

ESCUELA UNIVERSITARIA POLITECNICA
DE
LAS PALMAS

TITULO:
HARDWARE DEL SISTEMA DE DESARROLLO MDS 221 Y CONEXION
REMOTA A UN ORDENADOR HP-3000

AUTOR:

TUTOR:

Antonio F. Quintana Hernandez

Sebastián Suárez Gil

Octubre 1983

ORIGINAL

INDICE

	Página
INTRODUCCION	1
CAPITULO I	3
1. INTRODUCCION A LOS SISTEMAS DE DESARROLLO	4
1.1. EVOLUCION DE LOS METODOS DE DESARROLLO	4
2. SISTEMA DE DESARROLLO "INTELLEC SERIES II"	6
2.1. INFORMACION GENERAL	6
2.2. PLACA DEL PROCESADOR INTEGRADO (IPB)	13
2.2.1. SUBSISTEMA PROCESADOR MAESTRO	14
2.2.1.1. PROCESADOR MAESTRO Y GENERADOR DE RELOJ ..	14
2.2.1.2. CONTROLADOR DEL SISTEMA	15
2.2.1.3. CONTROLADOR DEL BUS	16
2.2.1.3.1. ADQUISICION DEL BUS	16
2.2.1.3.2. LIBERACION DEL BUS	17
2.2.1.3.3. SOBRECARGA DEL BUS	17
2.2.1.3.4. CICLO DE CONTROL DEL BUS	17
2.2.1.4. LOGICA DE RESOLUCION DE PRIORIDADES EN EL BUS	18
2.2.2. SUBSISTEMA DE MEMORIA DEL IPB	18
2.2.2.1. ROM DEL IPB	19
2.2.2.2. RAM DEL IPB	20
2.2.3. SUBSISTEMA DE E/S DEL IPB	21
2.3. TARJETA DEL PROCESADOR INTEGRADO (IPC)	24
2.3.1. SUBSISTEMA DEL PROCESADOR MAESTRO	25
2.3.1.1. PROCESADOR MAESTRO	26
2.3.1.2. CONTROLADOR DEL BUS	27
2.3.1.2.1. ADQUISICION DEL BUS	27
2.3.1.2.2. LIBERACION DEL BUS	28
2.3.1.2.3. SOBRECARGA DEL BUS	28
2.3.1.2.4. CICLO DE CONTROL DEL BUS	29
2.3.1.4. LOGICAL DE RESOLUCION DE PRIORIDAD EN EL BUS	29
2.3.2. SUBSISTEMA DE MEMORIA	30
2.3.2.1. ROM DEL IPC	31
2.3.2.2. RAM DEL IPC	32

	Página
2.3.3. SUBSISTEMA DE E/S DEL IPC	34
2.3.3.1. TRANSFERENCIAS DE E/S EXTERNAS (PIO/IOC) ..	36
2.3.3.2. TRANSFERENCIAS INTERNAS DE E/S	36
2.4. CONTROLADOR DE ENTRADA-SALIDA (IOC)	41
2.4.1. SUBSISTEMA PROCESADOR DEL IOC	44
2.4.1.1. PROCESADOR DEL IOC Y GENERADOR DE RELOJ ..	44
2.4.1.2. LOGICA DEL RESET	45
2.4.1.3. CONTROLADOR DE DMA	46
2.4.2. SUBSISTEMA DE MEMORIA DEL IOC	49
2.4.2.1. ROM DEL IOC	50
2.4.2.2. RAM DEL IOC	50
2.4.3. SUBSISTEMA DE E/S DEL IOC	51
2.4.3.1. DECODIFICACION DE LOS PORTS DEL IOC	52
2.4.3.2. BUFFER DE BYTE DE DATOS (DBB)	53
2.4.3.3. CIRCUITOS DE ENTRADA DEL TECLADO	53
2.4.3.4. CIRCUITOS DE CONTROL DEL CRT	54
2.4.3.5. CIRCUITOS DE CONTROL DEL DISQUETE	57
2.5. SUBSISTEMA DE ENTRADA/SALIDA PARALELO (PIO) ...	60
2.5.1. PROCESADOR DEL PIO	61
2.5.2. INTERFACE PIO-IPB/IPC	61
2.5.3. INTERFACE CON LOS DISPOSITIVOS PERIFERICOS ..	62
 CAPITULO II	 65
1. EVOLUCION	66
1.1. SISTEMAS DISTRIBUIDOS. CLASIFICACION	67
1.1.1. REDES DE COMPUTADORES	69
1.1.2. REDES LOCALES	69
1.1.3. SISTEMAS MULTICOMPUTADORES	69
1.1.4. SISTEMAS MULTIPROCESADORES	70
2. OBJETIVOS DE LAS REDES	71
3. COMPONENTES DE LA RED	73
3.1. SUBRED DEL USUARIO	73
3.2. SUBRED DE COMUNICACION	74
4. CONTROL Y TOPOLOGIA DE LA RED	76
4.1. CAMINOS FISICOS Y LOGICOS	76
4.2. ENLACES PUNTO A PUNTO Y MULTIPUNTO	76

