácil mover programas a nuevas localidades físicas, cambiando solamente la dirección base del segmento.

El espacio de direccionamiento (1 megabyte) está localizado en 16 segmentos de 64K cada uno.

### **II.2.4.- CONJUNTO DE INSTRUCCIONES.**

El juego de instrucciones del 8088 de Intel es comparable al de otros como son: el MC68000 de Motorola y Z8000 de Zilog. Los tres fabricantes han adoptado un amplio juego de instrucciones haciendo que sus máquinas de 16 bits sean mucho más potentes que sus equivalentes de 8 bits. Han añadido operaciones como multiplicación y división. Es importante realzar que los procesadores de Zilog y Motorola son realmente en muchos aspectos, máquinas de 32 bits encapsuladas en chips de 16 bits. El 8088 de 8 bits pueden compararse, tanto en velocidad como en flexibilidad, en cuanto operaciones estándar de 8 y 16 bits. Sin embargo por motivos como ser más lento en operaciones de multiplicar y dividir y alguna de las operaciones directas con bits, el 8088/8086 generalmente, se encuentran detrás de los otros dos en pruebas de velocidad.

### **II.2.5.- INTERFASES E INTERRUPTACIONES.**

La interfase de entrada o salida del microprocesador 8088 permiten comunicar a este, con el mundo exterior. La forma en la cual el 8088 trata con circuitos de entrada

o salida, es similar a la forma en la cual este hace interfase con los circuitos de memoria. Esto es, la transferencia de datos con la cual la multiplexación de direcciones y datos toma un lugar importante. Este bus paralelo permite una interfase con periféricos, tales como expansores de manejadores de periféricos y controladores de comunicación.

El 8088 puede direccionar 65,536 puertos diferentes. Hay un solo pin en el 8088 que permite conocer cuando es leído o escrito una localidad de memoria o un puerto. Accesar a un puerto significa la activación IO/M en estado "alto" y memoria en estado "bajo".

• **Interfases de Modo Mínimo y Máximo.**

El pin MN/-MX, es usado por el microprocesador para utilizar este en modo MINIMO o MAXIMO. Conectando este pin, se definen las líneas de control del 8088. Al configurarlo en modo máximo poniendo el pin en estado "BAJO", para diseñar un sistema en donde se necesita que varios circuitos acceden al bus de direcciones y bus de datos. Cuando el 8088 está en modo máximo, este tiene que ser usado con el Controlador de Bus "8288", el cual decodifica las líneas de entrada para expandir las líneas de control. Como resultado, se obtiene veinte señales de control en lugar de las 13 del modo MINIMO. Intel pensó en esta forma por la misma razón se necesita multiplexar las líneas de datos y direcciones. Hay solo 40 pines disponibles.

En la fig. 2.E, en donde observa la diferencia entre el modo MAXIMO y MINIMO de las señales de control del 8088.

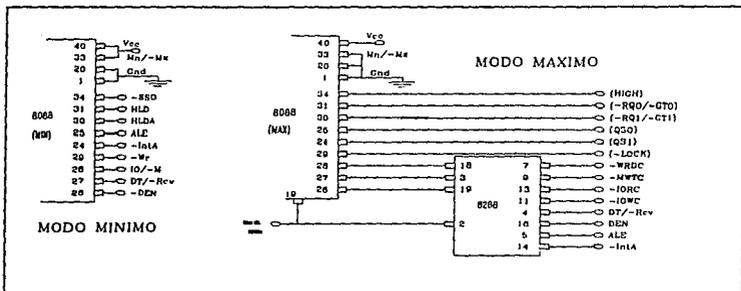


fig. 2.E .- Diferencias entre las señales de control en modo MAXIMO y MINIMO.

- **Interrupciones.**

Cada tipo de interrupción puede ser llamado, tanto hardware como por software. Hay un bit de control, llamado indicador de interrupción (IF) que controla si el 8088 responde o no las interrupciones externas. Este bit puede activarse o desactivarse por medio de órdenes de activar interrupciones (SSI) y el de borrar interrupción (CLE). Esto abre y cierra la puerta para interrupciones de el CPU. Cuando el bit de interrupción está desactivado es como si hubiera un cartel de "no molestar" en dicha puerta.

Las interrupciones de hardware actúan de la siguiente forma:

Cuando un dispositivo externo requiere servicio, produce un señal en la línea de petición de interrupción (INTR) del 8088. Si el 8088 puede responder (interrupción activa), envía una señal de recibido, bien de forma directa (en modo mínimo), o bien vía el controlador de bus 8288 (en modo máximo).

Es dispositivo externo indica, mediante un byte en el bus de datos, qué tipo de interrupción desea. El 8088 utiliza este número para localizar el puntero de la tabla de interrupciones. A continuación, el 8088 almacena las condiciones de sus indicadores y un puntero en la pila con la dirección de retorno, y carga el segmento código y el puntero de instrucciones desde la tabla de interrupción. Esto hace que el procesador ejecute la rutina de servicio. Al final de dicha rutina debe haber una instrucción de retorno de interrupción (IRET). Esta interrupción restaura los indicadores, el segmento código y el puntero a la dirección de retorno. Se debe usar la instrucción IRET ya que una instrucción de retorno "normal" no restauraría los indicadores, por lo que no sea adecuada para estos propósitos (de otra manera se rompería la pila). En este punto, el procesador ha vuelto a sus operaciones normales.

Hay una clase especial de interrupciones llamadas "interrupciones no enmascarables". El término "no mascarable" se refiere al hecho de que esta clase de interrupciones no pueden desactivarse limpiando el indicador de interrupciones. Esta clase de interrupciones genera una interrupción del tipo dos, o sea, su puntero se encuentra en la dirección  $2 + 4 = 8$  de memoria. Las interrupciones no enmascarables se reservan para casos de emergencia como fallos de potencia o errores de memoria.

## CAPITULO III

### ARQUITECTURA DEL SISTEMA DIGITAL

En este capítulo, se establece la arquitectura del sistema con la que se da solución al problema planteado, desde el punto de vista del "hardware" (parte física) y del "software" (programa). Con la finalidad de hacer una descripción completa, se dividirá la descripción en varios módulos, siguiendo una secuencia lógica en la formación del sistema.

#### III.1.-DIAGRAMAS DE BLOQUES DEL SISTEMA (hardware) Y DESCRIPCION DE CADA BLOQUE.

El sistema desde el punto de vista "hardware" está dividido en varios bloques, como se observa en el diagrama general de la figura 3.a :

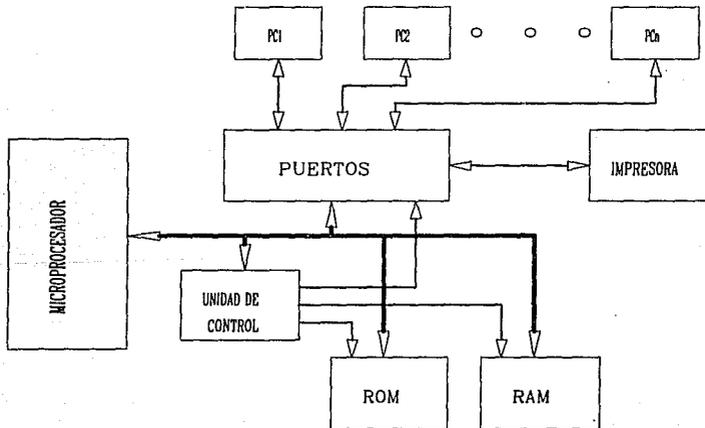


fig.3.a.- Diagrama de Bloques General.



- **El Microprocesador.**

En el diagrama 3.c, se muestra el principal punto de consideración, cuando se diseña con un microprocesador como es el 8088, en el cual se muestra la conexión básica. Una vez que se ha cubierto en forma general toda la parte teórica que la conforma, tanto al microprocesador como al sistema, se describirá la parte física.

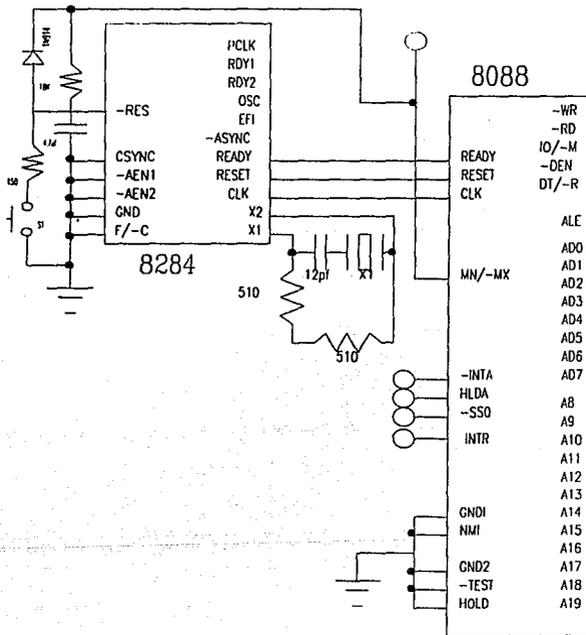
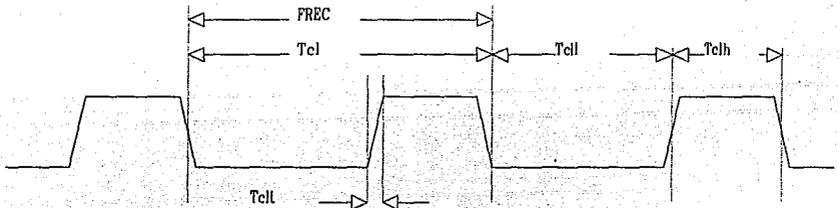


fig.3.c.- Diagrama de conexión del 8088.

El sistema debe contar con una alimentación o batería. Además de una señal de reinicialización (RESET), que indicará al microprocesador, el momento de restablecer los valores iniciales, como si hubiese sido apagado o encendido el sistema. Esta señal,

es indispensable cuando un programa se mete en un ciclo infinito, como es el caso del servidor. El regulador para la alimentación, es importante para evitar que el sistema sea desconectado y reconectado, con el peligro de provocar algún daño en los circuitos del sistema, al evitar sobretensiones (5 Vdc).

Con relación a la figura 3.c; El 8284 es un chip con 18 terminales que genera las pulsaciones del reloj, los cuales determinan la velocidad de funcionamiento de sistema. Para ello, el 8284 necesita un cristal oscilador cuya frecuencia es aproximadamente el triple de la salida del circuito. La velocidad mínima del 8088 es de 2Mhz y la máxima es de 5 MHz, dependiendo del valor del cristal. En la figura 3.d; se muestran los tiempos del ciclo de reloj del microprocesador 8088.



NOMBRE	DESCRIPCION	MAX	MIN	UNIDAD	CHIP
Tcl	Periodo de Ciclo de Reloj	500	200	ns	8088
FREC	Frecuencia de Reloj	5.00	2.00	MHz	8088
Tcll	Nivel Bajo de Ciclo	.5Tcl	.5Tcl	ns	8088
Tchl	Nivel Alto de Ciclo	.5Tcl	.33Tcl	ns	8088
Tcrl	Tiempo de Subida	10		ns	8088

fig. 3.d - Requerimientos del reloj del microprocesador 8088.

El Generador de Pulsos "8284", emite tres señales hacia el microprocesador. Estas son: el CLK (señal de reloj), el RESET y el READY. Las dos últimas, están sincronizadas con la señal de reloj.

El uso más común de la señal READY, es sincronizar al microprocesador con los dispositivos externos más lentos.

- **Demultiplexión del Bus.**

Uno de los beneficios de tener juntos los buses de direcciones y datos, es el tamaño físico del microprocesador. Y los perjuicios, son los tiempos que llegan a ser verdaderamente críticos, por lo hay que asegurar el usa dispositivos que eviten otro problema. Aumentando con ello el número de dispositivos del sistema.

En la figura 3.e. se observa la demultiplexación del bus. En donde el 8088 puede directamente controlar al 74373 con la señal de salida "ALE", activa "ALTA", que indica la existencia de una dirección válida en el bus. La señal "OC" (control de salida) del 74373 que controla las salidas "tres estados" del circuito, la cual permanecerá activada (activa "baja"). En conclusión, el 74373 retendrá a la salida la dirección válida que se encuentre a la entrada del dispositivo, por medio de la activación de la señal ALE.

El 74245 es un dispositivo bidireccional, que se utilizará para la tranferencia de datos, el cual aísla al bus de datos del resto de los circuitos. La dirección de transferencia, está bajo el control del pin No.1 "DIR". El control de la dirección será perfectamente manejado por la señal DT/-R del microprocesador, que indica si se quiere transmitir o recibir datos. El control de la habilitación del dispositivo, para la salida o entrada de datos, es directamente manejado por -DEN del 8088 (habilitación de datos).

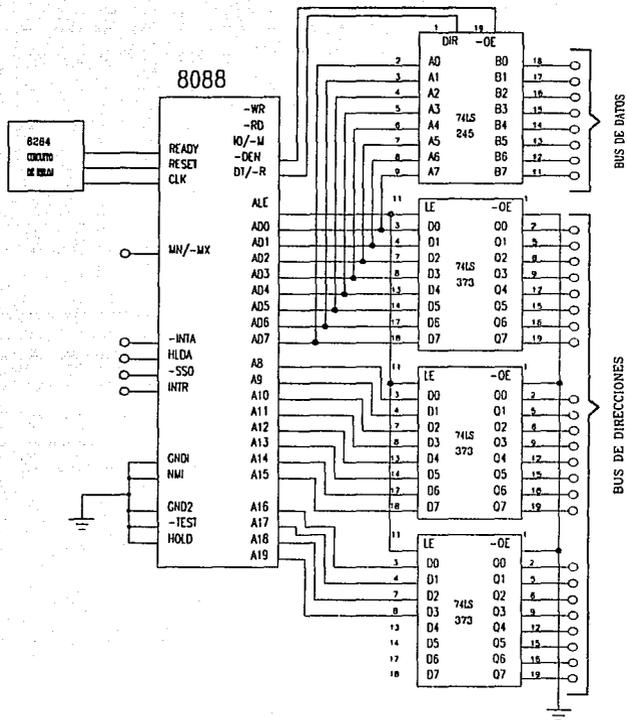


fig.3.c.- Demultiplexión del Bus.

- **Memoria.**

Algunos sistemas no tienen la necesidad de tener megabytes de memoria, para almacenar datos. En el caso del servidor, aunque no es indispensable para el funcionamiento del sistema, si se considera la existencia de memoria RAM la cual nos ayudará a mejorar el funcionamiento del sistema, al hacer más rápida la transferencia de caracteres de la Computadora Personal al Servidor.

Inicialmente se quiere que el sistema opere con pequeños archivos, por lo tanto la capacidad de memoria no deberá ser muy importante por el momento.

Una memoria RAM estática, es buena y fácil de manejar. Se pueden poner datos en ésta, tanto tiempo como se requiera, no hay refresco ni problemas de reloj, que se tiene cuando se trabaja con memorias dinámicas. La memoria dinámica tiene más capacidad de almacenamiento en el mismo empaque y se hace indispensable en sistemas que manejan bastante cantidad de datos.

La principal dificultad con las memorias dinámicas, se basa en el hecho de que la celda almacenadora es nada más que un capacitor. Siempre que se almacena un "1" en la celda de la memoria dinámica, lo que se hace es poner una carga en el capacitor y como se sabe, esto dista de ser permanente.

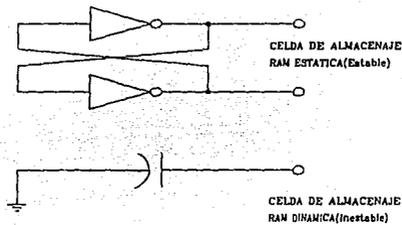


fig 3.f.- Representación de memoria Dinámica y Estática.

La forma como se resuelve este problema, es leyendo el valor de la celda mientras el dato permanece ahí y entonces reescribirlo. Este proceso de lectura y reescritura, es llamado "refresco". Algunas RAM's dinámicas requieren que el refresco de cada celda se haga por lo menos cada 2 milisegundos y algunas memorias grandes, sólo necesitan un refresco cada 4 milisegundos.

En lo que respecta a la memoria EPROM (Memoria la lectura, programable y borrrable) no existe ningún problema al manejarla.

