



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Estudios Superiores Cuautitlán

DISEÑO DE UN SISTEMA INTERFACE en base al
STANDARD HEWLETT PACKARD INTERFACE
BUS (HP-IB)

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N

MARCOS ROMERO HERNANDEZ
ARTURO ROBLES NAVARRETE

Director de Tesis: M. C. Juan Antonio Navarro Martínez
Premio Nacional de Ciencia y Tecnología Banamex

Cuautitlán Izcalli Edo. de Méx.

1986



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

	Pag.
INDICE -----	1
I- INTRODUCCION A LA COMUNICACION DIGITAL -----	1
II- LAS MICROCOMPUTADORAS EN LOS SISTEMAS DE COMUNICACION DIGITAL -----	7
i.- FUNDAMENTOS -----	8
ii.- SISTEMA EDUCACIONAL DE MICROPROCESADOR -----	10
III- SISTEMAS DE COMUNICACION DIGITAL -----	11
i.- FUNDAMENTOS -----	12
ii.- COMUNICACION SERIE -----	14
iii.- COMUNICACION PARALELO -----	21
iv.- DETECCION Y CORRECCION DE ERROR -----	22
v.- REDES DE COMUNICACION -----	23
IV- SISTEMAS OPERATIVOS -----	27
i.- FUNDAMENTOS -----	28
ii.- DESARROLLO DE SOFTWARE -----	33
iii.- PROGRAMACION -----	38
iv.- CRITERIO PARA LA EVALUACION DE PROGRAMAS -----	44
v.- CARACTER DE LOS SISTEMAS OPERATIVOS -----	45
vi.- COMPONENTES DE LOS SISTEMAS OPERATIVOS -----	46
vii.- REQUERIMIENTOS DE HARDWARE DEL SISTEMA OPERATIVO. -----	48

v.-	DISENO DEL SISTEMA INTERFACE	49
i.-	DESCRIPCION FISICA	50
ii.-	DESCRIPCION FUNCIONAL	50
iii.-	ESPECIFICACIONES	50
i.-	CARACTERISTICAS ELECTRICAS	50
iv.-	CAPACIDAD DE LA TARJETA HP-IB	52
v.-	FUNCIONES PROPIAS DE HP-IB	53
i.-	FUNCIONES DEL CONTROLADOR	53
ii.-	FUNCIONES DE LOS DISPOSITIVOS CONTROLADOS	54
vi.-	DIRECCIONAMIENTO DE PROTOCOLO	54
vii.-	FUNCIONES DE LOS DISPOSITIVOS EN LA INTERFACE HP-IB	55
viii.-	LINEAS DEL BUS DE LA INTERFACE HP-IB	56
ix.-	LINEAS DE CONTROL	56
x.-	LINEAS DE DATOS	59
xi.-	LINEAS DE TRANSFERENCIA	59
xii.-	TRANSFERENCIA DE DATOS	62
xiii.-	CARACTERISTICAS ELECTRICAS DE LA INTERFACE HP-IB	65
xiv.-	RELACION DE ATN Y LAS LINEAS DE TRANSFERENCIA NRFD, NDAC y DAV.	66
xv.-	DISENO DEL SISTEMA INTERFACE en base al standard HEWLETT-PACKARD INTERFACE BUS	67
i.-	COMPUERTAS DE TERCER ESTADO	69
ii.-	COMPUERTAS DE COLECTOR ABIERTO	71
iii.-	CALCULO DE LA RESISTENCIA DE CARGA DE LAS COMPUERTAS CUYA SALIDA ES DE COLECTOR ABIERTO	71

iv.- Z80 CPU	72
i.- CARACTERISTICAS	72
ii.- DESCRIPCION GENERAL	76
iii.- OPERACION GENERAL DE INTERRUPTONES	76
i.- INTERRUPCION NO MASCARABLE	77
ii.- INTERRUPCION MASCARABLE	78
iii.- MODO DE INTERRUPCION 0	78
iv.- MODO DE INTERRUPCION 1	79
v.- MODO DE INTERRUPCION 2	79
vi.- OPERACION DE HABILITACION/ DESHABILITACION DE INTERRUPCION	80
iv.- Z80 PIO	81
i.- CARACTERISTICAS	81
ii.- DESCRIPCION GENERAL	84
iii.- MODOS DE OPERACION	85
iv.- PROGRAMACION	88
v.- 74138 DECODIFICADOR	92
vi.- 7400 y 7408 NAND y AND	92
vii.- 7406 y 74241 INVERTERS (COLECTOR ABIERTO) BUFFERS (LINE DRIVERS/LINE RECEIVERS)	92
viii.- DISEÑO DEL HARDWARE DEL SISTEMA INTERFACE en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB)	93
i.- INTERFACE HP-IB EN UN SISTEMA TIPICO DE COMPUTADORA	99
ii.- ESTRUCTURA DEL BUS DE INTERFACE HP-IB	100

iii.- DIAGRAMA DE BLOQUES DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) EN EL QUE SE MUESTRA COMO CONTROLADOR DE SISTEMA -----	101
iv.- DIAGRAMA DE BLOQUES DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) EN EL QUE SE MUESTRA COMO EMISOR/RECEPTOR -----	102
v.- DIAGRAMA ESQUEMATICO LOGICO DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) -----	103
EN LA FIGURA XII SE MUESTRAN LOS REQUERIMIENTOS SECUENCIALES DE LA TRANSFERENCIA DE DATOS INTER-ASEGUNADA THREE WIRE HANDSHAKE TECHNIQUE -----	104
vi.- SECUENCIA DE TIEMPOS DEL PROTOCOLO (HANDSHAKE) -	105
vii.- SIMULACION DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) -	107
viii.- RELACION DE LAS OPERACIONES LOGICAS QUE SIMULAN LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB)-	110
ix.- CONECTOR DEL CABLE DE INTERFACE HP-IB -----	116
x.- LOCALIZACION DE LOS CIRCUITOS INTEGRADOS EN LA INTERFACE DISENADA en base al standard HEWLETT PACKARD INTERFACE BUS (HP-IB) -----	117
xi.- DISEÑO DEL FIRM-WARE -----	118
VI.- CONCLUSIONES -----	145
APENDICES -----	147
A.- DIAGRAMAS DE BASE DE LOS C.I. Y CONECTORES -----	148
B.- RELACION DE CONEXIONES -----	157
C.- DISEÑO DEL CIRCUITO IMPRESO -----	160
D.- RELACION DE COSTOS -----	161
E.- CONSIDERACIONES DE CONSTRUCCION Y OPERACION -----	162
F.- BIBLIOGRAFIA -----	167

X.- RELACION DE GRAFICOS

FIGURA I	DIAGRAMA DE BLOQUES DEL SISTEMA EDUCACIONAL DE MICROPROCESADOR	10
FIGURA II	SE MUESTRAN LOS REQUERIMIENTOS SECUENCIALES DE LA TRANSFERENCIA DE DATOS INTER-ASEGURADA THREE WIRE HANDSHAKE TECHNIQUE	63
FIGURA III	DIAGRAMA DE BLOQUES DE LA UNIDAD CENTRAL DE PROCESO Z80 CPU	73
FIGURA IV	REGISTROS DEL PRECESADOR Z80 CPU	74
FIGURA V	DIAGRAMA DE BLOQUES DEL CONTROLADOR DE ENTRADA/SALIDA PARALELO Z80 PIO	82
FIGURA VI	DIAGRAMA DE BLOQUES TIPICO DE LOS PUERTOS DE ENTRADA/SALIDA DEL Z80 PIO	83
FIGURA VII	INTERFACE HP-IB EN UN SISTEMA TIPICO DE COMPUTADORA	99
FIGURA VIII	ESTRUCTURA DEL BUS DE INTERFACE HP-IB	100
FIGURA IX	DIAGRAMA DE BLOQUES DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) EN EL QUE SE MUESTRA COMO CONTROLADOR DE SISTEMA	101
FIGURA X	DIAGRAMA DE BLOQUES DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) EN EL QUE SE MUESTRA COMO EMISOR/RECEPTOR	102
FIGURA XI	DIAGRAMA ESQUEMATICO LOGICO DE LA INTERFACE DISENADA en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB)	103

FIGURA XII SE MUESTRAN LOS REQUERIMIENTOS
SECUENCIALES DE LA TRANSFERENCIA DE DATOS
INTER-ASEGURADA

THREE WIRE HANDSHAKE TECHNIQUE ----- 104

FIGURA XIII CONECTOR DEL CABLE DE INTERFACE HP-IB - 116

FIGURA XIV LOCALIZACION DE LOS CIRCUITOS INTEGRADOS EN
LA INTERFACE DISEÑADA en base al standard
HEWLETT PACKARD INTERFACE BUS (HP-IB) ---- 117

INTRODUCCION

INTRODUCCION

Una de las características mas importantes que todos los seres vivos deben poseer es la habilidad de comunicarse con otros individuos de la misma especie. Tal comunicacion provee una ventaja de sobrevivencia definida en el sentido Darwiniano del termino y esta es ciertamente encontrada en la mayor parte de las criaturas multicelulares, desde insectos hasta el hombre mismo. En lo que se refiere a los insectos, existen varios modos de comunicacion, incluyendo la danza de una abeja y formas de comunicacion quimica a traves de agentes quimicos conocidos como 'pheromones'. El hombre puede comunicarse con sus semejantes a traves de sus cinco sentidos. Asumiendo que un individuo desea comunicarse con otro mediante el sentido del oido y el uso de la palabra, es claro que debe haber algun acuerdo concelniente a como un sonido hablado sera interpretado por el que escucha.

Al pasar del tiempo, diferentes regiones alrededor del mundo han desarrollado independientemente sus propias convenciones respecto a sonidos especificos y su transcripcion al papel. Nosotros llamamos tales convenciones lenguajes. Se conocen cientos de lenguajes diferentes, aunque solo un numero relativamente pequeno de ellos es usado por un porcentaje significativo de la poblacion mundial. Ademas, la popularidad de un lenguaje especifico crece y se desvanece en el desarrollo de

la civilización. El Latín, en algún tiempo lenguaje dominante en Europa, es ahora considerado una lengua muerta, aunque claramente ha influenciado la mayor parte de los lenguajes europeos en maneras no poco profundas. El Inglés puede considerarse ahora el lenguaje más común y es posible que dentro de algunos siglos no lo sea más.

En su sentido básico eléctrico, Comunicación se refiere al envío, recepción y procesamiento de información por medios eléctricos. Como tal, se inició con la telegrafía en los años 1840's, desarrollándose con la telefonía algunas décadas después y posteriormente la radio al inicio del presente siglo. La radio comunicación hecha posible por la invención del tubo de vacío (triode tube) fue grandemente estimulada por el trabajo realizado durante la Segunda guerra mundial. Posteriormente llegó a ser más ampliamente usado y refinado mediante la invención y uso del transistor, circuitos integrados y otros dispositivos semiconductores. Más recientemente, el uso de satélites y fibras ópticas han hecho la comunicación más amplia con un creciente énfasis en las computadoras. Un Sistema de Comunicación es primeramente relacionado con el proceso de información antes de su transmisión, posteriormente la filtración del ruido. Finalmente la recepción, la cual, incluye la decodificación, almacenamiento e interpretación. En este contexto, encontramos formas de comunicación incluyendo radio telefonía y telegrafía,

punto a punto y comunicaciones móviles (comerciales o militares), comunicaciones a través de computadoras, radar, ayuda de radio a la navegación y la radio telemetría.

Antes de profundizar en el sistema particular en estudio, hemos de tener en cuenta lo siguiente: los sistemas de comunicación existen para comunicar un mensaje. Este mensaje es recibido de una fuente de información, que la origina en el sentido de seleccionar un mensaje de un grupo de mensajes. El conjunto o número total de mensajes consiste de mensajes individuales, los cuales pueden ser distinguidos uno de otro. Estos pueden ser palabras, grupos de palabras, código de símbolos o cualquier unidad pre-establecida. La información misma es lo que se ha convenido, la cantidad de información contenida en cualquier mensaje es medida en bits. Cuando los bits son ensamblados en un grupo de longitud particular, estos forman una palabra binaria, algunas veces llamada byte. El número de posibles combinaciones aumenta en potencias de 2, de tal manera que palabras binarias de 8 bits en longitud, tienen 256 códigos diferentes.

La Comunicación Digital se define como la transmisión de una serie de formas de onda, seleccionadas cada una de un grupo específico y finito. La diferencia fundamental entre los Sistemas de Comunicación de Datos Digitales y de Datos Continuos (Datos Analógicos), consiste en que en la Comunicación Digital interviene la transmisión y detección de una forma de un grupo

finito de formas conocidas, mientras que en la Comunicación Continua, existe un número infinito de ondas y las formas de onda correspondientes no se conocen. En el proceso de Cuantificación se aproximan las señales continuas para que puedan tomar tan solo ciertas amplitudes discretas. Esencialmente esto es la digitalización de la señal continua. Los mensajes se pueden transmitir mediante un número finito de símbolos o niveles. En los sistemas digitales, el problema de detección es un poco más sencillo que en los sistemas continuos. Durante la transmisión, las formas de onda sufren los efectos del ruido del canal. Cuando la señal llega al receptor, es necesario decidir cual de las N formas de onda conocidas posibles se ha transmitido. Una vez adoptada tal decisión, se recupera exactamente la forma de onda transmitida, sin ningún ruido. Por lo tanto, en este sentido, el ruido del canal no ejerce influencia alguna sobre el mensaje. Sin embargo, provocará un error en nuestra decisión. Por ejemplo, podemos quedar desorientados por el ruido del canal y tomar una decisión incorrecta. De modo que debemos aceptar el hecho de que algunas decisiones serán equivocadas y que el error crezca con el incremento del ruido. Sin duda, la ocasión de cometer un error (probabilidad de error), constituye un criterio muy significativo para detectar señales en sistemas digitales. Existe un tipo de Comunicación Digital utilizado ampliamente en los Sistemas de Microcomputadoras que es conocido como Comunicación Binaria, la

cual emplea dos símbolos únicamente. Uno de los símbolos se representa por un pulso, y el otro por la ausencia del mismo, es decir, ausencia de señal, (presencia de un nivel lógico determinado). En este tipo de comunicación no importa detectar la forma de onda de la señal pues ya se conoce, por lo tanto, solo se requiere determinar si el pulso está presente o no. Esto nos lleva a definir una señal binaria, la cual es un voltaje o una corriente que lleva cierta información en la forma de cambios entre dos diferentes estados, los cuales se encuentran en un intervalo discreto; estos dos estados son llamados generalmente 0 lógico y 1 lógico.

El objeto de este breve prefacio, ha sido desarrollar una analogía entre los lenguajes hablados y lenguajes digitales, más comúnmente llamados Códigos Digitales. Así como existen numerosos lenguajes hablados, hay una variedad de códigos digitales entre los que se destacan los siguientes: Binario (binary), Binario Codificado Decimal (bcd), Gray, Exceso 3, American Standard Code for Information Interchange (ASCII) y Extended Binary Coded Decimal Interchange Code (EBCDIC). Estos lenguajes son en efecto verdaderos lenguajes de la Electrónica Digital.

LAS MICROCOMPUTADORAS EN LOS SISTEMAS DE COMUNICACION DIGITAL

LAS MICROCOMPUTADORAS EN LOS SISTEMAS DE COMUNICACION DIGITAL

FUNDAMENTOS

La función que desempeñan las microcomputadoras en un sistema depende de los programas de control de sistema y los programas de aplicación. La microcomputadora puede comunicarse con su operador y con otras microcomputadoras, esto es la microcomputadora acepta datos, los cuales son procesados de acuerdo al programa y produce los resultados de su proceso, a través de la operación de impresión o control de otras microcomputadoras. El programa juega un papel muy importante, puesto que establece lo que ha de ocurrir, el orden en que ha de ocurrir y quien lo realiza. La microcomputadora puede ejecutar operaciones aritméticas a gran velocidad y aceptable precisión, estas características son las mayores ventajas de la microcomputadora. La microcomputadora puede elegir entre varias alternativas, esta habilidad resulta de la capacidad de la computadora para realizar comparaciones a gran velocidad y precisión. El programa debe de incluir las reglas que la microcomputadora debe aplicar para la selección de tales alternativas y las especificaciones de acción de las mismas. Existen generalmente tres factores importantes en la limitación de una microcomputadora, estos son: el económico, la capacidad y el entendimiento del sistema en que la microcomputadora se

emplea. Desde el punto de vista organización la microcomputadora presenta tres aspectos funcionales elementales:

- a) recibir o aceptar datos que ha de procesar (función de entrada).
- b) procesamiento de información (función de procesamiento).
- c) emitir resultados (función de salida).

En un sentido fundamental, la microcomputadora es un equipo electrónico para el procesamiento de información. Esta habilidad lógica a menudo sirve como base para el uso de las mismas en un sistema.

La aplicación de una microcomputadora es el uso particular de una microcomputadora en un sistema por un usuario.

La aplicación de la red que se diseña, consiste en el desarrollo de HARDWARE, FIRMWARE y SOFTWARE en la FACULTAD DE ESTUDIOS SUPERIORES-C. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

A continuación se muestra un diagrama de bloques en el que se describen las características del SISTEMA EDUCACIONAL DE MICROPROCESADOR IBO (MKE) que es empleado como elemento de sistema, en el cual la red de microprocesadores será basada.

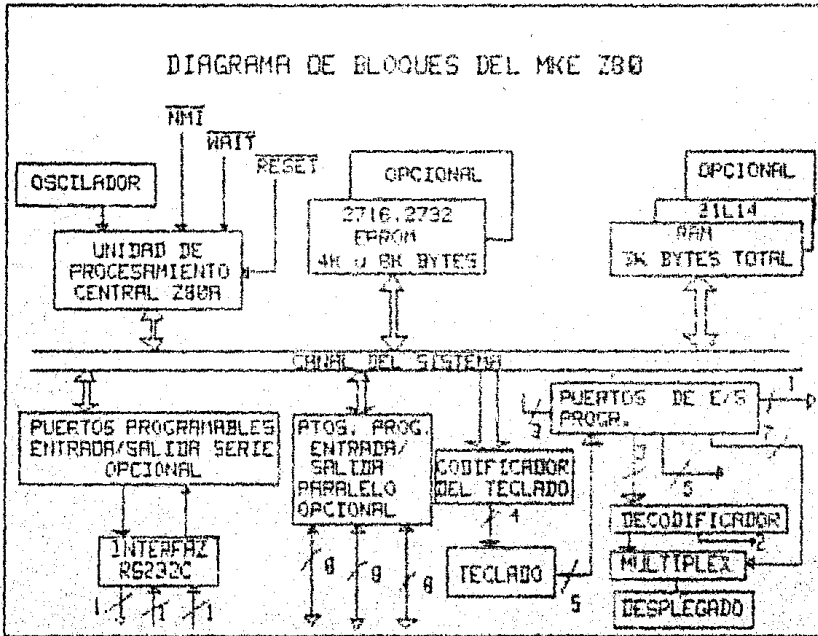


FIGURA I DIAGRAMA DE BLOQUES DEL SISTEMA EDUCACIONAL DE MICROPROCESADOR Z80 (MKE).

SISTEMAS DE COMUNICACION DIGITAL

SISTEMAS DE COMUNICACION DIGITAL

FUNDAMENTOS: Las comunicaciones de datos surgieron como resultado natural del sofisticado desarrollo de sistemas de computadoras. Que permiten una transferencia de datos rapida y eficiente. Fue la habilidad de manipular multiples tareas y numerosas terminales remotas quienes promovieron el crecimiento de la industria de transmision de datos. Inicialmente se pretendio la estandarizacion para las interconexiones necesarias entre la computadora y dispositivos perifericos, esta estandarizacion considero conectores standard, formatos de senales y niveles de senales. Tan pronto como estos 'standards' llegaron a ser reconocidos por la industria, luego a ser deseable el extenderlos a los medios de transmision media y larga. La necesidad de 'standards' de transmision llego a ser realmente apremiante cuando se presentaron las facilidades que permitian las computadoras de utilizar el sistema telefonico para sus requerimientos de transmision. Sin embargo, se presento un gran problema debido a que el sistema telefonico fue disenado para la comunicacion de voz, por lo que se tuvieron que hacer las modificaciones requeridas para la transmision de datos. La comunicacion de datos ahora posee su propio lenguaje, equipo y 'standards'. Y es una industria en si misma y ciertamente una parte integral de la sociedad computarizada de hoy.

Uno de los códigos más universales es el AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII), el cual está basado en una progresión binaria. El mismo es mostrado en la siguiente tabla en la que también se muestra su conversión a hexadecimal:

ASCII TO HEXADECIMAL CONVERSION TABLE

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

COMUNICACION SERIE

La característica de un flujo de datos serie es que los datos son transmitidos y recibidos como una sola señal a través de terminales de los dispositivos. Si un CPU va a transmitir o a recibir datos en serie, deberá tener una lógica de interfaz capaz de convertir los datos serie a datos paralelos y los datos paralelos a datos serie. Un dispositivo de recepción interpreta a una señal serie como cualquier otra señal digital, un nivel alto (+5 Volts) representando el dígito 1 y un nivel bajo (0 Volts) representando el dígito 0. El dispositivo receptor requiere de una evidencia que identifique los bits de datos y esta se encuentra en los extremos de los bits de datos. Se usa una señal de reloj para identificar el instante en el cual el dispositivo receptor debe interpretar el nivel de la señal como dato. La señal de datos serie se debe de generar en el dispositivo transmisor antes de que pueda ser interpretada por el dispositivo receptor. Si el dispositivo transmisor utiliza la frecuencia de un reloj para generar las señales de datos serie, entonces el dispositivo receptor debe usar un reloj con la misma frecuencia para interpretar las señales de datos serie. Las señales de reloj de transmisión y recepción no deben ser idénticas, debido a que se requiere de un tiempo finito para que la señal cambie de estado. El periodo de las señales de reloj proporciona el tiempo de duración de cada nivel lógico en un dato

serie. El dispositivo receptor debe esperar a que la señal de datos serie se asiente, suponiendo que ha cambiado de estado, antes de tratar de leer de nuevo el nivel de la señal. Un retardo para el asentamiento de la señal es una característica del dispositivo transmisor; la longitud de este retardo se describe en las especificaciones que proporcionan los fabricantes de los dispositivos. La transferencia de datos serie se mide en 'bits por segundos' y se refiere a este número como la RAZÓN DE BAUDS.

Los Sistemas de Microcomputadoras requieren de una señal de reloj para dar los tiempos para la ejecución de las instrucciones dentro del CPU. No debemos confundir el reloj del Sistema de Microcomputadora con el reloj de datos serie de transmisión y recepción; lo único que estas señales tienen en común es que ambas son señales de reloj. Las señales de reloj de los datos serie pueden o no ser derivadas del reloj del Sistema de Microcomputadora. Desde el punto de vista del usuario de microcomputadora, la velocidad es la diferencia principal entre el reloj del Sistema de Microcomputadora y el reloj de datos serie. La velocidad de transferencia de datos serie más rápida es aproximadamente 200 veces más lenta que la velocidad típica del reloj del CPU de una Microcomputadora. El reloj serie de E/S no necesariamente tiene que pulsar a la misma velocidad exacta a la de la transmisión de los bits aunque frecuentemente lo hace.

Es bastante común que la velocidad de reloj sea 16 veces la velocidad de bauds, la razón para tener relojes con frecuencias de 16 (x 16) veces la razón de bauds es para tratar de llegar tan cerca como sea posible al centro del intervalo de tiempo de cada dígito binario cuando se muestrean las señales de los datos serie. Cuando la señal de datos serie permanece en alto cuando no está transmitiendo, se le designa frecuentemente como 'marca'. Cuando el dispositivo de recepción detecta una transición de bajada en la línea, comenzará a identificar los datos binarios después de un periodo de duración de un dígito binario 0 (bit de arranque). También se puede definir una secuencia de bits especiales de sincronización, de tal manera que cada flujo de datos serie debe ser precedido por este patrón de sincronización. Este patrón de sincronización se conoce como carácter SYNC. Especificar que un flujo de datos serie debe de usar caracteres de sincronización es una de las reglas que se deben convenir en el flujo de datos serie con objeto de asegurar que los dispositivos receptores interpreten correctamente los datos transmitidos. Este conjunto de reglas es conocido como protocolo de comunicaciones. Cuando se transmiten datos serie sobre líneas telefónicas o directamente entre un dispositivo receptor y un transmisor se debe chequear por errores de transmisión. Algunas veces durante la transmisión se cuelan señales espurias de datos en las líneas de datos serie por lo que el dispositivo receptor debe tener algún medio para determinar

cuando existen errores dentro de los datos. Una técnica empleada para checar error en la transmisión es agregar un carácter de Redundancia Ciclica a continuación de los segmentos de flujo de datos. El carácter de redundancia ciclica es un número que se crea al dividir el segmento de datos transmitido por un número (polinomio) fijo. El receptor recibe el flujo de datos y el carácter de redundancia ciclica, entonces, el receptor divide el flujo de datos recibidos por el polinomio fijo para generar otra vez el carácter de redundancia ciclica. Si el resultado no es igual al carácter de redundancia ciclica que acompaña al dato, entonces, existe un error. Este método es simple para detectar errores en las transmisiones.

La comunicación de datos serie se puede dividir en las dos siguientes clasificaciones: transferencia sincrónica de datos serie cuya principal característica es que las frecuencias de los relojes de transmisión y recepción son las mismas. Habiendo establecido una vez la velocidad en bauds de transferencia de datos serie, el dispositivo transmisor debe transmitir un bit de datos en cada pulso de reloj; por lo tanto, el dispositivo receptor debe conocer inicialmente como interpretar la señal de datos serie. El carácter SYNC es una palabra fija previamente definida. Un dispositivo receptor entra en modo 'HUNT' mientras espera a que comiencen a llegar los datos serie sincrónicos. Durante la espera el receptor lee continuamente los datos serie

que recibe tratando de encontrar en el flujo de datos serie el patron 'standard' SYNC. Si su protocolo llama por un solo caracter SYNC, entonces el dispositivo receptor comenzara a interpretar la senal como datos en cuanto encuentre este caracter. La transmision de datos sincrona requiere que el dispositivo transmisor mande datos continuamente. Si el dispositivo transmisor no tiene datos listos para mandar debe meter relleno con caracteres SYNC hasta que el proximo caracter real esta listo para transmitir. Cuando el dispositivo receptor decodifica el caracter SYNC en medio de un mensaje, ignorara el caracter, pero permanecera en sincronizacion con el flujo de datos en serie, listo para interpretar el proximo caracter. La transferencia asincrona de datos en serie se distingue en que el dispositivo transmisor enviara una senal conocida como 'marca' (usualmente nivel alto) mientras no tenga un dato para transmitir. Para indicar que comenzara a transmitir un dato valido el transmisor envia un bit 0, el cual se conoce como 'senal o bit de arranque'. Despues del bit de arranque el transmisor envia un dato compuesto de una cantidad predefinida de bits. Para indicar que termino la transmision de un dato el transmisor envia una senal conocida como 'senal de parada'. Esta senal de parada consiste de uno, uno y medio o dos bits con nivel bajo. Cada dato en la transmision asincrona tiene el siguiente formato: 1) senal de arranque 2) dato 3) senal de parada. Existe una similitud en el concepto de los caracteres SYNC del flujo de

datos sincronos y los bits del formato de un flujo de datos asincronos, ambos se requieren para identificar a los datos. Durante la transferencia de datos asincronos el dispositivo receptor tiene un medio adicional de chequear error por transmision. El primer digito binario de cada unidad de datos debe ser un cero representando el bit de arranque y los ultimos dos digitos binarios de la unidad de datos pueden ser cero representando los bits de parada. Si el dispositivo receptor no detecta los bits adecuados de arranque y parada para cualquier unidad de datos, entonces, representara un error de formato. Un modem es un dispositivo que traduce senales de linea telefonica a niveles de logica digital y niveles de logica digital a senales de linea telefonica. El modo de operacion de transmision puede ser de alguna de las siguientes maneras: un modo que provee la transmision de datos en solo una direccion es considerado como modo simplex. Este tipo de grupo de datos usa solamente un canal de transmision, de tal manera que no existe una senal disponible en la direccion del receptor al transmisor. Este es un metodo economico de transferencia de datos pero es muy limitado en su aplicacion, obviamente no permite senalizacion de error ni requerimiento de retransmision. Algunos modems permiten la transferencia de datos en dos direcciones pero el flujo de datos es alternado, con el flujo en una direccion en un momento y en la direccion opuesta en otro momento. Este tipo de transmision es conocida como half-duplex, requiere de solo un canal de

transmision, el cual debe ser bidireccional. En este tipo de transmision la velocidad de transmision es reducida debido a la necesidad de compartir el mismo circuito mientras que los componentes del circuito de transmision toman turnos. El ultimo tipo de transmision conocido como full-duplex permite la transferencia de informacion al mismo tiempo aunque dos circuitos de transmision son requeridos, uno para cada direccion de transmision. Los modems son colocados en los extremos para la modulacion y demodulacion de las senales transmitidas.

COMUNICACION PARALELO

El microprocesador es considerado como un dispositivo orientado a la interconexión de varios dispositivos periféricos a través de una trayectoria sobre la cual información digital es transferida de cualquiera de varios emisores a cualquiera, o a todos de varios receptores (bus-oriented device), es decir, envía o recibe datos como palabras binarias (parallel binary words) sobre interconexiones comunes. Todos los dispositivos con quienes el microprocesador está conectado comparten el bus de datos. El dispositivo que no envía ni recibe información se encuentra en un estado de alta impedancia (tri-state). Los dispositivos periféricos son responsables de convertir los datos en una forma en que puedan ser utilizados por el microprocesador. Esto es, los dispositivos utilizados en un sistema de microprocesador deben ser orientados a través del bus (bus-oriented) y deben enviar y recibir datos como palabras binarias. El uso de palabras binarias significa que el sistema del microprocesador requiere datos paralelos. En un sistema de 8 bits como el que utilizamos en el proyecto, todos los bits deben llegar al puerto del dispositivo del sistema al mismo tiempo.

DETECCION Y CORRECCION DE ERROR

El ruido se refiere a formas de onda electricas no deseadas que distorsionan la senal que porta un mensaje. El ruido es inevitable en un sistema de comunicacion electrico, ya que existe un ruido termico asociado con la conduccion y radiacion. El ruido limita nuestra habilidad para identificar correctamente el mensaje de la senal y por lo tanto, limita la transmision de informacion. El proceso relacionado con la correccion de error normalmente resulta en un incremento en el numero de bits por segundo que son transmitidos y naturalmente esto aumenta el costo de transmision. Los procedimientos que permiten la correccion del error en el lugar de recepcion son complicados, por lo tanto, es necesario para los usuarios de datos determinar la importancia de los datos transmitidos y decidir que nivel de deteccion de error y correccion es adecuado para esos datos. La tolerancia que el usuario determine para los errores, decidira que sistema de control de error es adecuado para el circuito de transmision usado por los datos del usuario. El emisor y el receptor en un sistema de comunicacion deben ser cuidadosamente disenados para evitar la distorsion de la senal y minimizar los efectos del ruido en el receptor, para que sea posible una aceptable reproduccion del mensaje emitido por la fuente.

REDES DE COMUNICACION

Cuando creamos una asociación de computadoras, solemos llamarla red de computadoras. La mera existencia de una red de computadoras implica la habilidad de transmitir información de una computadora fuente a una computadora destino. El destino del mensaje y la ruta del mismo deben de ser completamente determinados por el emisor. En una red de computadoras una interface es un componente lógico (incluyendo hardware, software y procedimientos de operación) que permite a una computadora intercambiar información y controlar la operación de las otras, considerando el nivel físico eléctrico, nivel de transmisión y protocolos.

Datos que son transmitidos de un nodo en una red a otro nodo, necesariamente deben contener implícita o explícitamente información que especifique origen, destino y ruta.

El control de información debe incluir las facilidades siguientes:

- 1.- Identificar el origen y destino de cada mensaje.
- 2.- Transmitir mensajes de control.
- 3.- Evitar la transferencia de información a destinos no especificados.

- 4.- Guiar adecuadamente la información dentro de la red.
- 5.- Tratar adecuadamente los errores y mal funcionamiento.

TOPOLOGIAS DE RED

Varios esquemas han sido elaborados para clasificar las topologías de interconexión de las redes para permitir la modularidad y generalización de los efectos de falla.

Las topologías consideran tres elementos principales:

PROCESADOR una computadora de la cual pueden ser enviados mensajes a otros procesadores en la red y la cual puede recibir mensajes de otros procesadores en la red.