4.3. TOPOLOGIA Y CONTROL DE LOS ENLACES FISICOS.	
TIPOS DE REDES	77
4.3.1. TOPOLOGIAS NO RESTRINGIDAS	78
4.3.2. REDES EN ESTRELLA	80
4.3.3. REDES EN ANILLO	81
4.3.4. REDES TIPO BUS	81
5. TECNOLOGIA DE TRANSPORTE DE LOS MENSAJES	83
5.1. REDES DE CONMUTACION DE CIRCUITOS	83
5.2. REDES DE CONMUTACION DE MENSAJES	83
5.3. REDES DE CONMUTACION DE PAQUETES	84
6. MODOS DE TRANSMISION	85
6.1. TRANSMISION SIMPLEX	85
6.2. TRANSMISION HALF-DUPLEX	85
6.3. TRANSMISION FULL-DUPLEX	85
7. PROTOCOLOS DE LAS REDES	87
7.1. CONCEPTOS EN DISEÑO DE PROTOCOLOS	87
7.2. FUNCIONES DE LOS PROTOCOLOS	90
7.3. ORGANISMOS DE NORMALIZACION	91
7.3.1. LA ISO	91
7.3.2. EL CCITT	91
7.3.3. EL IFIP, LA ECMA Y OTROS	92
7.4. MODELO DE REFERENCIA PARA LA INTERCONEXION DE SISTEMAS ABIERTOS. RECOMENDACION X.25	93
7.4.1. LOS USUARIOS DEL BLOQUE DE TRANSPORTE	97
7.4.2. EL BLOQUE DE TRANSPORTE	99
7.4.3. EL BLOQUE DE TRANSMISION	100
7.5. LA RECOMENDACION X.25	101
8. ARQUITECTURAS DE LAS REDES MAS IMPORTANTES	104
8.1. ARQUITECTURA SNA	104
8.1.1. TOPOLOGIA DE UNA RED SNA	104
8.1.1.1. COMPONENTES BASICOS	104
8.1.1.2. DIRECCIONAMIENTO	107
8.1.1.3. SISTEMAS MULTIDOMINIO-NETWORKING	107
8.1.1.4. RED COMUN	108
8.2. ARQUITECTURA DECNET	110
8.2.1. ARQUITECTURAS DE REDES DIGITALES	112

	Páginas
8.2.1.1. ESTRATOS DNA	113
9. ESTUDIO DE LAS COMUNICACIONES SERIE. PROTOCOLO RS-232 C	115
9.1. COMUNICACIONES SERIE FRENTE A COMUNICACION EN PARALELO	115
9.2. TRANSMISION SERIE	116
9.2.1. CONVERSION PARALELO A SERIE	116
9.2.2. SINCRONIZACION DE LOS DISPOSITIVOS	117
9.2.3. PROTOCOLO SERIE ASINCRONO	120
9.2.4. PROTOCOLO SERIE SINCRONO	122
9.2.5. DETECCION DE ERRORES	124
9.3. PROTOCOLO SERIE RS-232 C	125
9.3.1. CARACTERISTICAS ELECTRICAS DE LAS SEÑALES ..	127
9.3.2. CARACTERISTICAS MECANICAS DEL INTERFACE	128
9.3.3. CARACTERISTICAS FUNCIONALES DE LOS CIRCUITOS .	129
9.3.4. CONFIGURACIONES MAS COMUNES	135
9.4. INTERFACE DEL LAZO DE CORRIENTE DE 20 MILI- AMPERIOS	138
9.4.1. VENTAJAS Y DESVENTAJAS	139
CAPTULO III	141
1. INTRODUCCION	142
2. ESTUDIO DEL PROGRAMA MODEM	144
2.1. ESTUDIO DEL FUNCIONAMIENTO DEL PROGRAMA	146
2.2. EXPLICACION DEL PROGRAMA	148
3. OBJETIVOS DEL PROGRAMA QUILEY	157
3.1. EXPLICACION DEL ALGORITMO	158
3.2. DESCRIPCION DEL FUNCIONAMIENTO DEL PROGRAMA	162
3.3. EXPLICACION DEL PROGRAMA QUILEY	166
3.3.1. ANALISIS DEL PROCESO DE COMUNICACION, Y DEL CONTROL DEL TECLADO DEL PROGRAMA QUILEY	167