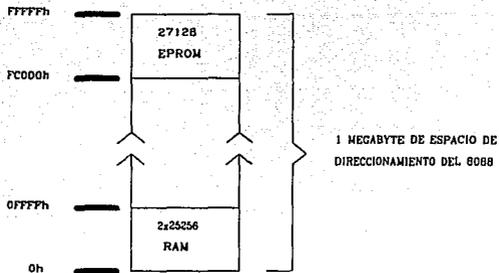
Tomando en cuenta todo lo anterior, inicialmente se manejará una memoria RAM estática de 64Kbyte (dos "26256" de 32k cada una) y una memoria EPROM de 16K (una "27128" donde se encontrará el programa. El tamaño de esta memoria es bastante grande ya que con una 2716 es suficiente, pensando en la posibilidad de crecimiento del sistema a su capacidad máxima: ocho PC.

Antes de iniciar el diseño del circuito de control, se tendrá que definir el mapa de memoria para el sistema. Esto significa, decidir dónde se localizarán los dispositivos de memoria, en el espacio de direccionamiento de 00000h a FFFFFh. Por el momento se utilizan 80Kbyte, entre memoria ROM y RAM sin tomar en consideración periféricos, lo que hace necesaria la utilización de un mapa de direccionamiento (figura 3.g).

Cuando inicia el microprocesador, tendrá que ir al lugar de la primera instrucción. Algunos CPU's, tales como el 8080 y Z80, esperan la primera instrucción en 0000h cuando son encendidos o reinicializados. Otros como el 8088, automáticamente ejecutan la instrucción que se encuentra en la dirección absoluta FFFF0h, por ello es conveniente poner la memoria permanente en la cima del rango de direccionamiento del CPU. Algo que se acostumbra hacer, es poner un salto en esa localidad hacia una localidad más baja.

En el sistema se usará una memoria ROM de 16Kbytes (3FFFh) de tamaño; el 8088 direccionará la ROM desde la localidad FC000h (FFFFFh menos 3FFF) a FFFFFh. La memoria RAM en el sistema, tiene un tamaño de 64Kbytes, ésta se localizará en la parte inferior del mapa. Siendo direccionable desde 00000h a 0FFFFh.

En esa situación es necesario tener señales de control para poder acceder a la memoria correcta. Con este objeto utilizaremos los pines A16 a A19, los cuales son '1's, cuando se direcciona a la memoria ROM y cuando son '0's, se direcciona a la RAM. Esto quiere decir, que las señales nos pueden servir para dar origen a las señales de control de memoria del sistema. Con este objetivo se deberán identificar las señales que hay que controlar.



**fig.3.g.- Mapa de Memoria.**

La RAM tiene dos pines de control y uno de habilitación de escritura. La señal "Chip Select" activa por completo al circuito (activa "baja"), cuando no está activo se encuentra en modo de "Stand by" de baja potencia (dormido) y las salidas están flotando. La línea de "-Output Enable" controla la salida o entrada de datos. Esta señal debe ser controlada, porque la memoria RAM emplea los mismos pines para entrada y salida de datos. Cuando la señal "-Out Enable" está en estado "bajo" los pines serán de salida y cuando éste en estado "alto" los pines serán de entrada. La señal de "Write Enable" (habilitación de escritura), cuando se encuentre en estado "alto" los pines de salida cargarán el dato, que será almacenado en la localidad direccionada.

Al utilizar todos o alguno de los pines de A16 a A19, el sistema puede diferenciar entre memoria RAM o ROM. Por ejemplo, el pin A19 y A18 divide el mapa en dos partes, 256Kbyte a memoria ROM con A19 y A18 en estado "alto" y 768K de memoria RAM con A19 y A18 en estado "bajo".

El espacio de direccionamiento de RAM, comprende dos dispositivos: los puertos E/S y desde luego la RAM. Para diferenciar dichos dispositivos el 8088, utiliza la señal "IO/-M", que indica el tipo de operación a realizar (puertos o memoria).

Además de las señales "-RD" y "-WR" con las que el 8088 indica si lee o escribe un dato en el bus de datos.

Con las señales anteriores, se construye el control lógico de los dispositivos que se van utilizar, como se observa en la fig. 3.h. En donde pueden notarse la tabla de estados de las señales de entrada y salida, incluyendo el esquemático.

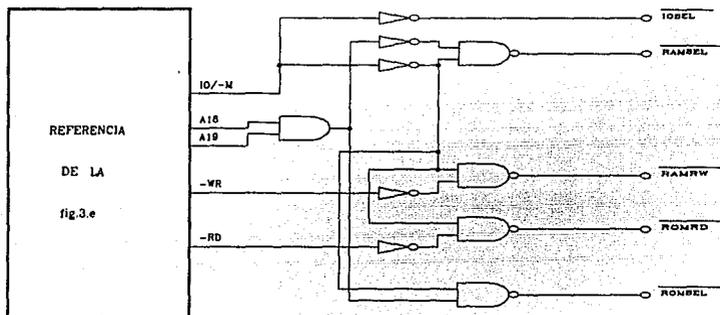


TABLA DE VERDAD DEL CONTROL LÓGICO DE MEMORIA									
ENTRADAS				SALIDAS					
IO/-M	A18 A19	WR	RD	RAMBEL	RAMRW	ROMBEL	ROMRD	IOBEL	OPERACION
BAJO	BAJO	ALTO	BAJO	BAJO	ALTO	ALTO	BAJO	ALTO	LECTURA RAM
BAJO	BAJO	BAJO	ALTO	BAJO	BAJO	ALTO	ALTO	ALTO	ESCRITURA RAM
BAJO	ALTO	ALTO	BAJO	ALTO	ALTO	BAJO	BAJO	ALTO	LECTURA ROM
BAJO	ALTO	BAJO	ALTO	ALTO	BAJO	BAJO	ALTO	ALTO	NO OPERACION
ALTO	X	X	X	ALTO	ALTO	ALTO	ALTO	BAJO	PUESTOS

fig. 3.h.- Control lógico de Memoria y Puertos.

- **Puertos E/S para Comunicación con las Computadoras Personales.**

Una de las partes importantes del sistema, es la comunicación con los dispositivos externos, como son las Computadoras Personales y la Impresora.

El primer paso para el control de periféricos, es dado con la utilización del circuito 74373, que contiene ocho "LATCHs". Este circuito tiene el objetivo de retener el dato válido, que se encuentra en los ocho pines de entrada, a través de la activación de la señal "C" en estado "alto". Las salidas del circuito, tienen la capacidad de "tres estados", permitiendo tener de varios circuitos conectados al mismo bus.

### III.1.1.- TRANSFERENCIA DE CARACTERES ENTRE COMPUTADORA (PC) Y SERVIDOR.

El proceso de *transferencia de caracteres entre la Computadora Personal y el Servidor*, es un poco complejo ya que envuelve varios procedimientos para su realización. Los cuales son :

1. Generación y Detección de Interrupción.
2. Detección de Carga de Caracter.
3. Lectura del Caracter.

- **La Generación y Detección de Interrupción**, se realizará cuando una PC inicia enviando el primer caracter al Servidor; la activación de la señal STROBE que se genera con la presencia del caracter, cambia de estado la salida de la compuerta NAND que integra todas las señales STROBE de las Computadoras Personales conectadas al Servidor. Cada una de estas señales puede ser enmascarada. La señal STROBE, se encuentra junto con la señal de "control de interrupción" (-CINT), a la entrada de una compuerta OR, en el caso en que esta señal se encuentre en estado "Alto" ("1"), la señal STROBE estará enmascarada y en caso en que la señal se encuentre en esta "Bajo"("0"), la señal STROBE no estará enmascarada. Todo esto con la finalidad de que las PCs interrumpan al procesador, sólo cuando se envíe el primer caracter. La compuerta NAND al recibir la activación de una señal STROBE no enmascarada, cambia el estado de la salida ("Alto") activando el pin "C" del 74373 y congelando el estado de las señales STROBE enmascarables. El dato retenido, indicará al sistema que existe una nueva PC pidiendo servicio. La señal que activa al pin "C" del 74373, es la misma que generará la interrupción al procesador. El proceso descrito anteriormente se observa en el esquema de la figura 3.i .

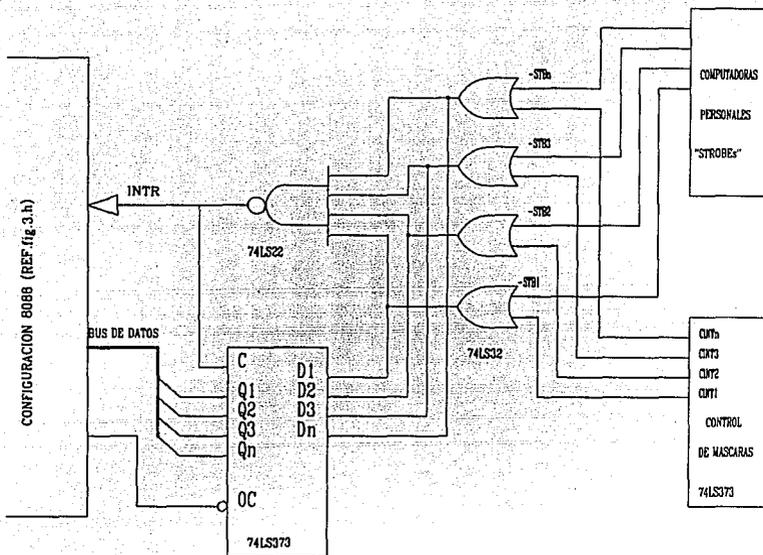


fig. 3.i - Generación y Detección de Interrupción.

### La Interrupción.

Antes de hacerse cualquier descripción del hardware de interrupción, es importante mencionar, por qué se eligió un sólo origen de interrupción teniéndose varias fuentes de interrupción. Esto se debió a que no hubiera sido práctico, el utilizar una interrupción por cada PC, ya que el número de dispositivos hubiera crecido, al integrar un Programador de Interrupciones "8259".

El pin INTR, es la puerta donde el 8088 da servicio a rutinas de interrupción externas. La utilización de éste, es ideal en la operación del sistema en "stand-alone". El 8088 puede manejar 256 vectores priorizados, verificándose el INTR al fin de cada ciclo de instrucción. Si el CPU ve que la línea a sido activada por un dispositivo externo, saltará (vector) a la subrutina localizada en la dirección almacenada en la memoria (tabla de vectores).

La localización de la tabla, está entre las direcciones 00000h-003FFh. La construcción de la tabla de vectores, es una de las primeras tareas que se deberá resolverse por el firmware del sistema.

La **-INTA** (Reconocimiento de Interrupción) es una línea de salida del procesador y su estado de activación es "Bajo", cuando un dispositivo externo hace una petición de interrupción, colocando la señal en el pin **INTR** en estado "Alto. Siempre y cuando el 8088 reciba esta petición, en cual esperará hasta que el ciclo de la instrucción en proceso haya terminado y es entonces cuando la línea **-INTA** colocará un estado "Bajo" en dos ciclos consecutivos. Intel llama a esta operación, ciclos de "Reconocimiento de Interrupción".

La primera activación permite al dispositivo saber que la interrupción a sido servida. Durante el segundo ciclo, el CPU espera que el dispositivo externo ponga el nivel de interrupción (0 a 255), en el bus de datos. El 8088 toma el código y encuentra el apropiado acceso en la tabla de interrupciones. A continuación, el 8088 salva las condiciones de sus indicadores y el puntero en la pila con la dirección de retorno y además carga el segmento de código y el puntero de instrucción. Lo expresado hace que el procesador ejecute la rutina de servicio.

Al final de la rutina debe haber una instrucción de retorno de interrupción (**IRET**). La instrucción restaura los indicadores, el segmento de código y el puntero a la dirección de retorno y ejecuta el código que encuentre ahí.

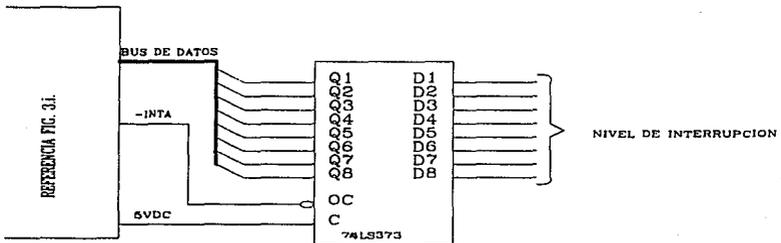


fig.3.1.- Colocación del Nivel de Interrupción.

La forma en que se coloca el nivel de interrupción en el bus de datos, se muestra en la figura 3.1.

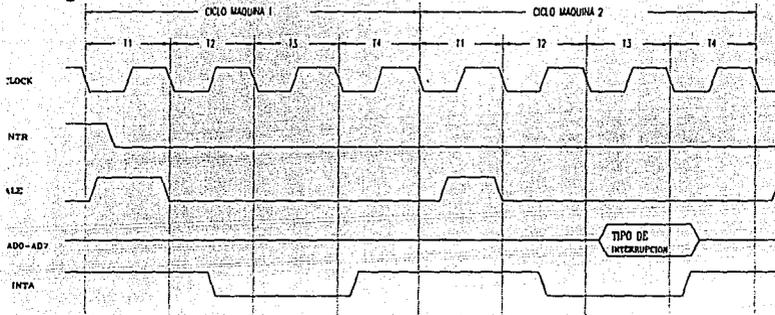


fig.3.z.- Diagrama de activación de INTA.

- **La Detección de Carga de Caracter**, se realizará una vez que se ha dado de "Alta" a una computadora y se proceda a cargar el archivo que se imprimirá. Para detectar la activación de la señal STROBE de la PC que se está sirviendo, se realizará por medio de un circuito 74373, el cual puede ser activado ("C") por cualquiera de las señales "STROBE" de las computadoras que están conectadas al sistema y a las entradas del circuito.

Las señales STROBE serán conectadas a la entrada de una compuerta "NAND" y su salida permanecerá en estado "Alto", mientras ninguna de las señales se encuentre activada. Al activarse cualquiera de las señales STROBE, se obligará al circuito a cambiar el estado de su salida ("bajo"), activando el pin "C" del 74373 y congelando los estados de los "STROBEs" de las PCs. Al ser leído el dato que contiene en el circuito, se detectará si el STROBE activado fue de la computadora que está dada de "ALTA" en el servidor. Existe la posibilidad, de que el pin "C" de éste circuito sea activado por otra señal, que tiene el objetivo de limpiar el dato ya procesado por el microprocesador, evitando que se repita la carga del mismo caracter. Esta señal se generará, al leer cualquiera de los circuitos que retienen los caracteres enviados por las PCs.

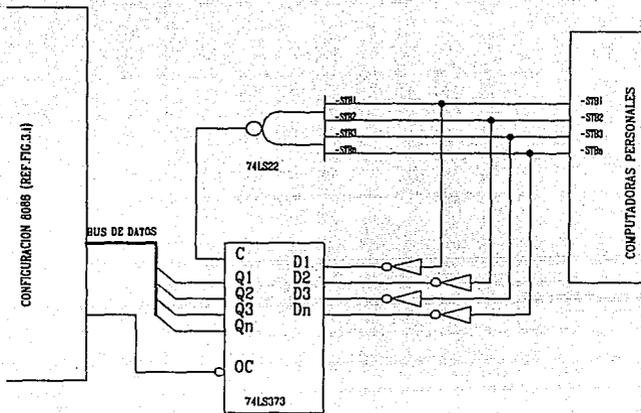


fig. 3.j.- Proceso de Carga de Caracter.

• **Lectura del Caracter**, se realiza una vez elaborado el proceso anterior, para la captura del caracter proveniente de la Computadora Personal *en servicio*. El circuito 74373, que está conectado al bus de datos del puerto de impresión de la computadora, abre una vía para la transferencia de caracteres, entre la PC y el servidor, conectando las entradas del circuito al bus de datos del puerto paralelo y conectando la señal STROBE, a un inversor que cambia el estado de activación de la señal, para ser conectada al pin "C" del 74373. Para que en el instante, en que la PC envíe un caracter válido, la activación de la señal "STROBE" retenga el caracter en el circuito, para ser leído por el sistema. El proceso de lectura del servidor, se realizará direccionando al dispositivo que activa el pin "-OC" (control de salida) del 74373. En ese momento el microprocesador tendrá en su bus de datos, el caracter proveniente de la computadora. El caracter permanecerá en el circuito hasta que se presente el siguiente.

Este proceso se muestra en el esquemático de la figura 3.k.