SWITCH un dispositivo capaz de recibir un mensaje y direccionarlo al destino especificado. La ruta específica a ser usada puede ser elegida por el switch para minimizar el congestionamiento de ruta alguna.

TRAYECTORIA DE DATOS pueden tener varias capacidades y disciplinas de línea y pueden ser simplex, half-duplex o full-duplex.

TOPOLOGIAS

REDES LINEALES .- Esta topología de red involucra un número de computadoras (nodos) interconectados por una simple línea común de comunicación. Los mensajes son enviados directamente de la computadora fuente a la línea para ser reconocidos y aceptados por el nodo designado como destino.

RED LINEAL CON SWITCH.- Esta topología de red es similar a la de las redes lineales, excepto que todos los mensajes son transmitidos al switch y posteriormente retransmitidos a su destino.

REDES ENTRELAZADAS.- Esta topología une varias redes lineales con switch, lo que permite que un mensaje sea transmitido a un nodo que se encuentre en otra red a través de varios switches que actúan correspondientemente.

REDES SIMPLES ANIDADAS.- Esta topología es esencialmente una red lineal con sus extremos conectados. Consiste de un número de nodos, cada uno conectado con dos nodos vecinos. Aunque la trayectoria de datos podría ser bidireccional, en la práctica es casi siempre unidireccional (simplex). Por lo tanto, un nodo vecino siempre es la fuente y el otro nodo vecino es el destino, de esta forma los mensajes circulan a través de la red de fuente a destino.

REDES ANIDADAS CON SWITCH.- Esta topología se logra adicionando un switch a la red, el cual es el destino de cada mensaje de un nodo y es la fuente de cada mensaje de otro nodo.

REDES ANIDADAS INTERCONECTADAS.- Esta topología de red se logra comunicando los switches de varias redes anidadas con switch.

REDES TIPO ESTRELLA.- En este tipo de red, se presenta un switch central conectado a N líneas bidireccionales (half-duplex o full-duplex) con N nodos. El switch central es el destino de cada mensaje de un nodo y la fuente de cada mensaje de otro nodo.

REDES TIPO ARBOL.- Este tipo de red varía grandemente con la configuración, sin embargo, existe cierta dependencia de cada uno de los nodos con los demás.

REDES IRREGULARES.- Consistentes de varias de las redes descritas anteriormente comunicadas a través de switches centrales que pueden ser combinados en diferentes formas.

SISTEMAS OPERATIVOS

SISTEMAS OPERATIVOS

FUNDAMENTOS

La microcomputadora no puede realizar un trabajo útil a menos que sea proveída con dos tipos de programas: Programas de sistema y Programas de aplicación.

Los programas de aplicación son aquellos que proporcionan utilidad inmediata. Los programas de sistema son aquellos que proveen las herramientas de programación sin las cuales los programas de aplicación no serían útiles.

Algunos sistemas tienen un programa llamado Monitor que se encuentra grabado en memoria ROM para la realización de tales funciones. Otros sistemas contienen programas del tipo bootstrap, también en memoria ROM, los cuales se encargan de cargar el sistema operativo que se encuentre ya sea en cassette o en disco. El Sistema Operativo es el primer componente de los programas de sistema, sin el cual no puede realizarse trabajo alguno de importancia. Los otros componentes son de ayuda a la programación, que facilitan la creación y ejecución de largos y complejos programas de aplicación. Entre los programas de sistema podemos distinguir las siguientes clases:

El Sistema Operativo que se encarga de las operaciones de entrada y salida de información entre periféricos, y del

sistema. El cual normalmente es cargado a la memoria principal cuando el sistema es inicializado y permanece hasta que este es deshabilitado. Proporciona facilidades para cargar y ejecutar programas que existen en almacenamiento interno, guardar programas recientemente creados o archivos de datos en unidades de almacenamiento externo.

Editores de texto que crean y modifican archivos de texto o archivos de programas en lenguaje de bajo nivel o lenguaje de alto nivel. Un buen editor es esencial para la creación de archivos de programas fuente. Los editores de texto se clasifican generalmente en dos tipos:

Editores de línea (Line-oriented)

Editores de página (Context-oriented)

Los dos tipos introducen los datos directamente a memoria, guardándolos en un archivo para permitir borrar, modificar o aumentar el texto en un archivo existente.

Los Editores de línea numeran cada cuerda de texto terminada por un delimitador de línea (carriage return). Una edición es llevada a cabo seleccionando el número de línea, la cual ha de ser afectada por un comando de edición.

Los Editores de página no numeran las líneas del texto ya que reconocen el carriage return como un delimitador de línea. Todos

los comandos son referidos a un apuntador de caracter visible, el cual es posicionable en cualquier lugar del texto y opera sobre un numero de caracteres o lineas anteriores o posteriores al mismo. Existen unos formateadores de texto auxiliares del editor los cuales procesan el texto para impresion. Algunos programas 'Procesadores de Palabras' combinan al editor y al formateador, aunque el editor y el formateador son mas comunmente programas separados.

Ensambladores (Assemblers) para convertir archivos de programas fuente en ensamblador en codigo de maquina ejecutable. Un ensamblador convierte un archivo de programa fuente consistente de comandos de lenguaje ensamblador en codigo ejecutable. Un ensamblador generalmente realiza dos pasos: El primero, reserva espacio en memoria para cada instruccion y considera las direcciones. Durante este paso el ensamblador construye una tabla de simbolos conteniendo constantes y sus valores definidos, asi como, etiquetas y las direcciones en las cuales se presentan. El segundo paso sustituye operandos simbolicos y direcciones simbolicas por sus correspondientes direcciones y/o valores encontrados en la tabla de simbolos. Dos nuevos archivos son creados: uno constituye el codigo objeto binario que cuando terminado resulta en el programa de lenguaje de maquina, el cual puede ser cargado en memoria y correrse; el otro es un listado que relaciona las sentencias del programa fuente en lenguaje

ensamblador con el valor equivalente ASCII Hexadecimal de la dirección donde aparece un comando y el comando ejecutable para cada operando fuente. Para crear un archivo ejecutable, se requiere de un programa que convierta el archivo hex en un binario en un lugar correcto en memoria. Este programa es llamado Linker. Una vez realizada esta operación, el programa puede ser guardado como un archivo binario ejecutable. Existen unos Macro-assemblers que permiten definir una secuencia de instrucciones y darles un nombre. Los parámetros sobre los cuales la secuencia del código opera son valores simbólicos. Una secuencia de este tipo es reconocida como una macro-instrucción o simplemente un macro. En cualquier momento que el ensamblador encuentra el nombre de un macro, sustituye el nombre con la secuencia de códigos contenida en la macro-definición.

Programas espulgadores y rastreadores (Debug and Trace) que muestran y permiten modificar el contenido de la memoria, detener la ejecución del programa en un punto determinado (Breakpoint) y desplegar el contenido de los registros y otras funciones de rastreo. Este tipo de programas opera sobre código de programas ejecutables, estática y dinámicamente. La característica estática incluye los siguientes comandos:

Mostrar, sustituir, mover y desensamblar.

Las características dinámicas permiten al programa de trabajo,

ser ejecutado bajo control del programa debug/trace. Es decir, el programa puede ser detenido despues de cada instruccion, despues de un número especificado de instrucciones o cuando una direccion especifica es localizada como un punto de ruptura (breakpoint). Siempre que el programa es suspendido, el programa debug/trace muestra el contenido de todos los registros del CPU.

Interpretes para ejecutar programas elaborados en lenguajes orientados a procedimientos (procedure-oriented language). Compiladores que convierten programas fuente en código de maquina ejecutable.

La principal diferencia entre el interpretador y el compilador es la manera en que cada uno de ellos considera las instrucciones fuente. El interpretador trabaja con una instruccion a la vez y la ejecuta si esta es ejecutable. El compilador traduce el programa entero en un modulo de lenguaje de maquina. El cual es cargado y ejecutado si el compilador no ha detectado instrucciones invalidas. El compilador realiza la traduccion solo una vez. El interprete traduce la instruccion cada vez que esta es encontrada mientras que el programa esta corriendo. La mayor ventaja de un interpretador consiste en que puede ser interactivo, es decir, el usuario introduce el programa y lo corre al momento. Si existen errores se corrigen las instrucciones que se encuentran en memoria. Un error de sintaxis detiene al interpretador, y produce un mensaje de error

proporcionando el número de línea donde el error fue detectado. Algunos intérpretes se colocan en modo de edición de línea y avisan al programador que puede realizar la corrección y correr nuevamente el programa. Esto reduce el tiempo de depuración. El compilador requiere de más pasos: indica los errores y su mensaje correspondiente al final de la traducción. El programador corrige dichas instrucciones usando un editor y recompila. Por cada error lógico posterior se requiere compilar, ligar y correr. Lo que lleva más tiempo en depuración. Sin embargo la versión de un programa compilado corre de 20 a 30 veces más rápido que la versión del programa fuente interpretada.

DESARROLLO DE SOFTWARE

El desarrollo de software puede ser dividido en varias etapas:

I.- Definición del problema: Esta etapa consiste en la definición formal del proyecto. Incluye la especificación de entradas/salidas, requerimientos de procesamiento, restricciones del sistema. La mayor parte de los diseños basados en microprocesadores involucran sistemas completos y no simples tareas, por lo que se requiere una considerable definición. Una pregunta básica es concerniente a la forma de entradas y salidas, es decir, que dispositivos se encuentran conectados al microprocesador y en que forma envían o reciben datos. Señales de control con las que los dispositivos de entrada y salida

indican la existencia de datos para lectura o escritura del puerto, longitud de la palabra, requerimientos de formato, relojes, strobes y protocolos. Debemos decidir precisamente que hacer con la información y en que orden deben realizarse las tareas. A menudo el orden de las operaciones es crítica, ya que las señales de entrada/salida deben ser enviadas o recibidas en una secuencia particular. El sistema puede tener restricciones de tiempo, tales como razones de transferencia de datos de los periféricos, tiempos de respuesta eléctricos o mecánicos, tiempos en que los datos deben de permanecer constantes, tiempos de colocación y retiro, así como, tiempos de habilitación y deshabilitación. Los requerimientos de memoria pueden limitar la longitud del programa o el área de datos en memoria y el tamaño de los buffers. Debemos de especificar el método de recuperación de secuencias incorrectas o señales erróneas. Errores comunes deben ser específicamente atendidos por el programa, otros errores requieran del reinicio del procedimiento de operación. Aun más, la definición del problema puede contener interacciones con otros programas o tablas, cuyos requerimientos deben ser considerados.

II.- Diseño del programa: En esta etapa se desarrolla el programa que ha de cubrir los requerimientos de la definición del problema. Utilizando alguna de las técnicas de programación como: programación modular, programación de arriba a abajo,

programación estructurada.

a) Programación modular: este es un método en el cual el programa se divide en pequeños módulos, los cuales pueden ser diseñados, codificados y probados separadamente con un mínimo de interacción. Los programas son casi siempre divididos en partes funcionales, en la programación de microprocesadores esto es particularmente útil, ya que los módulos pueden formar una librería de programas que pueden ser utilizados en trabajos posteriores. Este tipo de programación tiene ventajas obvias, porque limita el tamaño de los programas para ser probados y depurados. Entre sus desventajas se encuentra la realización de una etapa de interrelación, es decir, tiempo extra y memoria es necesaria para la transferencia de información de un módulo a otro, así como, la necesidad de dos niveles separados de prueba (modular y programa) y la escritura de programas que prueben los módulos.

b) Diseño de arriba-abajo: Un método en el que la tarea principal es definida en primera instancia en término de sub tareas generalizadas que en su momento son definidas completamente. El proceso continúa hacia abajo hasta que las sub tareas son definidas en una forma adecuada para ejecución por computadora.

El método opuesto es el diseño abajo-arriba en el cual todas

las sub tareas son codificadas y posteriormente integradas a partes mayores del diseño.

El metodo arriba-abajo se desarrolla del siguiente modo:

1. El programa principal que llama a los subprogramas es escrito y probado. Los programas indefinidos son reemplazados por subrutinas de estado que simulan que las subrutinas han sido llamadas para producir el resultado a un problema de prueba elegido. Esto es, estas produciran los mismos efectos que produciran las rutinas finales.

2. Cada subrutina es similarmente expandida, la prueba y depuracion pueden ocurrir a cada paso como cada una de ellas es reemplazada por un programa que realice su operacion correctamente.

3. El sistema entero es probado. Este metodo tiene la ventaja de que las pruebas e integracion ocurren conforme se desarrolla el trabajo y no al final. Tambien, este diseño combina el diseño, codificacion y etapas de prueba del desarrollo del programa.

La desventaja mayor de esta tecnica es que un error en la etapa inicial, puede ser de efectos catastróficos para el resto del proyecto.

c) Programacion estructurada: Este es un metodo en el que los

programas son escritos de acuerdo a formas definidas específicamente, esto es, solo ciertos tipos de lógica de programación son permitidos, ya que las formas involucradas serán encadenadas una con otra para desarrollar situaciones complejas. La programación estructurada algunas veces se refiere a métodos en los que secciones de programa siempre tienen una sola entrada y una sola salida.

Cualquier programa puede ser escrito utilizando las siguientes estructuras:

1. Una estructura secuencial en la que las instrucciones o programas son ejecutados consecutivamente en el orden escrito.

2. Una estructura condicional del tipo IF-THEN-ELSE, esto es: IF A THEN P1 ELSE P2 donde A es una expresión lógica y P1 y P2 son programas que pueden consistir de las tres estructuras permitidas.

3. Una estructura cíclica del tipo DO-WHILE, esto es: DO P WHILE A donde A es una expresión lógica y P es un programa que puede consistir de las tres estructuras permitidas.

Cada estructura tiene una sola entrada y una sola salida.

El diseñador de software puede utilizar una combinación de técnicas.

III.- Codificación: En esta etapa se efectúa la transcripción de las ideas del diseño del programa a instrucciones de computadora, estas instrucciones son propiamente el producto de la programación.

PROGRAMACION

Sugerencias para el estilo de programación:

a) Utilice nombres: en lugar de direcciones de memoria específicas, constantes, máscaras, números de dispositivos de entrada/salida o factores numéricos. Los nombres no solamente sugieren el propósito o significado de una dirección particular o datos, sino que también hace a los programas más fáciles de modificar.

b) Utilice nombres y etiquetas: que se identifiquen plenamente, puesto que pueden ser de gran ayuda en la documentación y mantenimiento, así como, en la depuración e integración. Los nombres deben ser tan simples y concisos como sea posible, esto evitará confusiones y ahorrará tiempo en el desarrollo del programa.

c) Coloque definiciones en grupos al inicio del programa; las cuales pueden ser fácilmente localizables, checadas y modificadas cuando sea necesario.

d) Mantenga nombres y etiquetas distintas (no parecidas) para evitar confusión.

e) Evite construcciones oscuras: el uso de expresiones complejas que dependen del orden en que las operaciones son realizadas o operadores que raramente son usados, y complejas instrucciones de condición.

f) Evite saltos siempre que sea posible: estos hacen que el programa sea difícil de seguir y probar.

g) Mantenga los módulos tan cortos como sea posible: divida módulos largos en secciones. Los módulos pueden ser implementados como macros o simplemente copiar la parte correspondiente para evitar un considerable número de llamadas (en los microprocesadores, tales llamadas requieren de tiempo extra y memoria y también pueden provocar una condición de sobreflujo limitada por la capacidad del stack para el almacenamiento de direcciones de retorno).

h) realice los módulos preferentemente de una manera general.

IV.- Depuración (Debugging): En esta etapa se descubren y corrigen los errores de programación. Por lo que la depuración es importante y es considerada una etapa en la que se invierte más tiempo en el desarrollo del programa.

Aquí se presentan dos consideraciones: verificación y validación. La verificación asegura que el programa realice el fin con el que el programador lo realizó. La validación asegura que el programa produzca los resultados correctos para un conjunto de datos de prueba.

V.- Pruebas: Esto consiste en la validación del programa, lo que asegura que el programa realice correctamente su función. Factores importantes que intervienen en este proceso son la selección de datos adecuados de prueba y el desarrollo de métodos de prueba.

Las siguientes son algunas reglas que pueden ayudar a la prueba del programa:

a) Haga el plan de prueba parte del diseño del programa: la prueba debe ser uno de los factores en la definición del problema, diseño del programa y etapa de codificación.

b) Cheque todos los casos triviales y especiales: tales casos pueden incluir entradas cero, no requerimiento de datos de las líneas de comunicación, señales especiales de alarma o advertencia y otras razones que pueden llevar a los errores más inesperados y misteriosos.

c) Seleccione datos de prueba sobre una base aleatoria: esto eliminará resultados condicionados.

d) Planee y documente las pruebas del programa como se hace con las pruebas del hardware. Obviamente las pruebas nunca pueden garantizar que no existan errores.

e) Utilice los valores máximos y mínimos de las variables como datos de prueba. Valores extremos son algunas veces fuente de errores especiales.

f) Utilice métodos estadísticos en la planeación y evaluación de pruebas complejas. Esto es para evaluar el significado de los resultados.

VI.- Documentación: En esta etapa se describe el programa para las personas que harán uso de él y se encargaran de mantenerlos, de manera que estos puedan ser extendidos para aplicaciones posteriores. Un programa debidamente documentado es fácilmente utilizado. Un programa con una estructura clara y nombres adecuadamente escogidos puede ser parcialmente explícito. Los comentarios deben explicar el propósito de las instrucciones de programa.

Las siguientes reglas deben ser tomadas en cuenta:

a) Los comentarios deben explicar el propósito de las secuencias de instrucciones no definir los códigos de operación.

b) Los comentarios deben ser tan claros como sea posible. Abreviaciones deben ser omitidas, aunque no es necesaria la explicación de la estructura completa.

c) Los comentarios deben localizarse en puntos importantes. Demasiados comentarios hacen al programa difícil de seguir.

d) Los comentarios deben estar relativamente cerca de las sentencias que describen. Y estos deben desarrollarse de acuerdo a un formato, puesto que las variaciones ocasionan confusiones.

e) Los comentarios deben de ser actualizados.

f) Los comentarios deben de ser breves

Los comentarios son una herramienta que los programadores deben de aprender a utilizar propiamente.

VII.- Mantenimiento: Concierne la actualización y corrección del programa de acuerdo a las condiciones de cambio y experiencia en el campo. Una adecuada prueba y apropiada documentación reducen significativamente este trabajo.

B. Extensión y rediseño: En esta etapa se contempla la extensión del programa para la solución de tareas o funciones no contempladas en la definición del problema. Obviamente los diseñadores siempre quieren aprovechar los programas

desarrollados para tareas previas. Cada una de las etapas del desarrollo de software afecta a las demas. El rediseño puede involucrar el sumarle nuevas características o adecuarse a nuevos requerimientos, tal rediseño debe de considerar las etapas de desarrollo de software.

Nota: Algunas reglas para lograr el incremento de velocidad son las siguientes:

- a) Concentrarse en los ciclos que se repiten mas constantemente.
- b) Trate de usar operaciones con registros cuando sea posible. Tales operaciones se realizan mas rapidamente, aunque requieren manipulacion e inicializacion extra.
- c) Trate las operaciones repetidas fuera de los ciclos.
- d) Utilice el modo de direccionamiento mas apropiado.
- e) Trate de eliminar los saltos.
- f) Repita una secuencia de instrucciones en lugar de utilizar subrutinas.
- g) Utilice el direccionamiento a traves del stack en lugar del direccionamiento directo para mover datos entre la memoria y los registros.

CRITERIO PARA LA EVALUACION DE PROGRAMAS

1. Realización: El criterio más importante para un programa, lo es su realización. Una estructura elegante, uso eficiente de tiempo y memoria, tiempo de diseño corto y completa documentación, no son válidas si el programa no funciona. El programa debe ser chequeado para ver que trabaja correctamente bajo condiciones de prueba que reflejen las condiciones de operación.

2. Velocidad: Un programa que ejecuta sus funciones rápidamente, puede realizar mayor trabajo que uno de menor velocidad. La velocidad puede determinar si el programa trabaja a los requerimientos de tiempo establecidos en la definición del problema.

3. Costo del hardware: Este es referido a la longitud del programa ya que cada chip de memoria necesitado le suma costo al sistema. Memoria adicional requiere de interconexiones adicionales, circuitería de decodificación, particularmente en aplicaciones en que el costo de un solo chip de memoria es significativo.

4. Tiempo y costo de programación: Estos factores en el desarrollo de software son muy importantes por lo que debe tenerse especial atención en las técnicas de programación que incrementen la productividad del programador.

5. Facilidad de uso: Un programa que es fácil de utilizar por otros programadores o usuarios será de mayor valor que aquellos que son relativamente difíciles de utilizar. El diseño y la documentación son etapas particularmente significativas en la determinación de si un programa es fácil de usar.

6. Tolerancia de error: Un programa que tolera errores es más fácil de mantener y usar que uno que no contempla esta posibilidad. Un programa debe ser diseñado para reaccionar de una manera razonable a errores cuya ocurrencia específica no sea prevista. El programa debe de tener al operador o al elemento del sistema pendiente de entradas erróneas o mal funcionamiento sin que se tenga que restablecer el sistema completo.

7. Ampliación: Un programa que puede ser extendido a la solución de otras tareas de las que fue especialmente diseñado es claramente superior a aquel que es utilizado para una tarea particular.

CARACTER DE LOS SISTEMAS OPERATIVOS

Los sistemas operativos son conocidos por el tipo de computadora para la cual fueron diseñados, o por el modo en que esta se encuentra equipada. Ejemplos de sistemas operativos son los siguientes: sistema operativo de disco, sistema operativo de cinta, sistema operativo de tiempo compartido, etc. Su carácter refleja en parte la manera deseada de operación de la

computadora, el caracter de la computadora y los tipos de trabajos que ejecuta. El sistema operativo consiste de programas que asisten al usuario en la mejor utilizacion de la computadora, una preparacion de programas mas facil y un manejo menos dificil de la disponibilidad de tiempo de la computadora. Los sistemas operativos exigen una considerable demanda de el hardware disponible'.

COMPONENTES DE LOS SISTEMAS OPERATIVOS

El componente basico del sistema operativo es el programa supervisor o Monitor. Este coordina las actividades de los otros componentes del sistema operativo y controla su operacion.

El segundo componente es el programa que organiza el trabajo (scheduler). Este programa establece las claves de los trabajos a realizar y los mantiene como el estado de varios cambios en los trabajos durante el curso, espera o ejecucion de los mismos. Toma en cuenta prioridades y condiciones de las unidades funcionales de la computadora reportadas por otros componentes del sistema operativo.

El tercer componente es el despachador (dispatcher), algunas veces llamado ejecutor. Este programa toma los trabajos identificados por el organizador (scheduler) como calificados para ejecucion y arregla la operacion siguiente necesitada en el proceso de terminacion de un trabajo. Es responsable de cargar

los programas, ligarlos, coordinar las estructuras de datos del programa, iniciar ejecución de programa y terminarla.

El cuarto componente es el controlador de interrupciones. Durante la ejecución de un programa, las interrupciones pueden suscitarse por una variedad de razones y pueden ocurrir, aun dentro de la ejecución de componentes del mismo sistema operativo. El controlador de interrupciones debe tener en cuenta el estado de los otros componentes del sistema, así como, la naturaleza de las operaciones actuales.

El quinto componente es el alimentador del periférico (peripheral driver). El sistema operativo debe llamar a las unidades de los periféricos de la computadora para satisfacer los requerimientos de acción especificados por el programa en ejecución.

El sexto componente es el localizador de almacenamiento (storage allocator). Este componente controla el uso de los dispositivos de almacenamiento. Esta es una de las partes más complejas del sistema operativo y su funcionamiento es crítico para la operación de los elementos del sistema.

El séptimo componente es el comunicador (communicator). La comunicación entre los componentes del sistema operativo es a través del uso de tablas, claves. El trabajo del comunicador es mantener estas tablas y claves para que cada componente que haga

referencia a ellas, las encuentre en el formato predescrito, reflejando propiamente el estado actual de los trabajos.

El octavo componente es el controlador de librerías. Este programa realiza inserciones en la librería de los programas y mantiene un directorio para localizar conjuntos en la librería cuando sean solicitados.

El noveno componente es la misma librería de programas. La librería consiste de programas para los otros ocho componentes e incluye programas para la translación de lenguajes. La librería es creada por el proceso de generación del sistema operativo.

REQUERIMIENTOS DE HARDWARE DEL SISTEMA OPERATIVO

El primero es la capacidad de almacenamiento (buffering).

El segundo es la capacidad de interrupción o algunos medios para la señalización de estado de hardware.

El tercero es que tenga al menos dos grupos diferentes de equipos de entrada.

El cuarto es que tenga al menos dos dispositivos diferentes de almacenamiento, uno de los cuales no se encuentre a nivel interno.

El quinto es la capacidad de protección del almacenamiento.

DISEÑO DEL SISTEMA INTERFACE

DISEÑO DEL SISTEMA INTERFACE

A continuación se presentan las características del Sistema Interface Hewlett-Packard Interface Bus (HP-IB), las cuales fueron consideradas para la realización del diseño que conforma esta tesis.

DESCRIPCION FISICA

La Interface Programable en base al STANDARD INTERNACIONAL HEWLETT-PACKARD INTERFACE BUS (HP-IB), el cual es una implementación del STANDARD IEEE 488-1978 y suplemento 488-1978A-1980, consiste de un Circuito Impreso ensamblado con los correspondientes Circuitos Integrados, un Cable de Interface y un Manual de Referencia.

DESCRIPCION FUNCIONAL

Esta Tarjeta se encuentra Bajo Control de SOFTWARE en TODO MOMENTO.

ESPECIFICACIONES

CARACTERISTICAS ELECTRICAS

LINEAS DEL HP-IB BUS: El bus consiste de las siguientes 16 líneas:

DIO1 Data Input/Output 1

DISEÑO DE UN SISTEMA INTERFACE
EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-IB).

TESIS

DIO2	Data Input/Output 2
DIO3	Data Input/Output 3
DIO4	Data Input/Output 4
DIO5	Data Input Output 5
DIO6	Data Input/Output 6
DIO7	Data Input Output 7
DIO8	Data Input/Output 8
DAV	Data Valid
NRFD	Not Ready for Data
NDAC	Not Data Accepted
IFC	Interface Clear
ATN	Attention
SRQ	Service Request
REN	Remote Enable
EOI	End or Identify

Niveles Logicos: High > 2.4 Volts
Low < 0.4 Volts

Todas las senales son activas en estado bajo excepto NRFD y NDAC.

Drivers de linea: Cada una de las lineas presenta uno de los siguientes estados tipicos a la salida.:

Tipo: Tri-state, Open Collector

Output Voltage Low State: < 0.4 V a 48 mA

Output Voltage High State: 2.5 V a -5.2 mA

Receivers de línea: Cada una de las líneas presenta las siguientes características típicas a la entrada.:

Tipo: Schmitt Trigger

Input current Low State: -1.3 mA a 0.4 V

Input current High State: 0.7 mA a 5.5 V

Maxima longitud de cable:

Operacion Standard: 2 metros por dispositivo con una longitud maxima de 20 metros con un tiempo de colocacion de 500 ns.

Operacion de alta velocidad: 2 metros por dispositivo con una longitud maxima de 15 metros y un tiempo de colocacion menor de 350 ns.

Temperatura de Operacion: 0 a +55 Grados Celsius.

CAPACIDAD DE LA TARJETA HP-IB

HP-IB Permite conectar de uno a catorce dispositivos compatibles. Los datos son transferidos bidireccionalmente a traves de HP-IB en bytes de 8 bits. Los datos pueden ser transferidos de un dispositivo a la computadora y a otros dispositivos simultaneamente o de la computadora a uno o mas dispositivos simultaneamente o de un dispositivo a otros dispositivos bajo la direccion de la computadora.

Algunas de las características de HP-IB son rigurosas, otras son opcionales. Por ejemplo todos los dispositivos deben ser capaces de ser direccionados, pero pueden o no ser operados por control remoto. Un sistema puede tener operando bajo control remoto varios dispositivos, mientras que otros responden a sus controles locales. Las mismas guías de todos los conectores HP-IB de todos los dispositivos están conectadas en paralelo, es decir, formando una Red de Comunicación en Paralelo. Esto permite el flujo de información en cualquier dirección sobre el BUS y permite a un dispositivo comunicarse con otro sin tener que hacerlo a través de una Unidad Central de Control.

FUNCIONES PROPIAS DE HP-IB

La tarjeta de la interface es diseñada para proporcionar las siguientes funciones definidas por el STANDARD IEEE 488-1978.

(La Interface HP-IB es una implementación del STANDARD 488-1978.).

FUNCIONES DEL CONTROLADOR

- | | |
|----|---|
| C1 | Controlador del Sistema |
| C2 | Envía Interface Clear y toma el mando |
| C3 | Envía Remote Enable |
| C4 | Responde a Service Request |
| C5 | Envía mensajes de interface, recibe el control, |

pasa el control, recupera el control, parallel
poll, toma el control sincronicamente.

FUNCIONES DE LOS DISPOSITIVOS CONTROLADOS

SR1	Service Request
RL2	Remote Local
SP1	Serial Poll
DC1	Device Clear
DT1	Device Trigger

Las siguientes funciones de utileria permiten las funciones
arriba mencionadas.

SH1	Source Handshake
AH1	Acceptor Handshake
T1	Basic Talker, Serial Poll, Talk Only
TE1	Basic Extended Talker, Serial Poll, Talk Only
L1	Basic Listener, Listen Only Mode
LE1	Basic Extended Listener, Listen Only mode

DIRECCIONAMIENTO DE PROTOCOLO Emittiendo/Recibiendo

Una tecnica de direccionamiento es usada para determinar que
dispositivo Emite y que dispositivos Reciben. Los datos son
enviados en un formato bit parallel byte serial de un dispositivo
a otro usando una tecnica de protocolo inter-asegurada.

"THREE WIRE HANDSHAKE TECHNIQUE"

Esta técnica garantiza que el emisor no retire el dato antes de que el receptor haya terminado de utilizarlo. Esto también asegura que el dato no se pierda cuando los dispositivos operen a diferentes velocidades en el mismo bus.

FUNCIONES DE LOS DISPOSITIVOS EN LA INTERFACE HP-IB

Los dispositivos conectados al bus deben ser direccionados por el controlador antes de que puedan funcionar en una o más de las siguientes formas:

TALKER - Cualquier dispositivo capaz de mandar o transmitir información por el bus. Solo puede haber un emisor (TALKER) a la vez en el bus.

LISTENER - Cualquier dispositivo capaz de recibir o aceptar información. Puede haber hasta 14 receptores (LISTENERS) al mismo tiempo en el bus.

TALKER-LISTENER - Cualquier dispositivo que tenga la capacidad tanto de transmitir como de recibir información de acuerdo a la definición previa del mismo, como receptor o emisor.

CONTROLLER - Cualquier dispositivo que haya sido programado para adquirir la responsabilidad del manejo del flujo de información entre los dispositivos conectados al bus, es capaz de direccionar uno de los dispositivos como emisor (TALKER) y uno o varios de los restantes como receptores (LISTENERS). La interface

HP-IB permite al sistema tener mas de un controlador, pero solo un controlador puede estar activo a la vez. (El controlador a cargo, debe ser el Controlador del Sistema).

SYSTEM CONTROLLER - El Diseñador del Sistema debe designar un dispositivo como el Controlador del Sistema al tiempo que el Sistema es configurado. El dispositivo realiza todas las funciones de un controlador, ademas tiene la habilidad de tomar el control absoluto de la Interface HP-IB para programar los modos de operacion de los dispositivos (colocando IFC y REN), tomando y procesando datos, etc.

LINEAS DEL BUS DE LA INTERFACE HP-IB

La estructura de la Interface HP-IB consiste de 16 lineas de senales. Existen 8 lineas adicionales: una tierra, una linea de blindaje y 6 lineas comunes mezcladas con 6 de las lineas de senales. A todas las 16 lineas se les ha asignado nombres y mnemonicos que describen el mensaje correspondiente a cada senal.

Existen 3 tipos de lineas: 8 de DATOS, 3 de Transferencia (HANDSHAKE), y 5 de CONTROL (manejo).

LINEAS DE CONTROL

Todos los dispositivos conectados al bus incluyendo al CONTROLADOR deben de ajustarse a estas descripciones.