INTRODUCCION

El objetivo de éste proyecto es efectuar la conexión a través de modem entre el Sistema de Desarrollo MDS 221 de Intel y un Ordenador HP-3000 utilizando un bus serie standard RS-232C.

Para ello se realiza un estudio detallado del hardware de dicho sistema de desarrollo, y de las comunicaciones serie, particularizando en el protocolo serie RS-232C.

Por tanto, el proyecto está dividido en tres capítulos claramente diferenciados. Un primer capítulo teórico en el que se realiza la descripción del hardware del sistema de desarrollo. Un segundo capítulo teórico en el que se estudian someramente las redes de ordenadores, particularizando en las comunicaciones serie y el protocolo RS-232C. Y un tercer capítulo práctico en el que se hace el estudio tanto teórico como práctico de toda la conexión.

En el primer capítulo se utilizó como referencia el "Hardware Reference Manual" del Sistema de desarrollo para efectuar el estudio de todo el hardware de éste. En éste primer capítulo se hace primero una pequeña introducción a los sistemas de desarrollo, antes de entrar a hablar del caso concreto que se estudia.

La descripción del sistema de Desarrollo está estructurada en cuatro temas. En el primer tema se hace una introducción a las características y especificaciones del sistema de desarrollo, y en él se describen básicamente las tres tarjetas básicas que lo componen. En los siguientes temas se hace un estudio más detallado de éstas. En el segundo tema se estudia la placa del procesador central ó maestro (IPB ó IPC). Esta placa es la que se encarga de gestionar la operación del

resto del sistema, y es la que controla, asimismo, los canales series de Entrada/Salida. En el siguiente tema se estudia la placa de E/S (IOC) que es la encargada de controlar la pantalla integral, la unidad de diskettes, y algunos otros dispositivos periféricos adicionales.

Y por fin, el último tema estudia la placa de E/S paralelo (PIO) que es la encargada de controlar la impresora paralelo (LINE PRINTER) y algunos otros dispositivos paralelo opcionales de los que ya hablaremos.

En el segundo capítulo, se hace primero un estudio de las Redes de Ordenadores en el que se describen los tipos de redes, componentes de éstas, y objetivos que deben cumplir. A continuación se habla de los protocolos de redes, entrando a hablar en detalle en la Recomendación X.25 del CCITT. Después de esto, se realiza una visión general de los 2 tipos de arquitecturas de redes más importantes. Una vez concluido esto se hace un estudio de las comunicaciones, profundizando en la comunicación Serie Asíncrona que es la que nos ocupa, y del protocolo estándar RS-232C.

Y por último, en el tercer capítulo se describe la conexión entre el canal serie 2 del Sistema de Desarrollo y el modem. Para ello se desarrolla un cable RS-232C del que ya hablaremos, y 2 programas que controlan la comunicación. Un primer programa llamado MODEM que tiene como misión mantener una comunicación sencilla con el ordenador. Es decir, que envía simplemente todos los caracteres tecleados al ordenador, y presenta en pantalla los caracteres devueltos por éste.

Y un segundo programa llamado QUILLEY en el que además de mantener la comunicación con el ordenador, controla la impresora paralelo del Sistema de Desarrollo y gestiona la Unidad de diskettes de éste.

CAPITULO I

HARDWARE DEL SISTEMA DE DESARROLLO MDS 221

1. INTRODUCCION A LOS SISTEMAS DE DESARROLLO

La necesidad de sistemas de desarrollo en la construcción y realización de sistemas basados en microprocesadores se debe a la posición intermedia que los microprocesadores ocupan entre el mundo del hardware y del software.