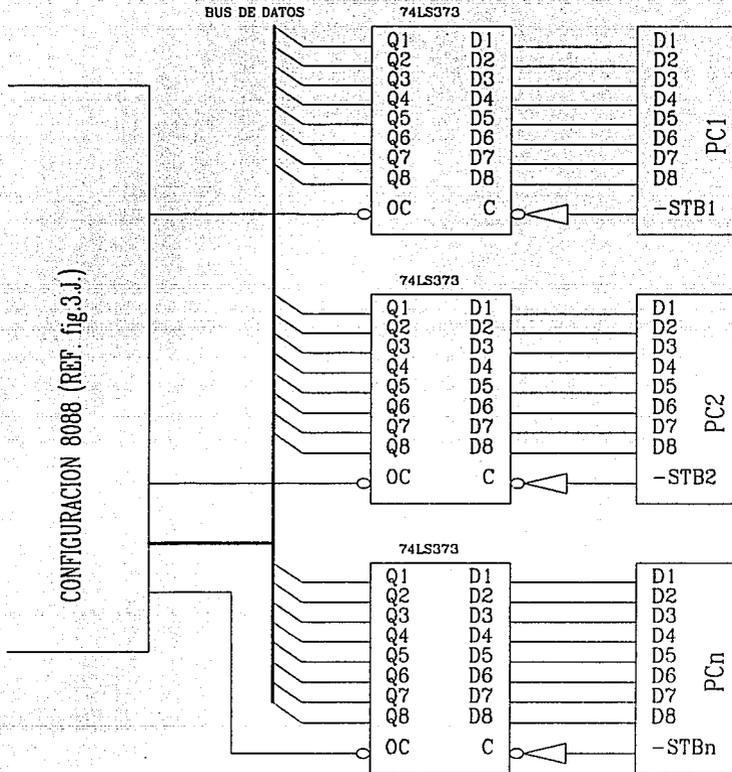


fig 3.k .- Proceso de Lectura de caracteres del Bus de las PCs.

Para la transferencia de caracteres de la computadoras al servidor, es necesario el control de las señales "ACK" y "CINTn". La señal "ACK", es un aviso para la PC para que ésta envíe un caracter y la señal "CINTn", que enmascara a la señal "STROBE" de la computadora "n", evite que las computadoras en "Lista de Espera", estén interrumpen al procesador.

Para la generación de estas señales, se utilizarán dos circuitos 74373, pero al contrario de los anteriores, los circuitos serán usados como puerto de salida. La señal que direcciona a este circuito será conectada al pin "C" y el pin "OC" se dejará permanentemente activado, conectándose a tierra (GND). De esta forma las señales siempre tendrán un estado que será determinado por el servidor.

La señal "BUSY", que evita que la computadora en servicio envíe caracteres, mientras el servidor procesa otro carácter, será generada directamente a través de las señales :

- "ACK" : generada por el sistema .
- "STROBE" : generada por las PCs .

Este proceso se muestra en el diagrama 3.m.

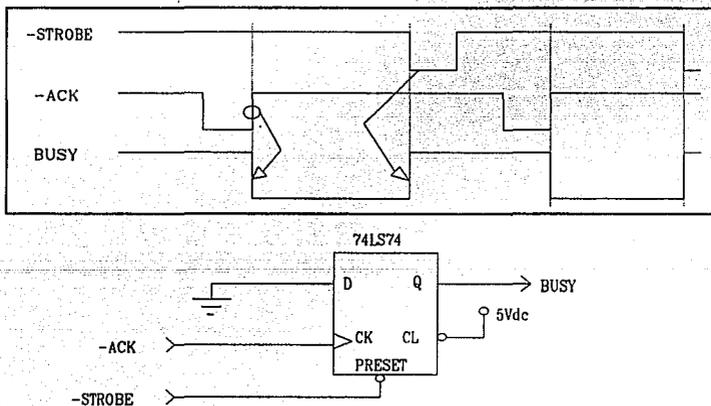


fig. 3.m.- Generación de la señal "BUSY".

### III.1.2.- TRANSFERENCIA DE CARACTERES ENTRE SERVIDOR E IMPRESORA.

- **EL 8255 (Interfase Periférica Programable).**

El 8255 de Intel, es un dispositivo de E/S, de propósito general para interfase periférica, tiene un registro de control y tres puertos de 8 bits A,B y C. Estos puertos están divididos en dos grupos de 12 bits cada uno. Los cuales pueden ser individualmente programados en un modo de operación. El registro de control y los tres registros de puertos se seleccionan por los pines A1 y A0.

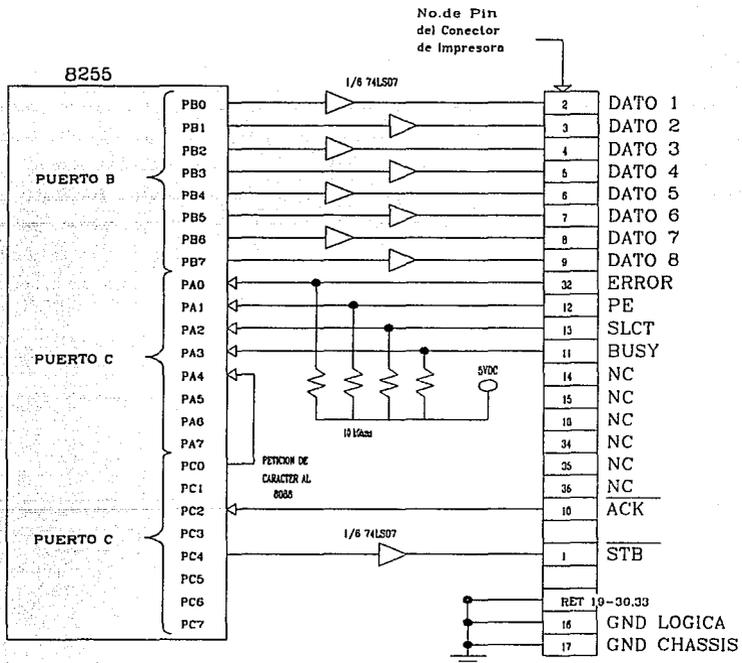


fig. 3.n .- Transferencia de Caracteres entre Servidor e Impresora.

Refiriéndose a la conexión del 8255 y la impresora, que serán parte fundamental de la transferencia de caracteres del servidor hacia la impresora, como se muestra en la figura 3.n, para esta interfase los "buffers" colector abierto "7407" se usan en las señales o líneas de datos del 8255, porque las salidas de este puerto no suficiente generación de corriente para la carga y descarga de la "capacitancia de la conexión" (cable). Las resistencias de "Pull-up", están construídas dentro de la impresora.

El puerto B es usa para la salida de datos. El bit PC0 funciona como una solicitud de caracter al microprocesador. La señal "ACK" de la impresora está conectada al "8255" en el bit PC2. La señal "STROBE" que generará el 8255 y entra a la impresora, sale del pin "PC4". Todas estas señales al igual que la habilitación de la señal "ACK" (PC2), se generan a través de un bit "set/reset" de estos pines.

Las cuatro señales que indican el estado de la impresora, se conectan al puerto A. Dicho puerto es leído en el programa, para determinar la condición de la impresora.

En la figura 3.p, se muestra la palabra de control para programar o inicializar por completo al 8255 y utilizarlo en la transferencia de caracteres (handshake). Más adelante se describirá claramente la programación del 8255. En la figura 3.q. se describe la palabra de "bit set/reset", necesaria para habilitar la señal de solicitud de dato al 8088.

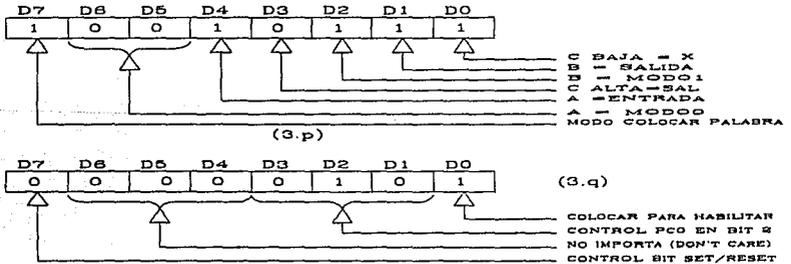
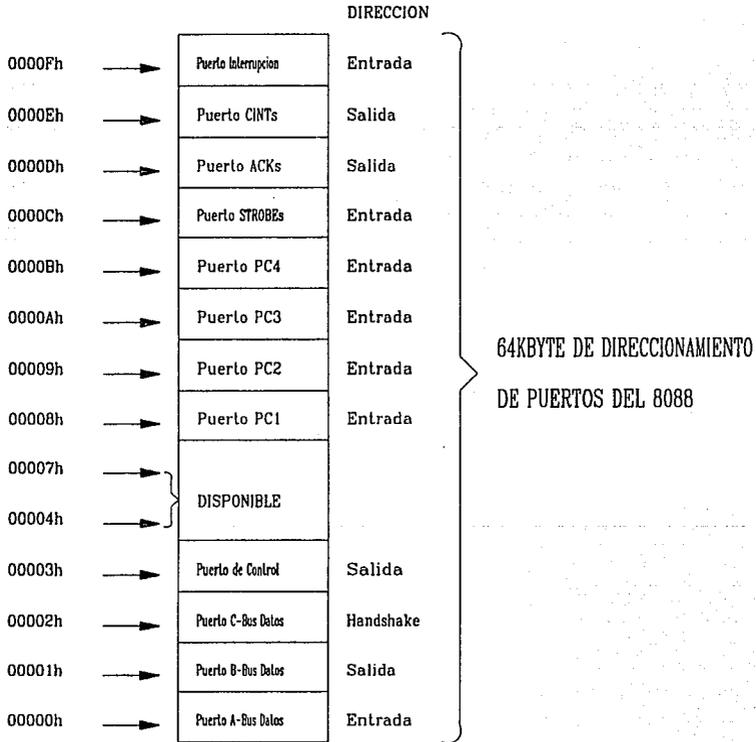


fig. 3.p,q .- Palabra de Control para 8255 (3.p) y Palabra de bit set/reset (3.q).

- **Decodificación de Puertos E/S.**

El 8088 puede direccionar más de 65,000 puertos. El microprocesador es capaz de comunicarse con el 8255, al generar y enviar varias señales de control de este circuito, como son las señales de "IOWR", "IORD" así como la señal de habilitación del circuito ("CHIP SELECT"). Un estado alto en éste último pin ocasiona que los pines de datos queden flotando.



**fig.3.r.- Mapa de Puertos.**

Para decodificar a los puertos, se selecciono el circuito 74138, que es un decodificador que entrega las salidas "activas bajas", necesarias para habilitar tanto al 8255 (-CS) como a los 74373 (-OC), que se utilizarán en la transferencia de caracteres entre PCs y sistema.

El mapa de direccionamiento de puertos se muestra en la figura 3.r.

Dentro del mapa de direccionamiento, los puertos están en las primeras direcciones. El 8255 ocupa las primeras cuatro direcciones. Esto significa que cuando cualquiera de los cuatro primeros pines de salida del 74138 estén en estado "bajo", el 8255 estará activado. Los demás puertos estarán en las direcciones siguientes, como se observa en el mapa de puertos de la figura 3.r.

Por último, el esquemático 3.s; en donde se integran todas los circuitos que conforman el *Servidor de Impresora para Computadoras Personales*.

### **III.2.- DIAGRAMA DE FLUJO DEL PROGRAMA Y DESCRIPCION DE CADA MODULO.**

Con la finalidad de hacer más sencilla la descripción del programa, se hará una división del programa., de acuerdo a las subrutinas que se manejarán en el programa. Estas son :

- Proceso de Inicialización del sistema
- Proceso de Preimpresión ( Revisión de Estado de Impresora ).
- Proceso de Petición de Caracter.
- Proceso de Almacenamiento de Caracter.
- Proceso de Impresión de Caracteres.
- Proceso de Actualización "Computadora en Servicio".
- Proceso de Interrupción.

#### **III.2.1.- PROCESO DE INICIALIZACION DEL SISTEMA.**

El proceso de inicialización del sistema, establecerá el estado "inicial" de las variables y constantes del sistema. En resumen, las condiciones con las cuales el sistema inicializará. Este proceso es muy importante, ya que de ello dependerá el buen funcionamiento del sistema. El proceso de inicialización, lo conforman :

- **Inicialización del PPI 8255.** Los puertos serán configurados enviando una palabra de control, mencionada anteriormente. La configuración de los puertos será:

Puerto "A" : Lectura del Estado de la Impresora.

Puerto "B" : La salida de Caracteres, hacia la Impresora.

Puerto "C" : Manejo de Señales. "Hanshake".

- **Estado Inicial de "STROBE".** Coloca la señal Strobe, que recibe la impresora, en estado "ALTO", para indicar que no está lista para enviar caracteres a la impresora.

- **Estado Inicial de señales "ACK"**. Coloca todas las señales ACK, que entran a las PCs en estado "ALTO", para que no envíen ningún carácter, hasta que les sea requerido.
- **Estado Inicial de " Máscaras de Interrupciones"**. Estas señales, sirven para enmascarar (deshabilitar) las señales "STROBES", para que no activen la línea de interrupción del procesador, cuando estén en proceso de *servicio*.

**III.2.2.- LA PETICION DE CARACTER A LAS PCs.** Se realiza a través de la activación de la señal "ACK" (activa "Baja"), generada por el servidor con el propósito de pedirle un carácter a la computadora en *servicio* y en general de las computadoras que no estén en Lista de Espera. La señal es colocada en estado "Alto" inmediatamente de la activación, no existiendo problemas con el tiempo de permanencia de activación. La activación de la señal ACK, dura aproximadamente 2.4us, lo cual resulta bastante sobrado del límite de 5us. El flanco de subida de la señal "ACK", activa a la señal "BUSY", permitiendo una respuesta de la PC. Todo este proceso la Petición de Caracter.

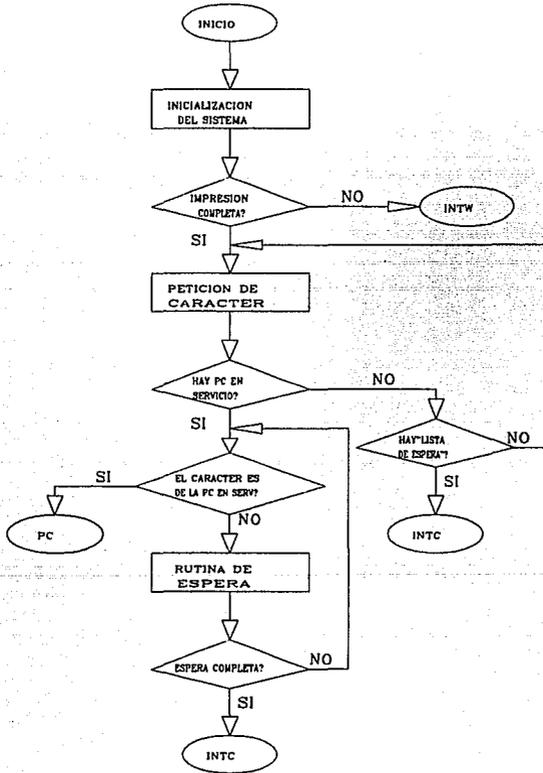
En caso de existir respuesta de la computadora en servicio, la señal "STROBE" con el flanco de bajada, coloca a la señal "BUSY" en estado "alto", evitando que la PC continúe enviando caracteres.

Una vez hecha la "petición de caracter", se revisará si la computadora en servicio envió carácter, a través del registro ( ESTI ). Este registro contiene la máscara de la PC (Strobe) que está en proceso de captura de caracteres. De ser correcto, el carácter se leerá y cargará al buffer del sistema.

En caso de haberse iniciado la operación del sistema y por lo tanto no existirá PC en servicio (ESTI=0), se estará revisando continuamente el registro de "Lista de Espera" (WAITO=0), para saber el momento en que exista una PC, esperando ser *servida*. Una vez logrado lo anterior, se ejecutará la Subrutina de actualización de "Computadora en Servicio". Este proceso será descrito más adelante.

De no existir PC, se continuará pidiendo carácter ("ACK"), esperando que exista PC en "Lista de Espera".

Cuando la PC en servicio, haya terminado de enviar caracteres, el sistema dejará pasar aproximadamente 500ms (CONTA=04), esperando una activación de la señal "STROBE". Transcurrido este tiempo, el sistema ejecutará la subrutina de actualización de "Computadora en Servicio"( INTC ), con la finalidad de dar de "Baja" a la PC que se terminó de servir y dar de Alta, si existiera PC en "Lista de Espera".