Las 5 líneas de control son usadas para manejar el flujo de información sobre las líneas de datos y transferencia. Estas comunican el control y el estado de la información entre el controlador activo y los dispositivos conectados al bus. Todos los dispositivos utilizan ATN e IFC. Un dispositivo puede o no utilizar REN, SRQ y EOI.

ATN (Attention) Es emitida por el controlador activo para colocar al bus en cualquiera de los modos COMMAND (low) o DATA (high). Todos los demás dispositivos deben monitorear ATN en todo momento. Cuando el controlador coloca ATN en estado bajo, el bus opera en modo comando (COMMAND MODE). El propósito principal del Modo Comando es permitir al controlador enviar comandos de interface y direcciones a aquellos dispositivos que se comunicaran cuando el bus sea puesto en modo de datos (DATA MODE). También el controlador puede enviar Comandos Universales mientras se encuentra en Modo Comando. Cuando el controlador coloca ATN en estado alto, el bus opera en modo de datos (DATA MODE). El dispositivo que fue direccionado como emisor y aquellos que fueron direccionados como receptores, ahora se comunican por el bus de datos.

ATN puede ser colocada en estado bajo o alto por el controlador en cualquier momento, mas es usualmente al final del ciclo de transferencia (HANDSHAKE) para que no se pierda el dato.

IFC (Interface Clear) Es utilizado por el controlador para Inicializar el Bus. Solo el Controlador del Sistema puede enviar esta señal, la cual ha de ser monitoreada por todos los demás dispositivos en la red. Cuando el Controlador del Sistema coloca IFC en estado bajo por lo menos 100 us, lo siguiente se presenta: todos los dispositivos de la red son detenidos, el modo de serial poll es deshabilitado y el control es retomado por el controlador. Cuando IFC se encuentra en estado alto, no tiene efecto sobre la operación del bus. El Controlador del Sistema puede colocar IFC en estado bajo en cualquier momento.

REN (Remote Enable) es una de las condiciones para que los dispositivos operen bajo Control Remoto. Solo los dispositivos capaces de esta operación utilizan REN y la monitorean en todo momento. Los dispositivos que no utilizan REN terminan la línea en una carga resistiva. Solo el Controlador del Sistema puede enviar esta señal y cambiar su estado en cualquier momento.

SRQ (Service Request) Es enviada a estado bajo por un dispositivo para indicar que requiere la atención del Controlador. Esta señal puede ser colocada en estado bajo por cualquier dispositivo en todo momento, excepto cuando IFC se encuentra en estado bajo. Solo el Controlador sensa SRQ. Algunos dispositivos no utilizan esta línea y la terminan en una carga resistiva.

EOI (End or Identify) Puede ser utilizada para indicar la terminación de transferencia de un dispositivo. Cuando el bus se encuentra en Modo de Datos (AIN high) el emisor direccionado puede indicar el fin de datos, colocando EOI en estado bajo al mismo tiempo que coloca el último byte en las líneas de datos.

LINEAS DE DATOS (DIO1-DIO8)

Las líneas de datos son usadas para comunicar toda la información incluyendo entrada, salida, códigos de programa, direcciones, control y estado de información entre dispositivos conectados al bus. El dato es enviado byte a la vez (i.e. byte serial y bit parallel) bajo control de las líneas de transferencia.

LINEAS DE TRANSFERENCIA

Las Líneas de Transferencia son utilizadas para ejecutar la Transferencia de cada byte de información en el bus de datos. Todos los dispositivos utilizan estas líneas y emplean una técnica de transferencia inter-asegurada para enviar la información. Esto permite una transferencia de datos asincrónica sin restricciones de tiempo para ninguno de los dispositivos conectados al bus. La Transferencia de cada byte es llevada a cabo a la velocidad del dispositivo más lento. Las 3 líneas de Transferencia, son las siguientes: NRFD, NDAC y DAV.

NRFD (Not Ready for Data) es la línea de transferencia que indica que todos los receptores están listos para aceptar información en las líneas de datos. Es enviada por todos los receptores. Es sensada por el emisor, el controlador cuando ATN está en estado bajo y el dispositivo direccionado para emitir cuando ATN está en estado alto.

Cuando NRFD se encuentra en estado alto, todos los receptores se encuentran incondicionalmente listos para recibir información. El emisor puede en su momento colocar un byte de información en las líneas de datos y colocar DAV en estado bajo. Cuando NRFD se encuentra en estado bajo, uno o más receptores no están listos para recibir datos.

Cuando el controlador coloca ATN en estado bajo, todos los dispositivos deben colocar NRFD en su estado alto dentro de 200 ns. (i.e. si un dispositivo está listo para recibir información entonces coloca NRFD en su estado alto y si no está listo coloca NRFD en estado bajo).

Un receptor no debe de colocar NRFD en estado bajo hasta que sensa DAV en estado bajo. Puede ser antes o al mismo tiempo que coloca NDAC en estado alto. También, no debe de regresar NRFD a su estado alto hasta que sensa DAV en estado alto y puede ser después o al mismo tiempo que coloca NDAC en estado bajo.

NDAC (Not Data Accepted) Es la línea de transferencia que indica la aceptación de información en las líneas de datos. NDAC es enviada por todos los receptores. Esto es, todos los dispositivos cuando ATN se encuentra en estado bajo. Es sensada por el emisor y el controlador cuando ATN se encuentra en estado bajo y por el dispositivo direccionado para recibir cuando ATN se encuentra en estado alto.

Cuando NDAC está en estado alto, todos los receptores han aceptado incondicionalmente el byte de información que se encuentra en las líneas de datos y el cual ya no es necesario.

El emisor puede colocar a su tiempo DAV en estado alto, remover ese byte de información y continuar. Cuando NDAC está en estado bajo, uno o varios receptores no han aceptado la información de las líneas de datos.

Cuando el controlador coloca ATN en estado bajo, cada dispositivo debe de colocar NDAC en estado alto dentro de 200 ns.

Un receptor no debe de colocar NDAC en estado bajo hasta que sense DAV en estado alto. Puede ser antes o al mismo tiempo que coloca NRFD en estado alto. No debe de recolocar NDAC en estado alto hasta que sense DAV en estado bajo y puede ser después o al mismo tiempo que coloca NRFD en estado bajo.

DAV (Data Valid) Es la línea de transferencia que indica la validez de la información en las líneas de datos. DAV es enviada por los emisores, el controlador cuando ATN está en estado bajo y el dispositivo direccionado para emitir cuando ATN está en estado alto. ATN es sensada por los receptores y por todos los dispositivos si ATN está en estado bajo, y por todos los dispositivos direccionados para recibir cuando ATN está en estado alto.

Cuando DAV está en estado bajo, los estados de las líneas de datos DIO1 a DIO8 son incondicionalmente válidos y pueden ser aceptados por todos los receptores a su propio tiempo. DAV puede solamente ser enviada a estado bajo si NRFD e IFC se encuentran en estado alto. Cuando DAV está en estado alto, la información en las líneas de datos no es válida. DAV no puede ser colocada en estado alto, a menos que NDAC esté en estado alto y NRFD en estado bajo.

TRANSFERENCIA DE DATOS

La transferencia de datos sobre el bus es asíncrona y por lo tanto, no establece restricciones en las razones de transferencia de datos de los dispositivos conectados al bus. Los tiempos y niveles requeridos para transferir un byte de datos sobre las líneas de datos se muestra en la figura 11. La transferencia se encuentra bajo el control de tres líneas DAV, NRFD y NDAC.

SEQUENTIAL REQUIREMENTS OF THE THREE WIRE TRANSFER

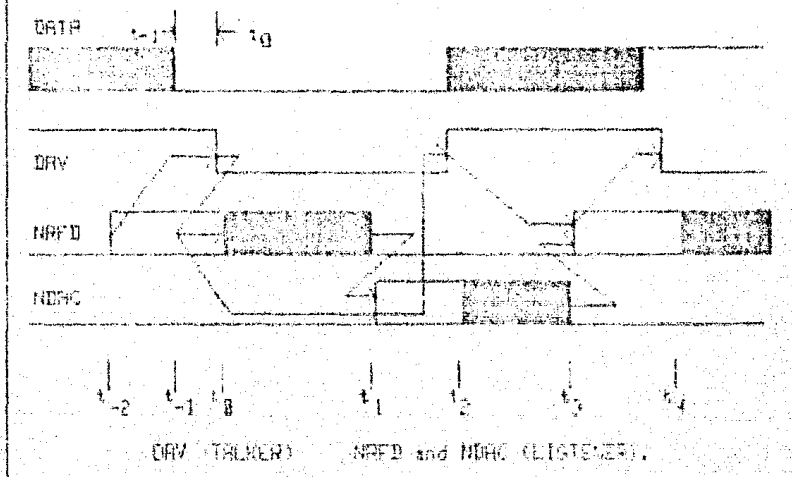


FIGURA II

EVENTOS

- t_{-2} : El receptor se encuentra listo para aceptar datos.
- t_{-1} : El emisor ha colocado el dato en el bus de datos.
- t_0 : Indicación de que el dato es valido.
- t_1 : El receptor ha aceptado el dato y no requiere que se mantenga aun valido el mismo.
- t_2 : El emisor indica que el dato ya no es valido y en este momento puede cambiarse.
- t_3 : El receptor indica que esta listo para un nuevo dato.
- t_4 : Se inicia un nuevo ciclo (equivalente a t_{-2}).
- t_{-1} to t_0 : Tiempo en que el dato es puesto en el bus de datos antes de que DAV es puesto en estado bajo.

NOTA: Por propósitos de ilustración se ha utilizado una sola línea para representar al bus de datos. Las curvas de seguimiento indican la secuencia de las señales de transferencia (HANDSHAKE).

El emisor (TALKER) envía el dato y DAV (Data Valid), el receptor (LISTENER) envía NRPD (Not Ready for Data) y NDAC (Not Data Accepted).

La transferencia de un byte de datos es iniciada por todos los receptores, indicando que están listos para recibir el dato colocando NRPD en estado alto. Cuando el emisor reconoce NRPD en estado alto y ha colocado el dato válido en las líneas de datos entonces coloca DAV en estado bajo. Cuando los receptores sientan DAV en estado bajo y han terminado de utilizar el dato, colocan NDAC en estado alto. Ya que todos los dispositivos en el bus tienen sus líneas correspondientes conectadas en paralelo (e.g. NRPD) todos los receptores deben estar en estado alto para que esa línea sea colocada en estado alto. Esta condición (Wired-AND) permite al emisor reconocer cuando el receptor más lento ha aceptado un byte de datos y se encuentra listo para el siguiente.

La figura II. también muestra el tiempo de transición del estado no activo de estas líneas. Un receptor puede colocar NRPD tan pronto como reconozca que DAV ha sido colocada en estado bajo y debe ser antes o al mismo tiempo en que coloca NDAC en estado alto. El emisor puede recolocar DAV a su estado alto, después de que detecta que NDAC está en estado alto.

Un receptor puede colocar NDAC en estado bajo tan pronto como reconozca que DAV esta en estado alto y debe ser antes o al mismo tiempo que coloca NRFD en su estado alto.

CARACTERISTICAS ELECTRICAS DE LA INTERFACE HP-IB

Todas las lineas del bus fueron disenadas para ser compatibles con circuitos integrados TTL .

Cada par de 'driver/receiver' forma una interface completa entre el bus y el dispositivo. Cualquier driver o receiver de cada canal es habilitado por su correspondiente senal habilitadora con la salida deshabilitada del par forzada a un estado de alta impedancia. Una opcion adicional permite al driver ser operado en colector abierto.

Los receptores tienen entradas Schmitt Trigger para proveer el margen de ruido y que la carga de entrada se ajuste al standard del bus.

En la tabla de la siguiente pagina se resume la relacion de la senal ATN y las lineas de transferencia DAV, NRFD y NDAC.

DISENO DE UN SISTEMA INTERFACE (ES15)
 EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-IB).

RELACION DE ATN Y LAS LINEAS DE TRANSFERENCIA NRFD, NDAC Y DAV.

		NRFD		NDAC		DAV	
MODE	ATN	LOW	HIGH	LOW	HIGH	LOW	HIGH
C O M M A N D	L	Una o mas unidades no estan listas para el dato.	Todas las unidades estan listas para el dato.	Una o mas unidades no han aceptado el dato.	Todas las unidades han aceptado el dato.	El controlador tiene un dato valido en las lineas de datos.	El dato del controlador no es valido.
	W	1.- enviada por todas las unidades excepto el monitor. 2.- sensada por el controlador. 3.- todas las unidades colocan NRFD y NDAC en estado valido en menos de 200 ns. despues que ATN va a estado bajo.				1.- enviada por el controlador. 2.- sensada por los receptores. 3.- vea DAV arriba para los tiempos.	
D A T A	H	Uno o mas receptores no estan listos para el dato.	Todos los receptores estan listos para el dato.	Uno o mas receptores no han aceptado el dato.	Todos los receptores han aceptado el dato.	El emisor tiene un dato valido en las lineas de datos.	El dato del emisor direccionado no es valida.
	G	1.- enviada por todas las unidades direccionadas para recibir. 2.- sensada por la unidad direccionada para emitir. 3.- todos los receptores direccionados colocan NRFD y NDAC en estado valido en menos de 200 ns despues que ATN va a estado alto.				1.- enviada por el dispositivo direccionado para emitir. 2.- sensada por todos los dispositivos direccionados para recibir. 3.- ver DAV para los tiempos.	

DISEÑO DEL HARDWARE

Una vez conocidas las características del sistema Interface que ha servido como base para la realización de este proyecto, procederemos a definir los elementos que lo conforman, es decir, establecer las características de los mismos.

Consideraciones preliminares, algunas definiciones:

BUS - Trayectoria sobre la cual información digital es transferida de cualquiera de varios Emisores a cualquiera de varios Receptores. Solo una transferencia de información puede realizarse a la vez, todos los otros Emisores conectados al bus deben estar deshabilitados.

TO BUS - Interconectar varios dispositivos digitales en el que cualquiera de ellos envía o recibe señales digitales por una serie de trayectorias digitales comunes llamada BUS, sobre la cual la información entre tales dispositivos es transferida.

RECEIVER - Un dispositivo que acepta información de un transmisor o fuente de información.

TRANSMITTER - Un dispositivo que provee información que es en última instancia aceptada por el receptor.

POSITIVE LOGIC - Una forma de logica en la que el voltaje mas positivo representa un 1 logico y el voltaje mas negativo representa un 0 logico.

NEGATIVE LOGIC - Una forma de logica en la que el voltaje mas positivo representa un 0 logico y el voltaje mas negativo representa un 1 logico.

POSITIVE VS. NEGATIVE LOGIC - Los dispositivos (circuitos integrados) son los mismos, lo que cambia es nuestra designacion para la tierra y alimentacion para los estados de entrada y salida.

Basicamente el proposito de un BUS es minimizar el numero de interconexiones y circuitos integrados requeridos para transmitir informacion entre dispositivos digitales, asi como, alambrado. BUSES son comunmente utilizados en computadoras, donde existe la necesidad de transferir informacion entre una Unidad Central de Proceso (CPU) y elementos aritmeticos, memorias de acceso directo (RAMs), memorias de solo lectura (ROMs), memorias programables de solo lectura (PROMs) y una variedad de dispositivos de entrada y/o salida (I/O), como impresoras de linea, CRTs, unidades de disco, unidades de cinta y otras microcomputadoras.

COMPUERTAS DE TERCER ESTADO (Tri-State).

Se le conoce como la logica de semiconductores en la que se pueden presentar tres posibles estados a la salida de la compuerta.

Estos son los siguientes:

- (1) Un estado logico 0
- (2) Un estado logico 1
- (3) Un estado en el que la salida se encuentra en efecto desconectada del resto del circuito y no tiene influencia sobre el.

El Tercer Estado es el que hace la logica de tercer estado unica.

Todos los dispositivos de logica de Tercer Estado tienen una senal de entrada adicional llamada entrada habilitadora de compuerta (enable/disable gating input), la cual permite al dispositivo logico comportarse normalmente o desconectar la salida de la compuerta del circuito.

Cuando el dispositivo logico es habilitado, la salida del mismo puede ser un 0 logico o un 1 logico, pero no los dos. Cuando el dispositivo logico es deshabilitado, entonces la salida es desconectada del resto del circuito y en efecto, el dispositivo logico se encuentra desconectado del resto del circuito.

La entrada habilitadora de compuerta acepta un 0 logico o un 1 logico y bien puede habilitar o deshabilitar el dispositivo logico. Algunos dispositivos de Tercer Estado son habilitados con un 0 logico, otros con un 1 logico.

Tal vez el dispositivo mas importante de tercer estado es el buffer de tercer estado, aunque es posible manufacturar AND, NAND, OR, NOR de tercer estado, asi como otros dispositivos. Los dispositivos de Tercer Estado han llegado a ser muy utilizados debido a que presentan poco ruido y velocidades considerablemente mayores que los dispositivos de colector abierto (Open Collector), quienes tambien son utilizados para interconectar varios dispositivos digitales.

Es posible interconectar hasta 128 dispositivos de tercer estado a temperatura de operacion.

TABLA DE VERDAD PARA UNA COMPUERTA DE TERCER ESTADO

Buffer Line Driver/Line Receiver

S/R	A	Output
0	0	0
0	1	1
1	0	Z
1	1	Z

NOTA: La compuerta es habilitada con un cero logico.
 Z Representa el Tercer Estado (Alta Impedancia).

Para el diseno se utilizan Buffers Drivers/Receivers de la familia TTL.

COMPUERTAS DE COLECTOR ABIERTO (Open Collector Output).

Así se le conoce a la salida de un dispositivo de circuito integrado en el que la resistencia de carga (pull-up) en el transistor de la etapa de salida no existe, y debe ser proveída por el usuario para que el circuito sea completo.

Debido a que la resistencia de carga de todos los dispositivos de Salida de Colector Abierto no existe, es posible conectar todas estas salidas a un mismo punto y utilizar una sola resistencia de carga.

TABLA DE VERDAD PARA UNA COMPUERTA DE COLECTOR ABIERTO

Inverter Buffer/Driver with Open Collector

A	B	C	Output
0	0	0	1
Todos los demás estados			0

NOTA: La interconexión de las salidas de esta compuerta es conocida como Wired-Nor.

Para el diseño se utilizan Inverters Buffers de Colector Abierto de la familia TTL.

CÁLCULO DE LA RESISTENCIA DE CARGA DE LAS COMPUERTAS CUYA SALIDA ES DE COLECTOR ABIERTO

Consideraciones:

VCC = 5	Volts.
Vohmin = 2.4	Volts.
Volmax = 0.4	Volts.

$I_{ih} = 40E-6$ Amps.
 $I_{oh} = 250E-6$ Amps.
 $I_{ol} = 16E-3$ Amps.
 $I_{il} = -1.6E-3$ Amps.
N ----> Numero de cargas
M ----> Numero de compuertas de Colector
Abierto

$$V_{rmax} = VCC - V_{ohmin} \text{ [V]}$$

$$I_{ri} = N * I_{ih} + M * I_{oh} \text{ [A]}$$

$$R_{imax} = V_{rmax} / I_{ri} \text{ [ohms]}$$

$$V_{ri} = VCC - V_{olmax} \text{ [V]}$$

$$I_{ri} = I_{ol} + N * I_{il} \text{ [A]}$$

$$R_{imin} = V_{ri} / I_{ri}$$

A continuación se muestran las características de los circuitos integrados que intervienen en el diseño del sistema interface.

Z8400

Z80 CPU UNIDAD CENTRAL DE PROCESO (Central Processing Unit)

El CPU es un circuito integrado de larga escala que ha revolucionado la tecnología de las computadoras. Este realiza las decisiones y controla las funciones necesarias para la operación del sistema, es decir, es el corazón del sistema.

CARACTERÍSTICAS:

* El grupo de instrucciones consta de 158 instrucciones, Las 78 instrucciones del 8080A son incluidas como un subgrupo; la compatibilidad del software con dicho procesador es mantenida.

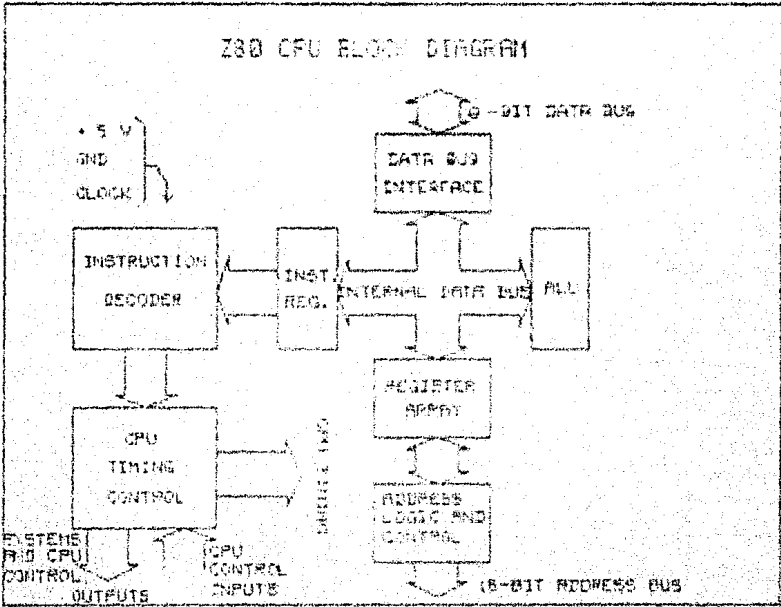


FIGURA III DIAGRAMA DE BLOQUES DE LA UNIDAD CENTRAL DE PROCESO Z80 CPU.

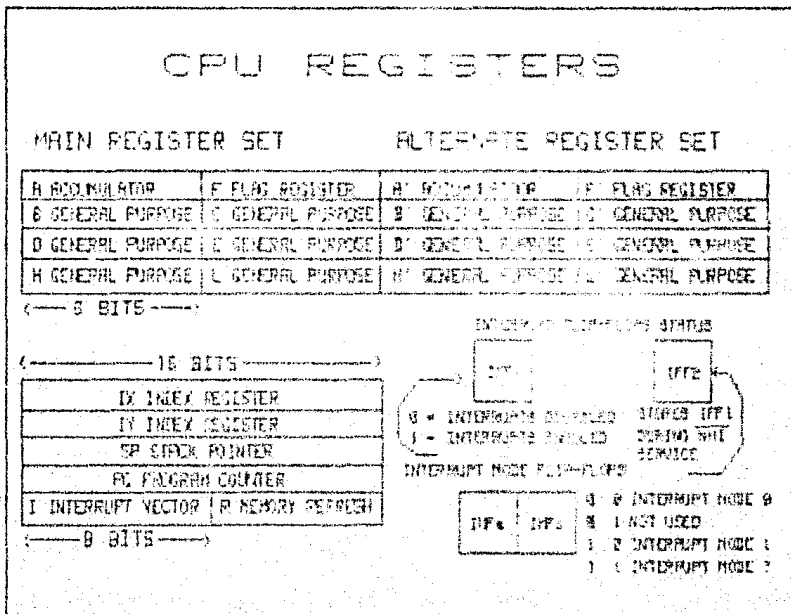


FIGURA IV REGISTROS DEL PROCESADOR Z80 CPU.

* Relojes de 6 Mhz, 4 Mhz and 2.5 Mhz para el Z80B, Z80A y Z80 CPU proveen una rapida ejecucion de instruccion y consecuentemente una alta razon de transferencia de datos.

* El amplio grupo de instrucciones incluye operaciones del tipo bit, byte. La transferencia de bloques y busqueda en bloques con direccionamiento indexado y relativo, resulta en la mas poderosa capacidad de manejo de datos en la industria de las microcomputadoras.

* Los microprocesadores Z80 y la familia asociada de controladores de perifericos son ligados por medio de un sistema de interrupcion via un vector.

El sistema puede ser implementado de manera que opere en 'daisy-chained' para permitir un esquema de prioridad de interrupcion. Pequena es la logica, si alguna logica adicional es requerida para el encadenamiento.

* Proporciona grupos de registros de proposito general y banderas alternos, facilitando el diseno y operacion de los programas del sistema. Procesamiento de interrupcion de nivel sencillo. Dos registros de indice de 16 bits facilitan el proceso de tablas y arreglos.

* Tiene tres modos de procesamiento de interrupcion de alta velocidad.

- * Contador de refresco de memoria dinamica integrado.

DESCRIPCION GENERAL

Los Z80, Z80A y Z80B CPUs son procesadores en un solo chip que pertenecen a la tercera generacion con poder computacional excepcional. Los registros internos contienen 208 bits de memoria read/write accesibles al programador. Estos registros incluyen dos grupos de seis registros de proposito general, los cuales pueden ser utilizados individualmente como registros de 8 bits o como registros pares de 16 bits.

Un grupo de instrucciones 'Exchange' permiten hacer uso de los registros principales como de los alternos. El Z80 tambien contiene un Apuntador de Stack, un contador de programa, dos registros indice, un registro contador de refresco y un registro de interrupcion. El CPU es facil de incorporar a un sistema, ya que solo requiere una fuente de voltaje de + 5 V. Todas las senales de salida son completamente decodificadas y checadas para controlar circuitos perifericos, memoria standard, y es respaldado por una amplia familia de controladores de perifericos.

OPERACION GENERAL DE INTERRUPCIONES

El CPU acepta dos senales de interrupcion: NMI y INT. La NMI es una interrupcion no mascarable y tiene la prioridad mas alta. La

INT es una interrupcion de menor prioridad y que requiere que la interrupcion sea habilitada a traves de software para que esta pueda operar.

Cualquiera de las lineas de interrupcion puede ser conectada a multiples dispositivos perifericos en un configuracion wired-OR. El Z80 tiene un modo de respuesta unico para la interrupcion no mascarable.

La interrupcion mascarable INT tiene tres modos de respuesta programables y son los siguientes:

Mode 0 - compatible con el microprocesador 8080.

Mode 1 - servicio de interrupcion a perifericos para uso con sistemas distintos al 8080/Z80.

Mode 2 - esquema de interrupcion via un vector, generalmente usado en interrupciones encadenadas 'daisy chain', entre dispositivos perifericos de la familia Z80 y el Z80 CPU.

El CPU atiende interrupciones, muestreando las senales NMI e INT en el ultimo filo de ascenso de la senal de reloj de una instruccion. El procesamiento posterior de la interrupcion depende del tipo de interrupcion que fue detectada.

INTERRUPCION NO MASCARABLE NMI

Este tipo de interrupcion no puede ser deshabilitada por

control de programa y por lo tanto es aceptada en todo momento por el CPU. NMI es generalmente reservada para tipo de interrupciones de mayor prioridad. Después del reconocimiento de la señal NMI (si BUSREQ no está activo), el CPU salta a la dirección de reinicio 0066H. Donde el software iniciado en esta localidad contiene la rutina de servicio de interrupción.

INTERRUPCIÓN MASCARABLE INT

Sin tomar en cuenta el modo de interrupción asignado por el usuario, la respuesta del Z80 a una señal de entrada de interrupción sigue un ciclo de tiempo común. Después de que la interrupción ha sido detectada por el CPU (si BUSREQ no está activa e interrupciones son permitidas) un ciclo de procesamiento de interrupción especial se inicia. Este es un ciclo fetch especial (M1) en el que IORQ es activa en lugar de MREQ como en un ciclo normal M1. Además este ciclo especial M1 es automáticamente extendido por dos estados de espera WAIT para proporcionar el tiempo requerido para reconocer el requerimiento de interrupción y colocar el vector de interrupción en el bus.

MODO DE INTERRUPCIÓN 0

En este modo el dispositivo que interrumpe coloca una instrucción en el bus de datos, la cual es ejecutada por el CPU. Esta es generalmente una instrucción de reinicio la cual realizará un salto incondicional a una de las 8 localidades de

reinicio en la pagina cero de la memoria.

MODO DE INTERRUPCION 1

La operacion de este modo de interrupcion es muy similar a la de NMI. La principal diferencia es que el modo de interrupcion 1 tiene una direccion de vector de 003BH unicamente.

MODO DE INTERRUPCION 2

Este modo de interrupcion ha sido disenado para utilizar mas eficientemente las capacidades del microprocesador 280 y su familia asociada de perifericos. El dispositivo periferico que interrumpe elige la direccion de inicio de la rutina de servicio, colocando un vector de direccion de 8 bits en el bus de datos durante el ciclo de reconocimiento de interrupcion. El byte de mayor orden de la direccion de la rutina de servicio a interrupcion es proporcionado por el registro I.

Eta facilidad de seleccion de direccion de rutina de servicio permite al dispositivo periferico usar diferentes tipos de rutina de servicio.

Estas rutinas deben estar localizadas en cualquier localidad disponible en memoria. Ya que el dispositivo periferico proporciona el byte de menor orden de un vector de 2-bytes, el bit 0 (A0) debe ser cero.

OPERACION DE HABILITACION/DESHABILITACION DE INTERRUPCION

Dos FLIP-FLOPS, IFF1 e IFF2 son usados para senalar al CPU el estado de interrupcion. Cuya operacion se muestra enseguida:

ACCION	IFF1	IFF2	COMENTARIOS
CPU reset	0	0	Interrupcion mascarable INT deshabilitada
Ejecucion de instruccion DI	0	0	Interrupcion mascarable INT deshabilitada
Ejecucion de instruccion EI	1	1	Interrupcion mascarable INT habilitada
Ejecucion de instruccion LD A,I	*	*	IFF2 ->bandera de paridad
Ejecucion de instruccion LD A,R	*	*	IFF2 ->bandera de paridad
Aceptacion de NMI	0	IFF1	IFF1 -> IFF2 Interrupcion mascarable INT deshabilitada
Ejecucion de instruccion RETN	IFF2	*	IFF2 -> IFF1 a la terminacion de servicio de interrupcion de NMI.

28420

280 PIO Controlador Entrada/Salida Paralelo (Parallel
Input/Output Controller)

CARACTERISTICAS:

- * Provee una interfaz directa entre sistemas de microcomputadora y dispositivos perifericos Z-80.
- * Dos puertos los cuales pueden interrumpir mediante control de protocolo para respuesta rapida.
- * Cuatro modos de operacion programable: Byte Input, Byte Output, Byte Input/Output (Puerto A solamente) y Bit Input/Output.
- * Interrupciones programables bajo condiciones de estado del dispositivo periferico.
- * Requerimiento de bus standard familia Z-80 y requerimiento de interrupcion prioritaria implementada sin logica externa, (daisy-chain).
- * Las ocho salidas del puerto B pueden llevar transistores darlington (1.5 mA at 1.5 V).

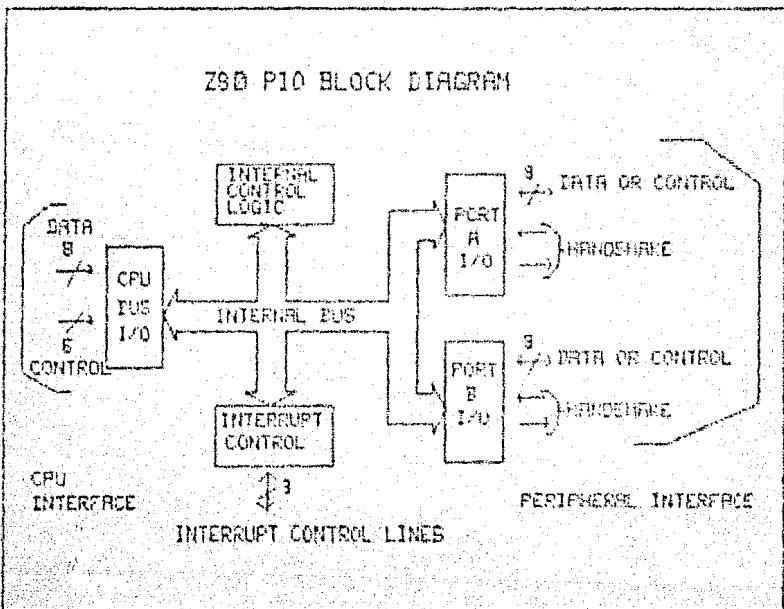


FIGURA V DIAGRAMA DE BLOQUES DEL CONTROLADOR DE ENTRADA/SALIDA PARALELO Z80 PIO.