Para el primer caso, la puesta a punto se puede hacer generalmente empleando polímetros, osciloscopios, y cualquier tipo de instrumentación analógica. En cambio, en una realización con microprocesadores, no es suficiente en general examinar las señales; es necesario, en primer lugar, introducir el programa que hará funcionar el sistema y a continuación verificar que cumple los requisitos planteados en el diseño.

Estas etapas de puesta a punto de un programa son las que se encuentran en los ordenadores clásicos. Sin embargo, no se habla de sistemas de desarrollo para ellos simplemente porque poseen ya todo lo necesario, tanto en periféricos como en lógico, para la puesta a punto. En cambio, en una realización industrial con microprocesador no existe nada de esto; los únicos periféricos de que dispone son sensores y accionadores útiles al automatismo que debe controlar.

Resulta pues, que un sistema con microprocesador va a existir sucesivamente bajo dos configuraciones. Antes de llegar a la configuración final, donde no tiene más que lo necesario para el proceso a controlar, es necesario pasar por una fase de desarrollo donde se le adjuntan los periféricos que permitan la introducción de datos y su examen y de programas de ayuda para la puesta a punto del software. Todo este material y esta lógica, utilizada temporalmente en el proceso de puesta a punto de un prototipo, constituye un SISTEMA DE DESARROLLO.

1.1 EVOLUCION DE LOS METODOS DE DESARROLLO

En los últimos diez años, los métodos de programar los microprocesadores han cambiado drásticamente. Cuando fueron introducidos, las facilidades del software aún no existían, los diseñadores trasladaban sus programas a código máquina y los introducían en el microprocesador. El primer paso en la evolución de las herramientas de desarrollo del software en microcomputador fue la introducción del "Cross assembler", un programa que funciona en un sistema distinto y ensambla en código máquina para el microcomputador. El código máquina es transferido entonces a la tarjeta del microprocesador para su ejecución. Pero era necesario todavía usar el microcomputador para corregir el programa. Cuando aparecían problemas, el programa

debía ser retocado en el microcomputador, o reensamblado y transferido al microcomputador de nuevo, consumiendo gran cantidad de tiempo.

A medida que la cantidad de memoria en el sistema microcomputador aumentó, estuvieron disponibles los ensambladores residentes. Esto eliminaba la necesidad de dos sistemas, si bien no existían todavía las facilidades de un editor de textos y de manejo de ficheros.

A mediados de la década empezaron a surgir los sistemas de desarrollo de una forma más o menos completa, incluyendo: sistema operativo, editor de textos, un ensamblador, y un "Debugger".

Los últimos sistemas consisten, en general, en: un CRT de alta velocidad, teclado, unidad a diskette o floppy, procesador central con dieciseis a sesenta y cuatro k bytes de memoria, impresora, ICE (emulador) y programador de PROMS.

El paquete de software incluye: un sistema operativo (personal o multitarea), editor de textos, manejo de ficheros, macroensamblador, linkador y lenguajes de alto nivel con sus correspondientes compiladores o intérpretes.

Los sistemas de desarrollo más corrientes en el mercado español son:

SISTEMA	CPU	RAM	ROM	SOPORTE HARDWARE	SOPORTE SOFTWARE
INTELLEC SERIE II (INTEL)	808X 802X 804X 8035	16/64	2/16	ICE (Para todas CPU's) Multi-ICE (dos CPU's emuladas simultán.) Programador PROM	BASIC FORTRAN COBOL PLM80 RMX-80
TEKTRONIC ROOZA	8048 (familia) 8080/85 TMS9900 6800 Z8, Z80 3870/72 J802	16/64	6	ICE Real Time Trace Programador PROM	BASIC FORTRAN 8080 8085 Z80
PDS 8000 (ZILOG)	Z8 Z80 Z8000	16	3	ICE Capacidad como analizador lógico Programador PROM EPROM	BASIC FORTRAN PASCAL COBOL Z8000 (Traductor) PLZ/ASM
SIGNETICS (PHILIPS)	2650	32/64	0.256/2	ICE Programador PROM	PL-uC
EXORCISER II (MOTOROLA)	6800 68000 (Soporta todas las CPU's de la compañía)	32/64	4/16	ICE para todas las CPU's incluyendo 6809 Programador PROM	BASIC COBOL FORTRAN MPL PASCAL (6809 68000) MACROASSEMBLER
TEXAS	9900	10/64	16	ICE para todas CPU's Real Time Trace Programador PROM	TI BASIC FORTRAN COBOL PASCAL AMPL
HP 64000 (HEWLETT-PACKARD)	6800 8080 8085 Z80	64/128	16	ICE Programador PROM	BASIC

2. SISTEMA DE DESARROLLO "INTELLEC SERIES II"

2.1 INFORMACION GENERAL

- Introducción

Los sistemas de desarrollo "Intellec Series II" son del tipo "bus orientado", es decir, todos sus bloques están orientados a un único bus principal, con sistemas multiprocesadores que emplean un software basado en ROM y en Diskette. Cada sistema de desarrollo de ésta serie provee interfaces de E/S series y paralelos capaces de soportar una gran variedad de periféricos externos.