D-1.- Inicialización y Petición de Caracter.

### **III.2.3.- PROCESO DE PREIMPRESION ( Revisión de Estado de Impresora ) (Diagrama D-2).**

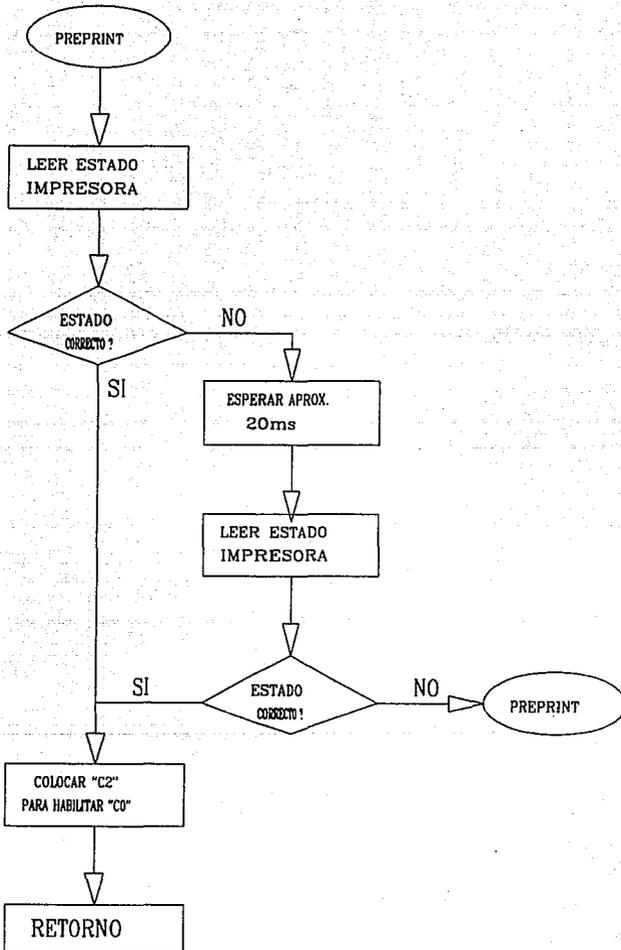
Este proceso del programa, forma parte de la inicialización del sistema, ya que solo se ejecutará una vez al momento de ser encendido o reinicializado el sistema. El proceso tiene el objetivo de conocer si la impresora está lista o en línea (On Line). Este proceso se realiza, leyendo el puerto A del 8255, en el cual están conectadas las señales "ERROR", "BUSY", "SLCT" y "PE" y el programa leerá y verificará el estado de estas señales, que fueron vistas en capítulos anteriores.

La lectura del puerto A, se realizará dos veces previendo cualquier error, durante la primera lectura. La segunda lectura, se realizará 20ms después de haberse detectado incorrecto el estado de la Impresora.

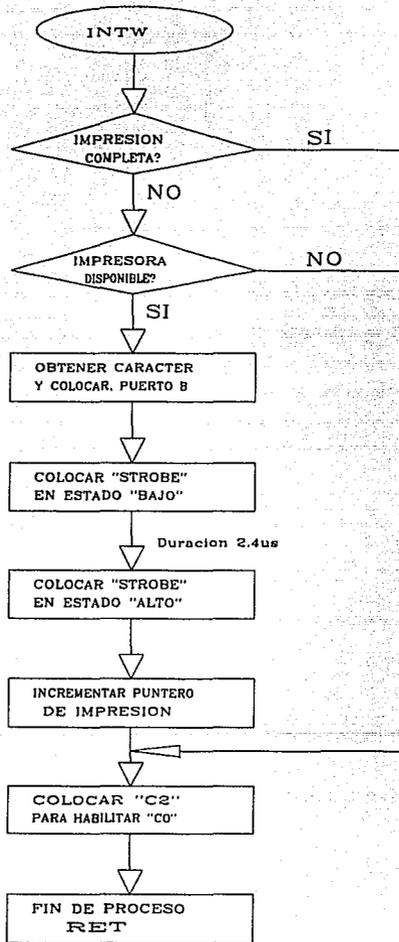
### **III.2.4 .- PROCESO DE IMPRESION DE CARACTER ( Diagrama D-4 ).**

Este proceso es muy sencillo como se puede observar en el diagrama. Al inicio, se debe verificar si la impresión se ha completado, comparando los punteros de almacenamiento e impresión. Si la impresión no ha finalizado, se verificará la disponibilidad de la impresora, que indicará si la impresora puede recibir un carácter. Este proceso se efectúa al leer el puerto A (pin 40) del 8255, el cual tendrá un estado "alto", en caso de que se haya recibido la señal "ACK" de la impresora. El estado permanecerá, hasta que se escriba algo al 8255. En caso de que el estado de PC0 no sea el correcto, se saldrá de la subrutina.

Si la impresora está lista para recibir el carácter, se procederá a enviarlo al puerto B del 8255. Una vez hecho esto, se activará la señal "STROBE" (estado "bajo"), por medio de un "bit SET/RESET" del pin PC4 del 8255, para que la impresora se dé cuenta de la presencia del carácter. Posteriormente, se levantará la señal y se incrementará la dirección del puntero de impresión. Por último se enviará un "bit SET/RESET" al pin PC2, con la finalidad de habilitar la señal ACK de la impresora.



**D-2.- Proceso de Revisión del "Estado de Impresora" o "PREIMPRESION"**



**D-4 .- Proceso de "Impresión de Caracter"**

### III.2.5 .- PROCESO DE ACTUALIZACION DE "Computadora en Servicio" ( Diagrama D-5 )

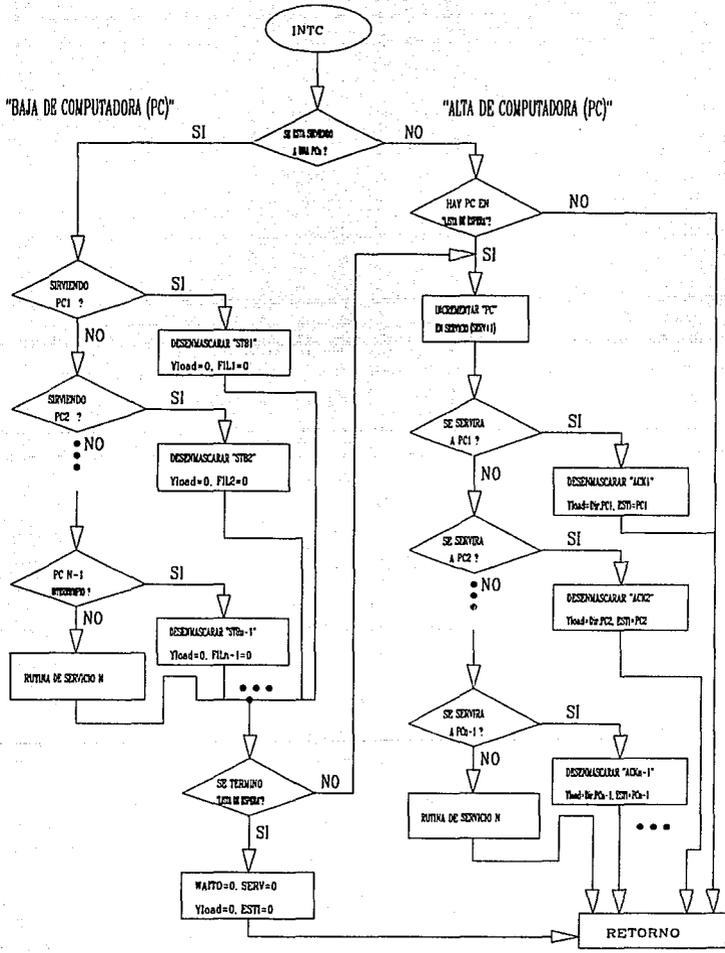
Este proceso, desarrollará una actualización de la "Lista de Espera", la cual trabajara, con base a un "Contador", que registrará el número de PCs que interrumpen al procesador.

Existen dos casos, en los cuales se accesa a este proceso. El primero sucede se iniciando operaciones y no hay PC en servicio, por lo tanto el sistema se encuentra en de que exista una PC para ser servida. La segunda forma, acontece cuando se completado la transferencia del archivo de impresión, de la Computadora que se estaba sirviendo.

El proceso inicia, cuando se revisa la existencia de alguna PC en servicio (  $SERV=0$  ); si no existe PC en servicio, se procede a revisar "Lista de Espera" (  $WAITO=0$  ); en el caso de que si existiera PC, se procedería a dar de "ALTA", incrementando el registro "SERV", que contiene el número de PC que ha servido el sistema ( $SERV+1$ ). Después se procede a localizar la PC que va a ser servida (  $FILn=SERV$  ). Una vez encontrada, se desenmascarar la señal "ACK" que fue enmascarada por la rutina de Interrupción y también se carga la dirección del puerto donde se buscan los caracteres de la PC en servicio y por último se colocará la máscara del Strobe de la PC, con la finalidad de que otras activaciones no afectar la operación del sistema.

Al inicio del proceso, se revisó la existencia de alguna PC en servicio, de existir se tendrá que dar de "BAJA". Este subproceso se realizará localizando la PC (  $SERV = FILn$  ) que se terminó de servir. Una vez encontrada, se limpiará el número asignado a la PC dentro de la Lista de Espera (  $FILn=0$  ) y además, se limpiará la dirección del puerto donde estuvo buscando los caracteres (  $Y_{load}=0$  ), y por último se quitará máscara la señal STROBE, para que pueda volver interrumpir al procesador.

Una vez dada de baja la PC, se procederá a dar de "ALTA" si existe alguna PC en "Lista de Espera" (  $WAITO=SERV$  ). En caso de que existir, se dará de "ALTA" la nueva PC. En caso contrario se saldrá del proceso.



D-5.- Proceso de Actualización de "Computadora en Servicio"

### III.2.6.- PROCESO DE INTERRUPCION. ( Diagrama D-6 ).

Considerando la cantidad de dispositivos que generan interrupción, se decidió utilizar sólo un tipo de interrupción que controlara a todas las PCs. De tal modo, que para saber cuál de todos los dispositivos había generado la interrupción, debiera ser determinado por un proceso llamado POLEO, a través de un *registro de petición de interrupción*, que contiene el estado de las señales que pudieran interrumpir al procesador.

Un ejemplo, de lo que sería el proceso de POLEO se observa en el diagrama de la figura 3.t.

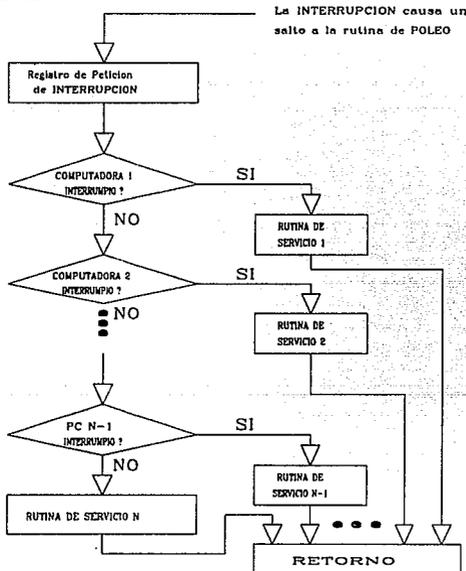
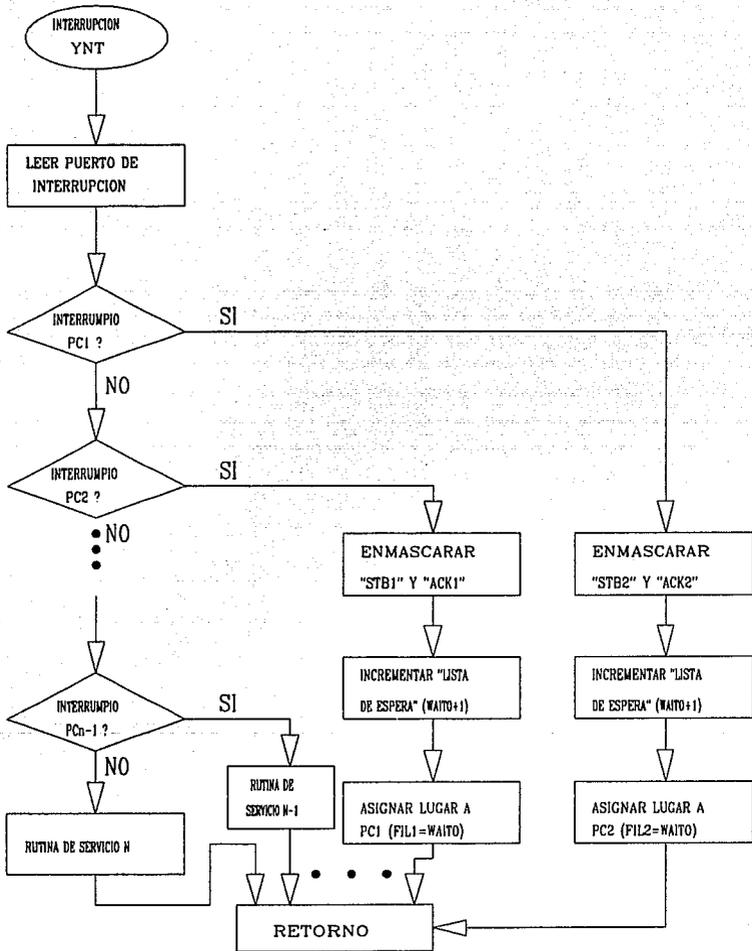


Fig. 3.t.- Proceso de poleo usando un registro de petición de interrupción.



D-6.- Proceso de "Interrupción del Microprocesador"

Este proceso se basa en el POLEO, que se puede definir como un procedimiento de identificación de los dispositivos que se encuentran en el "Registro de Interrupción", que permite identificar al generador de la interrupción al procesador. Una vez identificado el dispositivo, se salta a la rutina de servicio de ese dispositivo.

A esta rutina se accesa, cuando una PC envía el primer caracter al sistema, en respuesta a una petición de éste. Al inicio del proceso se guardan los registros que serán ocupados durante el mismo, una vez hecho esto se procederá a realizar el POLEO. En cuanto se identifica a la PC, se enmascara tanto la señal "Strobe" que llega al *registro de petición de interrupción* como la señal "ACK", con la finalidad de evitar que continúe interrumpiendo y enviando datos, sin haberse dado de "ALTA" en la subrutina de "Actualización de PC en Servicio". En este se proceso, se creará la "Lista de Espera" (WAITO+1), por medio de este registro que contiene el número de PCs que se encuentran en servicio y esperando a ser servidas. El valor de éste se asigna al registro de la PC que interrumpió (WAITO=FiLn). Aquí finaliza el proceso de interrupción.

Con el proceso de POLEO finaliza la descripción cada módulo del programa completo del sistema, cuya comprensión se apoyó con los diagramas de flujo de cada módulo.

## CAPITULO IV

### **DISEÑO DEL PROTOTIPO Y HERRAMIENTAS DEL "DISEÑO ASISTIDO POR COMPUTADORA" (CAD).**

Durante la realización de este proyecto, se utilizarán diversas herramientas de CAD, tanto de software como de hardware.

#### **¿Qué es el Diseño Asistido por Computadora?**

Es la tecnología basada en sistemas de cómputo, que auxilia en la creación, modificación, análisis y optimización de un producto.

Para la elaboración de diagramas, esquemáticos y dibujos que formarán parte esencial del proyecto, se utilizarán paqueterías de software, como son:

- **AutoCAD Versión 10** : Con la elegancia del lenguaje de sus comandos y la forma sensitiva, estructurada en que interactúan sus comandos, AutoCAD es uno de los paquetes de software para el dibujo y edición, que supera lo que se espera de él, por su facilidad y rapidez de funcionamiento. Se puede trabajar con el paquete a nivel superficial resultando muy productivo, pero si se necesita realizar tareas más complicadas, también se dispone de la potencia necesaria. La Versión 10 incorpora funciones adicionales que permiten trabajar en tres dimensiones.

A medida que el software avanza en tamaño y en sofisticación, el peligro mayor radica en la tendencia de volverse "críptico". Esto no ocurre con AutoCAD, aunque ofrece un mundo de funciones, es flexible y fácil de manejar. Se encuentran muchas formas de regresar después de introducir órdenes incorrectas o cometer errores involuntarios.