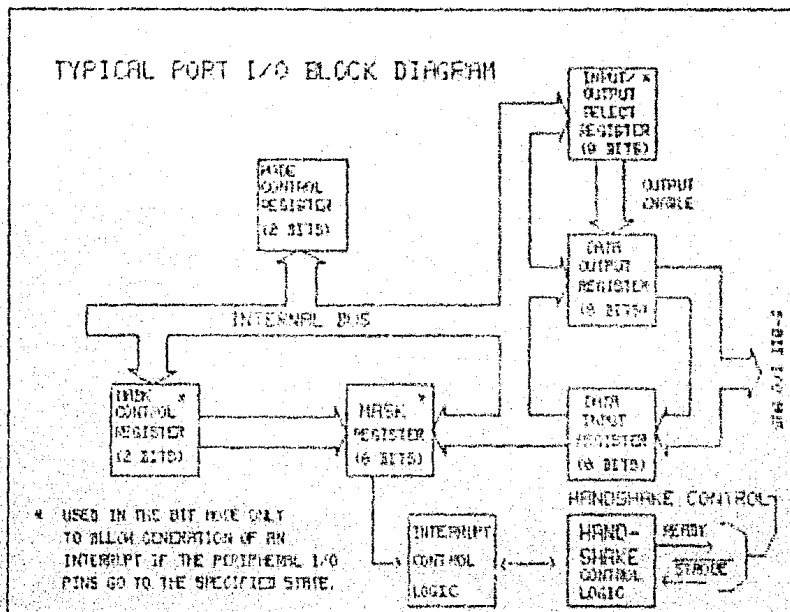


FIGURA VI DIAGRAMA DE BLOQUES TÍPICO DE LOS PUERTOS DE ENTRADA/SALIDA DEL Z80 P/D.

DESCRIPCION GENERAL

El controlador de Entrada/Salida I80 PIO es un dispositivo programable de dos puertos que provee una interface compatible TTL entre dispositivos perifericos y el I80 CPU. El cual configura al I80 PIO para interconectar dispositivos perifericos sin necesidad de logica externa. Algunos dispositivos perifericos que son compatibles con el I80 PIO son: teclados, impresoras, programadores de PROM, etc.

Una característica de los controladores de dispositivos I80 que los diferencia de otros controladores de interface, es que la transferencia de datos entre el dispositivo periferico y el CPU, es llevada a cabo bajo control de interrupcion. De tal manera que la logica de interrupcion del PIO permite un completo uso de las eficientes capacidades de interrupcion del I-80 CPU durante transferencias de entrada/salida. Toda la logica para implementar una estructura de interrupcion anidada es incluida en el PIO. Otra característica del PIO es la habilidad de interrumpir bajo condiciones especificas de estado del dispositivo periferico.

Esta capacidad de interrupcion reduce el tiempo que el procesador debe de emplear para checar el estado del dispositivo.

El Z-80 PIO permite la interfaz a dispositivos periféricos a través de dos puertos independientes de entrada/salida, designados Puerto A y Puerto B. Cada puerto tiene ocho bits de datos y dos señales de protocolo Ready y Strobe, las cuales controlan la transferencia. La señal de salida Ready indica al periférico que el puerto está listo para transferencia de datos. La señal de entrada Strobe del periférico indica que una transferencia de datos ha ocurrido.

MODOS DE OPERACION

El Z-80 PIO puede ser programado para operar en cuatro modos:

Byte Output (modo 0), Byte Input (modo 1), Byte Input/Output (modo 2) y Bit Input Output (modo 3).

En Modo 0, ya sea el puerto A o el puerto B puede ser programado para salida de datos. Los dos puertos tienen registros de salida que son individualmente direccionados por el CPU. Los datos pueden ser escritos en cualquier puerto en cualquier momento.

Cuando los datos son escritos en el puerto, Ready en estado activo indica al dispositivo externo que el dato está disponible en el puerto asociado y está listo para la transferencia al mismo.

Despues de la transferencia del dato, el dispositivo externo responde con una senal activa Strobe la cual genera la interrupcion si esta se encuentra habilitada.

En Modo 1, ya sea el puerto A o el puerto B puede ser programado para entrada de datos. Los puertos tienen registros de entrada que son individualmente direccionados por el CPU. Cuando el CPU lee datos del puerto asociado, el PIO coloca la senal Ready activa, la cual es detectada por el dispositivo externo. Entonces el dispositivo coloca el dato en las lineas de datos y avisa al puerto mediante Strobe activo, lo que transfiere el dato al registro de entrada del puerto y recoloca Ready a cero y genera la interrupcion, si esta habilitada. El CPU puede leer un puerto en cualquier momento, lo que coloca Ready en estado activo de nuevo.

El Modo 2 es bidireccional y usa el puerto A mas las senales de interrupcion y protocolo de los dos puertos. El puerto B debe ser programado en modo 3. En operacion el puerto A es usado tanto para entrada como para salida. La operacion de salida es similar al modo 0, excepto en que el dato es sacado por el puerto A solamente cuando ASTB esta en estado bajo. La operacion de entrada es similar al modo 1 excepto en que la entrada de datos usa las senales de protocolo del puerto B y la interrupcion del puerto B, si esta habilitada.

Los dos puertos pueden ser usados en modo 3. En este modo, los bits individuales son definidos como bits de entrada o salida. Lo que provee hasta ocho bits individualmente definidos por puerto.

Durante la operación de este modo, Ready y Strobe no son usadas. En cambio, una interrupción es generada si la condición de una entrada cambia. Los requerimientos para generar una interrupción son definidos durante la programación de operación; el nivel activo es especificado como alto o bajo y la condición lógica es especificada como una entrada activa (OR) o todas las entradas activas (AND).

Las salidas de datos son controladas por el CPU y pueden ser escritas o cambiadas en cualquier momento.

Bits individuales pueden ser seleccionados.

Las señales de protocolo no son usadas en modo 3, Ready es mantenida en estado bajo y Strobe es deshabilitada.

Cuando se utilizan interrupciones en el Z-80 PIO, el modo de interrupción del Z-80 CPU debe ser el modo 2.

PROGRAMACION

Modo 0, 1, 2. (Byte Output, Input or Bidireccional).

Programar a un puerto para modo 0, 1, 2 requiere dos palabras por puerto, estas son las siguientes:

Una palabra de control de modo: selecciona el modo de operacion del puerto y puede ser escrita en cualquier momento.

Un vector de interrupcion: El Z-80 PIO esta disenado para usarse con el Z-80 CPU en modo de interrupcion 2. Cuando la interrupcion es habilitada, el PIO debe de proveer un vector de interrupcion.

Modo 3. Programar un puerto para modo 3, requiere una palabra de control de modo, un vector de interrupcion y tres palabras adicionales siguientes:

Control de registro de Entrada/Salida: Cuando el modo 3 es elegido, la palabra de control de modo debe ser seguida por otra palabra que seleccione el registro de control de Entrada/Salida, el cual define cuales lineas del puerto son Entradas y cuales Salidas.

Palabra de control de interrupcion: En modo 3, el protocolo no es usado, Las interrupciones son generadas como una funcion logica de los niveles de las senales de entrada. La palabra de

control de interrupción establece las condiciones y los niveles lógicos requeridos para generar interrupción. Hay dos funciones posibles:

AND (si todas las entradas cambian a nivel activo) y OR (si cualquiera de las señales de entrada cambia a nivel activo).

El nivel activo de los bits de entrada puede ser alto o bajo.

Palabra de control de máscara: esta palabra establece el registro de control de máscara, permitiendo que los bits inutilizados sean deshabilitados.

Des-habilitación de interrupción: es otra palabra de control la cual puede ser usada para habilitar o deshabilitar la interrupción de un puerto. Puede ser usada sin necesidad de cambiar el resto de las palabras de control de interrupción.

En las siguientes figuras se muestra los niveles apropiados para la programación de las palabras de control.

PALABRA DE CONTROL DE MODO

! D7 ! D6 ! D5 ! D4 ! 1 ! 1 ! 1 ! 1 !

! Identifica a la palabra
de control de modo

! No importa su estado

! Seleccion de modo

0 0 Modo 0

0 1 Modo 1

1 0 Modo 2

1 1 Modo 3.

PALABRA DE VECTOR DE INTERRUPCION

! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! 0 !

! Identifica vector
de interrupcion

! Vector de interrupcion
proporcionado por el usuario.

PALABRA DE CONTROL DEL REGISTRO I/O

! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! D0 !

! 0 designa bit de salida
! 1 designa bit de entrada.

PALABRA DE CONTROL DE INTERRUPCION

! D7 ! D6 ! D5 ! D4 ! 0 ! 1 ! 1 ! 1 !

! ! ! ! !
! ! ! ! ! Identifica a la palabra
! ! ! ! ! de control de interrupcion.
! ! ! ! ! D4 = 0 no sigue palabra de mascara
! ! ! ! ! D4 = 1 sigue palabra de mascara
! ! ! ! ! D5 = 0 el nivel activo es bajo
! ! ! ! ! D5 = 1 el nivel activo es alto
! ! ! ! ! D6 = 0 interrupcion sobre funcion OR
! ! ! ! ! D6 = 1 interrupcion sobre funcion AND
! ! ! ! ! D7 = 0 interrupcion deshabilitada
! ! ! ! ! D7 = 1 interrupcion habilitada *

* NOTA : El puerto no es habilitado hasta que la habilitacion de
interrupcion es seguida por un activo Mi.

PALABRA DE CONTROL DE MASCARA

! D7 ! D6 ! D5 ! D4 ! D3 ! D2 ! D1 ! D0 !

! ! ! ! !
! ! ! ! ! MBO - MB7 bits de mascara
! ! ! ! ! un bit es monitoreado para
! ! ! ! ! una interrupcion si es definido
! ! ! ! ! como entrada y el bit de mascara
! ! ! ! ! es 0.

PALABRA DESHABILITADORA DE INTERRUPCION

! D7 ! D6 ! D5 ! D4 ! 0 ! 0 ! 1 ! 1 !

! Identifica a la palabra
deshabilitadora de interrupcion
! no importa su estado
! D7 = 0 interrupcion deshabilitada
! D7 = 1 interrupcion habilitada.

74138

74138 DECODIFICADOR

Este circuito asegura que el dispositivo periferico correcto sea habilitado para la transferencia de informacion.

La interface de cada dispositivo utiliza las direcciones 5CH-5FH.

7400 y 7408

Compuertas Standard NAND y AND respectivamente.

7406 y 74241

Inversores de colector abierto y buffers line drivers/line receivers respectivamente.

EL desarrollo del Diseño del Sistema Interface en Base al Standard HEWLETT-PACKARD INTERFACE BUS (HP-IB), se muestra mediante la sucesión de las siguientes graficas. Tambien se presenta una simulacion del circuito del Sistema Interface.

Inicialmente se muestran los graficos que sirvieron como base para la comprension de la funcion de una interface mediante su localizacion en un sistema de computadora (figura VII), asi como, la estructura que presenta el bus del 'sistema interface' (figura VIII).

Posteriormente se muestran los diagramas de bloques de la interface disenada en base al standard HEWLETT PACKARD INTERFACE BUS (HP-IB):

1) Como controlador de sistema (figura IX) en el que se pueden observar :

La representacion de una microcomputadora con las siglas C-T/L que indican que tal dispositivo puede funcionar como C Controlador, T Emisor, L receptor.

La representacion de un controlador de entrada/salida paralelo que muestra la existencia de las senales de control de sistema definidas por el standard utilizado: SRQ, REN, EOI, IFC, ATN y las senales auxiliares: RD1, RD2, WR. Las cuales se utilizan para la habilitacion del dispositivo como emisor o receptor (RD1, RD2)

del circuito de protocolo y bus de datos. Y emisor o receptor (WR) del bus de control de interface. Estas senales actuan sobre circuitos integrados conocidos como buffers (74241) [RD2, WR] y las senales ARDY y ASTRB definidas anteriormente cuando se describio el controlador de entrada/salida paralelo.

La representacion del circuito de protocolo, el cual es mostrado en la figura XI y finalmente las lineas del bus HP-IB.

2) Como emisor/receptor (figura X) en el que se puedan observar:

La representacion de una microcomputadora con las siglas T/L que indican que tal dispositivo puede funcionar como T Emisor, L receptor.

La representacion de un controlador de entrada/salida paralelo que muestra la existencia de las senales de control de sistema definidas por el standard utilizado: SRD, REN, EO1, IFC, ATN y las senales auxiliares: RD1, RD2, WR. Las cuales se utilizan para la habilitacion del dispositivo como emisor o receptor (RD1, RD2) del circuito de protocolo y bus de datos. Y emisor o receptor (WR) del bus de control de interface. Estas senales actuan sobre circuitos integrados conocidos como buffers (74241) [RD2, WR]. Y las senales ARDY y ASTRB definidas anteriormente cuando se describio el controlador de entrada/salida paralelo.

La representación del circuito de protocolo, el cual es mostrado en la figura XI y finalmente las líneas del bus HP-IB.

En seguida se muestra el diagrama esquemático lógico de la interface diseñada en base al standard HEWLETT PACKARD INTERFACE BUS (HP-IB) [figura XI] considerando los requerimientos secuenciales de la transferencia de datos inter-asegurada THREE WIRE HANDSHAKE TECHNIQUE (figura XII) y la inter-relación de la secuencia de tiempos del protocolo del dispositivo emisor y el dispositivo receptor (pags. 105-106).

Finalmente la simulación del circuito como dispositivo emisor (talker) y como dispositivo receptor (listener), acompañada por la relación de las operaciones lógicas que simulan la interface diseñada en base al standard HEWLETT PACKARD INTERFACE BUS (HP-IB) pags. 113-115, para las cuales se han de tomar en cuenta las siguientes consideraciones:

Auxiliarse de:

- 1) El diagrama esquemático lógico de la interface diseñada en base al standard HEWLETT PACKARD INTERFACE BUS (HP-IB) figura XI.
- 2) Los requerimientos secuenciales de la transferencia de datos inter-asegurada THREE WIRE HANDSHAKE TECHNIQUE figura XII.
- 3) Secuencia de tiempos de protocolo (HANDSHAKE) pags. 105-106.

Consideraciones:

a) La indicación de la función del dispositivo se encuentra en el ángulo superior derecho, acompañado por un número que permite el seguimiento de las tablas de resultado que provee el simulador (tengase cuidado de no confundir con el número de dispositivo en el sistema, es decir, dirección del dispositivo).

b) Especificación de que el controlador del sistema es el dispositivo designado por MICROO (ángulo superior izquierdo).

c) En el contexto de cada tabla de resultado se muestran las etiquetas que identifican a cada una de las compuertas que aparecen en el diagrama esquemático lógico de la figura XI. Así como, el estado actual, tanto a su entrada como, a la salida de las mismas inmediatamente después, ya sea debajo o a la derecha.

d) Todas y cada una de las compuertas, se encuentran asociadas a la simulación de las mismas por un número en la compuerta (número superior cuando se muestra más de uno).

e) Existen algunas etiquetas y valores asociados que no aparecen en el diagrama esquemático lógico, las cuales sirven como ayuda a la simulación y son explicadas a continuación:

AIDS	L/T		
1	DEFINEN ESTADOS	1	INDICA QUE DISPOSITIVO DEBE ACEPTAR
0	LOGICOS	1	INTERRUPCION

11111111 o -11111111 QUE ACOMPAÑAN A $\overline{\text{ASTRB}}$ PARA INDICAR EL
EMISOR RECEPTOR ESTADO DEL DISPOSITIVO QUE GENERA
INTERRUPCIÓN

and4' AUXILIAR QUE REFLEJA EL VALOR DE and4

MICRO1-MICROn ETIQUETAS QUE REPRESENTAN DISPOSITIVOS CONECTADOS AL
BUS.

not', not51-not5n AUXILIARES QUE INVIERTEN EL VALOR DE $\overline{\text{ARDY}}$
UTILIZADOS PARA DESARROLLAR LA IMPLEMENTACION DE
LA FUNCION LOGICA DE COLECTOR ABIERTO, ES DECIR,

$\overline{\text{ARDY1}}$	-----	not51		
$\overline{\text{ARDY2}}$	-----	not52	and	OPCO
$\overline{\text{ARDYn}}$	-----	not5n		

f) Existen unas senales repetidas, esto es debido a que unas se utilizan para introducir los estados a los cuales el circuito ha de responder y otras para seguir el funcionamiento del circuito de interface y observar la respuesta del mismo en las salidas correspondientes.

SIMULACION DEL CIRCUITO COMO EMISOR

TALKER -- EN ESTA PRIMERA TABLA SE MUESTRAN LOS ESTADOS INICIALES
(1)

TALKER -- EN ESTA SEGUNDA TABLA SE MUESTRA EL ESTADO DE LISTO POR
(2) UNO O MAS RECEPTORES INDICANDO QUE ESTAN LISTOS PARA RECIBIR INFORMACION (OBSERVAR PENULTIMA LINEA)

TALKER -- EN ESTA TERCERA TABLA SE MUESTRAN TODOS LOS RECEPTORES
(3) EN ESTADO DE LISTO Y CONSIGUIENTEMENTE NRFD VA A 1 Y DAV VA A 0 LO QUE OCACIONA QUE ALGUNO O TODOS LOS RECEPTORES ACEPTEN INTERRUPCIÓN COLOCANDO SU CORRESPONDIENTE ARDY EN 0 LO QUE OCACIONA QUE NRFD VAYA A 0 LO CUAL SE MUESTRA EN LA SIGUIENTE TABLA.

TALKER -- EN ESTA CUARTA TABLA SE MUESTRA UN ESTADO ESPECIAL QUE
(4) SE PRESENTA SOLO EN LA SIMULACION DEBIDO A QUE NO SE
PUEDEN INTRODUCIR DATOS NI SE PUEDE IMPRIMIR LA TABLA
MIENTRAS SE EFECTUAN CALCULOS Y PUESTO QUE NDAC VA A 1
DEBE PRODUCIR UNA INTERRUPCION QUE MANDARIA A ARDY A 0
Y PROVOCARIA QUE DAV FUERA A 1, ESTO SE COMPROBABA CON
EL ESTADO DE LA ETIQUETA AUXILIAR L/T EN 0

TALKER -- EN ESTA QUINTA TABLA SE MUESTRA LOS ESTADOS QUE
(5) DEBIERON PRESENTARSE EN LA TABLA ANTERIOR Y QUE SE
ESTABLECIERON UNA VEZ QUE COLOCAMOS ARDY EN 0 COMO
ACEPTACION DE INTERRUPCION VOLVIENDO NDAC A 0

TALKER -- EN ESTA SEXTA Y ULTIMA TABLA SE PRESENTAN LOS ESTADOS
(6) QUE CORRESPONDEN AL INICIO DEL CICLO (TALKER (1)).

SIMULACION DEL CIRCUITO COMO RECEPTOR

LISTENER -- EN ESTA PRIMERA TABLA SE MUESTRAN LOS ESTADOS
(1) INICIALES

LISTENER -- EN ESTA SEGUNDA TABLA UNO O VARIOS DE LOS RECEPTORES
(2) ESTAN EN ESTADO LISTO PARA RECIBIR INFORMACION

LISTENER -- EN ESTA TERCERA TABLA TODOS LOS RECEPTORES ESTAN EN
(3) ESTADO LISTO POR LO QUE NRFD VA A 1 Y DAV A 0, LO
QUE INDICARIA QUE UNO O TODOS LOS RECEPTORES
ACEPTAN INTERRUPCION Y SU CORRESPONDIENTE SENAL
DE ARDY FUERA A 0 PROVOCANDO QUE NRFD FUERA A 0

LISTENER -- EN ESTA CUARTA TABLA SE MUESTRA UN ESTADO ESPECIAL QUE
(4) SE PRESENTA SOLO EN LA SIMULACION DEBIDO A QUE NO SE
PUEDEN INTRODUCIR DATOS NI SE PUEDE IMPRIMIR LA TABLA
MIENTRAS SE EFECTUAN CALCULOS Y PUESTO QUE NDAC VA A 1
DEBE PRODUCIR UNA INTERRUPCION QUE MANDARIA A ARDY A 0
(CORRESPONDIENTE AL EMISOR) Y PROVOCARIA QUE DAV FUERA
A 1, ESTO SE COMPROBABA CON EL ESTADO DE LA ETIQUETA
AUXILIAR L/T EN 0

LISTENER -- EN ESTA QUINTA Y ULTIMA TABLA SE MUESTRA LOS ESTADOS
(5) QUE DEBIERON PRESENTARSE EN LA TABLA ANTERIOR Y QUE SE
ESTABLECIERON UNA VEZ QUE COLOCAMOS ARDY EN 0 COMO
ACEPTACION DE INTERRUPCION VOLVIENDO NDAC A 0

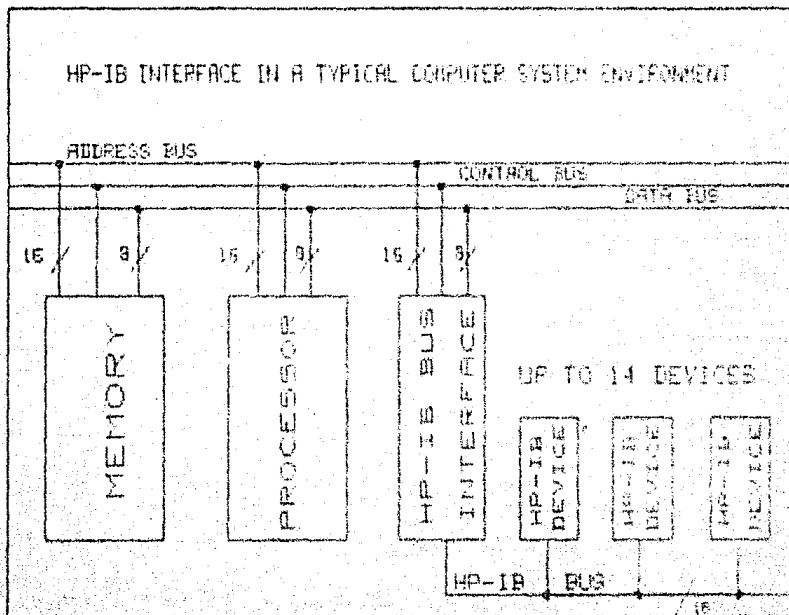


FIGURA VII INTERFACE HP-IB EN UN SISTEMA TIPICO DE COMPUTADORA.

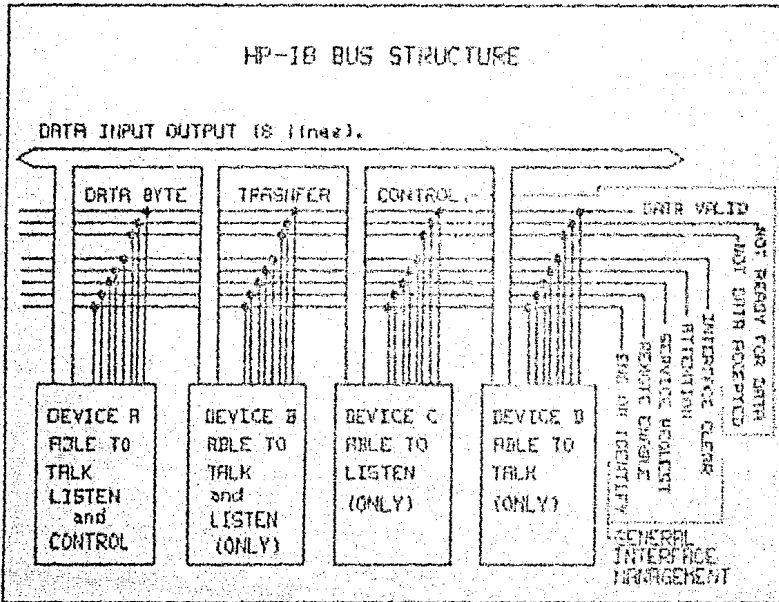


FIGURA VIII ESTRUCTURA DEL BUS DE INTERFACE HP-IB.

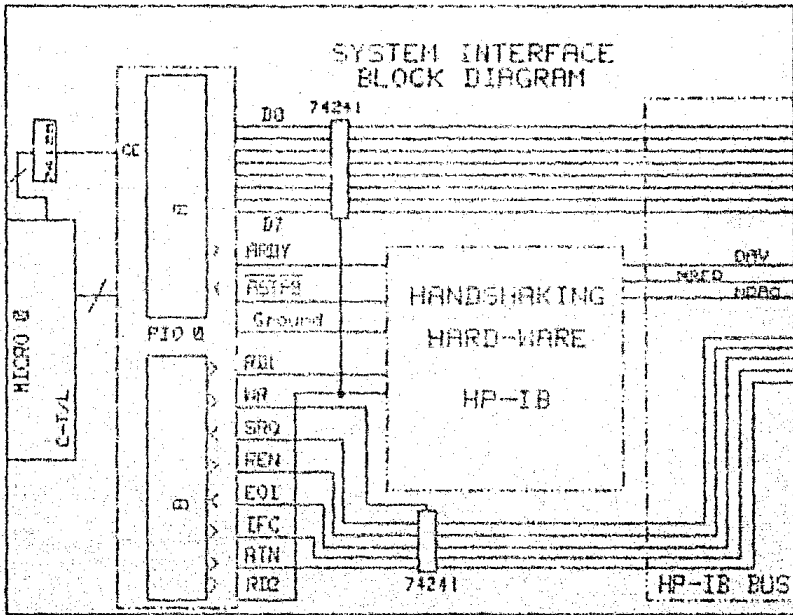


FIGURA IX DIAGRAMA DE BLOQUES DE LA INTERFACE DISEÑADA

en base al standard

HEWLETT-PACKARD INTERFACE BUS (HP-IB)

EN EL QUE SE MUESTRA COMO CONTROLADOR DE SISTEMA.

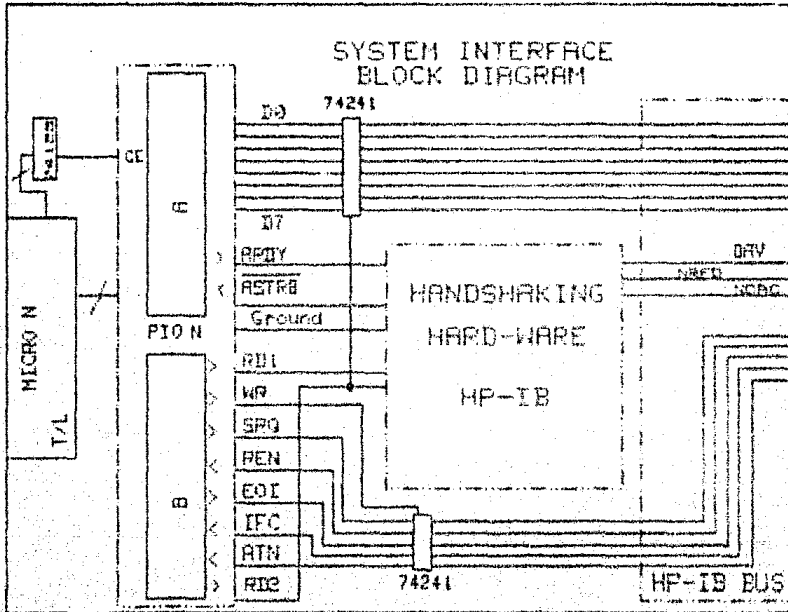


FIGURA X DIAGRAMA DE BLOQUES DE LA INTERFACE DISEÑADA
 en base al standard
 HEWLETT-PACKARD INTERFACE BUS (HP-IB)
 EN EL QUE SE MUESTRA COMO EMISOR/RECEPTOR.

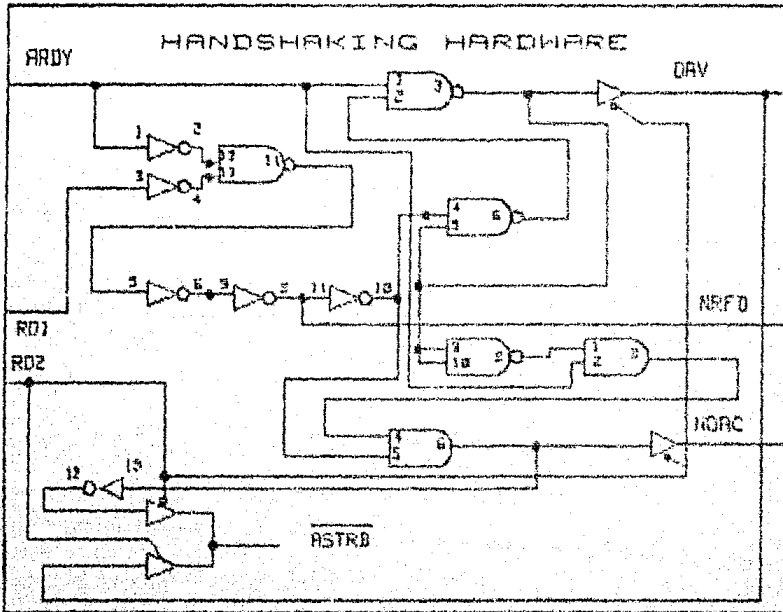
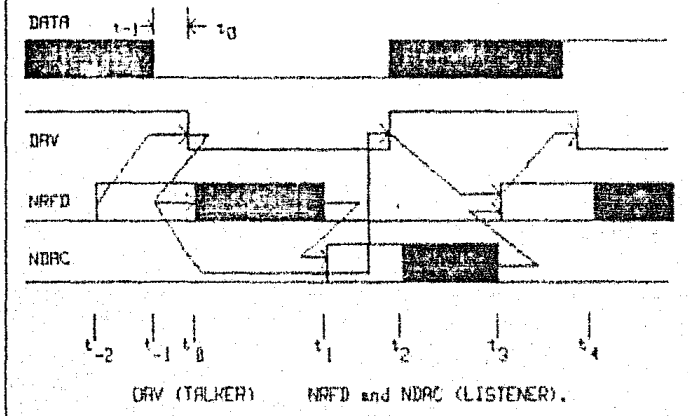


FIGURA XI DIAGRAMA ESQUEMATICO LOGICO DE LA INTERFACE
DISENADA en base al standard HEWLETT-PACKARD
INTERFACE BUS (HP-IB).

EN LA FIGURA XII SE MUESTRAN LOS REQUERIMIENTOS
SECUENCIALES DE LA TRANSFERENCIA DE DATOS INTERASEGURADA

THREE WIRE HANDSHAKE TECHNIQUE.

SEQUENTIAL REQUIREMENTS OF THE THREE WIRE TRANSFER



EVENTOS

FIGURA XII

- t_{-2} : El receptor se encuentra listo para aceptar datos.
- t_{-1} : El emisor ha colocado el dato en el bus de datos.
- t_0 : Indicación de que el dato es válido.
- t_1 : El receptor ha aceptado el dato y no requiere que se mantenga aun válido el mismo.
- t_2 : El emisor indica que el dato ya no es válido y en este momento puede cambiarlo.
- t_3 : El receptor indica que esta listo para un nuevo dato.
- t_4 : Se inicia un nuevo ciclo (equivalente a t_0).
- t_{-1} to t_0 : Tiempo en que el dato es puesto en el bus de datos antes de que DAV es puesto en estado bajo.

NOTA: Por propósitos de ilustración se ha utilizado una sola línea para representar al bus de datos. Las curvas de seguimiento indican la secuencia de las señales de transferencia (HANDSHAKE).

DISEÑO DE UN SISTEMA INTERFACE
EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-1B).

TESIS

OPERACION DEL RECEPTOR

- 01) Inicio
- 1L) Coloca NRFD y NDAC en estado bajo
- 2L) Listo para transferencia y responder a datos si estos son
aceptados ? 'no' ----> 2L)
- 'si'
- 3L) Listo para aceptar datos ? 'no' ----> 3L)
- 'si'
- 4L) Coloca NRFD en estado alto * la línea de la señal
NRFD va a estado alto
cuando todos los receptores
están listos ----> 6T)
- 5L) Es DAV sensada en estado bajo ? 'no' ----> 5L)
- 'si'
- 6L) Coloca NRFD en estado bajo
- 7L) Acepta el byte de dato
- 8L) Coloca NDAC en alto * NDAC permanece en estado
bajo hasta que todos los
receptores han aceptado
el byte de dato ----> 8T)
- 9L) Ha cambiado de estado bajo a alto DAV ? 'no' ----> 9L)
- 'si'
- 10L) Coloca NDAC en estado bajo
- 11L) ----> 3L)

DISENO DE UN SISTEMA INTERFACE
 EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-IB).