Existen tres modelos básicos de sistemas de desarrollo en la serie: los modelos 220, 225 y 230. Por otro lado, existen dos variaciones para cada modelo básico, según el voltaje de red con que operan. Así, el modelo 220 está configurado para operar a 115 v. y el modelo 221 a 230 v.

Todos los modelos de la serie incorporan una pantalla integral, un teclado, una fuente de alimentación integral, uno o dos drivers de disco flexible, y seis conectores para conectar las placas deseadas según la configuración. Existen dos placas de circuito impreso que son imprescindibles, y que contienen los tres principales elementos del sistema de desarrollo: el procesador maestro (IPB o IPC), el subsistema controlador de E/S (IOC), y el subsistema de E/S paralelo (PIO).

Todos los modelos de la serie incorporan el siguiente software basado en Diskette:

- Sistema operativo de disco (ISIS)
- Ensamblador 8080/8085

Asimismo incorporan el siguiente software basado en ROM

- Programa monitor
- Programa de diagnóstico y testeo

Los modelos 225 y 230 incluyen también un editor de textos orientado a la pantalla basado en Diskette (CREDIT).

-Organización funcional

Como ya hemos comentado antes, los sistemas de desarrollo de esta serie están formados por tres elementos claramente diferenciados: el IPB o IPC, el IOC y el PIO. Cada elemento posee su propio microprocesador, dispositivos de I/O y memoria suficiente para desarrollar tareas en tiempo real.

El IPB o IPC es el procesador maestro, y como tal, ejecuta el sistema operativo (Monitor o ISIS), los programas de soporte (ensamblador, compiladores, emuladores, etc.) y los programas del usuario. Con la excepción de los dos canales series

de I/O, el IPB/IPC solamente se dirige al IOC o al PIO para transferir uno o más bytes a o desde el dispositivo periférico asociado. Toda la información de control desde el IPB/IPC al PIO o al IOC se suministra en la forma de simples comandos que pueden o no estar asociados a un byte de datos.