En AutoCAD, se encontrarán todas las herramientas de interfase que el usuario espera, incluyendo menús instantáneos, menús encadenados, menús laterales, una línea de órdenes y menús de tablero.

El AutoCAD será una herramienta importante para la elaboración de gráficos (esquemáticos, mapas de memoria, etc.), que ayudarán a soportar la

parte teórica del diseño del servidor, utilizando un comando (dxfout) que permite generar un archivo que es fácilmente importado por procesadores de texto.

- **HI-WIRE Versión 2.0** : Tradicionalmente, el diseño electrónico ha necesitado del uso de diagramas esquemáticos, donde los componentes son representados con símbolos gráficos y son interconectados con líneas de interconexión. Aún con los avances de los métodos de diseño, el diseño de esquemáticos es un método que impera como de los más entendibles y aceptados para especificar documentación electrónica.

La mayoría de los equipos eléctricos dependen de una "Tablilla de Circuito Impreso". Tal tablilla consiste en una delgada pieza de material aislante que tiene pistas de cobre, en una o ambas caras, en donde en una cara se montan los componentes electrónicos. Estos son los pasos para muchos diseñadores en la creación de una tarjeta de circuito impreso.

El Hiwire es una herramienta del Diseño Asistido por Computadora (CAD), para la creación de esquemáticos y diagramas de circuitos impresos. Esto permite al diseñador, concluir con más rapidez el diseño del esquemático del sistema y también permite la creación rápida de diagramas de circuito impreso. En suma, HiWIRE es un producto versátil, como otros sistemas de CAD, obteniéndose flexibilidad en el proceso de diseño.

Esta herramienta es básica, para cualquier diseño que se piense implementar físicamente, siendo parte importante para la realización de modificaciones e inclusive mantenimiento del Servidor.

#### **IV.1- IMPLEMENTACION DEL PROTOTIPO, BASANDOSE EN UNA COMPUTADORA PERSONAL.**

La herramienta principal para la creación, modificación, análisis y optimización del *Servidor de Impresora para Computadoras Personales*, es la misma Computadora Personal, en la cual se desarrollará un tarjeta de expansión, que será conectada en uno de los Canales E/S o "slots" de la PC. De esta forma, se satisfacen los objetivos que persigue el CAD.

### **¿ Qué Objetivo Persigue el Prototipo en la Computadora Personal ?**

El prototipo es una visualización útil para demostración y captura de ideas principales, de operación de un sistema. Todo es un mecanismo que permite al miembro del equipo de diseño, ver una representación literal del sistema.

Pensando en la visualización como un "ejemplo" generador de ideas, muchas veces una idea tiene sólo una descripción verbal y puede ser capaz de hacer tomar una mejor decisión de diseño. Sin embargo, la visualización puede también mostrar algo real para obtener una retroalimentación. Aún si no se puede interactuar con la visualización, el concepto del sistema implementado en la Computadora Personal puede ser demostrado con el desarrollo de tarjeta prototipo.

Un beneficio importante que se obtiene, es la velocidad con la que el diseñador puede integrar todos los resultados de la visualización (Prototipo). La interactividad del prototipo, permite al diseñador someter a prueba al sistema.

Todo esto provee una forma de probar las teorías, al poder implementarlas y modificarlas o corregirlas, tantas veces como sea necesario.

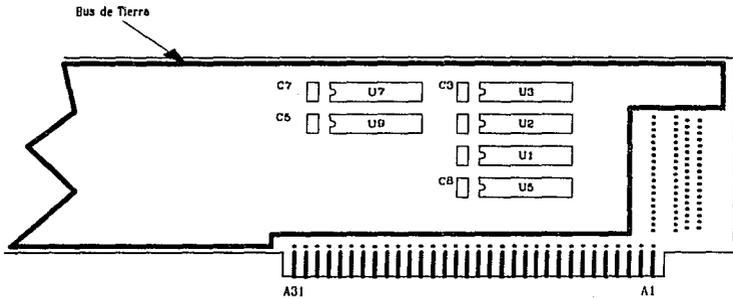
### **IV.1.1- DISEÑO DE LA TARJETA PROTOTIPO.**

La tarjeta del prototipo mide aproximadamente, 10 cm de ancho por 23.3cm de largo, soportada por una unidad de expansión o "Slot" del sistema. Todas las señales de control y voltajes requeridos, se proporcionan en una peinetas de 2 x 31 posiciones.

El espacio de direccionamiento, que proporciona la Computadora Personal para desarrollar cualquier prototipo, se encuentra entre las direcciones 300h a 31Fh, exclusivamente.

El prototipo tiene dos caras (parte frontal y parte trasera). En la cara de componentes (parte frontal) se tiene el bus de tierra de 1.27 mm de ancho. En esta cara se encuentra marcado el peine con las etiquetas que van de A1 hasta A31, cada

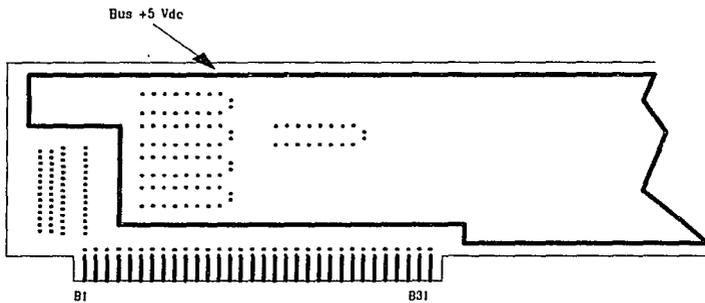
una de ellas contiene una señal de entrada o de salida. Esta cara se observa en la figura 4.a.



LADO COMPONENTES

Fig.4.a - Lado de Componentes de la tarjeta Prototipo

En la otra cara de la tarjeta, se tienen en las peinetas las etiquetas desde B1 hasta B31; en esta cara se encuentra la alimentación de +5 Vdc, en una pista de 1.27mm de ancho. Como se observa en la figura 4.b



LADO SOLDADURA

Fig. 4.b.- Lado de Soldadura de la Tarjeta Prototipo

#### IV.1.2.- CANAL DE ENTRADA/SALIDA (Slot).

El canal de entrada y salida (Slots de expansión), es una expansión del bus del procesador (8088), está demultiplexado y adicionado con interrupciones y funciones de acceso directo a memoria (DMA).

El canal de E/S, contiene un bus bidireccional de 8 bits, 20 líneas de direccionamiento, 6 niveles de interrupción, líneas de control para memoria y puertos de lectura y escritura, reloj, líneas de tiempo, además provee de +5 Vdc, -5 Vdc, +12 Vdc y -12 Vdc entre otras cosas. Todo esto incluido en un conector de 62 pines (2 x 31).

El canal de E/S, provee de suficiente potencia para manejar los cinco unidades de expansión, asumiendo baja potencia de cargas "Schottky" por "Slot"

En la figura 4.c se muestra el Canal de E/S, además de describir cada señal en las siguientes páginas.

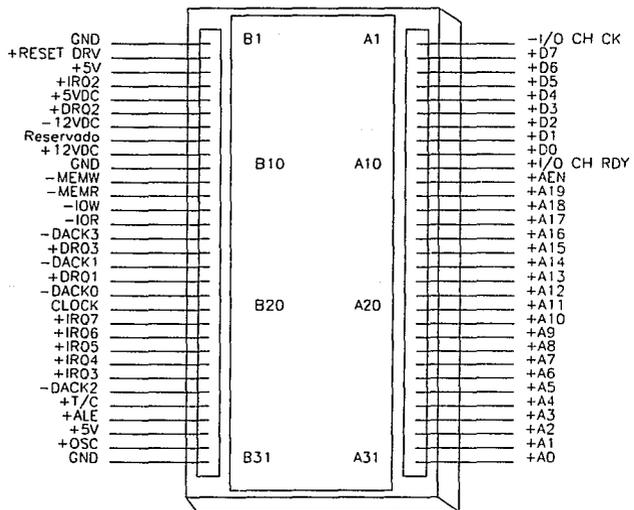


Fig. 4.c.- Canal de E/S de la PC (slot).

- **Descripción del Canal E/S.**

A continuación se describirá el Canal de E/S de una Computadora Personal. (8088). Todas las líneas son compatibles con TTL.

<b>Señal</b>	<b>E/S</b>	<b>Descripción.</b>
•OSC-----	Salida-----	Oscilador: Alta velocidad del reloj, con un período de 70 ns (14.311818 Mhz).
•CLK-----	Salida-----	Reloj del sistema: Este se dividido entre tres, del valor oscilador y tiene un período de 210 ns (4.77 MHz). El reloj tiene un ciclo del 33%.
•RESET DRV---	Salida---	Esta línea es usada para inicializar el sistema lógico, al encender ó durante un bajo voltaje. Esta señal es sincronizada por el flanco de subida del reloj y es activo "alto".
•A0-A19-----	Salida-----	Bits de direcciones del 0 al 19: Estas líneas se usan para direccionamiento de memoria y dispositivos de E/S. Las 20 líneas de direcciones, permiten acceder a 1 megabyte de memoria. La menos significativa, es A0 (LSB) y el bit más significativo es A19 (MSB). Estas líneas son generadas para el procesador o controlador de DMA. Las señales son activas "altas".
•D0-D7-----	E/S-----	Bits de datos (0 al 7): Estas líneas suministran datos al procesador, memoria y dispositivos E/S. El bit menos significativo es D0 y el más significativo es D7. Las líneas son activas "altas".
•ALE-----	Salida-----	Habilitación del "Latch" de direcciones: Esta línea es suministrada por el Controlador del Bus 8288 y se usa en la tarjeta para congelar la dirección válida para el procesador. Está disponible para el canal de E/S, como un indicador de una dirección válida (similar al AEN).
•I/O CH CK---	Entrada-----	-Checa al Canal de E/S. Esta línea suministra información con paridad (error) en memoria o dispositivos en el canal de E/S. Cuando esta señal es activa "baja", indica un error de paridad.

- **I/O CH RDY**-----Entrada-----Listo Canal E/S: Esta línea, normalmente está en estado "alto"; es puesta en estado "bajo" (No listo) por memoria o por dispositivos de E/S. Lo cual permite a los dispositivos lentos, fijar el canal de E/S con un mínimo de dificultad.
- **IRQ2-IRQ7**-----Entrada-----Petición de Interrupción de la 2 a la 7. Estas líneas se usan por el procesador, cuando un dispositivo de E/S requiere atención. Las señales tienen su prioridad, IRQ2 con la más alta prioridad y IRQ7 con la más baja. Una petición de interrupción, se genera por el flanco de subida de la línea IRQ (bajo- alto) y congelándolo en alto, hasta que sea reconocida por el procesador (Servicio de rutina de interrupción).
- **IOR**-----Salida----- -Comando de Lectura de E/S. Este comando funciona con una instrucción en línea, para leer datos de dispositivos de E/S. Puede ser manejado por el procesador o por el controlador de DMA. La señal es activa "baja".
- **IOW**-----Salida-----Comando de Escritura de E/S. Este comando funciona con una instrucción para escribir datos en el bus. Puede ser manejado por el procesador o el controlador de DMA. La señal es activa "baja".
- **MEMR**-----Salida-----Comando de Lectura de Memoria: Este comando de instrucción es para manejar datos en el bus de datos. La señal puede ser manejada por el procesador o controlador de DMA, es activa "baja".
- **MEMW**-----Salida-----Comando de Escritura de Memoria: es comando de instrucción para almacenar el dato presente en el bus de datos. Al igual que los anteriores, puede ser manejado por el procesador o el controlador de DMA. La señal es activa "baja".
- **AEN**-----Salida-----Dirección habilitada. Esta línea se usa para compuertas del procesador y otros dispositivos del canal E/S. Cuando esta señal está activa "alta", el controlador de DMA tiene el control del bus de direcciones, bus de datos, líneas de comandos de lectura (memoria y E/S), y líneas de comandos de escritura.

- DRQ1-DRQ3-----Entrada----Petición DMA de la 1 a la 3: Las líneas son peticiones asíncronas del canal, usadas para dispositivos periféricos para servicio de DMA. Están priorizadas, con DRQ3 siendo la menor y DRQ1 la mayor.

Existen otras señales para el controlador de DMA, como son "DACK0-DACK3" las cuales son señales usadas para reconocer peticiones de DMA y refresco de memoria dinámica. Así como la señal "T/C", la cual suministra pulsos. Todas las señales utilizadas para el controlador de DMA, no serán manejadas en el diseño del prototipo del servidor ya que se utilizarán los recursos de memoria de que disponga la Computadora.

Los siguientes voltajes están disponibles en la tarjeta del Canal de E/S:

+5 Vdc, localizado en dos pines del conector.

-5 Vdc, localizado en un pin del conector.

+12 Vdc, localizado en un pin del conector.

-12 Vdc, localizado en un pin del conector.

GND (Tierra), localizada en 3 pines del conector.

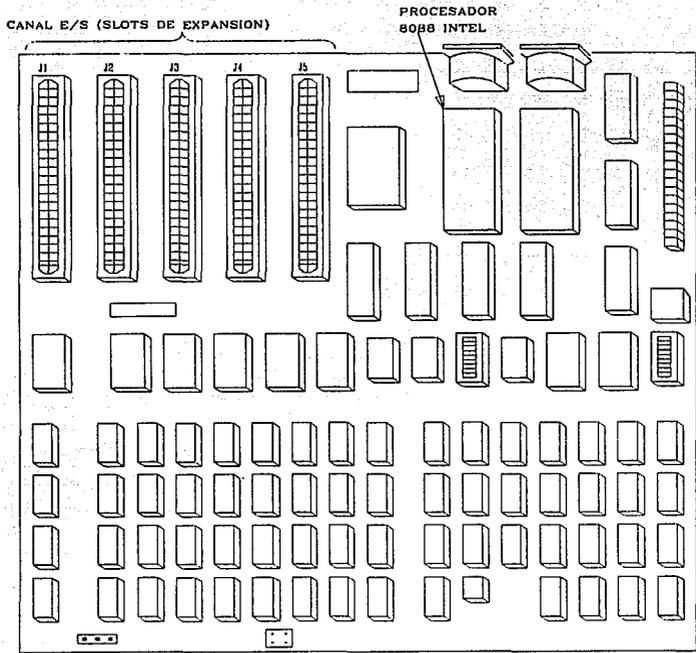


Fig. 4.d.- Localización de los "Slots de Expansión" en la "Tarjeta Madre".

#### IV.1.3.- IMPLEMENTACION DEL PROTOTIPO.

El objetivo de éste capítulo, no es reiterar lo visto en el capítulo anterior, sino describir los elementos que contribuirán al diseño del prototipo, por medio de una Computadora Personal.

Las base teóricas bajo las cuales operará el prototipo serán las mismas, que fueron explicadas en el capítulo anterior, para el diseño del *Servidor de Impresora para Computadoras Personales*. Todo ello, aprovechando las ventajas del ahorro de componentes, tiempo, etc.

Uno de los principales puntos de atención en la utilización del canal de E/S, es la necesidad de realizar una decodificación completa de puertos (64K), con el objetivo de aislar el prototipo de todos los dispositivos que integran la Computadora, para evitar cualquier error de direccionamiento, debido a que no se tiene control total del direccionamiento de la PC.

A continuación se muestra el mapa de puertos del sistema. La inclusión de memoria no es importante, ya que se reservará un espacio de memoria dentro de la misma Computadora Personal.

Dirección	Dispositivo
300h	Puerto A (8255)
301h	Puerto B (8255)
302h	Puerto C (8255)
303h	Puerto de Control (8255)
304h	Puerto de Caracteres PC1
305h	Puerto de Caracteres PC2
306h	Puerto de Estados "Strobes"
307h	Puerto de "ACKs" y "CINTRn".

### Mapa de puerto del Prototipo

El prototipo está diseñado para controlar a dos Computadoras, considerándose este número suficiente para demostrar el funcionamiento del sistema y de la "Lista de Espera", la cual ayudará al dar servicio a más de una Computadora Personal.

Se cuenta con 6 niveles de interrupción, los cuales están asignados a dispositivos y periféricos de la Computadora Personal. Es por ello que cada uno de los respectivos vectores de interrupción, ya tienen una dirección asignada. Al utilizar una de las interrupciones para el proceso descrito en el capítulo anterior (Interrupción), obliga a interceptar la dirección real, para cambiarla por la dirección de la rutina de "Servicio de Interrupción del Servidor".