16915

HANDSHAKING HARDWARE SIMULATION

MICROO
 CONTROLLER

BASED ON HP-IB

TALKER
 (1)

ARDY	0	not1	1	and	nand12	not	and	nand1	not
RD1	1	not3	0	0	0	1	0		1
					nand4	not			
not'	not5	OPCD	not11	and4	not				and4'
1	0	0	1	1	0		0		0
L/T	and	nand9	not	and1	and4	not13	AIDS		BUS
1	1		0	0	0	1	1		ENABLE
1							0		RD2
ARDY	RD1	DAV	NRFD	NDAC	ASTRB		111111111		0
0	1	1	0	0	1				
		-111111111		111111111					
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn		LISTENERS	
not51	not52	not53	not54	not55	not56	not5n		NOT READY	
1	1	1	1	1	1	1		1	
0	0	0	0	0	0	0		and opco	
nots'	0	0	0	0	0	0		0	

HANDSHAKING HARDWARE SIMULATION

MICROO
 CONTROLLER

BASED ON HP-IB

TALKER
 (2)

ARDY	1	not1	0	and	nand12	not	and	nand1	not
RD1	1	not3	0	0	0	1	0		1
					nand4	not			
not'	not5	OPCD	not11	and4	not				and4'
1	0	0	1	1	0		0		0
L/T	and	nand9	not	and1	and4	not13	AIDS		BUS
1	1		0	0	0	1	1		ENABLE
1							0		RD2
ARDY	RD1	DAV	NRFD	NDAC	ASTRB		111111111		0
1	1	1	0	0	1				
		-111111111		111111111					
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn		LISTENERS	
not51	not52	not53	not54	not55	not56	not5n		NOT READY	
0	1	1	0	1	0	1		1	
1	0	0	1	0	1	0		and opco	
nots'	0	0	0	0	0	0		0	

HANDSHAKING HARDWARE SIMULATION

MICRO0 CONTROLLER		BASED ON HP-IB						TALKER (3)
ARDY	1	not1	0	and	not	and	nand1	
RD1	1	not3	0	0	1	1	not	
							0	
not	not5	OPCO	not11	and4	not		and4'	
1	0	1	0	0	1	1	1	
L/T	and	not	and1	and4	not13	AIDS	BUS	
0	0	1	1	0	1	1	ENABLE	
1						0	RD2	
ARDY	RD1	DAV	NRFED	NDAC	ASTRB	11111111	0	
1	1	0	1	0	1			
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS	
not51	not52	not53	not54	not55	not56	not5n	NOT READY	
0	0	0	0	0	0	0	1	
1	1	1	1	1	1	1	and opco	
nots	1	1	1	1	1	1	1	

HANDSHAKING HARDWARE SIMULATION

MICRO0 CONTROLLER		BASED ON HP-IB						TALKER (4)
ARDY	1	not1	0	and	not	and	nand1	
RD1	1	not3	0	0	1	1	not	
							0	
not	not5	OPCO	not11	and4	not		and4'	
1	0	0	1	0	1	1	1	
L/T	and	not	and1	and4	not13	AIDS	BUS	
0	0	1	1	1	0	1	ENABLE	
0						0	RD2	
ARDY	RD1	DAV	NRFED	NDAC	ASTRB	11111111	0	
1	1	0	0	1	0			
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS	
not51	not52	not53	not54	not55	not56	not5n	NOT READY	
1	1	1	1	1	1	1	1	
0	0	0	0	0	0	0	and opco	
nots	0	0	0	0	0	0	0	

DISEÑO DE UN SISTEMA INTERFACE
 EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-IB).

TESIS

HANDSHAKING HARDWARE SIMULATION

MICRO0 CONTROLLER		BASED ON HP-IB						TALKER (S)
ARDY	0	not1	1	nand12	not	nand1		
RD1	1	not3	0	and	1	and	not	
				nand4	0	0	1	
not	not5	OPCD	not11	and4	not			
1	0	0	1	1	0	0	and4	
							0	
L/T	nand9	not	and1	and4	not13	AIDS		
1	and	0	0	0	1	1	BUS	
1	1	0	0	0	0	0	ENABLE	
ARDY	RD1	DAV	NRF0	NDAC	ASTRB		RD2	
0	1	1	0	0	1	11111111	0	
MICRO1	MICRO2	-11111111	MICRO4	11111111	MICRO6	MICROn	LISTENERS	
not51	not52	MICRO3	not54	MICRO5	not56	not5n	NOT_READY	
0	0	not53	not54	not55	not56	not5n	1	
1	1	0	1	0	1	1	and opco	
not6	not6	1	0	0	0	0	0	

HANDSHAKING HARDWARE SIMULATION

MICRO0 CONTROLLER		BASED ON HP-IB						TALKER (S)
ARDY	0	not1	1	nand12	not	nand1		
RD1	1	not3	0	and	1	and	not	
				nand4	0	0	1	
not	not5	OPCD	not11	and4	not			
1	0	0	1	1	0	0	and4	
							0	
L/T	nand9	not	and1	and4	not13	AIDS		
1	and	0	0	0	1	1	BUS	
1	1	0	0	0	0	0	ENABLE	
ARDY	RD1	DAV	NRF0	NDAC	ASTRB		RD2	
0	1	1	0	0	1	11111111	0	
MICRO1	MICRO2	-11111111	MICRO4	11111111	MICRO6	MICROn	LISTENERS	
not51	not52	MICRO3	not54	MICRO5	not56	not5n	NOT_READY	
0	0	not53	not54	not55	not56	not5n	1	
1	1	0	1	0	1	1	and opco	
not6	not6	1	0	0	0	0	0	

TESTS

HANDSHAKING HARDWARE SIMULATION

BASED ON HP-16

DISEÑO DE UN SISTEMA INTERFACE
EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-16).

HANDSHAKING HARDWARE SIMULATION										LISTENER (1)	
BASED ON HP-16											
MICROCONTROLLER											
ARDY	0	not1	1	and	not	and	not	and	not		
RD1	0	not2	1	1	0	0	1	0	1		
				nand12				nand1			
not	1	OPCD	not11	and4	not						
0	0	0	1	1	0				0		
L/T	and	not	and1	and4	not13	AIDS					
1	1	0	0	0	1	1					
ARDY	RD1	DAV	NRFD	NDAC	ASTRB	BUS					
0	0	1	0	0	1	ENABLE					
						RD2					
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS				
not51	not52	not53	not54	not55	not56	not5n	NOT READY				
1	1	1	1	1	1	1	1				
0	0	0	0	0	0	0	and opcd				
not6	0	0	0	0	0	0	0				

HANDSHAKING HARDWARE SIMULATION										LISTENER (2)	
BASED ON HP-16											
MICROCONTROLLER											
ARDY	1	not1	0	and	not	and	not	and	not		
1	0	not3	1	0	1	0	1	0	1		
				nand12				nand1			
not	not5	OPCD	not11	and4	not						
1	0	0	1	1	0				0		
L/T	and	not	and1	and4	not13	AIDS					
1	1	0	0	0	1	1					
ARDY	RD1	DAV	NRFD	NDAC	ASTRB	BUS					
1	0	1	0	0	1	ENABLE					
						RD2					
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS				
not51	not52	not53	not54	not55	not56	not5n	NOT READY				
0	1	1	0	1	0	1	1				
1	0	0	1	0	1	0	and opcd				
not6	0	0	0	0	0	0	0				

DISENO DE UN SISTEMA INTERFAZ
 EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-IB).

TESIS

HANDSHAKING HARDWARE SIMULATION

MICRO CONTROLLER		BASED ON HP-10						LISTENER (3)
ARDY	1	not1	0	and	not	and	not	
RD1	0	not3	1	0	1	1	0	
				nand12		nand1		
				and	not	and	not	
				0	1	1	0	
				nand4				
not	not5	OPCO	not11	and4	not		and4	
1	0	1	0	0	1	1	1	
L/T	and	not	and1	and4	not13	AIDS	BUS	
0	0	1	1	0	1	1	ENABLE	
1						0	RD2	
ARDY	RD1	DAV	NRF0	NDAC	ASTRB	-11111111	1	
1	0	0	1	0	0			
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS	
not51	not52	not53	not54	not55	not56	not5n	NOT READY	
0	0	0	0	0	0	0	1	
1	1	1	1	1	1	1	and opco	
nots	1	1	1	1	1	1	1	

HANDSHAKING HARDWARE SIMULATION

MICRO CONTROLLER		BASED ON HP-1B						LISTENER (4)
ARDY	1	not1	0	and	not	and	not	
RD1	0	not3	1	0	1	1	0	
				nand12		nand1		
				and	not	and	not	
				0	1	1	0	
				nand4				
not	not5	OPCO	not11	and4	not		and4	
1	0	0	1	0	1	1	1	
L/T	and	not	and1	and4	not13	AIDS	BUS	
0	0	1	1	1	0	1	ENABLE	
1						0	RD2	
ARDY	RD1	DAV	NRF0	NDAC	ASTRB	-11111111	1	
1	0	0	0	1	0			
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS	
not51	not52	not53	not54	not55	not56	not5n	NOT READY	
1	1	1	1	1	1	1	1	
0	0	0	0	0	0	0	and opco	
nots	0	0	0	0	0	0	0	

DISENO DE UN SISTEMA INTERFACE
 EN BASE AL STANDARD HEWLETT-PACKARD INTERFACE BUS (HP-IB).

YESIS

HANDSHAKING HARDWARE SIMULATION

MICROO
 CONTROLLER

BASED ON HP-IB

LISTENER
 (5)

ARDY	0	not1	1	and	not	and	nand1	not
RD1	0	not3	1	1	0	0		1
		not5	OPCO	not11	and4	not		
not	1	0	1	1	0		and4	
0						0	0	0
		nand9						
L/T	and	not	and1	and4	not13	AID5		
1	1	0	0	0	1	1		BUS
1						0		ENABLE
ARDY	RD1	DAV	NRF0	NDAC	AST00			RD2
0	0	1	0	0	1	-11111111		1
		-11111111		11111111				
MICRO1	MICRO2	MICRO3	MICRO4	MICRO5	MICRO6	MICROn	LISTENERS	
not51	not52	not53	not54	not55	not56	not5n	NOT READY	
0	0	0	1	0	1	0	1	
1	1	1	0	1	0	1	and opco	
nots	1	1	0	0	0	0	0	

RELACION DE LAS OPERACIONES LOGICAS QUE SIMULAN LA INTERFACE
 DISEÑADA EN BASE AL STANDARD HEWLETT PACKARD INTERFACE BUS
 (HP-IB). En una microcomputadora PC utilizando una hoja de
 calculo Perfect Calc Copyright 1983 by Perfect Software Inc.

```

\dg0
\jgc
>c1:" HANDSHAKING HARDWARE SIMULATION
>a3:"MICRO0
>d3:" BASED ON HP-IB
>a4:"CONTROLLER
>e5:" nand12
>g5:" nand1
>a6:" ARDY
>b6:=a17#0.E+00
>06:"not1
>d6:=not(b6)#1E+00
>e6:"and
>f6:"not
>g6:"and
>h6:"not
>a7:"RD
>b7:=b17#0.E+00
>c7:"not3
>d7:=not(b7)#1.E+00
>e7:=and(d6,d7)#1.E+00
>f7:=not(e7)#0.E+00
>g7:=and(b6,h11)#0.E+00
>h7:=not(g7)#1.E+00
>e8:" nand4
>b9:"not5
>c9:"DPC0
>d9:"not11
>e9:"and4
>f9:"not
>a10:"not
>b10:=not(f7)#1.E+00
>c10:=h23#0.E+00
>d10:=not(c10)#1.E+00
>e10:=and(d10,h7)#1.E+00
>f10:=not(e10)#0.E+00
>h10:"and4
>a11:=not(b10)#0.E+00
>g11:=f10#0.E+00
>h11:0.E+00
>b12:" nand9
    
```

```
>a13:"L/T
>b13:"and
>c13:"not
>d13:"and1
>e13:"and4
>f13:"not13
>g13:"AIDS
>a14:if(c17,g14,g15)#1.E+00
>b14:=and(h7,h7)#1.E+00
>c14:=not(b14)#0.E+00
>d14:=and(c14,b6)#0.E+00
>e14:=and(d14,d10)#0.E+00
>f14:=not(e14)#1.E+00
>g14:1.E+00
>a15:=if(e17,g15,g14)#1.E+00
>f15:"-----
>g15:0.E+00
>h15:"BUS
>a16:"ARDY
>b16:"RD
>c16:"DAV
>d16:"NRFD
>e16:"NDAC
>f16:"ASTRB
>h16:"ENABLE
>a17:0.E+00
>b17:0.E+00
>c17:=h7#1.E+00
>d17:=h23#0.E+00
>e17:=e14#1.E+00
>f17:=if(h17,c17,f14)#1.E+00
>g17:=if(h17,c18,e18)#1.1111111E+08
>h17:0.E+00
>c18:-1.1111111E+07
>e18:1.1111111E+08
>a19:"MICRO1
>b19:"MICRO2
>c19:"MICRO3
>d19:"MICRO4
>e19:"MICRO5
>f19:"MICRO6
>g19:"MICROn
>h19:"LISTENERS
>a20:"not51
>b20:"not52
>c20:"not53
>d20:"not54
>e20:"not55
>f20:"not56
>g20:"not5n
```

```
>h20:"NOT READY  
>a21:1.E+00  
>b21:1.E+00  
>c21:1.E+00  
>d21:1.E+00  
>e21:1.E+00  
>f21:1.E+00  
>g21:1.E+00  
>h21:1.E+00  
>a22:=not (a21)#0.E+00  
>b22:=not (b21)#0.E+00  
>c22:=not (c21)#0.E+00  
>d22:=not (d21)#0.E+00  
>e22:=not (e21)#0.E+00  
>f22:=not (f21)#0.E+00  
>g22:=not (g21)#0.E+00  
>h22:" and opco  
>a23:"nots'  
>b23:=and (a22,b22)#0.E+00  
>c23:=and (b23,c22)#0.E+00  
>d23:=and (c23,d22)#0.E+00  
>e23:=and (d23,e22)#0.E+00  
>f23:=and (e23,f22)#0.E+00  
>g23:=and (f23,g22)#0.E+00  
>h23:=and (g23,a11)#0.E+00
```

HP-IB CABLE CONNECTOR PIN CONNECTIONS

		DESCRIPTION		
MICRO RIBBON CONNECTOR TYPE 57	DI01	1	13	DI05
	DI02	2	14	DI06
	DI03	3	15	DI07
	DI04	4	16	DI08
	E0I	5	17	REN
	DRY	6	18	P/O TWISTED PAIR WITH 6
	NRFI	7	19	P/O TWISTED PAIR WITH 7
	NORC	8	20	P/O TWISTED PAIR WITH 8
	IFC	9	21	P/O TWISTED PAIR WITH 9
	SRQ	10	22	P/O TWISTED PAIR WITH 10
	ATN	11	23	P/O TWISTED PAIR WITH 11
	GROUND ONLY → SHIELD AT SYSTEM CONTROLLER		12	24

SHOULD BE GROUNDED NEAR TERMINATION OF OTHER WIRE OF TWISTED PAIR

FIGURA XIII CONECTOR DEL CABLE DE INTERFACE HP-IB.

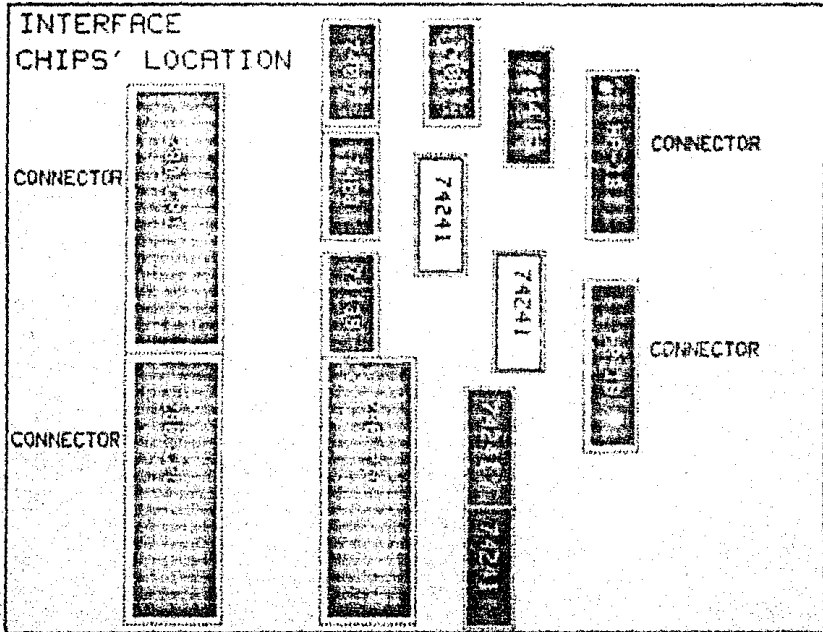


FIGURA XIV LOCALIZACION DE LOS CIRCUITOS INTEGRADOS EN LA INTERFACE DISEÑADA en base al standard HEWLETT PACKARD INTERFACE BUS (HP-IB).

DISEÑO DEL FIRM-WARE

En las siguientes paginas se muestran los programas y rutinas de servicio que satisfacen la operacion del Sistema Interface disenado en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB) y al Sistema Educacional de Microprocesador I80.

ADDR	OP	SYSTEM INTERFACE	CONTROL	CONTROL PROGRAM
0000	0000	START		INITIAL TEST
0001	0001	START		INITIAL TEST
0002	0002	START		INITIAL TEST
0003	0003	START		INITIAL TEST
0004	0004	START		INITIAL TEST
0005	0005	START		INITIAL TEST
0006	0006	START		INITIAL TEST
0007	0007	START		INITIAL TEST
0008	0008	START		INITIAL TEST
0009	0009	START		INITIAL TEST
0010	0010	START		INITIAL TEST
0011	0011	START		INITIAL TEST
0012	0012	START		INITIAL TEST
0013	0013	START		INITIAL TEST
0014	0014	START		INITIAL TEST
0015	0015	START		INITIAL TEST
0016	0016	START		INITIAL TEST
0017	0017	START		INITIAL TEST
0018	0018	START		INITIAL TEST
0019	0019	START		INITIAL TEST
0020	0020	START		INITIAL TEST
0021	0021	START		INITIAL TEST
0022	0022	START		INITIAL TEST
0023	0023	START		INITIAL TEST
0024	0024	START		INITIAL TEST
0025	0025	START		INITIAL TEST
0026	0026	START		INITIAL TEST
0027	0027	START		INITIAL TEST
0028	0028	START		INITIAL TEST
0029	0029	START		INITIAL TEST
0030	0030	START		INITIAL TEST
0031	0031	START		INITIAL TEST
0032	0032	START		INITIAL TEST
0033	0033	START		INITIAL TEST
0034	0034	START		INITIAL TEST
0035	0035	START		INITIAL TEST
0036	0036	START		INITIAL TEST
0037	0037	START		INITIAL TEST
0038	0038	START		INITIAL TEST
0039	0039	START		INITIAL TEST
0040	0040	START		INITIAL TEST
0041	0041	START		INITIAL TEST
0042	0042	START		INITIAL TEST
0043	0043	START		INITIAL TEST
0044	0044	START		INITIAL TEST
0045	0045	START		INITIAL TEST
0046	0046	START		INITIAL TEST
0047	0047	START		INITIAL TEST
0048	0048	START		INITIAL TEST
0049	0049	START		INITIAL TEST
0050	0050	START		INITIAL TEST
0051	0051	START		INITIAL TEST
0052	0052	START		INITIAL TEST
0053	0053	START		INITIAL TEST
0054	0054	START		INITIAL TEST
0055	0055	START		INITIAL TEST
0056	0056	START		INITIAL TEST
0057	0057	START		INITIAL TEST
0058	0058	START		INITIAL TEST
0059	0059	START		INITIAL TEST
0060	0060	START		INITIAL TEST
0061	0061	START		INITIAL TEST
0062	0062	START		INITIAL TEST
0063	0063	START		INITIAL TEST
0064	0064	START		INITIAL TEST
0065	0065	START		INITIAL TEST
0066	0066	START		INITIAL TEST
0067	0067	START		INITIAL TEST
0068	0068	START		INITIAL TEST
0069	0069	START		INITIAL TEST
0070	0070	START		INITIAL TEST
0071	0071	START		INITIAL TEST
0072	0072	START		INITIAL TEST
0073	0073	START		INITIAL TEST
0074	0074	START		INITIAL TEST
0075	0075	START		INITIAL TEST
0076	0076	START		INITIAL TEST
0077	0077	START		INITIAL TEST
0078	0078	START		INITIAL TEST
0079	0079	START		INITIAL TEST
0080	0080	START		INITIAL TEST
0081	0081	START		INITIAL TEST
0082	0082	START		INITIAL TEST
0083	0083	START		INITIAL TEST
0084	0084	START		INITIAL TEST
0085	0085	START		INITIAL TEST
0086	0086	START		INITIAL TEST
0087	0087	START		INITIAL TEST
0088	0088	START		INITIAL TEST
0089	0089	START		INITIAL TEST
0090	0090	START		INITIAL TEST
0091	0091	START		INITIAL TEST
0092	0092	START		INITIAL TEST
0093	0093	START		INITIAL TEST
0094	0094	START		INITIAL TEST
0095	0095	START		INITIAL TEST
0096	0096	START		INITIAL TEST
0097	0097	START		INITIAL TEST
0098	0098	START		INITIAL TEST
0099	0099	START		INITIAL TEST

```

0000) 0000 DEAR EQU 0000H ADDRESS POINTER (PNAF+2)
0001) 0001 DEAR EQU 0000H DEVICE ENABLE ADDRESSES
0002) 0002 DEAR EQU 0000H POINTER (PNAF+3)
0003) 0003 DEAR EQU 0000H DEVICE ADDRESS
0004) 0004 DEAR EQU 0000H
0005) 0005 DEAR EQU 0000H
0006) 0006 DEAR EQU 0000H
0007) 0007 DEAR EQU 0000H
0008) 0008 DEAR EQU 0000H
0009) 0009 DEAR EQU 0000H
0010) 0010 DEAR EQU 0000H
0011) 0011 DEAR EQU 0000H
0012) 0012 DEAR EQU 0000H
0013) 0013 DEAR EQU 0000H
0014) 0014 DEAR EQU 0000H

```

MACROS DEFINITION

```

0015) 0015 MACRO INTCON
0016) 0016 INTCON EQU 0000H
0017) 0017 INTCON EQU 0000H
0018) 0018 INTCON EQU 0000H
0019) 0019 INTCON EQU 0000H
0020) 0020 INTCON EQU 0000H
0021) 0021 INTCON EQU 0000H
0022) 0022 INTCON EQU 0000H
0023) 0023 INTCON EQU 0000H
0024) 0024 INTCON EQU 0000H
0025) 0025 INTCON EQU 0000H
0026) 0026 INTCON EQU 0000H
0027) 0027 INTCON EQU 0000H
0028) 0028 INTCON EQU 0000H
0029) 0029 INTCON EQU 0000H
0030) 0030 INTCON EQU 0000H
0031) 0031 INTCON EQU 0000H
0032) 0032 INTCON EQU 0000H
0033) 0033 INTCON EQU 0000H
0034) 0034 INTCON EQU 0000H
0035) 0035 INTCON EQU 0000H
0036) 0036 INTCON EQU 0000H
0037) 0037 INTCON EQU 0000H
0038) 0038 INTCON EQU 0000H
0039) 0039 INTCON EQU 0000H
0040) 0040 INTCON EQU 0000H
0041) 0041 INTCON EQU 0000H
0042) 0042 INTCON EQU 0000H
0043) 0043 INTCON EQU 0000H
0044) 0044 INTCON EQU 0000H
0045) 0045 INTCON EQU 0000H
0046) 0046 INTCON EQU 0000H
0047) 0047 INTCON EQU 0000H
0048) 0048 INTCON EQU 0000H
0049) 0049 INTCON EQU 0000H
0050) 0050 INTCON EQU 0000H
0051) 0051 INTCON EQU 0000H
0052) 0052 INTCON EQU 0000H
0053) 0053 INTCON EQU 0000H
0054) 0054 INTCON EQU 0000H
0055) 0055 INTCON EQU 0000H
0056) 0056 INTCON EQU 0000H
0057) 0057 INTCON EQU 0000H
0058) 0058 INTCON EQU 0000H
0059) 0059 INTCON EQU 0000H
0060) 0060 INTCON EQU 0000H
0061) 0061 INTCON EQU 0000H
0062) 0062 INTCON EQU 0000H
0063) 0063 INTCON EQU 0000H
0064) 0064 INTCON EQU 0000H
0065) 0065 INTCON EQU 0000H
0066) 0066 INTCON EQU 0000H
0067) 0067 INTCON EQU 0000H
0068) 0068 INTCON EQU 0000H
0069) 0069 INTCON EQU 0000H
0070) 0070 INTCON EQU 0000H
0071) 0071 INTCON EQU 0000H
0072) 0072 INTCON EQU 0000H
0073) 0073 INTCON EQU 0000H
0074) 0074 INTCON EQU 0000H
0075) 0075 INTCON EQU 0000H
0076) 0076 INTCON EQU 0000H
0077) 0077 INTCON EQU 0000H
0078) 0078 INTCON EQU 0000H
0079) 0079 INTCON EQU 0000H
0080) 0080 INTCON EQU 0000H
0081) 0081 INTCON EQU 0000H
0082) 0082 INTCON EQU 0000H
0083) 0083 INTCON EQU 0000H
0084) 0084 INTCON EQU 0000H
0085) 0085 INTCON EQU 0000H
0086) 0086 INTCON EQU 0000H
0087) 0087 INTCON EQU 0000H
0088) 0088 INTCON EQU 0000H
0089) 0089 INTCON EQU 0000H
0090) 0090 INTCON EQU 0000H
0091) 0091 INTCON EQU 0000H
0092) 0092 INTCON EQU 0000H
0093) 0093 INTCON EQU 0000H
0094) 0094 INTCON EQU 0000H
0095) 0095 INTCON EQU 0000H
0096) 0096 INTCON EQU 0000H
0097) 0097 INTCON EQU 0000H
0098) 0098 INTCON EQU 0000H
0099) 0099 INTCON EQU 0000H
0100) 0100 INTCON EQU 0000H
0101) 0101 INTCON EQU 0000H
0102) 0102 INTCON EQU 0000H
0103) 0103 INTCON EQU 0000H
0104) 0104 INTCON EQU 0000H
0105) 0105 INTCON EQU 0000H
0106) 0106 INTCON EQU 0000H
0107) 0107 INTCON EQU 0000H
0108) 0108 INTCON EQU 0000H
0109) 0109 INTCON EQU 0000H
0110) 0110 INTCON EQU 0000H

```

MACROS DEFINITION IS OVER

MAIN PROGRAM STARTS HERE

```

0111) 0111 ORG 0000H SEGMENT
0112) 0112 ORG 0000H CPU INTERRUPT RESPONSE
0113) 0113 ORG 0000H MODE 3
0114) 0114 ORG 0000H
0115) 0115 ORG 0000H
0116) 0116 ORG 0000H
0117) 0117 ORG 0000H
0118) 0118 ORG 0000H
0119) 0119 ORG 0000H
0120) 0120 ORG 0000H
0121) 0121 ORG 0000H
0122) 0122 ORG 0000H
0123) 0123 ORG 0000H
0124) 0124 ORG 0000H
0125) 0125 ORG 0000H
0126) 0126 ORG 0000H
0127) 0127 ORG 0000H
0128) 0128 ORG 0000H
0129) 0129 ORG 0000H
0130) 0130 ORG 0000H
0131) 0131 ORG 0000H
0132) 0132 ORG 0000H
0133) 0133 ORG 0000H
0134) 0134 ORG 0000H
0135) 0135 ORG 0000H
0136) 0136 ORG 0000H
0137) 0137 ORG 0000H
0138) 0138 ORG 0000H
0139) 0139 ORG 0000H
0140) 0140 ORG 0000H
0141) 0141 ORG 0000H
0142) 0142 ORG 0000H
0143) 0143 ORG 0000H
0144) 0144 ORG 0000H
0145) 0145 ORG 0000H
0146) 0146 ORG 0000H
0147) 0147 ORG 0000H
0148) 0148 ORG 0000H
0149) 0149 ORG 0000H
0150) 0150 ORG 0000H
0151) 0151 ORG 0000H
0152) 0152 ORG 0000H
0153) 0153 ORG 0000H
0154) 0154 ORG 0000H
0155) 0155 ORG 0000H
0156) 0156 ORG 0000H
0157) 0157 ORG 0000H
0158) 0158 ORG 0000H
0159) 0159 ORG 0000H
0160) 0160 ORG 0000H
0161) 0161 ORG 0000H
0162) 0162 ORG 0000H
0163) 0163 ORG 0000H
0164) 0164 ORG 0000H
0165) 0165 ORG 0000H
0166) 0166 ORG 0000H
0167) 0167 ORG 0000H
0168) 0168 ORG 0000H
0169) 0169 ORG 0000H
0170) 0170 ORG 0000H
0171) 0171 ORG 0000H
0172) 0172 ORG 0000H
0173) 0173 ORG 0000H
0174) 0174 ORG 0000H
0175) 0175 ORG 0000H
0176) 0176 ORG 0000H
0177) 0177 ORG 0000H
0178) 0178 ORG 0000H
0179) 0179 ORG 0000H
0180) 0180 ORG 0000H
0181) 0181 ORG 0000H
0182) 0182 ORG 0000H
0183) 0183 ORG 0000H
0184) 0184 ORG 0000H
0185) 0185 ORG 0000H
0186) 0186 ORG 0000H
0187) 0187 ORG 0000H
0188) 0188 ORG 0000H
0189) 0189 ORG 0000H
0190) 0190 ORG 0000H
0191) 0191 ORG 0000H
0192) 0192 ORG 0000H
0193) 0193 ORG 0000H
0194) 0194 ORG 0000H
0195) 0195 ORG 0000H
0196) 0196 ORG 0000H
0197) 0197 ORG 0000H
0198) 0198 ORG 0000H
0199) 0199 ORG 0000H
0200) 0200 ORG 0000H