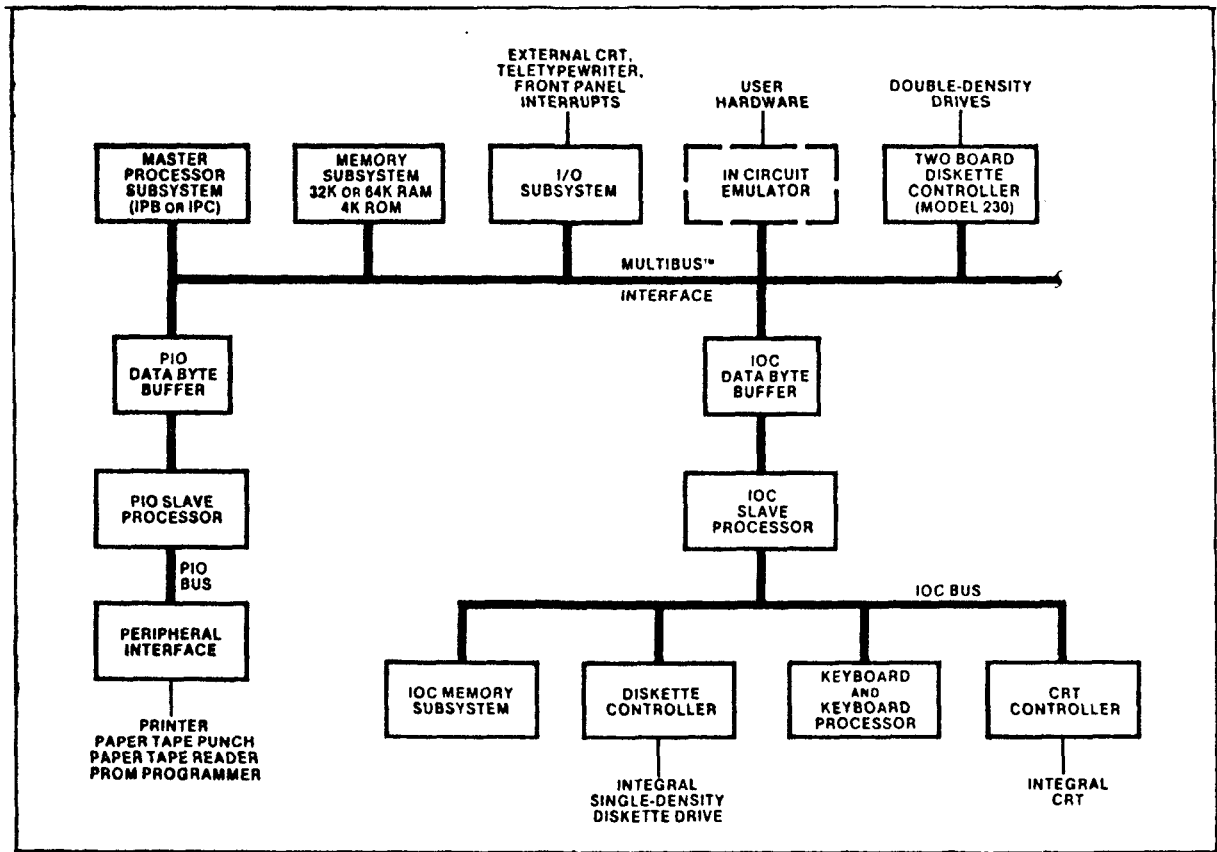


DIAGRAMA DE BLOQUES DEL SISTEMA
DE DESARROLLO INTELLEC DE LA SERIE II

Los subsistemas IOC y PIO están generalmente inactivos, excepto cuando responden a un comando del IPB/IPC. Durante la ejecución de un comando, el IOC y el PIO revisan sus respectivos flags del buffer de datos (DBB) para indicar la aceptación del comando, la aceptación de cualquier byte de datos asociado, o la conclusión del proceso asociado al comando. El sistema

operativo (monitor o ISIS) revisa los flags del DBB para determinar el estado de la ejecución del comando por el IOC o el PIO. Hay que reseñar que el único byte devuelto al IPB/IPC como byte de estados es el DBB. Toda la demás información de estados es devuelta como bytes de datos, y es reconocida como de estado sólo por software.

Pocas interrupciones son implementadas por el hardware del sistema de desarrollo "Intellec Series II". Las interrupciones que son implementadas normalmente no son activadas por el software del sistema (monitor o ISIS). Sin embargo, el IPB/IPC hace uso de los requerimientos de servicios que aparecen como bits dentro de los bytes de estados retornados del IOC o del PIO.

- Arquitectura del multibus

La arquitectura del "Multibus" es tal que permite a cualquier procesador maestro instalado en el sistema de desarrollo asumir un control total del bus. Esto permite a cada procesador maestro tener acceso a todos los recursos de cualquier otro procesador maestro. En términos del sistema de desarrollo "Intellec Series II", cualquier procesador maestro puede añadir recursos no disponibles en el sistema standard. En términos de otro procesador maestro, los recursos del sistema de desarrollo están disponibles para su uso, y por lo tanto, no tienen por qué ser duplicados. Los recursos disponibles del sistema de desarrollo incluyen toda la RAM del IPB/IPC, la ROM del monitor, todo el IOC con su CRT subordinada y el Diskette integral y todo el PIO con su lector/perforador de cinta de papel, impresora de línea y el programador de PROM. Pocos de los recursos del IPB/IPC no están disponibles a otros procesadores maestros. En el IPB estos recursos son los controladores del bus y del sistema, el puerto de control (port FF), los dos canales series, el temporizador de intervalos y los dos controladores de interrupciones. En el IPC sólo son el controlador del bus, el puerto de control, y el controlador de interrupciones del sistema.