---

<sup>1</sup> Señal que puede enmascara a la señal "Strobe n", para deshabilitar interrupción de PCn

Tománse en consideración que en el "Slot", no se cuenta con una señal "INTR", pero que en su lugar se contará con varias señales de interrupción, entradas del 8259. La selección de la interrupción es bastante importante, debido a que hay interrupciones que controlan a dispositivos esenciales para la computadora, es por ello que se podría afectar el funcionamiento de la computadora.

Atendiendo a lo anterior, se utilizará la interrupción del "Puerto Paralelo de Impresión", pues no afectará la operación normal de la computadora.

Para resolver el problema de la sustitución de la dirección del vector de interrupción, se utilizará interrupción de DOS.

- **Servicio de Interrupción.**

En esta parte se demuestra como se puede escribir una rutina de servicio de interrupción y se explorará el manejo de interrupciones del 8088, en lenguaje ensamblador.

El programa del servidor demuestra la forma de manejar interrupciones, que vienen vía 8259. La interrupción de la Impresora Paralela es utilizada teniendo especial cuidado, para evitar disturbios en la operación de la computadora.

El puerto paralelo de impresión en una PC IBM, PC/XT y PC/AT, es un dispositivo con un ancho de 8 bits, que puede ser usado para controlar varias impresoras compatibles.

El primer paso para controlar la interrupción del puerto paralelo de impresión, es salvar el vector de interrupción 0Fh, llamando a la función de DOS 035h para obtener la dirección del vector en el registro `es:bx`, las cuales son salvadas con las variables `IntSeg` y `IntOfs`. El siguiente paso, es llamar la función de DOS 025h con la dirección del nuevo vector de interrupción, igualmente el offset del segmento del proceso `YNT` (proceso de servicio de interrupción del servidor). Este sustituye el vector original con la dirección del nuevo ISR.

```

mov ax,350Fh      ;obtener vector de interrupción 0Fh
int 21h          ;llamada de DOS para Vector.
mov [IntSeg],es  ;Salvar valor de Segmento.
mov [IntOfs],bx  ;Salvar valor offset.
push cs         ;hacer ds=cs para dirección.
pop ds         ;el nuevo ISR.
mov dx,offset YNT ;Nueva dirección ds:dx.
mov ax,250Fh    ;Colocar vector de interrupción.
int 21h

```

#### • El Comando de Fin de Interrupción (EOI).

En el programa, se activan interrupción con sti, permitiendo al puerto paralelo de interrupción (0Fh) continuar funcionando durante todo el programa. Como la interrupción viene vía el 8259 PIC, sti(POPF) no es suficiente para permitir futuras interrupciones, esto quiere decir, que debe haber refresco de interrupción para ser procesado. Para lo cual debe ejecutarse el comando fin de interrupción (EOI), para el puerto 8259.

```

EOI EQU 020h      ;valor Fin de Interrupción.
Ctrl8259_0 EQU 020h ;dirección de puerto 8259.

mov al,EOI        ;al= fin de interrupción.
out Ctrl8259_0,al ;Colocar fin de interrupción.

```

Ambos EOI y Ctrl8259\_0, son igual al valor 020h, siendo una coincidencia.

## CAPITULO V

### PRUEBAS Y CONCLUSIONES

Durante el desarrollo del diseño del "Servidor de Impresora para PCs", fue necesario realizar pruebas, que ratificaban o rectificaban tanto el "hardware" como el programa del sistema. Para la realización de estas pruebas, fue necesario contar con herramientas fundamentales para el diseño del prototipo. Tales como :

- "Analizador de Estados Lógicos". Con este dispositivo se observaron los estados reales de las señales, que permitieron corregir cualquier error en el funcionamiento.
- Editor "Turbo C". Permitió escribir el código en ensamblador del servidor.
- Turbo Ensamblador "TASM" de Borland. Versión 1. Sirvió para compilar el programa.
- Turbo Debugger. Con este paquete, fue posible correr el programa paso a paso; observando el estado de los registros, banderas, segmentos de memoria del microprocesador; con el fin de observar la ejecución del programa y corregirlo, en caso necesario.

#### V.1.- PRUEBAS QUE ANTECEDIERON AL DISEÑO.

Una de las primeras pruebas que se realizaron, fue la verificación del estado de las señales, que indican a la PC el estado en que se encuentra la impresora: Busy, Error, PE , Slct Out. En atención a la importancia que representan las señales, para el funcionamiento correcto del sistema. El resultado se muestra en el diagrama 5.a, que confirma la teoría examinada en capítulos anteriores.

P.END	0	_____
SLO	1	_____
AFEED	1	_____
ERROR	1	_____
SLCT IN	0	_____

Fig.5.a.- Señales que indican el Estado normal de la Impresora.

Otra de las pruebas, fue la verificación de la transferencia de caracteres entre la PC y la impresora, con el fin de detectar el computamiento real de las señales en la "Transferencia de Caracteres en Paralelo", entre una Computadora Personal y una impresora (Panasonic modelo P1624). Para ello se observaron las señales "ACK", "Busy" y "Strobe", durante la transferencia de caracteres, dando por resultado el diagrama 5.b, en el cual se observa que la señal ACK, permanece activada al mismo tiempo que la señal BUSY. Además pueden suceder varias activaciones de la señal ACK, antes de que haya respuesta de la computadora (Strobe).

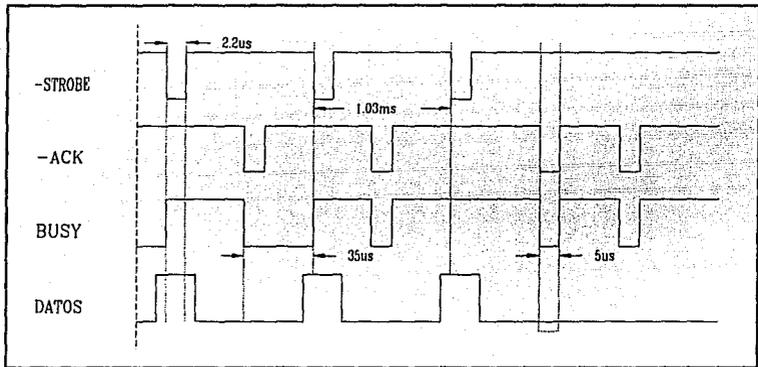


fig.5.b.- Transferencia entre la Computadora e Impresora.

## V.2 - PRUEBAS DEL PROTOTIPO DE "Servidor de Impresora para PCs".

Una vez diseñado el prototipo, se procede a probar la decodificación del mapa de puertos, realizando pruebas de direccionamiento. Esta prueba es fundamental debido a que la computadora está constantemente direccionando, lo cual puede interferir en nuestro sistema, si no está decodificado correctamente. La prueba, se realiza con ayuda del comando "debug", para ejecutar instrucciones específicas. El procedimiento solucionó los primeros errores de decodificación. Ejemplo:

### - IN 300 h

Una vez que se tiene la seguridad de la decodificación, se procede a dividir las pruebas, con la finalidad de probar cada sección integrante del servidor, para que sea

más sencilla y agil la detección y corrección de errores, que pudieran llegarse a cometer.

Una vez diseñado el hardware y software, se realizaron pruebas sobre ambas partes, en forma conjunta.

La primera sección que se pone a prueba, es el Proceso de Impresión. En éste intervienen los subprocesos de verificación del Estado de la Impresora; inicialización del 8255 y el subproceso de "transferencia de caracteres" del servidor a la impresora. Por lo tanto, los problemas que se pudieran haber presentado, serian: la errónea programación del 8255 que enmarcaría, la lectura del estado de la impresora; la generación de la señal "STROBE" y la salida del caracter hacia la impresora. El único problema que se presentó, fue la necesidad de resistencias de "Pull-op" de las señales que indican "Estado de la Impresora", que salen de la misma. Una de las pruebas importantes, fue la verificación del mensaje de "ERROR IMPRESORA", cuando la impresora se encontraba apagada o fuera de línea.

La siguiente prueba, fue el proceso de "Captura de Caracteres" de una PC. Donde los primeros problemas, se presentaron debido a que inicialmente se había planteado la idea, de que el servidor generara directamente la señal BUSY, lo cual fue un error. Pudiéndose comprobar, que una vez enviado el primer carácter, la computadora unicamente requiere para enviar los siguientes caracteres, que la señal BUSY permanezca activada en estado "bajo", lo cual provoca pérdida de caracteres.

En conclusión, la activación y desactivación de la señal ACK, marca el punto de la habilitación de la señal BUSY, la cual permite que se envíe el o los caracteres solicitados. Tomando en cuenta lo anterior, se procedio a implementar un arreglo que evite la pérdida de caracteres, haciendo que la señal BUSY dependa de la señal ACK, generada por el servidor y la señal STROBE, generada a su vez por la Computadora Personal, todo a través del uso de un flip-flop como se describió en el capítulo III. Aunque la solución fue sencilla, no fue tan rápida, pues se tuvieron que probar otras opciones. Cabe mencionar que estas opciones fueron tomadas en cuenta, a partir de la investigación que se realizó en manuales y libros, que tratan el tema de "transferencia de datos en paralelo", además de lo observado en pruebas directas entre una computadora y una impresora.

La primera opción fue a través de la modificación del software, por medio del cual se activaba por espacios pequeños de tiempo, la señal "ACK" y "Busy", como lo muestra la figura 5.d, Lo cual disminuyó en gran cantidad la pérdida de caracteres, pero no la evitó.

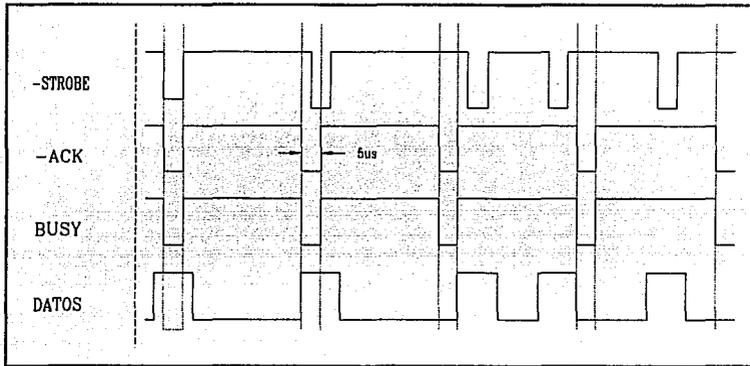


fig.5.d.- Diagrama de la primera Opción de Transferencia de Caracteres.

La segunda y tercera opción, fue a través de la modificación del "hardware", con las cuales se obtuvieron mejores resultados, que con la anterior opción. La diferencia entre la segunda y la tercera solución radica, en que en la segunda opción la señal "Busy" es activada con el flanco de bajada de la señal "ACK"(activada por el servidor) y permaneciendo activada hasta que el flanco de bajada de la señal "Strobe", que desactivará la señal BUSY, como se observa en la figura 5.e. Esta opción probó ser buena, ya que la pérdida de caracteres fue nula.

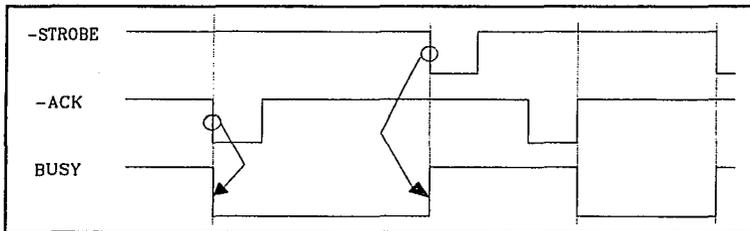


fig.5.e.- Segunda Opción de Transferencia de Caracteres.

En la tercera opción, se planteaba que la señal BUSY se activara ahora no con el flanco de bajada de la señal ACK, que marcaba el punto de activación de dicha señal. Sino con el flanco de súbida, el cual marca el punto de desactivación de la señal ACK. Esta opción tuvo los mismos resultados que la anterior, pero en cuando a tiempo de respuesta a la petición de caracter, resultó más rápida que la segunda opción.

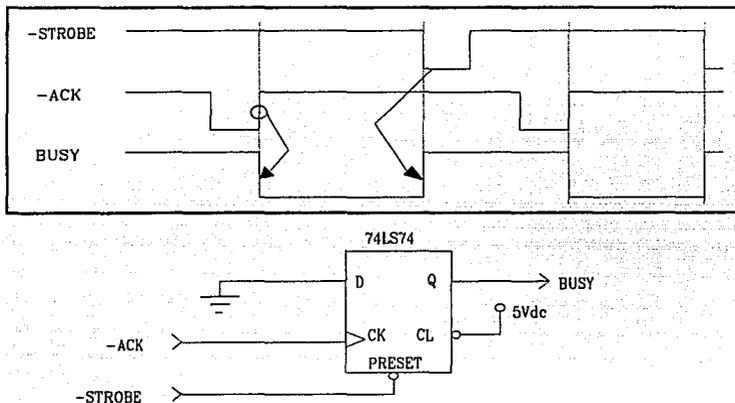


fig.5.f.- Tercera Opción de Transferencia de Caracteres.

Una vez que funcionó, la tranferencia de caracteres de la PC al servidor, se procedió a unir los dos procesos anteriores (Impresión y Captura de caracteres). Esta unión hizo necesaria la realización de modificaciones en el programa, con el objetivo de ejecutar, el proceso completo de captura e impresión de caracteres. Lo cual es retivamente sencillo, una vez asegurando el funcionamiento correcto de los dos procesos, tanto del hardware como software. Dando por resultado, la transferencia de caracteres de la *PC-Servidor-Impresora*.

Funcionando el proceso anterior, faltaría realizar el proceso de *Interrupción al Miprocesador*, para la creación de la "Lista de Espera". Para probarlo, se construyó una tarjeta prototipo, implementándose con un circuito "flip-flop", el cual se observa en la figura 5.i. Conjuntamente se elaboró un programa de prueba, el cual en su proceso normal colocaba asteriscos en la pantalla de la Computadora y al momento de

presentarse la interrupción por "hardware" (oprimimiento botón PRESET), el proceso de servicio a la interrupción, ponía en el monitor una letra "A", con la cual se aseguraba el funcionamiento de la *Rutina Servicio a Interrupción*. Una vez probado, se incorporaba el proceso de "Servicio de Interrupción" al programa del Servidor de Impresora para Computadoras Personales, realizándose pequeñas modificaciones al proceso de servicio, en lo que refiere a localización de la PC que interrumpe. Lo más importante del proceso, es la sustitución de la dirección de la rutina de "Servicio de Interrupción", en el vector que se está utilizando.

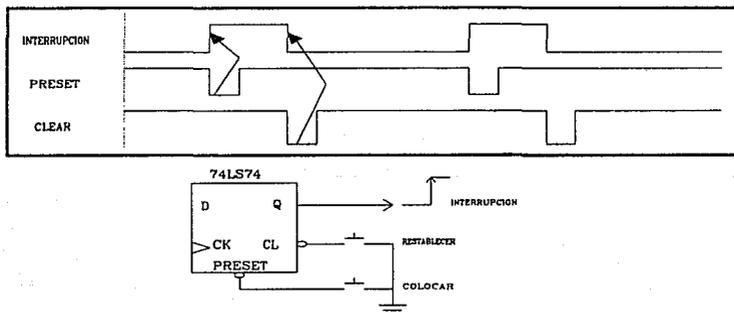


fig.5.i.- Proceso de Interrupción al Microprocesador.

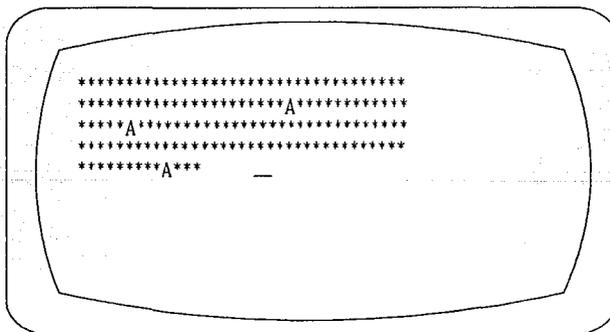


fig.5.h.- Monitor de la Computadora, durante la Prueba de Interrupción.