```

DEFINE INTERRUPT SERVICE ROUTINE TABLE

```

0201) 0201 LD A,00H
0202) 0202 LD (<0000>,A) PORT A (LSB)
0203) 0203 LD A,00H
0204) 0204 LD (<0000>,A)
0205) 0205 LD A,00H
0206) 0206 LD (<0000>,A)
0207) 0207 LD A,00H
0208) 0208 LD (<0000>,A) PORT B (LSB)
0209) 0209 LD A,00H
0210) 0210 LD (<0000>,A)
0211) 0211 LD A,00H
0212) 0212 LD (<0000>,A)
0213) 0213 LD A,00H
0214) 0214 LD (<0000>,A)
0215) 0215 LD A,00H
0216) 0216 LD (<0000>,A)
0217) 0217 LD A,00H
0218) 0218 LD (<0000>,A)
0219) 0219 LD A,00H
0220) 0220 LD (<0000>,A)
0221) 0221 LD A,00H
0222) 0222 LD (<0000>,A)
0223) 0223 LD A,00H
0224) 0224 LD (<0000>,A)
0225) 0225 LD A,00H
0226) 0226 LD (<0000>,A)
0227) 0227 LD A,00H
0228) 0228 LD (<0000>,A)
0229) 0229 LD A,00H
0230) 0230 LD (<0000>,A)
0231) 0231 LD A,00H
0232) 0232 LD (<0000>,A)
0233) 0233 LD A,00H
0234) 0234 LD (<0000>,A)
0235) 0235 LD A,00H
0236) 0236 LD (<0000>,A)
0237) 0237 LD A,00H
0238) 0238 LD (<0000>,A)
0239) 0239 LD A,00H
0240) 0240 LD (<0000>,A)
0241) 0241 LD A,00H
0242) 0242 LD (<0000>,A)
0243) 0243 LD A,00H
0244) 0244 LD (<0000>,A)
0245) 0245 LD A,00H
0246) 0246 LD (<0000>,A)
0247) 0247 LD A,00H
0248) 0248 LD (<0000>,A)
0249) 0249 LD A,00H
0250) 0250 LD (<0000>,A)
0251) 0251 LD A,00H
0252) 0252 LD (<0000>,A)
0253) 0253 LD A,00H
0254) 0254 LD (<0000>,A)
0255) 0255 LD A,00H
0256) 0256 LD (<0000>,A)
0257) 0257 LD A,00H
0258) 0258 LD (<0000>,A)
0259) 0259 LD A,00H
0260) 0260 LD (<0000>,A)
0261) 0261 LD A,00H
0262) 0262 LD (<0000>,A)
0263) 0263 LD A,00H
0264) 0264 LD (<0000>,A)
0265) 0265 LD A,00H
0266) 0266 LD (<0000>,A)
0267) 0267 LD A,00H
0268) 0268 LD (<0000>,A)
0269) 0269 LD A,00H
0270) 0270 LD (<0000>,A)
0271) 0271 LD A,00H
0272) 0272 LD (<0000>,A)
0273) 0273 LD A,00H
0274) 0274 LD (<0000>,A)
0275) 0275 LD A,00H
0276) 0276 LD (<0000>,A)
0277) 0277 LD A,00H
0278) 0278 LD (<0000>,A)
0279) 0279 LD A,00H
0280) 0280 LD (<0000>,A)
0281) 0281 LD A,00H
0282) 0282 LD (<0000>,A)
0283) 0283 LD A,00H
0284) 0284 LD (<0000>,A)
0285) 0285 LD A,00H
0286) 0286 LD (<0000>,A)
0287) 0287 LD A,00H
0288) 0288 LD (<0000>,A)
0289) 0289 LD A,00H
0290) 0290 LD (<0000>,A)
0291) 0291 LD A,00H
0292) 0292 LD (<0000>,A)
0293) 0293 LD A,00H
0294) 0294 LD (<0000>,A)
0295) 0295 LD A,00H
0296) 0296 LD (<0000>,A)
0297) 0297 LD A,00H
0298) 0298 LD (<0000>,A)
0299) 0299 LD A,00H
0300) 0300 LD (<0000>,A)
0301) 0301 LD A,00H
0302) 0302 LD (<0000>,A)
0303) 0303 LD A,00H
0304) 0304 LD (<0000>,A)
0305) 0305 LD A,00H
0306) 0306 LD (<0000>,A)
0307) 0307 LD A,00H
0308) 0308 LD (<0000>,A)
0309) 0309 LD A,00H
0310) 0310 LD (<0000>,A)
0311) 0311 LD A,00H
0312) 0312 LD (<0000>,A)
0313) 0313 LD A,00H
0314) 0314 LD (<0000>,A)
0315) 0315 LD A,00H
0316) 0316 LD (<0000>,A)
0317) 0317 LD A,00H
0318) 0318 LD (<0000>,A)
0319) 0319 LD A,00H
0320) 0320 LD (<0000>,A)
0321) 0321 LD A,00H
0322) 0322 LD (<0000>,A)
0323) 0323 LD A,00H
0324) 0324 LD (<0000>,A)
0325) 0325 LD A,00H
0326) 0326 LD (<0000>,A)
0327) 0327 LD A,00H
0328) 0328 LD (<0000>,A)
0329) 0329 LD A,00H
0330) 0330 LD (<0000>,A)
0331) 0331 LD A,00H
0332) 0332 LD (<0000>,A)
0333) 0333 LD A,00H
0334) 0334 LD (<0000>,A)
0335) 0335 LD A,00H
0336) 0336 LD (<0000>,A)
0337) 0337 LD A,00H
0338) 0338 LD (<0000>,A)
0339) 0339 LD A,00H
0340) 0340 LD (<0000>,A)
0341) 0341 LD A,00H
0342) 0342 LD (<0000>,A)
0343) 0343 LD A,00H
0344) 0344 LD (<0000>,A)
0345) 0345 LD A,00H
0346) 0346 LD (<0000>,A)
0347) 0347 LD A,00H
0348) 0348 LD (<0000>,A)
0349) 0349 LD A,00H
0350) 0350 LD (<0000>,A)
0351) 0351 LD A,00H
0352) 0352 LD (<0000>,A)
0353) 0353 LD A,00H
0354) 0354 LD (<0000>,A)
0355) 0355 LD A,00H
0356) 0356 LD (<0000>,A)
0357) 0357 LD A,00H
0358) 0358 LD (<0000>,A)
0359) 0359 LD A,00H
0360) 0360 LD (<0000>,A)
0361) 0361 LD A,00H
0362) 0362 LD (<0000>,A)
0363) 0363 LD A,00H
0364) 0364 LD (<0000>,A)
0365) 0365 LD A,00H
0366) 0366 LD (<0000>,A)
0367) 0367 LD A,00H
0368) 0368 LD (<0000>,A)
0369) 0369 LD A,00H
0370) 0370 LD (<0000>,A)
0371) 0371 LD A,00H
0372) 0372 LD (<0000>,A)
0373) 0373 LD A,00H
0374) 0374 LD (<0000>,A)
0375) 0375 LD A,00H
0376) 0376 LD (<0000>,A)
0377) 0377 LD A,00H
0378) 0378 LD (<0000>,A)
0379) 0379 LD A,00H
0380) 0380 LD (<0000>,A)
0381) 0381 LD A,00H
0382) 0382 LD (<0000>,A)
0383) 0383 LD A,00H
0384) 0384 LD (<0000>,A)
0385) 0385 LD A,00H
0386) 0386 LD (<0000>,A)
0387) 0387 LD A,00H
0388) 0388 LD (<0000>,A)
0389) 0389 LD A,00H
0390) 0390 LD (<0000>,A)
0391) 0391 LD A,00H
0392) 0392 LD (<0000>,A)
0393) 0393 LD A,00H
0394) 0394 LD (<0000>,A)
0395) 0395 LD A,00H
0396) 0396 LD (<0000>,A)
0397) 0397 LD A,00H
0398) 0398 LD (<0000>,A)
0399) 0399 LD A,00H
0400) 0400 LD (<0000>,A)

```

INITIALIZE PORTS

```

0401) 0401 CALL CONL
0402) 0402 COMMAND IFCOMT
0403) 0403
0404) 0404
0405) 0405
0406) 0406
0407) 0407
0408) 0408
0409) 0409
0410) 0410
0411) 0411
0412) 0412
0413) 0413
0414) 0414
0415) 0415
0416) 0416
0417) 0417
0418) 0418
0419) 0419
0420) 0420
0421) 0421
0422) 0422
0423) 0423
0424) 0424
0425) 0425
0426) 0426
0427) 0427
0428) 0428
0429) 0429
0430) 0430
0431) 0431
0432) 0432
0433) 0433
0434) 0434
0435) 0435
0436) 0436
0437) 0437
0438) 0438
0439) 0439
0440) 0440
0441) 0441
0442) 0442
0443) 0443
0444) 0444
0445) 0445
0446) 0446
0447) 0447
0448) 0448
0449) 0449
0450) 0450
0451) 0451
0452) 0452
0453) 0453
0454) 0454
0455) 0455
0456) 0456
0457) 0457
0458) 0458
0459) 0459
0460) 0460
0461) 0461
0462) 0462
0463) 0463
0464) 0464
0465) 0465
0466) 0466
0467) 0467
0468) 0468
0469) 0469
0470) 0470
0471) 0471
0472) 0472
0473) 0473
0474) 0474
0475) 0475
0476) 0476
0477) 0477
0478) 0478
0479) 0479
0480) 0480
0481) 0481
0482) 0482
0483) 0483
0484) 0484
0485) 0485
0486) 0486
0487) 0487
0488) 0488
0489) 0489
0490) 0490
0491) 0491
0492) 0492
0493) 0493
0494) 0494
0495) 0495
0496) 0496
0497) 0497
0498) 0498
0499) 0499
0500) 0500

```

0010	FE	0111	01	001 INTERRUPT DISABLE
0018	FFFF	0110	LD A TWR	INITIATE
0020	1000	0110	OUT STADR	DATA ADDRESS
0028	FFFF	0110	LD A STADR	01 001
0030	1000	0110	OUT STADR	
		0110		
0038	100000	0110	CALL ITR	
		0110		
		0110		
		0111	CONSEC	COMMAND RECEPTION
		0111		
		0100	DISPLAY MESSAGE 'SI'	
		0101		
0048	000000	0101	CALL ITR	MICROKIT ROUTINE
0050	011000	0100	LD A STDR	SI VALUE
0058	000100	0101	CALL STDR	MICROKIT ROUTINE
0060	007000	0100	CALL ITR	MICROKIT ROUTINE
		0100		
		0100	COMMAND RECEPTION	
		0111		
0070	000100	0100	CALL ITR	MICROKIT ROUTINE
0078	000000	0100	LD A STDR	
0080	000100	0101	LD STADR	FILE NAME (HEX)
0088	000100	0101	LD A STDR	FUNCTION CODE
0090	000000	0100	LD STADR	
0098	000000	0101	CALL ITR	MICROKIT ROUTINE
0100	000000	0101	LD STADR	INDEX
0108	000000	0100	LD STADR	IDENTITY
0110	000000	0100	LD STADR	DESTINE
0118	000000	0101	LD STADR	LOAD QUANTITY
0120	000000	0101	LD STADR	
0128	000000	0101	LD STADR	
0130	000000	0101	LD STADR	IF QUANTITY = 0
0138	000000	0100	LD STADR	
0140	000000	0100	LD STADR	COMPUTE QUANTITY
0148	000000	0101	LD STADR	0 - DESTINE - SOURCE
0150	000000	0101	LD STADR	
0158	000000	0100	LD STADR	DEVICE FUNCTIONS ADDRESS
0160	000000	0100	LD STADR	POINTER
0168	000000	0101	CALL STDR	MICROKIT ROUTINE
0170	000000	0100	LD STADR	
0178	000000	0101	LD STADR	DESTINE TAKEN
0180	000000	0100	LD STADR	FUNCTION CODE
0188	000000	0101	LD STADR	
0190	000000	0100	LD STADR	NUMBER OF DEVICES
0198	000000	0100	LD STADR	MICROKIT ROUTINE
0200	000000	0100	LD STADR	DEVICE ADDRESS
0208	000000	0101	LD STADR	LAST ADDRESS
0210	000000	0101	LD STADR	
0218	000000	0100	LD STADR	DESTINE LISTENER
0220	000000	0101	LD STADR	FUNCTION CODE
0228	000000	0100	LD STADR	

0001	0100	0100	DEC 00	
0002	0101	0101	DEC 1	
0003	0007	0100	JR NEXT IN	NEXT ADDRESS
0004	1000	0100	JR CHDLE	
0005	0000010	0170	READY LD 0100, 0100	DEVICE DISABLE ADDRESS
0006	0000	0071	DEC 00	
0007	00	0170	DEC 1	
0008	0007	0170	JR NR READY	NEXT DEVICE
0009	0100010	0170	CHDLE LD 0100, 0100	DEVICE FUNCT POINTER-1
0010	0000	0170	LD 0100	OPTIONS
0011	00	0170	OTHER INC 01	
0012	00	0170	LD 0100	INTERFACE ADDRESS
0013	0010	0170	AND 010	
0014	0000	0170	CP 0100	CHECK DEVICE ADDRESS
0015	0000	0170	JR 0100	
0016	1000	0101	COND OTHER	NEXT OPTION
0017	1000	0100	JR LOCAL	FILE STATE
0018	00	0100	LD 0100	
0019	0000	0101	BIT 010	CHECK DEVICE FUNCTION
0020	0000	0100	IN NO RECEIVE	RECEIVE
0021	0007	0100	BIT 010	
0022	0000	0101	JR NO INITI	TRANSMIT
0023	00	0100	LOCAL 01	
0024	0000	0100	CALL 0000	
0025	0000	0100	CALL 0000	
0026	0000	0100	CALL 0000	MONITOR
0027	0000	0100		
0028	0000	0100		
0029	0000	0100		
0030	0000	0100		
0031	0000	0100		
0032	0000	0100		
0033	0000	0100		
0034	0000	0100		
0035	0000	0100		
0036	0000	0100		
0037	0000	0100		
0038	0000	0100		
0039	0000	0100		
0040	0000	0100		
0041	0000	0100		
0042	0000	0100		
0043	0000	0100		
0044	0000	0100		
0045	0000	0100		
0046	0000	0100		
0047	0000	0100		
0048	0000	0100		
0049	0000	0100		
0050	0000	0100		
0051	0000	0100		
0052	0000	0100		
0053	0000	0100		
0054	0000	0100		
0055	0000	0100		
0056	0000	0100		
0057	0000	0100		
0058	0000	0100		
0059	0000	0100		
0060	0000	0100		
0061	0000	0100		
0062	0000	0100		
0063	0000	0100		
0064	0000	0100		
0065	0000	0100		
0066	0000	0100		
0067	0000	0100		
0068	0000	0100		
0069	0000	0100		
0070	0000	0100		
0071	0000	0100		
0072	0000	0100		
0073	0000	0100		
0074	0000	0100		
0075	0000	0100		
0076	0000	0100		
0077	0000	0100		
0078	0000	0100		
0079	0000	0100		
0080	0000	0100		
0081	0000	0100		
0082	0000	0100		
0083	0000	0100		
0084	0000	0100		
0085	0000	0100		
0086	0000	0100		
0087	0000	0100		
0088	0000	0100		
0089	0000	0100		
0090	0000	0100		
0091	0000	0100		
0092	0000	0100		
0093	0000	0100		
0094	0000	0100		
0095	0000	0100		
0096	0000	0100		
0097	0000	0100		
0098	0000	0100		
0099	0000	0100		
0100	0000	0100		
0101	0000	0100		
0102	0000	0100		
0103	0000	0100		
0104	0000	0100		
0105	0000	0100		
0106	0000	0100		
0107	0000	0100		
0108	0000	0100		
0109	0000	0100		
0110	0000	0100		
0111	0000	0100		
0112	0000	0100		
0113	0000	0100		
0114	0000	0100		
0115	0000	0100		
0116	0000	0100		
0117	0000	0100		
0118	0000	0100		
0119	0000	0100		
0120	0000	0100		
0121	0000	0100		
0122	0000	0100		
0123	0000	0100		
0124	0000	0100		
0125	0000	0100		
0126	0000	0100		
0127	0000	0100		
0128	0000	0100		
0129	0000	0100		
0130	0000	0100		
0131	0000	0100		
0132	0000	0100		
0133	0000	0100		
0134	0000	0100		
0135	0000	0100		
0136	0000	0100		
0137	0000	0100		
0138	0000	0100		
0139	0000	0100		
0140	0000	0100		
0141	0000	0100		
0142	0000	0100		
0143	0000	0100		
0144	0000	0100		
0145	0000	0100		
0146	0000	0100		
0147	0000	0100		
0148	0000	0100		
0149	0000	0100		
0150	0000	0100		
0151	0000	0100		
0152	0000	0100		
0153	0000	0100		
0154	0000	0100		
0155	0000	0100		
0156	0000	0100		
0157	0000	0100		
0158	0000	0100		
0159	0000	0100		
0160	0000	0100		
0161	0000	0100		
0162	0000	0100		
0163	0000	0100		
0164	0000	0100		
0165	0000	0100		
0166	0000	0100		
0167	0000	0100		
0168	0000	0100		
0169	0000	0100		
0170	0000	0100		
0171	0000	0100		
0172	0000	0100		
0173	0000	0100		
0174	0000	0100		
0175	0000	0100		
0176	0000	0100		
0177	0000	0100		
0178	0000	0100		
0179	0000	0100		
0180	0000	0100		
0181	0000	0100		
0182	0000	0100		
0183	0000	0100		
0184	0000	0100		
0185	0000	0100		
0186	0000	0100		
0187	0000	0100		
0188	0000	0100		
0189	0000	0100		
0190	0000	0100		
0191	0000	0100		
0192	0000	0100		
0193	0000	0100		
0194	0000	0100		
0195	0000	0100		
0196	0000	0100		
0197	0000	0100		
0198	0000	0100		
0199	0000	0100		
0200	0000	0100		

```

08E7 C0709 0001 RECEIVE CALL COM TRANSMITE DATA
08E8 FB 0002 RECEIVED DI
08E9 C00F09 0003 CALL COML
08EA C04109 0004 CALL READ
08EB FB 0005 EI
08EC E00F00 0006 LD 00, (000A) ; DE DESTINE ADDRESS
08ED E01000 0007 LD 00, (000A+0) ; QUANTITY
08EE C000 0008 NEAR IN A, EDAS ; DATA BYTE RECEIVED
08EF 7A 0009 HALT
08F0 00 0010 INC C
08F1 00 0011 DEC C
08F2 C005 0012 JR NZ, NEAR ; NEXT DATA BYTE
08F3 C00009 0013 CALL I0L ;
08F4 E00000 0014 JP 0010 ; MONITOR
0015
0016 ; INTERRUPT SUBROUTINE RESPONSE PORT A
0017
0018
08F7 FB 0019 DI ; CPU INTERRUPT DISABLE
08F8 E010 0020 IN A, EDAS
08F9 C047 0021 BIT 0, A ; CHECK DEVICE FUNCTION
08FA 2000 0022 JR Z, TALIA ;
08FB E000 0023 LISR IN A, EDAS ; DATA BYTE RECEIVED AND
08FC 10 0024 LD 00, A ; STORED
08FD 10 0025 INC 00
08FE 00 0026 DEC 00 ; CPU INTERRUPT ENABLE
08FF FB 0027 EI ;
0900 E040 0028 RETI
0029
0901 10 0029 TALIA INC HL ; NEXT DATA BYTE LOCATION
0902 00 0030 DEC 00
0903 FB 0031 EI ; CPU INTERRUPT ENABLE
0904 E040 0032 RETI
0033
0905 ; INTERRUPT SUBROUTINE RESPONSE PORT B
0906
0907
0908 FB 0034 DI ; CPU INTERRUPT DISABLE
0909 0000 0035 IN A, EDAS
090A C00F 0036 SET 1, A ;
090B C007 0037 SET 2, A ; STABILIZE
090C C007 0038 SET 4, A ;
090D E000 0039 OUT EDAS, A ;
090E C00F 0040 BIT 0, A ; CHECK
090F C007 0041 JR C, SRQ ; DEVICES
0910 C00F 0042 BIT 1, A ; STATUS
0911 FB 0043 JR Z, ROI ;
0912 FB 0044 EI ;
0913 E040 0045 RETI ;
0046
0914 ; SERVICE ROUTINES
0915
0916
0917 0001 SRQ EPC SRQ1
0918 0002
  
```

0211	AL	0274+	POP HL	EXCHANGE
0212	LDHRA	0277+	LD HL, BRC1	PROGRAM
0213	RR	0278+	PUSH HL	COUNTER
0214	RAND	0279+	RTI	
		0280+		
0215	LDHRA	0281	BRC1 CALL ORG	
0216	LDHRA	0282	CALL WRAC	
0217	RR	0283	SI	
0218	LDHRA	0284	LD DE, FNRP	COMMAND POINTER
0219	LDHRA	0285	LD EC, 17H	OPTIONS
0220	RR	0286	IN R, BLAR	INTERFACE ADDRESS OR
0221	RR	0287	HALT	COMMAND
0222	RR	0288	INC C	
0223	RR	0289	DEC C	
0224	LDHRA	0290	JR NC, NEX	NEXT ADDR. OF COMMAND
0225	LDHRA	0291	CALL IDLE	
0226	LDHRA	0292	JR CONREC	
		0293		
0227		0294	BRC1 BRC W011	
		0295+		
0228	SI	0296+	POP HL	EXCHANGE
0229	LDHRA	0297+	LD HL, W011	PROGRAM
0230	RR	0298+	PUSH HL	COUNTER
0231	RAND	0299+	RTI	
		0300+		
0232	LDHRA	0301	W011 LD HL, DEFP-1	DEVICE FUNCT. POINTER-1
0233	RR	0302	LD B, GEN	OPTIONS
0234	RR	0303	OTHER INC HL	
0235	RR	0304	LD A, CHL	INTERFACE ADDRESS
0236	RR	0305	RND 17H	
0237	RR	0306	CP NSIMO	CHECK DEVICE ADDRESS
0238	RR	0307	JR Z, OK	
0239	RR	0308	OTHER	NEXT OPTION
0240	LDHRA	0309	JP LOCAL	IDLE STATE
0241	RR	0310	OK LD A, CHL	
0242	LDHRA	0311	BIT 0, A	CHECK DEVICE FUNCTION
0243	LDHRA	0312	JR NC, RECEIVE1	RECEIVE
0244	RR	0313	BIT 0, A	
0245	LDHRA	0314	JP NC, W011	TRANSMITE
0246	LDHRA	0315	JP LOCAL	IDLE STATE
		0316		
		0317		
		0318	UTILITY ROUTINES	
		0319		
		0320	CON	TRANSMITE COMMANDS
0247		0321	COMMAND ATNPT	INTERFACE COMMAND
		0322+		
0248	RR	0323+	SI	CPU INTERRUPT DISABLE
0249	LDHRA	0324+	LD A, TRAK	TR1-STATE
0250	RR	0325+	OUT EDRA, A	DATA ADDRESS
0251	RR	0326+	LD A, ATNPT	COMMAND
0252	RR	0327+	OUT EDRA, A	
		0328+		
0253	LDHRA	0329	CALL CONT	
0254	LDHRA	0330	CALL WRAC	

0888	FE	0011	DI	CPU INTERRUPT ENABLE
0889	LIANR	0012	LD R,ANR	COMMAND POINTER
0890	SLTRC	0013	LD R,C,17-	OPTIONS
0891	FE	0014	DI	INTERPRET ADDRESS ON
0892	0000	0015	OUT R,R,A	COMMAND
0893	TS	0016	HALT	
0894	00	0017	INC C	
0895	00	0018	DEC C	
0896	00FB	0019	SR R,C,HEMT	HEMT ADDR. OF COMMAND
0897		0020	COMMAND PENALT	
0898		0021+		
0899	FE	0022+	DI	CPU INTERRUPT DISABLE
0900	TEPW	0023+	LD R,TEPW	TRI-STATE
0901	0000	0024+	OUT R,R,A	DATA ADDRESS
0902	0000	0025+	LD R,REMPM	COMMAND
0903	0000	0026+	OUT R,R,A	
0904		0027+		
0905	000000	0028	ORL R,000	
0906	00	0029	RET	
0907		0030		
0908		0031	0000	COMMAND PENALT
0909		0032+		ENABLE
0910	FE	0033+	DI	CPU INTERRUPT DISABLE
0911	TEPW	0034+	LD R,TEPW	TRI-STATE
0912	0000	0035+	OUT R,R,A	DATA ADDRESS
0913	0000	0036+	LD R,REMPM	COMMAND
0914	0000	0037+	OUT R,R,A	
0915		0038+		
0916	TEPW	0039	LD R,TEPW	PID INTERRUPT ENABLE
0917	0000	0040	OUT R,R,A	CONTROL ADDRESS
0918	00	0041	RET	
0919		0042		
0920		0043	WRITE	COMMAND WRITPM
0921		0044+		ENABLE
0922	FE	0045+	DI	CPU INTERRUPT DISABLE
0923	TEPW	0046+	LD R,TEPW	TRI-STATE
0924	0000	0047+	OUT R,R,A	DATA ADDRESS
0925	0000	0048+	LD R,WRITPM	COMMAND
0926	0000	0049+	OUT R,R,A	
0927		0050+		
0928	TEPW	0051	LD R,TEPW	PID INTERRUPT ENABLE
0929	0000	0052	OUT R,R,A	CONTROL ADDRESS
0930	00	0053	RET	
0931		0054		
0932	FE	0055	DI	CPU INTERRUPT DISABLE
0933	TEPW	0056	LD R,TEPW	PID INTERRUPT ENABLE
0934	0000	0057	OUT R,R,A	CONTROL ADDRESS
0935	0000	0058	LD R,TEPW	TRI-STATE
0936	0000	0059	OUT R,R,A	DATA ADDRESS
0937	0000	0060	LD R,TEPW	PID INTERRUPT ENABLE
0938	0000	0061	OUT R,R,A	CONTROL ADDRESS
0939	00	0062	DI	CPU INTERRUPT DISABLE
0940	00	0063	RET	
0941		0064		
0942	0000	0065	ORL R,000	LISTENER

		0154		
		0157	P10 PROGRAMMING	PORT A
		0158		
015F	015F	0159	LD A, 0001	INPUT MODE
0160	0160	0159	OUT ED08, A	CONTROL ADDRESS
0161	0161	015A	LD A, 100A	INTERRUPT VECTOR ADDRESS
0162	0162	015B	OUT ED08, A	
0163	0163	015C	LD A, 1000	INTERRUPT DISABLE
0164	0164	015D	OUT ED08, A	
		015E		
		015F	P10 PROGRAMMING	PORT B
		0160		
0165	0165	0166	LD A, 0000	RIT INPUT/OUTPUT MODE
0166	0166	0167	OUT ED08, A	CONTROL ADDRESS
0167	0167	0168	LD A, 10000A	10 REGISTER
0168	0168	0169	OUT ED08, A	RIN, INC, PEN, OUTPUTS
		016A		SFD, SDI, INPUTS
016B	016B	016B	LD A, 1000	INTERRUPT VECTOR
016C	016C	016C	OUT ED08, A	
016D	016D	016D	LD A, 100000	INTERRUPT ON
016E	016E	016E	OUT ED08, A	ON FUNCTION
016F	016F	016F	LD A, 1000	ACTIVE LEVEL LOW
		0170		MARK CONTROL WORD
0171	0171	0170	LD A, 1000	
0172	0172	0170	OUT ED08, A	
0173	0173	0171	LD A, 1000	
0174	0174	0171	OUT ED08, A	
0175	0175	0172	LD A, 100000	
0176	0176	0172	OUT ED08, A	
0177	0177	0173	LD A, 1000	
0178	0178	0173	OUT ED08, A	
0179	0179	0174	LD A, 1000	
017A	017A	0174	OUT ED08, A	
017B	017B	0175	LD A, 1000	
017C	017C	0175	OUT ED08, A	
017D	017D	0176	LD A, 1000	
017E	017E	0176	OUT ED08, A	
017F	017F	0177	LD A, 1000	
		0178		
		0179	CONT	TALKER
		017A		
		017B	P10 PROGRAMMING	PORT A
		017C		
017D	017D	017D	LD A, 0000	OUTPUT MODE
017E	017E	017D	OUT ED08, A	CONTROL ADDRESS
017F	017F	017E	LD A, 1000A	INTERRUPT VECTOR ADDRESS
0180	0180	017F	OUT ED08, A	
0181	0181	0180	LD A, 1000	INTERRUPT DISABLE
0182	0182	0181	OUT ED08, A	
		0183		
		0184	P10 PROGRAMMING	PORT B
		0185		
0186	0186	0186	LD A, 0000	RIT INPUT/OUTPUT MODE
0187	0187	0187	OUT ED08, A	CONTROL ADDRESS
0188	0188	0188	LD A, 10000A	10 REGISTER
0189	0189	0189	OUT ED08, A	RIN, INC, PEN, OUTPUTS
		018A		SFD, SDI, INPUTS
018B	018B	018B	LD A, 1000	INTERRUPT VECTOR
018C	018C	018C	OUT ED08, A	
018D	018D	018C	LD A, 100000	INTERRUPT ON
018E	018E	018C	OUT ED08, A	ON FUNCTION
018F	018F	018D	LD A, 1000	ACTIVE LEVEL LOW
		0190		MARK CONTROL WORD
0191	0191	018E	LD A, 1000	
0192	0192	018E	OUT ED08, A	

0418	DEFD	0441	LD A, IEN	PI0 INTERRUPT ENABE
0419	DDI A	0442	OUT STAB, A	
041C	DEBF	0443	LD A, TRSL	TR1-STATE
041E	DDDD	0444	OUT ENAB, A	DATA ADDRESS
0420	CA	0445	RET	
		0446		
		0448		
		0449		
		044B		
		044C		
		044D		
		044E		
		044F		
		0450		
		0451		
		0452		
		0453		
		0454		
		0455		
		0456		
		0457		
		0458		
		0459		
		045A		
		045B		
		045C		
		045D		
		045E		
		045F		
		0460		
		0461		
		0462		
		0463		
		0464		
		0465		
		0466		
		0467		
		0468		
		0469		
		046A		
		046B		
		046C		
		046D		
		046E		
		046F		
		0470		
		0471		
		0472		
		0473		
		0474		
		0475		
		0476		
		0477		
		0478		
		0479		
		047A		
		047B		
		047C		
		047D		
		047E		
		047F		
		0480		
		0481		
		0482		
		0483		
		0484		
		0485		
		0486		
		0487		
		0488		
		0489		
		048A		
		048B		
		048C		
		048D		
		048E		
		048F		
		0490		
		0491		
		0492		
		0493		
		0494		
		0495		
		0496		
		0497		
		0498		
		0499		
		049A		
		049B		
		049C		
		049D		
		049E		
		049F		

```

0496          BUFFERS 9 Y 8
0497          ENTRADA, TECLADO
0498          SALIDA, DESPLAZADO Y BUFFERS 9 Y 8
0499          MODIFIC. A.P.C.D.E.H.L.
0500          LLAMA: CICALL, ARLO, 6FF98, LLAC1
0501
(0600) 0502          CICALL DL 0400H
0503
0504          0410H
0505          CICALL SOLICITA 2 DATOS DE 16 BITS
0506          FUENTE, CANTIDAD Y DESTINO, COLOCANDOS
0507          EN ROL, EC Y DE RESPECTIVAMENTE
0508          ENTRADA, TECLADO
0509          SALIDA, HL, 70 Y DE
0510          MODIFICA. A.P.C.D.E.H.L. Y REGISTROS ALTERNOS
0511          LLAMA: CICALL, LIMPIS
0512
(0610) 0513          DATOS DL 0A17H
0514
0515          0A17H
0516          DATOS ESPERA HASTA QUE 2 TECLAS SON PRESIONADAS
0517          (MINUTOS). LAS DOS PRIMERAS DEBEN SER
0518          DATOS Y LA TERCERA "COMI" SI "COMI" O
0519          "COMI" SON PRESIONADAS PUEDE SE DESPLAZAR
0520          ATRÁS, PERO SI ES "COMI" DEBE ESPERAR
0521          A UNA "EN ATENS CASO" (COMI, COMI O COMI)
0522          QUEDA A LA ESPERA DE RECONOCIMIENTO
0523          DE COMANDO.
0524          ENTRADA, TECLADO
0525          SALIDA, DESPLAZADO, MODIFICADO, CAMPO DE DATOS
0526          Y BUFFERS 7
0527          MODIFIC. A.P.C.D.E.H.L. Y REGISTROS ALTERNOS
0528          LLAMA: LLAC1, 6F90, LLAC, CORREL, CONVIC,
0529          LLAC2
0530
(0620) 0531          LIMPIS DL 0650H
0532
0533
0534          TABLES
0535
0536          TSI DE SCH, 04H, 00H, 00H
0537
0538          END START

```

0539 0
0540 RANGE 0001H 0

SYMBOL	VALUE	DEFN	REFERENCES
ATNPWT	0060	0042	0028
EGG	0008	0008	0091
BUFF1	0008	0020	0122 0128 0162
BUFF2	0008	0081	0122
BUFF3	0004	0183	0122
CALL	01F1	0435	0122
CHOICE	00A1	0171	0122
ENDCALL	01C0	0122	0127
CON	0079	0122	0127 0211
CONC	0080	0021	0121 0124
CONHAND	MACRO	0067	0110 0228 0231 0240 0251 0262
CONKRD	0027	0121	0228
CONL	00CF	0007	0122 0124 0228 0231
CONT	00F8	0416	0228
DATOS	0017	0017	0121 0128
DEAF	00A0	0037	0124 0221
DEPAL	0074	0435	0127
DONE	0045	0124	
DCAR	000E	0020	0028 0070 0077 0228 0232 0234 0237 0122 0123 0124
DCMB	00DF	0028	0121 0228 0401 0424 0426 0428 0431 0432 0433 0434 0437
DCMA	0000	0013	0122 0128 0042 0128 0228
DCMB	0000	0007	0114 0118 0112 0114 0240 0228 0231 0122 0227 0044 0048 0007 0007 0007 0228 0279 0431 0431
EMIT1	0008	0128	0127 0114
EMIT2	0029	0021	
EMIT3	0000	0000	
END	0020	0091	0027
END1	0002	0081	0027
FOIPWT	0027	0048	
FR	MACRO	0077	0274 0124
FRFR	20A4	0034	0124 0126 0224 0222
ICED	0017	0021	
ICMR	0027	0023	0426 0428
LINE	0000	0122	0118 0120 0020 0121 0124
LEN	0020	0029	0276 0128 0424
LEN	00F1	0021	0128 0271 0228 0410 0411
INFRNT	0028	0044	0117
IOACON	0028	0017	0120 0121
IOATLCH	0018	0018	
IYA	0028	0122	0027
IYH	0026	0010	0021 0422
TYMS	0009	0012	0432 0434
LINPR	003F	0435	0127
LTER	000E	0010	
LLC	0021	0121	0122
LOCAL	006F	0128	0122 0216 0228 0215
MCMB	000F	0014	0420
MCMA	004F	0010	0228
MCMB	000F	0016	0276 0422
MCMA	0007	0021	0426 0428
MCNTL	0028	0017	
NEA	0040	0126	0228
NEARA	00FA	0228	0222

STATEMENT OF FINANCIAL POSITION

ACCOUNT	2011	2010	2009
ASSETS			
CASH	1114	1114	1114
RENT	1114	1114	1114
DEBT	1114	1114	1114
INVEST	1114	1114	1114
OTHER	1114	1114	1114
TOTAL ASSETS	5570	5570	5570
LIABILITIES			
DEBT	1114	1114	1114
OTHER	1114	1114	1114
TOTAL LIABILITIES	2228	2228	2228
EQUITY			
RETAINED EARNINGS	1114	1114	1114
CONTRIBUTED CAPITAL	1114	1114	1114
TOTAL EQUITY	3332	3332	3332
TOTAL LIABILITIES & EQUITY	5570	5570	5570

```

0001 .
0002 . SYSTEM INTERFACE TALKER-LISTENER
0003 . CONTROL PROGRAM
0004 .
0005 . VER# 01 10 12-FEB-86 ACM
0006 .
0007 .
0008 START
(0000) 0009 B80 EQU 800H INITIALISATION
(0000) 0010 1FA EQU 0FH PROGRAM START ADDRESS
(0000) 0011 1V8A EQU 04H INTERRUPT VECTOR ADDRESS
(0000) 0012 1V8B EQU 04H INTERRUPT VECTOR HIFF
(0000) 0013 1V8C EQU 04H PORT A R/W
(0000) 0014 1V8D EQU 04H INTERRUPT VECTOR HIFF
(0000) 0015 1V8E EQU 04H PORT B
(0000) 0016 1V8F EQU 04H MODE CONTROL WORD 0 PIC
(0000) 0017 1V8G EQU 04H MODE CONTROL WORD 1 PIC
(0000) 0018 1V8H EQU 04H MODE CONTROL WORD 2 PIC
(0000) 0019 1V8I EQU 04H IO REGISTER CONTROLLER
(0000) 001A 1V8J EQU 04H CONTROL WORD
(0000) 001B 1V8K EQU 04H IO REGISTER TALKER-LISTENER
(0000) 001C 1V8L EQU 04H CONTROL WORD
(0000) 001D 1V8M EQU 04H INTERRUPT CONTROL WORD
(0000) 001E 1V8N EQU 04H PORT B DISABLE
(0000) 001F 1V8O EQU 04H INTERRUPT CONTROL HIFF
(0000) 0020 1V8P EQU 04H PORT B ENABLE
(0000) 0021 1V8Q EQU 04H MASK CONTROL WORD
(0000) 0022 1V8R EQU 04H CONTROLLER
(0000) 0023 1V8S EQU 04H MASK CONTROL WORD IN USE
(0000) 0024 1V8T EQU 04H LISTENER
(0000) 0025 1V8U EQU 04H INTERRUPT DISABLE WORD
(0000) 0026 1V8V EQU 04H A AND B PORTS
(0000) 0027 1V8W EQU 04H INTERRUPT ENABLE WORD
(0000) 0028 1V8X EQU 04H A AND B PORTS
(0000) 0029 1V8Y EQU 04H ENABLE DATA ADDRESS
(0000) 002A 1V8Z EQU 04H PORT A
(0000) 002B 1V8AA EQU 04H ENABLE CONTROL ADDRESS
(0000) 002C 1V8AB EQU 04H PORT A
(0000) 002D 1V8AC EQU 04H ENABLE DATA ADDRESS
(0000) 002E 1V8AD EQU 04H PORT B
(0000) 002F 1V8AE EQU 04H ENABLE CONTROL ADDRESS
(0000) 0030 1V8AF EQU 04H PORT B
(0000) 0031 1V8AG EQU 04H 1P-STATE PROGRAM WORD
(0000) 0032 1V8AH EQU 04H ATTENTION PROGRAM WORD
(0000) 0033 1V8AI EQU 04H TRANSMITE
(0000) 0034 1V8AJ EQU 04H INTERFACE CLEAR PROGRAM
(0000) 0035 1V8AK EQU 04H WORD TRANSMITE
(0000) 0036 1V8AL EQU 04H REMOTE ENABLE PROGRAM
(0000) 0037 1V8AM EQU 04H WORD TRANSMITE
(0000) 0038 1V8AN EQU 04H END OR IDENTIFY PROGRAM
(0000) 0039 1V8AO EQU 04H WORD TRANSMITE
(0000) 003A 1V8AP EQU 04H SERVICE REQUEST PROGRAM
(0000) 003B 1V8AQ EQU 04H WORD TRANSMITE
(0000) 003C 1V8AR EQU 04H READ PROGRAM WORD
(0000) 003D 1V8AS EQU 04H WRITE PROGRAM WORD
(0000) 003E 1V8AT EQU 04H FILE NAME ADDRESS POINTER

```

*** TITLE ***

```

(00A6) 0056 EQU 0000 SOURCE QUANTITY DESTINE
0007 , ADDRESS POINTER (FNPF+0)
(00AC) 0008 DEAF EQU 0000 DEVICE ENABLE ADDRESSES
0009 , POINTER (FNPF+8)
(0001) 0060 NSIND EQU 0000 DEVICE ADDRESS
(0006) 0061 BUFF CL 0000H
(0007) 0062 BUFFS IL 0000H
(0004) 0063 BUFFS TL 0000H
0064
0065
0066
0067

```

MACROS DEFINITION

COMMAND MACRO WINTCON

```

0070          DI          CPU INTERRUPT DISABLE
0071          LD R,TRAP    TRI-STATE
0072          OUT ED08,A    DATA ADDRESS
0073          LD R,WINTCON  COMMAND
0074          OUT ED08,A
0075
0076          MEND
0077
0078 EFC MACRO WAF0
0079
0080          FOR HL          EXCHANGE
0081          LD HL,WAF0     PROGRAM
0082          PUSH HL        COUNTER
0083          RETI
0084
0085          MEND
0086
0087 , MACROS DEFINITION IS OVER
0088
0089
0090 , MAIN PROGRAM STARTS HERE
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111

```

```

(0000) 0092 ORG 0000          BEGIN
0000 ED0E 0093 SI          INC          CPU INTERRUPT RESPONSE
0002 1E46 0094          LD R,IYA    MODE 3
0004 ED47 0095          LD R,I,A
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111

```

DEFINE INTERRUPT SERVICE ROUTINE TABLE

```

0005 1E4E 0099          LD R,0FEH
0006 00A0B 0100          LD (167AH),A    PORT A (16E)
0008 1E00 0101          LD R,0EH
0009 00A71E 0102          LD (0007H),A
0010 0E1E 0103          LD R,12H
0012 12061E 0104          LD (000EH),A    PORT B (16E)
0014 1E09 0105          LD R,0AH
0017 00A0B 0106          LD (000AH),A
0107
0108
0109
0110
0111

```

INITIALIZE PORTS

```

001A 00A00 0110          CALL LISTEN
0111

```


*** SIFLA ***

0410	CD0P05	0111	CALL IOLE	
		0112		
		0113		
		0114	CONRES	CONSOLE RECEPTION
		0115		
		0116	DISPBY MESSAGE 'SI'	
		0117		
0420	CD0P06	0118	CALL LIMPTR	MICRO KIT ROUTINE
0421	CD0P06	0119	LD R0, #1	SI TRAIL
0422	CD0P06	0120	CALL DESR1	MICRO KIT ROUTINE
0423	CD0P06	0121	CALL LLAS	MICRO KIT ROUTINE
		0122		
		0123	COMMANDS RECEPTION	
		0124		
0430	CD0P07	0125	CALL CTRLCL	MICRO KIT ROUTINE
0431	CD0P07	0126	LD R0, #0	
0432	CD0P07	0127	LD R0, #0	FILE NAME TRIM
0433	CD0P07	0128	LD R0, #0	(#0) (0) (0) (0)
0434	CD0P07	0129	LD R0, #0	
0435	CD0P07	0130	CALL CMCALL	MICRO KIT ROUTINE
0436	CD0P07	0131	LD R0, #0	SOURCE
0437	CD0P07	0132	LD R0, #0	QUANTITY
0438	CD0P07	0133	LD R0, #0	DESTINY
0439	CD0P07	0134	LD R0, #0	LOAD QUANTITY
0440	CD0P07	0135	LD R0, #0	
0441	CD0P07	0136	SEC R0, 0	
0442	CD0P07	0137	IF R0, #0	IF QUANTITY = 0
0443	CD0P07	0138	LD R0, #0	
0444	CD0P07	0139	LD R0, #0	COMPUTE QUANTITY
0445	CD0P07	0140	SEC R0, 0	0 - DESTINY - SOURCE
0446	CD0P07	0141	LD R0, #0	
0447	CD0P07	0142	LD R0, #0	DEVICES FUNCTIONS ADDRESS
0448	CD0P07	0143	LD R0, #0	PRINTER
0449	CD0P07	0144	CALL DATOS	MICRO KIT ROUTINE
0450	CD0P07	0145	LD R0, #0	
0451	CD0P07	0146	SET S0, A	DEFINE TRAFF
0452	CD0P07	0147	SET S0, A	FUNCTION NODE
0453	CD0P07	0148	LD R0, #0	
0454	CD0P07	0149	SEC IN	
0455	CD0P07	0150	LD R0, #0	NUMBER OF DEVICES
0456	CD0P07	0151	CALL DATOS	MICRO KIT ROUTINE
0457	CD0P07	0152	LD R0, #0	DEVICE ADDRESS
0458	CD0P07	0153	IF R0, #0	LAST ADDRESS
0459	CD0P07	0154	IF R0, #0	
0460	CD0P07	0155	LD R0, #0	
0461	CD0P07	0156	SET S0, A	DEFINE LISTENER
0462	CD0P07	0157	SET S0, A	FUNCTION NODE
0463	CD0P07	0158	LD R0, #0	
0464	CD0P07	0159	SEC IN	
0465	CD0P07	0160	LD R0, #0	
0466	CD0P07	0161	IF R0, #0	NEXT ADDRESS
0467	CD0P07	0162	IF R0, #0	
0468	CD0P07	0163	LD R0, #0	DEVICE DISABLE ADDRESS
0469	CD0P07	0164	SEC IN	
0470	CD0P07	0165	LD R0, #0	

*** START ***

0896	08FF	0165	OR NO READY	NEXT DEVICE
0898	0900	0167	CHOICE LD HL (DE+)	DEVICE SELECT POINTER -1
0899	089F	0168	LD B,0FA	OPTION
089C	08	0169	OTHERS INC HL	
089E	FE	0170	LD A,(HL)	INTERFACE ADDRESS
089F	EV1F	0171	AND 1FH	
08A1	8801	0172	CF NRING	ORHDY DEVICE ADDRESS
08A2	8801	0173	CF 0,0FA	
08A5	10FC	0174	COND OTHERS	NEXT OPTION
08A7	100A	0175	JR LOCAL	SOLE STATE
08A9	FE	0176	OR LD A,(HL)	
08AA	08FF	0177	RIT 0,0	ORHDY DEVICE FUNCTION
08AC	1000	0178	JR NO,RECEIVE	RECEIVE
08AE	08FF	0179	BIT 0,A	
08B0	100A	0180	JR NO,EMIT	TRANSMITE
08B2	FE	0181	LOCAL DI	
08B3	0D0000	0182	CALL LISTEN	
08B6	0D0000	0183	CALL IDLE	
08B8	0D0000	0184	IF 0,0FA	MONITOR
		0185		
		0186		
		0187	TRANSMITE INFORMATION	
		0188		
		0189		
08C0	FE	0189	EMIT DI	TRANSMITE DATA
08C2	0D0000	0190	CALL COM	
08C3	0D0000	0191	CALL WRITE	
08C4	FE	0192	EI	
08C4	8801	0193	EMIT LD HL (DE+)	HL SOURCE ADDRESS
08C5	ED180000	0194	LD BC,(DE+0)	BC QUANTITY
08C6	78	0195	NEXT LD A,(HL)	BYTE TRANSFERRED
08C8	0800	0196	OUT 0FA,A	DATA BYTE
08CA	78	0197	RLT	
08CB	00	0198	INC C	
08CD	00	0199	DEC C	
08CE	08FA	0200	OR NO,EMIT	NEXT DATA BYTE
08D0		0201	COMMAND EQUIP	
		0202		
		0203		
08D7	FE	0203	DI	CPU INTERRUPT DISABLE
08D8	08FF	0204	LD A,TFW	TRI-STATE
08DA	0100	0205	OUT 0FA,A	DATA ADDRESS
08DB	0800	0206	LD A,EMIT	COMMAND
08DA	0800	0207	OUT 0FA,A	
		0208		
08E0	100A	0209	JR LOCAL	
		0210		
		0211		
		0212	RECEIVE INFORMATION	
		0213		
		0214		
08F0	8801	0214	RECEIVE CALL COM	TRANSMITE DATA
08F1	FE	0215	RECEIVE DI	
08F2	0D0000	0216	CALL LISTEN	
08F3	0D0000	0217	CALL READ	
08F4	FE	0218	EI	
08F5	8801	0219	LD BC (DE+)	BC LISTING ADDRESS
08F6	ED180000	0220	LD BC (DE+0)	BC QUANTITY

```

08F1 0800      0221  NEARR      IN A,EDAA      DATA BYTE RECEIVED
08F2 76        0222      ANLT
08F3 00        0223      INC C
08F4 00        0224      DEC C
08F5 20FF     0225      JR NZ,NEARR    NEXT DATA BYTE
08F6 000000    0226      CALL 10E      CALL 10E
08F7 010000    0227      JP 0000       MONITOR
                                0228
                                0229
                                0230      INTERRUPT SUBROUTINE RESPONSE PORT A
                                0231
08FE 76        0232      DI           CPU INTERRUPT DISABLE
08FF 0E00     0233      IN A,EDAB
0901 0B47     0234      BIT 0,A      CHECK DEVICE FUNCTION
0902 2008     0235      JR Z,TALKR
0903 2E0C     0236      LTR, IN A,EDAB DATA BYTE RECEIVE AND
                                STORED
0907 10      0237      LD 10E,A
0908 10      0238      INC DE
0909 0E      0239      DEC EC
090A 76      0240      EI           CPU INTERRUPT ENABLE
090B 0D40     0241      RETI
                                0242
090D 23      0243      TALKR, INC HL NEXT DATA BYTE LOCATION
090E 0E      0244      DEC EC
090F 76      0245      EI           CPU INTERRUPT ENBLE
0910 0D40     0246      RETI
                                0247
                                0248
                                0249      INTERRUPT SUBROUTINE RESPONSE PORT B
                                0250
0912 76      0251      DI           CPU INTERRUPT DISABLE
0913 0E00     0252      IN A,EDAB
0914 0E0F     0253      SET 0,A
0917 0E0F     0254      SET 3,A      STABILIZE
0918 0E0D     0255      OUT ED06,A
091B 0E4F     0256      BIT 1,A
091C 2008     0257      JR Z,ATN     CHECK
091F 0E37     0258      BIT 2,A
0921 2023     0259      JR Z,IFC     DEVICE
0923 0E57     0260      BIT 4,A
0925 2024     0261      JR Z,REN     STATUS
0927 76      0262      EI
0928 0D40     0263      RETI
                                0264
                                0265
                                0266      SERVICE ROUTINES
                                0267
092A      0268      ATN, EPC ATN1
                                0269+
092B 76      0270+     POP HL      EXCHANGE
092C 2110F    0271+     LD HL,ATN1  PROGRAM
092E 0E      0272+     PUSH HL    COUNTER
092F 0D40     0273+     RETI
                                0274+
0971 000000    0275      ATN, CALL LISTEN
  
```

0214	00A000	0076	CALL ADDR	
0217	00	0077	BT	
0218	11A000	0078	LD R0, ADDR	COMMANDS POINTER
0219	011700	0079	LD R0, ITR	OPTIONS
0220	0000	0080	MOV R0, ADDR	INTERFACE ADDRESS OR
0241	75	0081	HALT	COMMAND
0241	00	0082	INC C	
0242	00	0083	DEC C	
0243	00F0	0084	JR NEXTADR	NEXT ADDR OR COMMAND
0240	000000	0085	CALL ITR	
0240	000000	0086	JR COMPED	
		0087		
0248		0088	JFO	8FC IFO1
		0089+		
0248	01	0090+	POP HL	EXCHANGE
0249	000000	0091+	LD HL, IFO1	PROGRAM
0249	00	0092+	PUSH HL	COUNTER
0250	0000	0093+	RET	
		0094+		
0250	000000	0095	CALL IFO1	CALL IFO1
0250	000000	0096	CALL ITR	CALL ITR
0250	000000	0097	JR COMPED	JR COMPED
		0098		
0250		0099	MOV	8FC R0R1
		009A+		
0250	01	009B+	POP HL	EXCHANGE
0250	000000	009C+	LD HL, R0R1	PROGRAM
0250	00	009D+	PUSH HL	COUNTER
0250	0000	009E+	RET	
		009F+		
0250	11A000	00A0	MOV R0, ADDR	DEVICE FUNCT. POINTER
0250	0000	00A1	LD R0, ADDR	OPTIONS
0250	00	00A2	OTHER	
0250	75	00A3	LD R0, ADDR	INTERFACE ADDRESS
0250	0000	00A4	AND IFO	
0250	0000	00A5	CP ADDR	CHECK DEVICE ADDRESS
0250	0000	00A6	TR 0, 0	
0250	00F0	00A7	CONE, OTHER	NEXT OPTION
0271	000000	00A8	JR LOCAL	IDLE STATE
0271	75	00A9	OR	LD R0, ADDR
0270	0000	00B0	BIT 0, R0	CHECK DEVICE FUNCTION
0270	000000	00B1	JR NZ, RECEIVE	RECEIVE
0270	0000	00B2	LD R0, ADDR	
0270	000000	00B3	BIT 0, R0	TRANSMIT
0270	000000	00B4	JR NZ, ENITEL	IDLE STATE
		00B5		
		00B6		
		00B7		
		00B8		
		00B9		
		00BA		
		00BB		
		00BC		
		00BD		
		00BE		
		00BF		
		00C0		
		00C1		
		00C2		
		00C3		
		00C4		
		00C5		
		00C6		
		00C7		
		00C8		
		00C9		
		00CA		
		00CB		
		00CC		
		00CD		
		00CE		
		00CF		
		00D0		
		00D1		
		00D2		
		00D3		
		00D4		
		00D5		
		00D6		
		00D7		
		00D8		
		00D9		
		00DA		
		00DB		
		00DC		
		00DD		
		00DE		
		00DF		
		00E0		
		00E1		
		00E2		
		00E3		
		00E4		
		00E5		
		00E6		
		00E7		
		00E8		
		00E9		
		00EA		
		00EB		
		00EC		
		00ED		
		00EE		
		00EF		
		00F0		
		00F1		
		00F2		
		00F3		
		00F4		
		00F5		
		00F6		
		00F7		
		00F8		
		00F9		
		00FA		
		00FB		
		00FC		
		00FD		
		00FE		
		00FF		

0987	0987	0014+	LD A, EDPINT	COMMAND
0988	0988	0015+	OUT EDPAB, A	
		0016+		
0989	00810A	0017	CALL TRM	
0990	008B09	0018	CALL WRTE	
0991	FB	0019	EI	CPU INTERRUPT ENABLE
0992	01A1E8	001F	ENTEL LD HL, ENAP	COMMAND POINTER
0993	011700	0020	LD BC, 17H	OPTIONS
0994	78	0021	MENT LD A, 78H	INTERFACE ADDRESS OR
0995	0000	0022	OUT EDAB, A	COMMAND
0996	78	0023	HALT	
0997	80	0024	INC C	
0998	80	0025	DEC D	
0999	00F3	0026	JR NO, NEXT	NEXT ADDS OR COMMAND
09A0		0027	COMMAND EQUENT	
		0028+		
09A1	FB	002F	DI	CPU INTERRUPT DISABLE
09A2	00FF	0030+	LD A, 7FH	TRI-STATE
09A3	0000	0031+	OUT EDAB, A	DATA ADDRESS
09A4	0087	0032+	LD A, 80H	COMMAND
09A5	0000	0033+	OUT EDAB, A	
		0034+		
09A6	000009	0035	CALL 00E	
09A7	09	0036	RET	
		0037		
09A8		0038	REHD COMMAND REHDW	ENABLE
		0039+		
09A9	FB	003E+	DI	CPU INTERRUPT DISABLE
09AA	00FF	003F+	LD A, 7FH	TRI-STATE
09AB	0000	0040+	OUT EDAB, A	DATA ADDRESS
09AC	000F	0041+	LD A, 0FH	COMMAND
09AD	0000	0042+	OUT EDAB, A	
		0043+		
09AE	0000	0044	LD A, 15H	FID INTERRUPT ENABLE
09AF	000E	0045	OUT EDAB, A	CONTROL ADDRESS
09B0	0E	0046	RET	
		0047		
09B1		0048	WRITE COMMAND WRTEW	ENABLE
		0049+		
09B2	FB	0050+	DI	CPU INTERRUPT DISABLE
09B3	00FF	0051+	LD A, 7FH	TRI-STATE
09B4	0000	0052+	OUT EDAB, A	DATA ADDRESS
09B5	000E	0053+	LD A, 0EH	COMMAND
09B6	0000	0054+	OUT EDAB, A	
		0055+		
09B7	0000	0056	LD A, 15H	FID INTERRUPT ENABLE
09B8	000E	0057	OUT EDAB, A	CONTROL ADDRESS
09B9	0E	0058	RET	
		0059		
09BA	FB	005A	FILE DI	CPU INTERRUPT DISABLE
09BB	00FF	005B	LD A, 15H	FID INTERRUPT DISABLE
09BC	0000	005C	OUT EDAB, A	CONTROL ADDRESS
09BD	000E	005D	LD A, 0EH	TRI-STATE
09BE	0000	005E	OUT EDAB, A	DATA ADDRESS
09BF	0000	005F	LD A, 15H	FID INTERRUPT ENABLE

0001	0001	0100	OUT SCAR.A	CONTROL ADDRESS
0002	0002	0101	BT	INTERRUPT DISABLE
0003	0003	0102	RET	
		0103		
		0104		
		0105		
		0106		
		0107		
		0108		
		0109		
		0110		
		0111		
		0112		
		0113		
		0114		
		0115		
		0116		
		0117		
		0118		
		0119		
		0120		
		0121		
		0122		
		0123		
		0124		
		0125		
		0126		
		0127		
		0128		
		0129		
		0130		
		0131		
		0132		
		0133		
		0134		
		0135		
		0136		
		0137		
		0138		
		0139		
		0140		
		0141		
		0142		
		0143		
		0144		
		0145		
		0146		
		0147		
		0148		
		0149		
		0150		
		0151		
		0152		
		0153		
		0154		
		0155		
		0156		
		0157		
		0158		
		0159		
		0160		
		0161		
		0162		
		0163		
		0164		
		0165		
		0166		
		0167		
		0168		
		0169		
		0170		
		0171		
		0172		
		0173		
		0174		
		0175		
		0176		
		0177		
		0178		
		0179		
		0180		
		0181		
		0182		
		0183		
		0184		
		0185		
		0186		
		0187		
		0188		
		0189		
		0190		
		0191		
		0192		
		0193		
		0194		
		0195		
		0196		
		0197		
		0198		
		0199		
		0200		

```

0414 DEPT 0411 LE A 1000A . INTERRUPT ON
0415 DDFP 0412 OUT 8000A . OR PRODUCTION
0416 0413 . ACTIVE LEVEL LOW
0417 DEEP 0414 LE A 1001A . DASH CONTROL WORD
0418 DDFP 0415 OUT 8000A .
0419 DDFP 0416 LE A 1001A . PIC INTERRUPT ENABLE
0420 DDFP 0417 OUT 8000A .
0421 DDFP 0418 LE A 1001A . TRI-STATE
0422 DDFP 0419 OUT 8000A . DATA ADDRESS
0423 DDFP 0420 RET .
0424
0425 . MICRO KIT ROUTINES
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900

```

```

0096 . DIRECCIONES. SI SE TECLEA. MAS DATOS
0097 . LOS VA RECONOCIENDO A LA IGUALDA
0098 . DEJANDO AL NUEVO A LA CATEGORIA. PERO SI
0099 . LA TECLA ES "OTRO" REGRESA CON LOS
0100 . DATOS QUE ESTAN DESPLEGADOS EN LOS
0101 . BUFFERS 9 Y 8
0102 . ENTRADA. TECLADO
0103 . SALIDA. DESPLEGADO Y BUFFERS 9 Y 8
0104 . MODIFICA. A.F.P.C.D.E.H.
0105 . LLAMA. CUALI. APID. BARRA. LLACI
0106
(0040) 0107 . CUALI. DE. BARRA
0108
0109 . CUALI
0110 . CUALI. SOLICITA. Y DATOS DE 14 BITS
0111 . FUENTE. CANTIDAD Y DESTINO. CULCANDOLC
0112 . EN. AL. 80 Y DE. RESPECTIVAMENTE
0113 . ENTRADA. TECLADO
0114 . SALIDA. AL. 80 Y DE
0115 . MODIFICA. A.F.P.C.D.E.H. Y REGISTROS ALTERNOS
0116 . LLAMA. DEPL. CUALI. LIMP
0117
(0017) 0118 . DATOS. DE. BARRA
0119
0120 . BARRA
0121 . DRIVE. ESPERA HASTA QUE 3 TECLAS SON PRESIONADAS
0122 . (MINIMO). LAS DOS PRIMERAS DEBE. SER
0123 . DATOS Y LA TERCERA "OTRO". SI "OTRO" O
0124 . "OTRO" SON PRESIONADAS ANTI. SE DESPLEGA
0125 . BARRA. PERO SI ES "CUALI" DESPLEGARA
0126 . HOLA Y EN AMBOS CAROS (80. COMO 0. COMO
0127 . BARRA A LA RUTINA DE RECONOCIMIENTO
0128 . DE. COMANDOS
0129 . ENTRADA. TECLADO
0130 . SALIDA. DESPLEGADO (MODIFICANDO CAMPO DE DATOS)
0131 . Y BUFFER 7
0132 . MODIFICA. A.F.P.C.D. Y REGISTROS ALTERNOS
0133 . LLAMA. LLACI. ALD80. LLACI. CUALI. CONVAI.
0134 . LLACI
0135
(0000) 0136 . COMI. DE. BARRA
0137
0138 . BARRA
0139 . TABLES
0140
0141 . DE. BARRA. COMI. BARRA. BARRA
0142
0143 . END START
0144
ERRORS . 0
RANGE COUNT . 0
  
```


CONCLUSIONES

CONCLUSIONES

La Comunicacion de Datos ha llegado a ser importante con el incremento del uso de computadoras, y se han continuado desarrollando hacia una mayor industria, proporcionando la interconexión de perifericos y la transmisión de datos a sitios distantes.

La comunicación de datos depende grandemente de la electrónica digital, el sistema binario puede considerarse virtualmente sinonimo de electrónica digital. La comunicación de datos es una industria de soporte para el procesamiento de datos y una industria independiente que permite a otros campos tomar ventaja de los avances de la transmisión que desarrolla. Los datos son el conjunto básico de comunicaciones de datos.

El valor de las técnicas digitales se deriva de la habilidad de construir códigos únicos para representar diferentes grupos de información. Estos códigos son los lenguajes de las computadoras y de otro tipo de equipo electrónico que ha revolucionado la sociedad moderna. El procesamiento digital es usado tan ampliamente puesto que provee una manipulación de datos rápida y eficiente. Si la información es reducida a un conjunto de códigos 'standard' los cuales pueden ser representados por palabras binarias, la información puede ser llevada por dispositivos electrónicos digitales. Lo que es realizado en

computadoras y sistemas digitales de proceso.

Desde el punto de vista de hacer buen uso de los recursos, el diseñador de sistemas tiene normalmente la obligación de utilizar como componentes de sistema a aquellos que funcionan satisfactoriamente al menor precio efectivo.

La selección de un sistema operativo adecuado es crucial para el eficiente funcionamiento del sistema. Ya sea que el usuario pretenda crear sus propios programas de aplicación o se decida a comprarlos. El hardware de la computadora por sí solo, no hace un sistema y no tiene gran valor sin un sistema operativo y sus librerías. Los programas de aplicación hacen uso extensivo de las facilidades que proporciona el sistema operativo, especialmente el manejo de archivos.

Los diseñadores deben considerar el amplio rango de problemas concernientes con el desarrollo de software y deben elegir los métodos más simples y los más eficientes.

El mayor costo en la mayoría de los proyectos de microprocesadores es el tiempo de programación, por lo que los métodos para minimizar el tiempo requerido para completar el programa son especialmente importantes.

Los sistemas operativos tienen un efecto considerable sobre los programas y la manera en que los programadores los diseñan. Ya

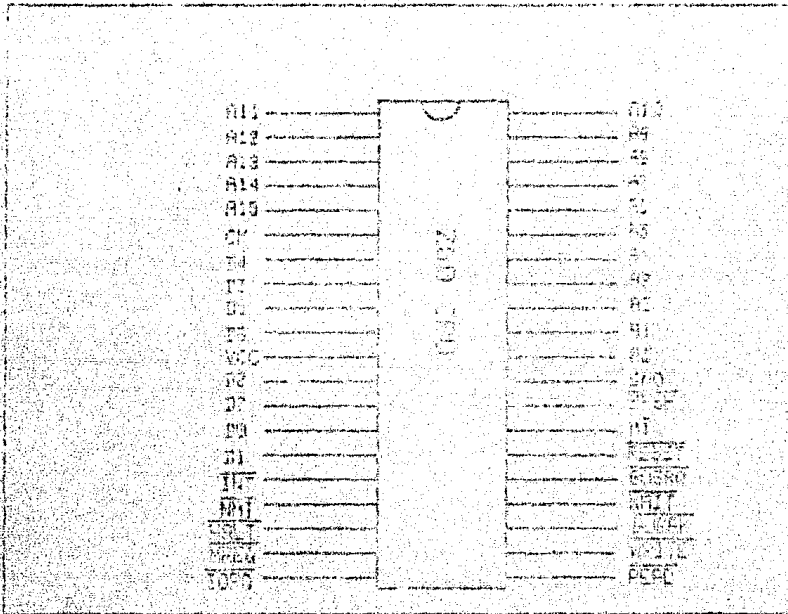
que el manejo del equipo periférico es una de las mayores responsabilidades del sistema operativo. El uso de sistemas de control de entrada/salida es una de las maneras más significativas en que el programador logra ventaja del sistema operativo.

Una de las características de mayor importancia del uso de sistemas operativos es la necesidad de personal especializado (programadores de sistemas) para mantenerlo y mejorarlo, así como, ayudar a los programadores de aplicaciones y operadores a obtener un buen servicio de él.

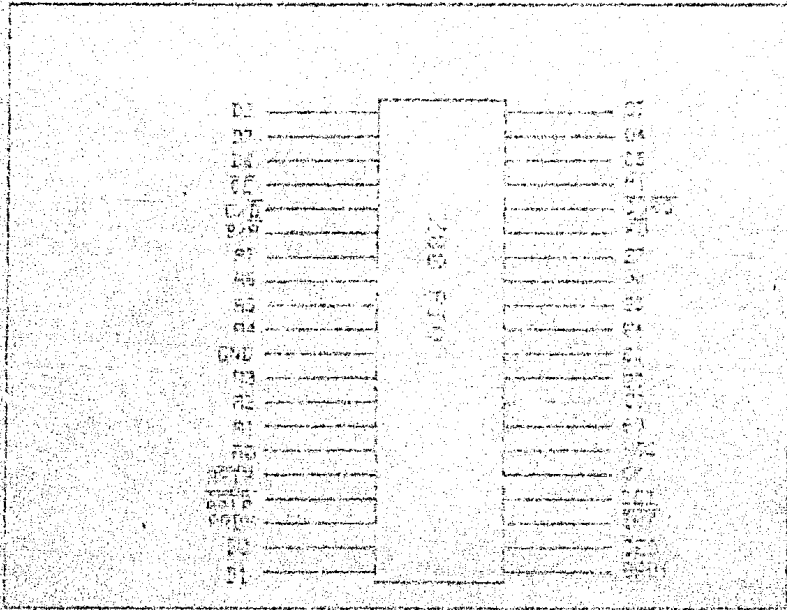
El análisis de un sistema de comunicación consiste en la evaluación de un sistema de comunicación dado, mientras que el diseño consiste en satisfacer los detalles de un sistema para realizar una tarea satisfactoriamente. Ya que no es posible definir claramente donde el análisis y el diseño comienzan, uno puede decir con seguridad que uno tiene que aprender a analizar un sistema de comunicación antes de aprender a diseñarlo.