La siguiente prueba que se realizó, fue la verificación del tiempo de permanencia de la interrupción de la PC, para ser atendida por la rutina de "Servicio de Interrupción". Pudiendo comprobar, que el tiempo que permanecía esta señal es el suficiente, debido también a que la señal es activada con el flanco de subida de la señal de interrupción. Conjuntamente se realizaron las pruebas del proceso, que da de "alta" y "baja" a la PC que está en "Lista de Espera" en el sistema.

Después de varias pruebas, con una y con dos PCs, se finalizó la parte de experimental del hardware del "Servidor de Impresora para Computadoras Personales" y casi en su totalidad el software.

### **V.3.- CONCLUSIONES**

La idea de diseñar un "Servidor de Impresora para PC", surgió de la necesidad que se tiene de este tipo de equipos, ya que es casi imposible tener un número igual de computadoras e impresoras en una determinada empresa. Estos equipos funcionan en base a multiplexión, esto es: de varias entradas (PCs) del equipo, se selecciona una ruta de salida (Impresora). Existen dos tipos de equipos más comunes: los que realizan este proceso en forma manual, por medio de un "Switch" de varios tiros (PCs) y un polo (Impresora) y los que realizan el proceso en forma automática. Sin embargo ninguno cuenta con una lista de espera ("Cuenta de Espera"), como la que ofrece el sistema propuesto, que además de tendrá una capacidad máxima, de servir hasta 8 PCs a un precio relativamente bajo y desde luego contar con ventaja con la disponibilidad de refacciones en el país.

**NOTA:** La comparación anterior se realiza en base a equipos que se fabrican o bien son más vendidos en México.

A continuación se muestra la tabla de características del multiplexor automático y que se fabricado por la compañía SAHUARO ELECTRONICA DE MEXICO S.A DE C.V.

MODELO CONFIG.	AUTO-001	AUTO-002	AUTO-003
PC/PS	2	3	4
IMPRESORA	1	1	1
PROTOCOLO	PARALELO	PARALELO	PARALELO
SOFTWARE	Strobe Polling	Strobe Polling	Strobe Polling
DETECCION	Spooler-Espera	Spooler-Espera	Spooler-Espera

fig.5.j.- Tabla de Características del Multiplexor "AUTOMUX".

Una de las conclusiones que finaliza el diseño del software, es la diferencia entre una impresora de matriz de puntos y una lasser, ya que se pudo comprobar que la capacidad de impresión (caracter por segunda), también determina el tiempo que hay entre envío de un caracter y otro, de parte del servidor.

Aunque el diseño se basa en una impresora de matriz de punto, las pruebas se realizaron también con una impresora laser, modificando unicamente el tiempo que hay entre el envío de un caracter y otro a la impresora.

Un detalle que pudo haber quedado sin explicar, fue la necesidad o no de una gran capacidad de memoria de parte del sistema, a lo cual no se le dio mucha importancia, debido a que en el mercado existe más necesidad de un servidor o bien de un multiplexor para impresoras laser, por su gran calidad y velocidad de impresión y también lo alto de su precio. La gran capacidad de memoria lograda por éste tipo de impresoras (16 Megabytes), han hecho tomar esta decisión.

Uno de los principales conocimientos que se adquirieron, fue el haber utilizado una Computadora Personal para auxiliarse en la creación, modificación, análisis y optimización del diseño, al haber logrado controlar la transferencia de caracteres de varias PCs a la Impresora. Lo cual abrió un gran panorama para la aplicación en diseños de sistemas que dependan directamente de una Computadora Personal, como por ejemplo: Un sistema de Seguridad industrial, que cuente con control de "Accesores

Digitales", de sensores de todo tipo (magnéticos, infrarrojos, de calor etc.), inclusive contar con el control de cámaras de circuito cerrado. Esto es tan sólo un ejemplo de la aplicación de una Computadora Personal, al ser vista también, como parte de un diseño, además de ser una herramienta para el desarrollo de "software" y de "hardware".

## APENDICE A

### GUIA DE OPERACION DEL SISTEMA

El servidor está diseñado de forma tal, que el operador no intervenga directamente en la operación del sistema. Quiere decir que, una vez conectada la o las computadoras, el sistema debe hacerse cargo totalmente del monitoreo y servicio de cada una de ellas.

El único requisito del servidor, es que la impresora se encuentre conectada y encendida antes de que el sistema (Servidor) este encendido. Ello en atención a que el servidor, dentro del proceso de inicialización del sistema, verifica previamente si el estado de la impresora es el correcto. En caso de que no sea así, podrá deberse a que la impresora carezca de papel, o que no se encuentre en línea (ON LINE) y por supuesto que se encuentre apagada. En el caso en que se haya incurrido en cualquiera de dichos errores, tendrán que corregirse para después reinicializar al "servidor".

Superada la etapa de inicialización, el servidor seguira monitoreando a la impresora mediante la señal "ACK", que permanecerá en estado "alto" si se hubiera terminado el papel de la impresora o ésta estuviera fuera de línea. En éste caso el sistema no enviará caracter a la impresora hasta que la señal "ACK", se normalice.

Cumplido lo anterior, el sistema será responsable de la transferencia de caracteres, de las computadoras que requieran servicio al *Servidor de Impresora para Computadoras Personales*.

## **APENDICE B**

### **LISTADO DEL PROGRAMA.**

```

1
2
3
4
5
6
7 0000
8 0000
9
10 =
11 =
12 =
13 =
14 =
15 =
16 =
17 =
18
19 =
20 =
21 =
22 =
23 =
24
25
26 0100
27 0000 00
28 0001 00
29 0002 00
30 0003 00
31 0004 00
32 0005 00
33 0006 0000
34 0008 00
35 0009 00
36 000A 00
37 0008 45 52 52 4F 52 20 45 +
38 44 4F 24
39 0015 45 52 52 4F 52 20 49 +
40 4E 54 43 24
41 0020 45 52 52 4F 52 20 49 +
42 4D 50 52 45 53 4F 52 +
43 41 24
44 0030 00
45
46 0031 ????
47 0033 ????
48
49 0035
50 0000
51 0000 8B 0000s
52 0003 8E D8
;*****
;***** UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO *****
;***** FACULTAD DE INGENIERIA *****
;*****
IDEAL
DOSSEG
MODEL large
STACK 256
;----- IGUALDADES
Y0 EQU 300H ;Puerto A 8255
Y1 EQU 301H ;Puerto B 8255
Y2 EQU 302H ;Puerto C 8255
Y3 EQU 303H ;Palabra de Control 8255
Y4 EQU 304H ;PC1 datos
Y5 EQU 305H ;PC2 datos
Y6 EQU 306H ;"ED0" (Estado)
Y7 EQU 307H ;"ACK"s
;CONTANTES DE RUTINA INTERRUPCION EN PROTOTIPO (8259)
Ctr(8259_0 EQU 020H ;puerto del 8259.
Ctr(8259_1 EQU 021H ;puerto del 8259.
EnableIRQ EQU 07FH ;Colocar mascara a IRQ.
DisableIRQ EQU 080H ;Quitar Mascara 8259.
EOI EQU 020H ;Valor de fin de interrupción.
DATASEG
BUACK DB 0 ;Mascara de Interrupción y ACKs
WAITO DB 0 ;Cuenta de Espera
SERV DB 0 ;Cuenta de PC en servicio
EDO DB 0 ;Estado de interrupciones
FIL1 DB 0 ;No. en lista de espera pc1
FIL2 DB 0 ;No. en lista de espera pc2
Yload DW 0 ;Direccion de PC en servicio
CONTA DB 0 ;Contador Espera A.
CONTB DB 0 ;Contador Espera B.
ESTI DB 0 ;Mascara de localidad en serv.
err0 DB 'ERROR EDOs'
err1 DB 'ERROR INTCS'
err2 DB 'ERROR IMPRESORAS'
CARACT DB 0 ;Caracteres impresos
;Mensajes de la interrupción de las PCs
IntSeg DW ? ;Salvar el vector original.
IntOfs DW ? ;Rutina de serv. de Interrupción.
CODESEG
START:
mov ax,@data ;Inic. ds para direcciones
mov ds,ax ;data segmet

```

```
53 ;*****  
54 ;*****PROCESO DE INICIALIZACION DE*****  
55 ;*****TODO EL SISTEMA*****  
56 ;*****  
57  
58 ;INICALIZACION 8255  
59 ;Puerto A:modo 0 in/p. Puerto B:modo 1 o/p.  
60 0005 BA 0303 mov dx,Y3 ;direccion de control  
61 0008 B0 94 mov al,10010100B ;Palabra de control  
62 000A EE out dx,al ;Enviar palabra de control  
63 ;Enviar STROBE en estado C4 ="ALTO",8255  
64 000B B0 09 mov al,00001001b ;bit set  
65 000D EE out dx,al ;enviar bit set/reset  
66  
67 000E BA 0307 ;Enviar ACK's "ALTO" ;direccion ack  
68 0011 B0 0F mov dx,Y7 ;direccion ack  
69 0013 0A 06 0000r or al,[BUACK] ;ack=1  
70 0017 EE out dx,al  
71 ;CHECAR ESTADO DE IMPRESORA  
72 0018 9A 00000097sr CALL FAR PREPRINT  
73 ;Activar latch status, colocar C6 "BAJO"  
74 001D BA 0303 mov dx,Y3 ;direccion de control  
75 0020 B0 0C mov al,00001100b ;bit reset  
76 0022 EE out dx,al ;enviar.  
77 ;INICIALIZAR 8259 DE PC  
78 0023 9A 000000C3sr CALL FAR INI8259  
79  
80 ;*****  
81 ;*****PETICION DE DATOS DEL SISTEMA A LAS PCs*****  
82 ;*****  
83 0028 POL_A:  
84 0028 BA 0307 mov dx,Y7 ;Cargar dirección acks  
85 002B B0 00 mov al,0 ;Colocar ack="Bajo"  
86 002D 0A 06 0000r or al,[BUACK]  
87 0031 EE out dx,al ;enviar estado a puerto.  
88 0032 B0 0F mov al,00001111b ;Colocar ack="Alto"  
89 0034 0A 06 0000r or al,[BUACK]  
90 0038 EE out dx,al ;enviar estado a puerto.  
91 ;*****  
92 ;*****POCESO DE REVISION DE LISTA DE ESPERA (LE)*****  
93 ;*****  
94 0039 80 3E 000Ar 00 cmp [ESTI],0 ;Existe PC en Servicio.  
95 003E 75 13 jne CICLO_A ;Si. Buscar caracter.  
96 0040 80 3E 0001r 00 cmp [WAITO],0 ;No.Buscar en LE.  
97 0045 74 05 je POL_B ;No existe LE. ESPERAR.  
98 0047 9A 00000118sr call far INTC ;Si existe LE, entonces buscar en LE.  
99 004C B9 16EA POL_B: mov cx,16EAH ;Rutina de ESPERA al no existir LE.  
100 004F E2 FE Time_A: loop Time_A  
101 0051 EB D5 jmp POL_A  
102 ;*****  
103 ;*****LECTURA DE ESTADO DE STROBE's DE PCs*****  
104 ;*****  
105 ;Leer estado de PC's  
106 0053 CICLO_A:  
107 0053 BA 0306 mov dx,Y6 ;cargar direccion de STROBES  
108 0056 EC in al,dx ;leer dato.  
109 0057 22 06 000Ar and al,[ESTI] ;toma en cuenta PC en servicio.  
110 0058 38 06 000Ar cmp [ESTI],al ;El dato es de la PC en servicio?.
```

```
111 005F 75 11 jne RELOJ_B ;No. Entoces esperar
112 0061 C6 06 0008r 00 mov [CONTA],0 ;Si. Limpiar Contadores de espera.
113 0066 C6 06 0009r 00 mov [CONTB],0
114 0068 9A 00000089sr call far PC ;Llamar a rutina de lectura de caracter.
115 0070 EB B6 jmp POL_A ;Pedir un nuevo caracter.
116
117 0072 RELOJ_B: ;Proceso espera de caracter.
118 0072 B9 00EA mov cx,00EAH
119 0075 E2 FE TOLE_5: loop TOLE_5
120 0077 FE 06 0009r inc [CONTB] ;Incrementar espera de caracter STB.
121 0078 80 3E 0009r 04 cmp [CONTB],04h ;Espera Completa?
122 0080 75 D1 jne CICLO_A ;No. Leer de nuevo dato de STB.
123 0082 9A 00000118sr CALL FAR INTC ;Si. Entonces ir a proceso de ALTA y BAJA
124 0087 EB 9F jmp POL_A ;Retorno de rutina. Peticion de caracter.
125
```

```
;*****
;*****PROCESO DE LECTURA DE CARACTER*****
;*****
```

```
130 0089 PROC PC
131 Load:
132 0089 8B 16 0006r mov dx,[Yload] ;direccion de PC en servicio.
133 0080 EC in al,dx ;Leer caracter.
134 008E A2 0030r mov [CARACT],al
135 0091 9A 000000E8sr CALL FAR INTW ;Imprimir caracter.
136 0096 CB ret
137 0097 ENDP PC
```

```
;*****
;***RUTINA DE INICIALIZACION DE IMPRESORA PARA PODER***
;***IMPRIMIR. ESTADO (A3=BUSY=0,A2=SLCT=1,A1=PE=0,*****
;*****A0=ERROR=1*****
;*****
```

```
145 0097 PROC PREPRINT
146 0097 BA 0300 mov dx,Y0 ;Puerto A
147 009A EC in al,dx ;obtener estado impresora
148 0098 24 0F and al,0fh ;4 bits BMS no son usados
149 009D 3C 05 cmp al,00000101b ;dato correcto?
150 009F 74 0F je INTU ;Si.habilitar interrupcion.
151
```

```
;Rutina de reintento de lectura de estado de impresora.
152 mov cx,16EAH ;cargar contador con 20ms
153 00A1 B9 16EA TOLE: loop TOLE
154 00A4 E2 FE in al,dx
155 00A6 EC and al,0fh
156 00A7 24 0F and al,0fh
157 00A9 3C 05 cmp al,00000101b
158 00AB 74 03 je INTU
159 00AD EB 08 90 jmp EIMP ;ERROR EN IMPRESORA.
160
```

```
;INTU es la peticion de impresion. Esto es colocando
;C2 para obtener una respuesta de la impresora C0
161 INTU:
```

```
162 mov dx,Y3 ;direccion palabra de Control
163 mov al,00000101b ;colocando C2
164 00B0 BA 0303 out dx,al ;enviar a puerto
165 00B3 80 05 ret
166 00B5 EE
167 00B6 CB
168 00B7 ENDP PREPRINT
```

```
169
170 ;*****
171 ;Proceso de ERROR de IMPRESORA*
172 ;*****
173 00B7 B4 09 EIMP: mov ah,09h
174 00B9 BA 0020r mov dx,offset err2
175 00BC CD 21 int 21h
176
177 00BE B8 4C00 mov ax,04c00h
178 00C1 CD 21 int 21h
179 ;*****
180 ;*****PROCESO DE INICIALIZACION DEL 8259*****
181 ;*****DE LA PC.*****
182 ;*****
183
184 00C3 PROC INI8259
185
186 00C3 1E push ds ;salvar registros.
187 00C4 06 push es
188 00C5 B8 350F mov ax,350Fh ;Obtener vector de interrupción 0Fh.
189 00C8 CD 21 int 21h ;llamada de DOS para Vector.
190 00CA 8C 06 0031r mov [IntSeg],es ;salvar valor del segmento.
191 00CE 89 1E 0033r mov [IntOfs],bx ;salvar valor del offset.
192 00D2 0E push cs ;dirección YNT.
193 00D3 1F pop ds ;con ds:dx
194 00D4 BA 01CEr mov dx,offset YNT ;Nueva dirección en ds:dx .
195 00D7 B8 250F mov ax,250Fh ;Colocar vector de interrupción.
196 00DA CD 21 int 21h ;Colocar nueva dirección de interrupción.
197 00DC 07 pop es ;Restaurar registros.
198 00DD 1F pop ds
199
200 ;----- habilitar interrupción 8259 -----
201 00DE 9C pushf ;almacenar banderas
202 00DF FA cli
203 00E0 E4 21 in al,Ctrl8259_1 ;Leer registro de mascaras.
204 00E2 24 7F and al,EnableIRQ ;limpiar bit de mascara.
205 00E4 E6 21 out Ctrl8259_1,al ;Escribir nueva mascara al 8259.
206 00E6 9D popf
207 00E7 CB ret
208 00E8 ENDP INI8259
209
210
211 ;*****
212 ;*****PROCESO DE IMPRESION DE CARACTERES*****
213 ;*****
214
215 00E8 PROC INTW
216 REIN_C:
217 00EB BA 0300 mov dx,Y0 ;puerto A 8255
218 00EB EC in al,dx ;leer puerto
219 00EC 24 10 and al,10h ;Tomar en cuenta PA4
220 00EE 3C 10 cmp al,10h ;PA4 = "1"?
221 00F0 75 1F jne REIN_B
222 00F2 A0 0030r mov al,[CARACT] ;Si, Cargar caracter.
223 00F5 BA 0301 mov dx,Y1 ;Enviar puerto.
224 00F8 EE out dx,al ;enviar caracter a impresión
225 00F9 BA 0303 mov dx,Y3 ;dir. palabra de control
226 00FC 80 08 mov al,00001000b ;Colocar strobe en estado "bajo"
```

```

227 00FE EE out dx,al
228 00FF 80 09 mov al,00001001b ;Colocar strobe en estado"alto"
229 0101 EE out dx,al
230 ;Tiempo de retardo de envio entre caracter y caracter.
231 0102 B9 0EEA mov cx,0EEAH ;cargar contador de retardo
232 0105 E2 FE TOLE_i: loop TOLE_i
233
234 ;*****
235 ;RESETEAR PUERTO DE STROBE Y COLOCAR C2*****
236 ;*****
237 ;Activar Reset de latch status, colocar C6 "ALTO"
238 0107 80 0D mov al,00001101b ;bit set C6
239 0109 EE out dx,al ;enviar.
240 ;Desactivar Reset latch status, colocar C6 "BAJO"
241 010A 80 0C mov al,00001100b ;bit reset C6
242 010C EE out dx,al ;enviar.
243 ;Activando PC2 para activación de petición de caracter de la impresora.
244 010D 80 05 mov al,00000101b ;colocando c2
245 010F EE out dx,al ;enviar a puerto
246 0110 CB ret
247 0111 ENDP INTW
248 0111 REIN_B:
249 0111 B9 16EA mov cx,16EAH ;cargar contador con 20ms
250 0114 E2 FE TOLE_a: loop TOLE_a
251 0116 EB D0 jmp REIN_C
252 ;*****
253 ;**PROCESO DE "BAJA" DE COMPUTADORA QUE SE TERMINO DE****
254 ;****SERVIR Y "ALTA" DE LA PC QUE SE ENCUENTRA EN *****
255 ;***** LISTA DE ESPERA *****
256 ;*****
257 0118 PROC INTC
258 ;NEW PROCESS
259 0118 80 3E 0002r 00 cmp [SERV],0 ;Hay PCs en servicio?.
260 011D 75 0A jne INTC_A ;Si. Proceso dar de "BAJA" de PC en serv.
261 011F 80 3E 0001r 00 cmp [WAITO],0 ;Hay PC esperando ha ser servidas?.
262 0124 75 38 jne SERVI_1 ;Si. Saltar a rutina de dar "ALTA" PCs.
263 0126 E9 008E jmp SAL_2 ;No. Entonces salir de rutina.
264
265 ;PROCESO DE DAR DE "BAJA" LA PC QUE SE TERMINO DE SERVIR.
266 0129 A0 0002r INTC_A: mov al,[SERV] ;cargar valor de serv
267 012C 38 06 0004r cmp [FIL1],al ;Son iguales FIL1=SERV ?.
268 0130 74 09 je Rest_1 ;Si. habilitar interrupcion de PC1.
269 0132 38 06 0005r cmp [FIL2],al ;No. Son iguales FIL2=SERV ?.
270 0136 74 16 je Rest_2 ;Si. habilitar interrupcion de PC2.
271 0138 E9 0087 jmp ESTORE ;No. NO HAY MAS PCs "ERROR".
272
273 ;Restablecer interrupción de PC1.
274 0138 80 36 0000r 10 Rest_1: xor [BUACK],00010000b ;Restaurar INTERRUPCION.
275 0140 C6 06 0004r 00 mov [FIL1],0 ;Restaurar registro de PC1.
276 0145 C7 06 0006r 0000 mov [Yload],0 ;Limpiar dirección de Servicio
277 0148 EB 14 90 jmp SERVI_1 ;Ir a Proceso de "ALTAS".
278
279 ;Restablecer interrupción y busy de PC2.
280
281 014E 80 36 0000r 20 Rest_2: xor [BUACK],00100000b ;Restaurar Int.
282 0153 C6 06 0005r 00 mov [FIL2],0 ;Restaurar registro de PC2.
283 0158 C7 06 0006r 0000 mov [Yload],0 ;Limpiar dirección de Servicio.
284 015E EB 01 90 jmp SERVI_1 ;Ir a Proceso de "ALTAS".