APENDICES

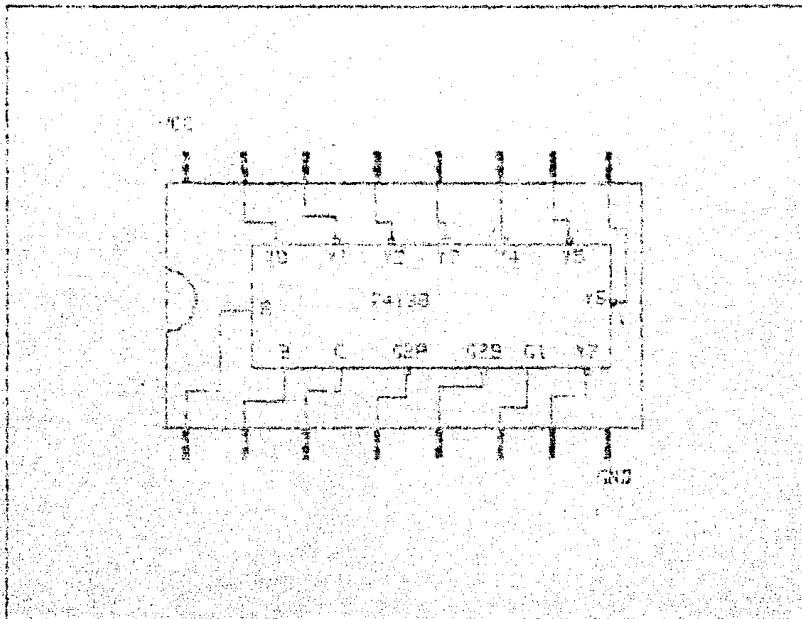
DIAGRAMAS DE BASE DE LOS CIRCUITOS INTEGRADOS Y CONECTORES
UTILIZADOS EN EL DISEÑO DEL SISTEMA INTERFACE en base al standard
HEWLETT-PACKARD INTERFACE BUS (HP-IB).



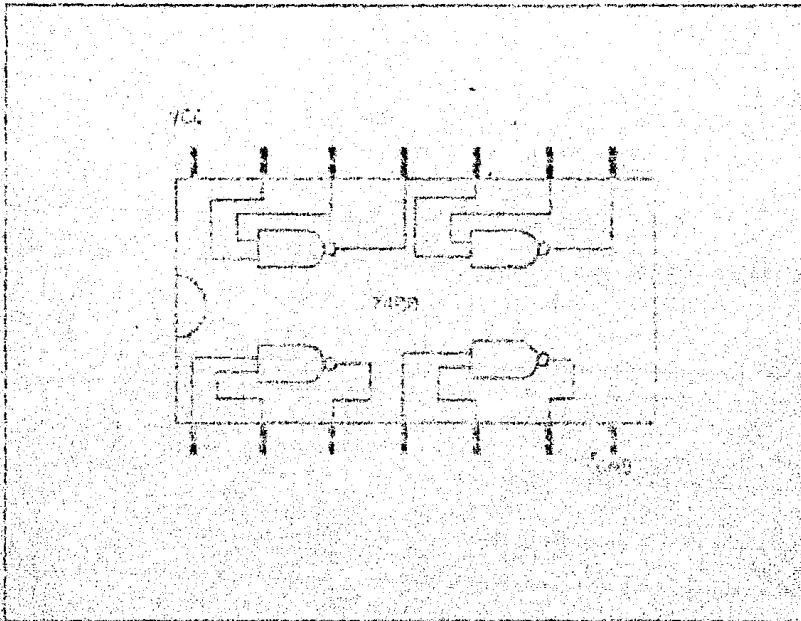
CENTRAL PROCESSING UNIT Z80 CPU



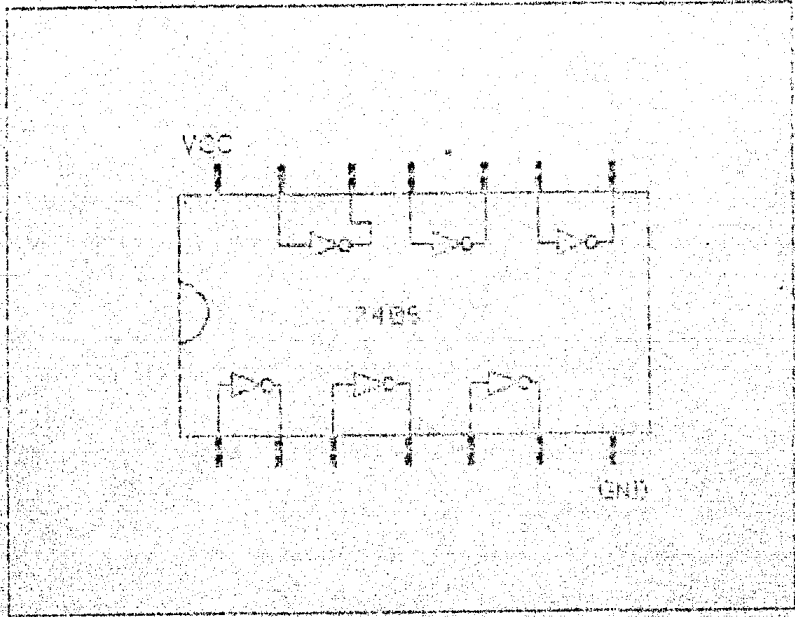
PARALLEL I/O CONTROLLER 280 PIO



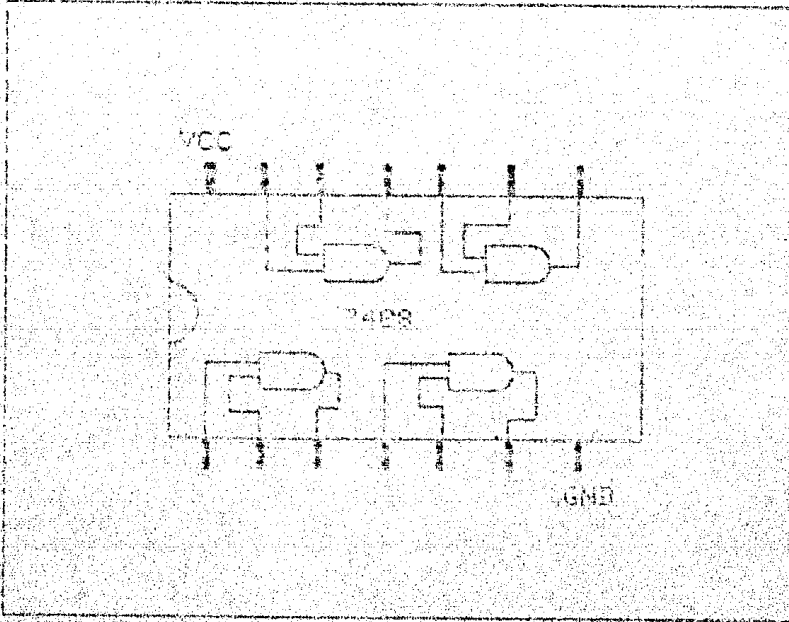
3 TO 8 LINE DECODERS/MULTIPLEXERS



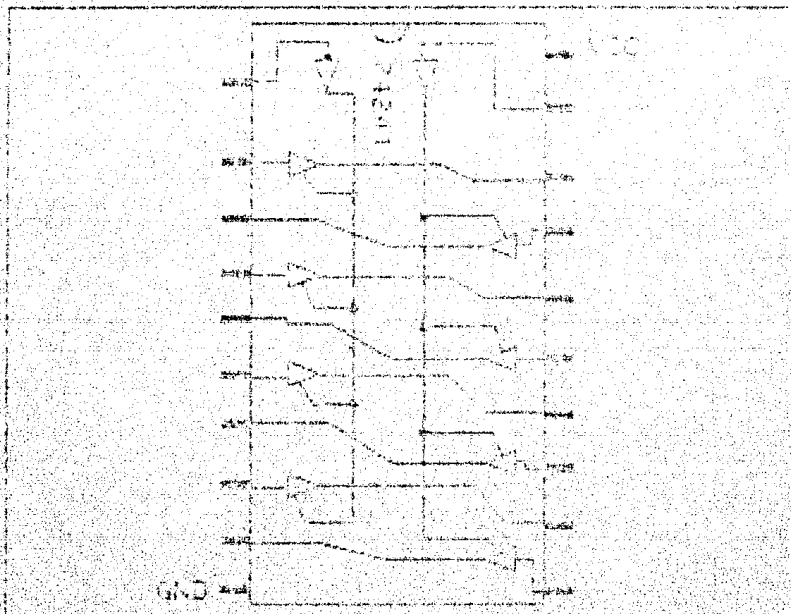
QUADRUPLA 2 INPUT POSITIVE NAND GATES



HEX INVERTER BUFFERS/DRIVERS
WITH OPEN COLECTOR
HIGH VOLTAGE OUTPUTS



QUADRUPLE 2 INPUT POSITIVE AND GATES



OCTAL BUFFERS/LINE DRIVERS/LINE RECEIVERS

HP-IB CABLE CONNECTOR PIN ASSIGNMENT

		DEFINITION	
GROUND	0100	1 1 10	GROUND
DATA 0	0101	2 1 11	DATA 0
DATA 1	0110	3 1 12	DATA 1
DATA 2	0111	4 1 13	DATA 2
DATA 3	1000	5 1 14	DATA 3
DATA 4	1001	6 1 15	DATA 4
DATA 5	1010	7 1 16	DATA 5
DATA 6	1011	8 1 17	DATA 6
DATA 7	1100	9 1 18	DATA 7
DATA 8	1101	10 1 19	DATA 8
DATA 9	1110	11 1 20	DATA 9
DATA 10	1111	12 1 21	DATA 10
DATA 11	0000	13 1 22	DATA 11
DATA 12	0001	14 1 23	DATA 12
DATA 13	0010	15 1 24	DATA 13
DATA 14	0011	16 1 25	DATA 14
DATA 15	0100	17 1 26	DATA 15
DATA 16	0101	18 1 27	DATA 16
DATA 17	0110	19 1 28	DATA 17
DATA 18	0111	20 1 29	DATA 18
DATA 19	1000	21 1 30	DATA 19
DATA 20	1001	22 1 31	DATA 20
DATA 21	1010	23 1 32	DATA 21
DATA 22	1011	24 1 33	DATA 22
DATA 23	1100	25 1 34	DATA 23
DATA 24	1101	26 1 35	DATA 24
DATA 25	1110	27 1 36	DATA 25
DATA 26	1111	28 1 37	DATA 26
DATA 27	0000	29 1 38	DATA 27
DATA 28	0001	30 1 39	DATA 28
DATA 29	0010	31 1 40	DATA 29
DATA 30	0011	32 1 41	DATA 30
DATA 31	0100	33 1 42	DATA 31
DATA 32	0101	34 1 43	DATA 32
DATA 33	0110	35 1 44	DATA 33
DATA 34	0111	36 1 45	DATA 34
DATA 35	1000	37 1 46	DATA 35
DATA 36	1001	38 1 47	DATA 36
DATA 37	1010	39 1 48	DATA 37
DATA 38	1011	40 1 49	DATA 38
DATA 39	1100	41 1 50	DATA 39
DATA 40	1101	42 1 51	DATA 40
DATA 41	1110	43 1 52	DATA 41
DATA 42	1111	44 1 53	DATA 42
DATA 43	0000	45 1 54	DATA 43
DATA 44	0001	46 1 55	DATA 44
DATA 45	0010	47 1 56	DATA 45
DATA 46	0011	48 1 57	DATA 46
DATA 47	0100	49 1 58	DATA 47
DATA 48	0101	50 1 59	DATA 48
DATA 49	0110	51 1 60	DATA 49
DATA 50	0111	52 1 61	DATA 50
DATA 51	1000	53 1 62	DATA 51
DATA 52	1001	54 1 63	DATA 52
DATA 53	1010	55 1 64	DATA 53
DATA 54	1011	56 1 65	DATA 54
DATA 55	1100	57 1 66	DATA 55
DATA 56	1101	58 1 67	DATA 56
DATA 57	1110	59 1 68	DATA 57
DATA 58	1111	60 1 69	DATA 58
DATA 59	0000	61 1 70	DATA 59
DATA 60	0001	62 1 71	DATA 60
DATA 61	0010	63 1 72	DATA 61
DATA 62	0011	64 1 73	DATA 62
DATA 63	0100	65 1 74	DATA 63
DATA 64	0101	66 1 75	DATA 64
DATA 65	0110	67 1 76	DATA 65
DATA 66	0111	68 1 77	DATA 66
DATA 67	1000	69 1 78	DATA 67
DATA 68	1001	70 1 79	DATA 68
DATA 69	1010	71 1 80	DATA 69
DATA 70	1011	72 1 81	DATA 70
DATA 71	1100	73 1 82	DATA 71
DATA 72	1101	74 1 83	DATA 72
DATA 73	1110	75 1 84	DATA 73
DATA 74	1111	76 1 85	DATA 74
DATA 75	0000	77 1 86	DATA 75
DATA 76	0001	78 1 87	DATA 76
DATA 77	0010	79 1 88	DATA 77
DATA 78	0011	80 1 89	DATA 78
DATA 79	0100	81 1 90	DATA 79
DATA 80	0101	82 1 91	DATA 80
DATA 81	0110	83 1 92	DATA 81
DATA 82	0111	84 1 93	DATA 82
DATA 83	1000	85 1 94	DATA 83
DATA 84	1001	86 1 95	DATA 84
DATA 85	1010	87 1 96	DATA 85
DATA 86	1011	88 1 97	DATA 86
DATA 87	1100	89 1 98	DATA 87
DATA 88	1101	90 1 99	DATA 88
DATA 89	1110	91 1 100	DATA 89
DATA 90	1111	92 1 101	DATA 90
DATA 91	0000	93 1 102	DATA 91
DATA 92	0001	94 1 103	DATA 92
DATA 93	0010	95 1 104	DATA 93
DATA 94	0011	96 1 105	DATA 94
DATA 95	0100	97 1 106	DATA 95
DATA 96	0101	98 1 107	DATA 96
DATA 97	0110	99 1 108	DATA 97
DATA 98	0111	100 1 109	DATA 98
DATA 99	1000	101 1 110	DATA 99
DATA 100	1001	102 1 111	DATA 100
DATA 101	1010	103 1 112	DATA 101
DATA 102	1011	104 1 113	DATA 102
DATA 103	1100	105 1 114	DATA 103
DATA 104	1101	106 1 115	DATA 104
DATA 105	1110	107 1 116	DATA 105
DATA 106	1111	108 1 117	DATA 106
DATA 107	0000	109 1 118	DATA 107
DATA 108	0001	110 1 119	DATA 108
DATA 109	0010	111 1 120	DATA 109
DATA 110	0011	112 1 121	DATA 110
DATA 111	0100	113 1 122	DATA 111
DATA 112	0101	114 1 123	DATA 112
DATA 113	0110	115 1 124	DATA 113
DATA 114	0111	116 1 125	DATA 114
DATA 115	1000	117 1 126	DATA 115
DATA 116	1001	118 1 127	DATA 116
DATA 117	1010	119 1 128	DATA 117
DATA 118	1011	120 1 129	DATA 118
DATA 119	1100	121 1 130	DATA 119
DATA 120	1101	122 1 131	DATA 120
DATA 121	1110	123 1 132	DATA 121
DATA 122	1111	124 1 133	DATA 122
DATA 123	0000	125 1 134	DATA 123
DATA 124	0001	126 1 135	DATA 124
DATA 125	0010	127 1 136	DATA 125
DATA 126	0011	128 1 137	DATA 126
DATA 127	0100	129 1 138	DATA 127
DATA 128	0101	130 1 139	DATA 128
DATA 129	0110	131 1 140	DATA 129
DATA 130	0111	132 1 141	DATA 130
DATA 131	1000	133 1 142	DATA 131
DATA 132	1001	134 1 143	DATA 132
DATA 133	1010	135 1 144	DATA 133
DATA 134	1011	136 1 145	DATA 134
DATA 135	1100	137 1 146	DATA 135
DATA 136	1101	138 1 147	DATA 136
DATA 137	1110	139 1 148	DATA 137
DATA 138	1111	140 1 149	DATA 138
DATA 139	0000	141 1 150	DATA 139
DATA 140	0001	142 1 151	DATA 140
DATA 141	0010	143 1 152	DATA 141
DATA 142	0011	144 1 153	DATA 142
DATA 143	0100	145 1 154	DATA 143
DATA 144	0101	146 1 155	DATA 144
DATA 145	0110	147 1 156	DATA 145
DATA 146	0111	148 1 157	DATA 146
DATA 147	1000	149 1 158	DATA 147
DATA 148	1001	150 1 159	DATA 148
DATA 149	1010	151 1 160	DATA 149
DATA 150	1011	152 1 161	DATA 150
DATA 151	1100	153 1 162	DATA 151
DATA 152	1101	154 1 163	DATA 152
DATA 153	1110	155 1 164	DATA 153
DATA 154	1111	156 1 165	DATA 154
DATA 155	0000	157 1 166	DATA 155
DATA 156	0001	158 1 167	DATA 156
DATA 157	0010	159 1 168	DATA 157
DATA 158	0011	160 1 169	DATA 158
DATA 159	0100	161 1 170	DATA 159
DATA 160	0101	162 1 171	DATA 160
DATA 161	0110	163 1 172	DATA 161
DATA 162	0111	164 1 173	DATA 162
DATA 163	1000	165 1 174	DATA 163
DATA 164	1001	166 1 175	DATA 164
DATA 165	1010	167 1 176	DATA 165
DATA 166	1011	168 1 177	DATA 166
DATA 167	1100	169 1 178	DATA 167
DATA 168	1101	170 1 179	DATA 168
DATA 169	1110	171 1 180	DATA 169
DATA 170	1111	172 1 181	DATA 170
DATA 171	0000	173 1 182	DATA 171
DATA 172	0001	174 1 183	DATA 172
DATA 173	0010	175 1 184	DATA 173
DATA 174	0011	176 1 185	DATA 174
DATA 175	0100	177 1 186	DATA 175
DATA 176	0101	178 1 187	DATA 176
DATA 177	0110	179 1 188	DATA 177
DATA 178	0111	180 1 189	DATA 178
DATA 179	1000	181 1 190	DATA 179
DATA 180	1001	182 1 191	DATA 180
DATA 181	1010	183 1 192	DATA 181
DATA 182	1011	184 1 193	DATA 182
DATA 183	1100	185 1 194	DATA 183
DATA 184	1101	186 1 195	DATA 184
DATA 185	1110	187 1 196	DATA 185
DATA 186	1111	188 1 197	DATA 186
DATA 187	0000	189 1 198	DATA 187
DATA 188	0001	190 1 199	DATA 188
DATA 189	0010	191 1 200	DATA 189
DATA 190	0011	192 1 201	DATA 190
DATA 191	0100	193 1 202	DATA 191
DATA 192	0101	194 1 203	DATA 192
DATA 193	0110	195 1 204	DATA 193
DATA 194	0111	196 1 205	DATA 194
DATA 195	1000	197 1 206	DATA 195
DATA 196	1001	198 1 207	DATA 196
DATA 197	1010	199 1 208	DATA 197
DATA 198	1011	200 1 209	DATA 198
DATA 199	1100	201 1 210	DATA 199
DATA 200	1101	202 1 211	DATA 200
DATA 201	1110	203 1 212	DATA 201
DATA 202	1111	204 1 213	DATA 202
DATA 203	0000	205 1 214	DATA 203
DATA 204	0001	206 1 215	DATA 204
DATA 205	0010	207 1 216	DATA 205
DATA 206	0011	208 1 217	DATA 206
DATA 207	0100	209 1 218	DATA 207
DATA 208	0101	210 1 219	DATA 208
DATA 209	0110	211 1 220	DATA 209
DATA 210	0111	212 1 221	DATA 210
DATA 211	1000	213 1 222	DATA 211
DATA 212	1001	214 1 223	DATA 212
DATA 213	1010	215 1 224	DATA 213
DATA 214	1011	216 1 225	DATA 214
DATA 215	1100	217 1 226	DATA 215
DATA 216	1101	218 1 227	DATA 216
DATA 217	1110	219 1 228	DATA 217
DATA 218	1111	220 1 229	DATA 218
DATA 219	0000	221 1 230	DATA 219
DATA 220	0001	222 1 231	DATA 220
DATA 221	0010	223 1 232	DATA 221
DATA 222	0011	224 1 233	DATA 222
DATA 223	0100	225 1 234	DATA 223
DATA 224	0101	226 1 235	DATA 224
DATA 225	0110	227 1 236	DATA 225
DATA 226	0111	228 1 237	DATA 226
DATA 227	1000	229 1 238	DATA 227
DATA 228	1001	230 1 239	DATA 228
DATA 229	1010	231 1 240	DATA 229
DATA 230	1011	232 1 241	DATA 230
DATA 231	1100	233 1 242	DATA 231
DATA 232	1101	234 1 243	DATA 232
DATA			

RELACION DE CONEXIONES

Z80 conector en paralelo con Z80 conector
 'HP-IB' conector en paralelo con 'HP-IB' conector

	Z80		Z80	
	CPU		PIO	
	6	CK	25	
	7	D4	39	
	8	D3	40	
	9	D5	38	
	10	D6	3	
	11	VCC	26	
	12	D2	1	
	13	D7	2	
	14	D0	19	
	15	D1	20	
*	16	INT	23	
*	20	IORD	36	
*	21	RD	35	
*	27	M1	37	
	29	GND	11	
A0	30		6	B/A
A1	31		5	C/D

NOTA: TODAS LAS SEÑALES ACTIVAS EN CERO SE SENALAN CON UN ASTERISCO

	Z80		74138	
	CPU			
A2	32		1	A
A3	33		2	B
A4	34		3	C
A5	35		4	G2A *
A6	36		6	G1
A7	37		5	G2B *
	74138		Z80	
			PIO	
* Y7	7		4	CE *

	Z80 PIO	74241 (1)	74241 (2)
A7	7	8,9	
A6	8	6,7	
A5	9	4,5	
A4	10	2,3	8,9
A3	12		6,7
A2	13		4,5
A1	14		2,3
A0	15		

* ASTB ARDY
 IEI
 Z80 PID 16 18
 74241 (5) 9,16
 7408 2
 7406 1
 7400 1
 24 a VCC (26 PIO) a través de una resistencia

	Z80 PIO	74241 (1)	74241 (2)	74241 (3)	74241 (4)	74241 (5)
B0	27	1,19	1,19	7	8	1,19
B1	28			5	6	
B2	29			8	9	
B3	30			3	4	
B4	31			6	7	
B5	32			1,19	1,19	
B6	33					
B7	34					

RD2 ATN IFC EQ1 REN SRQ WR
 RD1 a Pin 3 del C.I. 7406
 7400 2,6
 3,5,9,10
 4
 5
 11
 12
 13
 7406 10
 5
 2
 4
 6,9
 8,11
 12
 13
 7408 5
 1
 6
 3,4
 74241 (5) 8
 4
 2
 11,12
 18
 HP-IB conector
 7 NRFD
 6 DAV
 8 NDAC

74241 (1)	74241 (2)	74241 (3)	74241 (4)	74241 (5)	'HP-IB conector'
11,12					16 D108
13,14					15 D107
15,16					14 D106
17,18					13 D105
	11,12				4 D104
	13,14				3 D103
	15,16				2 D102
	17,18				1 D101
		12	11		5 EOI
		13	12		11 ATN
		14	13		10 SRQ
		15	14		9 IFC
		17	16		17 REN

'HP-IB conector'

18	TWISTED PAIR WITH	6
19	TWISTED PAIR WITH	7
20	TWISTED PAIR WITH	8
21	TWISTED PAIR WITH	9
22	TWISTED PAIR WITH	10
23	TWISTED PAIR WITH	11
24	GND	

EL DISEÑO DEL CIRCUITO IMPRESO ES PROVEIDO POR SEPARADO
PARA CUALQUIER CONSULTA DIRIGIRSE AL LABORATORIO DE ELECTRONICA
DE LA FACULTAD.

FES-C. U.N.A.M.

RELACION DE COSTOS

MATERIAL NECESARIO PARA CONSTRUIR EL SISTEMA INTERFACE DISENADO en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-IB).

FECHA DE ELABORACION 8 de MARZO de 1986.

CANTIDAD POR INTERFACE	DESCRIPCION	PRECIO UNITARIO (\$)	TOTAL PARCIAL (\$)
1	BASE 40 TERMINALES	385.00	385.00
4	BASE 24 TERMINALES	232.00	928.00
5	BASE 20 TERMINALES	194.00	970.00
3	BASE 16 TERMINALES	129.00	387.00
3	BASE 14 TERMINALES	129.00	387.00
6	RESISTENCIA 1000 ohms	7.50	45.00
1	RESISTENCIA 470 ohms	5.00	5.00
2	CONECTOR P/CABLE PLANO 24 PINS	3115.00	6230.00
1 Mt.	CABLE PLANO 16 HILOS	875.00	875.00
1	C.I. 7400	414.00	414.00
1	C.I. 7406	570.00	570.00
1	C.I. 7408	480.00	480.00
1	C.I. 7413B	365.00	365.00
5	C.I. 74241	1141.00	5705.00
1	C.I. Z80 PIO	5000.00	5000.00
1	C.I. 2716	3140.00	3140.00
TOTAL \$			25886.00

NOTA: NO SE INCLUYE COSTO POR ELABORACION DE LAS TARJETAS DOBLE CARA THROUGH HOLE.

CONSIDERACIONES DE CONSTRUCCION Y OPERACION

Una vez elaborado el circuito impreso (APENDICE C), hemos de tener en cuenta al momento de ensamblar los componentes lo siguiente:

La tarjeta diseñada permite al dispositivo operar, ya sea como Controlador del sistema o como Emisor/Receptor.

La seleccion de una u otra funcion de la misma, depende del Programa monitor grabado en memoria C.I. 2716 (vease Desarrollo del Firmware). En la tarjeta existen una serie de C.I. 74241 que complementan la seleccion de la funcion de la tarjeta de la manera siguiente:

Vease figura XIV considerando un orden arbitrario para c/u de los C.I. 74241 de la tarjeta de abajo hacia arriba.

El CUARTO y TERCER C.I. 74241 complementan la seleccion de la funcion de la interface.

Solo uno de los dos C.I. 74241 debe estar presente en la tarjeta.

EL CUARTO debe de estar presente si la tarjeta se selecciono primeramente como Controlador.

EL TERCERO debe de estar presente si la tarjeta se selecciono

primeramente como Emisor/Receptor.

La tarjeta seleccionada como controladora del sistema, es la unica que utiliza la resistencia de carga 1000 ohms a la salida B del C.I. 7406, las demas tarjetas seleccionadas como Emisor/Receptor, no la requieren. Esto es por la condicion de salida de las compuertas 'Colector Abierto'. Las demas resistencias 1000 ohms a la salida de las compuertas correspondientes son requeridas.

La tarjeta del Sistema Interface disenada en base al standard HEWLETT-PACKARD INTERFACE BUS (HP-1B), obtiene su alimentacion del Sistema Educacional de Microprocesador 180 MKE a traves de un conector.

Toda interaccion entre la Interface, el Procesador y la memoria es a traves de este conector.

La seleccion de direccion para c/u de los dispositivos en la red, se hace por software y es asignada al momento de programar la memoria C.I. 2716 en el valor de la variable NSIMD, la cual debe de encontrarse en el rango decimal (0-30) y debe ser introducido en su equivalente Hexadecimal. Siendo definidas las direcciones 0H para el Controlador y 1H para el primer Emisor/Receptor.

Esto exige que c/u de los dispositivos sean etiquetados

correspondientemente.

Una vez direccionados correctamente los dispositivos, la red del Sistema Interface puede conformarse.

Esta debe de establecerse antes de que cualquier dispositivo conectado a la red sea encendido.

Tenga cuidado en colocar los conectores del Bus de Interface correctamente, es decir uno en paralelo con otro.

Posteriormente se encenderan todos y c/u de los dispositivos conectados a la red.

Para permitir el funcionamiento del Sistema Interface, todos los dispositivos conectados a la red deben ser inicializados mediante la ejecucion del comando TV, es decir, oprimir en el teclado del Sistema Educacional de Microprocesador 260 MKE las teclas COM2 TV. Esto producira el desplegado SI que significa acceso al Sistema Interface.

Al oprimir la tecla DTR0 podremos introducir el nombre del archivo de nuestro programa en Hexadecimal seguido de DTR0.

Inmediatamente despues introduciremos la Direccion de inicio de nuestro Programa seguido de DTR0.

La Cantidad de bytes de nuestro Programa seguido de DTR0. (0000 si se desconoce).

La Direccion Final de nuestro Programa seguido de OTRO. Esta debe de ser incrementada en una unidad, ya que si no se ha dado la cantidad de bytes del Programa, esta se calcula como la diferencia de las Direcciones Final-Inicio. Si la cantidad de bytes es diferente de cero este calculo es omitido.

Inmediatamente despues se introduce la Direccion del dispositivo que funcionara como Emisor seguido de OTRO. Para inicializacion introduzcase 00IF.

A continuacion se introducen una a una las Direcciones de los dispositivos que funcionaran como receptores seguidas de OTRO. No podran introducirse mas de 14 Direcciones de dispositivos receptores. Si se introduce una cantidad menor de 14 Direcciones de dispositivos Receptores, la ultima debe de ser seguida de la Direccion 00IF.

A partir de esta ultima Direccion introducida, el Sistema Interface se comportara de acuerdo a las especificaciones introducidas.

Nota: cualquier error en la introduccion de datos provocara un salto a la rutina de recepcion de comandos del Monitor Hala.

RECAPITULACION

	ACCION	DESPLIEGADO
1	ENCENDIDO	HOLA 00
2	COM2	COM2 00
3	TV	SI00 00
4	OTRO (FILE NAME)	XXXX XX
5	AAAA	AAAA 00
6	OTRO (DIRECCION DE INICIO)	0000 0F
7	2000	2000 00
8	OTRO (CANTIDAD DE BYTES)	0000 0C
9	000A	000A 0C
10	OTRO (DIRECCION FINAL)	0000 0d
11	2000B	200B 0d
12	OTRO (DIRECCION DEL DISP. EMISOR)	XXXX XX
13	001F	001F 00
14	OTRO (DIRECCIONES DE LOS DISP. RECEPTORES)	XXXX XX
15	00NN	00NN 00
16	OTRO	XXXX XX
17	001F	001F 00
18	OTRO	HOLA 00

BIBLIOGRAFIA

- * Logic & Memory Experiments Using TTL ICs
Books 1 and 2 by Peter R. Rony and David G. Larsen,
Howard W. Sams & Co. Inc.
- * The TTL Data Book for Design Engineers
by Texas Instruments.
- * Designing with TTL Integrated Circuits
by Texas Instruments.
- * Microcomputer Interfacing
by Harold S. Stone, Addison Wesley.
- * Operating Systems
by Madnick Donovan, Mc. Graw Hill.
- * Computer Architecture and Parallel Processing
by Kai H. Wang, Faye A. Briggs, Mc. Graw Hill.
- * Digital System Design and Micro-processors
by Jhon P. Hayes, Mc. Graw Hill.

- * Software Engineering Concepts
by Richard Fairley, Mc. Graw Hill.
- * Interfacing to Microprocessors and Microcomputer
and Microprocessor Interfacing
by J.C. Cluley, Mc. Graw Hill.
- * Digital Integrated Electronics
by Herbert Taub / Donald Schilling, Mc. Graw Hill.
- * - Computer Organization Hardware/Software
by G. W. Gorsline, Prentice Hall.
- * - Introduction to Microprocessors:
Software, Hardware, Programming
by Lance A. Leventhal, Prentice Hall.
- * - Digital Computers Fundamentals
by Thomas C. Bartee, Mc. Graw Hill.
- * - Digital and Analog Communication Systems
by K. Sam Shanmugam, John Wiley and Sons.

- * Tutorial Description of the
HEWLETT-PACKARD INTERACE BUS, Hewlett-Packard.
- * Cromenco Macro Assembler Instruction Manual,
CROMENCO INC.
- * MKE Users Manual Programming, MICROCALLI SA. CV.
- * Z80 Family, ZILOG.