```

C:\TC\TESIS\PARTE.B

SERVIDOR DE IMPRESORA PARA PC's

```

285
286
287
288
289
290 0161 A0 0001r          SERVI_1: mov     al,[WAITO]      ;Obtener valor de "Cuenta de Espera"(CE).
291 0164 38 06 0002r      cmp     [SERV],al          ;Hay PC en "Cuenta de Espera" (CE)?
292 0168 74 3E           je      CLR_A             ;No. Entonces limpiar registros de CE.
293 016A FE 06 0002r      inc     [SERV]           ;Si. Nueva PC en servicio.
294 016E A0 0002r          mov     al,[SERV]        ;Obtener valor de Servicio.
295
;PROCESO DE "ALTAS" DE PCs
296 0171 38 06 0004r      cmp     [FIL1],al        ;Es PC1 la nueva.FIL1=SERV
297 0175 74 09           je      Rest_A           ;Si.Restablecer ack,Yload,ESTI
298 0177 38 06 0005r      cmp     [FIL2],al        ;Es PC2 la nueva. FIL2=SERV
299 017B 74 17           je      Rest_B           ;Si.Restablecer ack,Yload,ESTI
300 017D EB 43 90         jmp     ESTORE           ;Error en proceso de SERV.
301
Rest_A:
302 0180 80 36 0000r 01  xor     [BUACK],00000001b ;Restaurar ACK1
303 0185 BA 0304         mov     dx,Y4           ;cargar dirección PC1
304 0188 89 16 0006r      mov     [Yload],dx      ;dirección de servicio
305 018C C6 06 000Ar 01  mov     [ESTI],00000001b ;Mascara de PC1 en serv.
306 0191 EB 24 90         jmp     SAL_2           ;Salir de Subrutina.
307
Rest_B:
308 0194 80 36 0000r 04  xor     [BUACK],00000100b ;Restaurar ACK2.
309 0199 BA 0305         mov     dx,Y5           ;cargar dirección PC2
310 019C 89 16 0006r      mov     [Yload],dx      ;dirección de servicio
311 01A0 C6 06 000Ar 02  mov     [ESTI],00000010b ;Mascara de PC2 en serv.
312 01A5 EB 10 90         jmp     SAL_2           ;Salir de Subrutina.
313
314 01A8 C6 06 0001r 00  CLR_A: mov     [WAITO],0      ;limpiar Registros.
315 01AD C6 06 0002r 00  mov     [SERV],0
316 01B2 C6 06 000Ar 00  mov     [ESTI],0
317
SAL_2:
318 01B7 C6 06 0008r 00  mov     [CONTA],0
319 01BC C6 06 0009r 00  mov     [CONTB],0
320 01C1 CB           ret
321 01C2          ENDP      INTC
322
323
324
325
;*****
;Proceso de ERROR de No. de buffer en servicio*
;*****
326 01C2 B4 09          ESTORE: mov     ah,09h
327 01C4 BA 0015r      mov     dx,offset err1
328 01C7 CD 21         int     21h
329
330 01C9 B8 4C00         mov     ax,04c00h
331 01CC CD 21         int     21h

```

332

333

334

335

336

337

338

339

340

341

342

343 01CE

344 01CE 50

345 01CF 53

346 01DD 52

347

348 01D1 8A 0306

349 01D4 EC

350

351 01D5 24 03

352 01D7 A2 0003r

353

354 01DA 80 3E 0004r 00

355 01DF 75 18

356 01E1 A0 0003r

357 01E4 24 01

358 01E6 3C 01

359 01E8 75 12

360

361

362

363 01EA 80 36 0000r 11

364 01EF FE 06 0001r

365 01F3 A0 0001r

366 01F6 A2 0004r

367 01F9 EB 01 90

368

369

370 01FC

371 01FC 80 3E 0005r 00

372 0201 75 18

373 0203 A0 0003r

374 0206 24 02

375 0208 3C 02

376 020A 75 12

377

378

379

380 020C 80 36 0000r 24

381 0211 FE 06 0001r

382 0215 A0 0001r

383 0218 A2 0005r

384 021B EB 01 90

385

386 021E

387 021E 80 20

388 0220 E6 20

389 0222 5A

```
-----  
; YNT RUTINA DE INTERRUPCION PARA ASIGNACION EN LISTA DE  
; ESPERA  
-----  
; En la rutina se asigna un número a la PC que Interrupcio  
; y enmascarar por hardware a la interrupción de esa PC, además  
; de deshabilitar la señal ACK(peticion de caracter), de esa PC  
; para dejar de pedir caracteres hasta que sea su turno dentro de  
; de la lista de espera.  
-----
```

```
YNT:  
    push ax ;Salvar registros.  
    push bx  
    push dx  
  
    mov dx,Y6 ;cargar direccion estado de Interrupciones  
    in al,dx ;leer puerto  
  
;Prototipo de dos PCs  
    and al,00000011b ;no se usan los mas sig.  
    mov [ED0],al ;almacenar ESTADOS  
  
;Preguntar a PC1 (para saber quien interrupcio)  
    cmp [FIL1],0 ;Ya está en Lista de espera?.  
    jne PREG2 ;Si. Entonces fue la PC2  
    mov al,[ED0] ;Obtener ESTADOS.  
    and al,00000001b ;preguntar a PC1  
    cmp al,00000001b ;Interrupcio la PC1?  
    jne PREG2 ;No, preguntar a PC2  
  
;Fue PC1 la que interrupcio por primera vez, por lo tanto  
;desabilitamos ACK1 y INT1  
    xor [BUACK],00010001b ;Si, Desabilitar Int. y ack  
    inc [WAIT0] ;Incrementar lista de espera  
    mov al,[WAIT0] ;cargar valor de serv  
    mov [FIL1],al ;Asignar valor a FIL1=wait0+1  
    jmp PREG2  
  
;Pregunta a PC2 (computadora 2)  
PREG2:  
    cmp [FIL2],0 ;Ya está en fila de espera?.  
    jne SAL_4 ;Si. Entoces salir de proceso.  
    mov al,[ED0] ;Obtener ESTADOS.  
    and al,00000010b ;preguntar a PC2  
    cmp al,00000010b ;Interrupcio la PC2?  
    jne SAL_4 ;No, preguntar a PC3 o ERROR  
  
;Fue PC1 la que interrupcio por primera vez, por lo tanto  
;desabilitamos ACK2 y INT2  
    xor [BUACK],00100100b ;Si, Desabilitar Int. y ack  
    inc [WAIT0] ;Incrementar lista de espera  
    mov al,[WAIT0] ;cargar valor de serv  
    mov [FIL2],al ;Asignar valor a FIL2=wait0+1  
    jmp SAL_4  
  
SAL_4:  
    mov al,E01 ;al = valor de fin de interrupción  
    out CtrI8259_0,al  
    POP DX
```

390 0223 58  
391 0224 58  
392 0225 CF  
393  
394

POP BX  
POP AX  
IRET

END START

## Symbol Table

Symbol Name	Type	Value
??DATE	Text	"08/11/93"
??FILENAME	Text	"PARTEB "
??TIME	Text	"08:50:39"
??VERSION	Number	0100
@CODE	Text	PARTEB_TEXT
@CODESIZE	Text	1
@CPU	Text	0101H
@CURSEG	Text	PARTEB_TEXT
@DATA	Text	DGROUP
@DATASIZE	Text	1
@FILENAME	Text	PARTEB
@WORDSIZE	Text	2
BUACK	Byte	DGROUP:0000
CARACT	Byte	DGROUP:0030
CICLO_A	Near	PARTEB_TEXT:0053
CLR_A	Near	PARTEB_TEXT:01AB
CONTA	Byte	DGROUP:0008
CONTB	Byte	DGROUP:0009
CTRLB259_0	Text	020H
CTRLB259_1	Text	021H
DISABLEIRQ	Text	080H
EDO	Byte	DGROUP:0003
EIMP	Near	PARTEB_TEXT:00B7
ENABLEIRQ	Text	07FH
EDI	Text	020H
ERR0	Byte	DGROUP:000B
ERR1	Byte	DGROUP:0015
ERR2	Byte	DGROUP:0020
EST1	Byte	DGROUP:000A
ESTORE	Near	PARTEB_TEXT:01C2
FIL1	Byte	DGROUP:0004
FIL2	Byte	DGROUP:0005
INIB259	Far	PARTEB_TEXT:00C3
INTC	Far	PARTEB_TEXT:0118
INTC_A	Near	PARTEB_TEXT:0129
INTOFS	Word	DGROUP:0033
INTSEG	Word	DGROUP:0031
INTU	Near	PARTEB_TEXT:0080
INTW	Far	PARTEB_TEXT:00E8
LOAD	Near	PARTEB_TEXT:0089
PC	Far	PARTEB_TEXT:0089
POL_A	Near	PARTEB_TEXT:002B
POL_B	Near	PARTEB_TEXT:004C
PREG2	Near	PARTEB_TEXT:01FC
PREPRINT	Far	PARTEB_TEXT:0097
REIN_B	Near	PARTEB_TEXT:0111
REIN_C	Near	PARTEB_TEXT:00E8
RELOJ_B	Near	PARTEB_TEXT:0072
REST_1	Near	PARTEB_TEXT:013B
REST_2	Near	PARTEB_TEXT:014E
REST_A	Near	PARTEB_TEXT:0180
REST_B	Near	PARTEB_TEXT:0194
SAL_2	Near	PARTEB_TEXT:01B7
SAL_4	Near	PARTEB_TEXT:021E
SERV	Byte	DGROUP:0002
SERVI_1	Near	PARTEB_TEXT:0161

START	Near	PARTEB_TEXT:0000
TIME_A	Near	PARTEB_TEXT:004F
TOLE	Near	PARTEB_TEXT:00A4
TOLE_5	Near	PARTEB_TEXT:0075
TOLE_A	Near	PARTEB_TEXT:0114
TOLE_I	Near	PARTEB_TEXT:0105
WAITD	Byte	DGROUP:0001
Y0	Text	300H
Y1	Text	301H
Y2	Text	302H
Y3	Text	303H
Y4	Text	304H
Y5	Text	305H
Y6	Text	306H
Y7	Text	307H
YLOAD	Word	DGROUP:0006
YNT	Near	PARTEB_TEXT:01CE

Groups & Segments

Bit Size Align Combine Class

DGROUP	Group				
STACK	16	0100	Para	Stack	STACK
_DATA	16	0035	Word	Public	DATA
PARTEB_TEXT	16	0226	Word	Public	CODE

## BIBLIOGRAFIA

- PC, *Technical Reference Manual*, IBM Inc. 1986
- 8255, *Microprocessor and Peripheral Handbook*. Vol II. Intel. 1988.
- 8088, *Microprocessor and Peripheral handbook*. Vol I, Intel. 1988.
- Russ Davies. *Mapping The IBM PC*.
- Robert Grossblat. *The 8088 Proje Book*. Tab Books. 1989.
- Tom Swan. *Mastering Turbo Assembler*. Hayden Books. 1989.
- Douglas V. Hall. *Microprocesor and Interfacing*. Mc.Graw Hill. 1986.
- Impresora de Matriz de Punto. *Operating Instructions Manual*. Panasonic. 1990.
- Morgan-Waite. *Introducción al Microprocesador 8086/8088*. Mc. Graw. Hill. 1986.
- Singh, Walter A. Triebel. *The 8088 Microprocessor Programing, Interfacing, software, hardware and Applications*. Prentice Hall. 1990.
- Texas Instrument. *The TTL Data Book for Desing Engineers*. Segunda Edición. 1976.
- Hiwire II. User's Guide. Wintek Corporations.
- José Salazar, Mauricio Alvarez. *Apuntes del Diseño Asistido por Computadora*. Departamento de Estudios de Posgrado, Facultad de Ingeniería. 1990.