Board Component Side			Board Circuit Side		
Pin	Mnemonic	Description	Pin	Mnemonic	Description
1	GND	Signal ground	2	GND	Signal ground
3	+5	+5 VDC	4	+5	+5 VDC
5	+5	+5 VDC	6	+5	+5 VDC
7	+12	+12 VDC	8	+12	+12 VDC
9	-5	-5 VDC	10	-5	-5 VDC
11	GND	Signal ground	12	GND	Signal ground
13	BCLK/	Bus Clock	14	INIT/	Initialize
15	BPRN/	Bus Priority In	16	BPRO/*	Bus Priority Out
17	BUSY/	Bus Busy	18	BREQ/	Bus Request
19	MRDC/	Memory Read Command	20	MWTC/	Memory Write Command
21	IORC/	I/O Read Command	22	IOWC/	I/O Write Command
23	XACK/	Transfer Acknowledge	24	INH1/	Inhibit (disable) RAM
25	AACK/	Advanced Acknowledge	26	INH2/	Inhibit (disable) ROM
27	BHEN/	Byte High Enable	28	ADR10/	Address Extension Lines
29	CBRQ/**	Common Bus Request	30	ADR11/	
31	CCLK/	Constant Clock	32	ADR12/	
33	INTA/*	Interrupt Acknowledge	34	ADR13/	
35	INT6/	Interrupt Requests	36	INT7/	Interrupt Requests
37	INT4/				
39	INT2/				
41	INT0/				
43	ADRE/	Address Lines	44	ADRF/	Address Lines
45	ADRC/				
47	ADRA/				
49	ADR8/				
51	ADR6/				
53	ADR4/				
55	ADR2/				
57	ADR0/				
59	DATE/	Data Lines	60	DATF/	Data Lines
61	DATC/				
63	DATA/				
65	DAT8/				
67	DAT6/				
69	DAT4/				
71	DAT2/				
73	DAT0/				
75	GND	Signal ground	76	GND	Signal ground
77	-10	-10 VDC	78	-10	-10 VDC
79	-12	-12 VDC	80	-12	-12 VDC
81	+5	+5 VDC	82	+5	+5 VDC
83	+5	+5 VDC	84	+5	+5 VDC
85	GND	Signal ground	86	GND	Signal ground

ASIGNACION DE PINES DEL MULTIBUS

Signal Mnemonic	Functional Description
AACK/	<i>Advanced Acknowledge.</i> A memory acknowledge signal that is generated in advance of the normal transfer acknowledge (XACK/) signal to allow a bus master to shorten its memory access timing.
ADR0/-ADRF/	<i>Address.</i> These 16 lines are used to specify the address of the memory location or I/O port to be accessed within a 64k range. ADRF/ is the least significant bit.
ADR10/-ADR13/	<i>Extended Address.</i> These four lines are appended to the 16-bit address to increase the addressing range to 1 megabyte. The IPB/IPC monitors these inputs to determine if the address on the bus corresponds to its memory segment (0-64k). ADR10/ is the least significant bit of the extended address.
BCLK/	<i>Bus Clock.</i> An asynchronous clock signal used to synchronize bus contention resolving circuits among bus masters. The BCLK/ signal from the IPB/IPC has a minimum period of 100 ns with a 35%-65% (minimum) duty cycle.
BHEN/	<i>Byte High Enable.</i> A signal that indicates the presence of a data byte on the DAT8/-DATF/ data lines. This signal is supported only by the memory on the IPC.
BPRN/	<i>Bus Priority In.</i> A bus contention resolving signal that indicates to a bus master that no higher-priority bus master is requesting the bus. The IPB/IPC incorporates the bus resolution logic and provides an individual BPRN/ signal to each card-cage slot. BPRN/ is synchronized by BCLK/.
BPRO/	<i>Bus Priority Out.</i> A bus contention resolving signal for use with serial (daisy chain) bus priority resolution schemes (BPRO/ is passed to the BPRN/ input of the next lower-priority bus master). BPRO/ is not supported by the IPB/IPC.
BREQ/	<i>Bus Request.</i> A bus contention resolving signal for use with parallel-priority resolution schemes. A bus master uses BREQ/ to indicate that it requires the bus. The IPB/IPC monitors the individual BREQ/ inputs and outputs BPRN/ to the highest-priority bus master requesting the bus. BREQ is synchronized by BCLK/.
BUSY/	<i>Bus Busy.</i> A common bus contention resolving signal to indicate that the bus is in use. When a bus master gains access to the bus, it uses BUSY/ to prevent any other bus master from acquiring the bus. BUSY/ is synchronized by BCLK/.
CBRQ/	<i>Common Bus Request.</i> A bus contention resolving signal to allow bus requests by lower-priority bus masters to override bus requests by high-priority bus masters. CBRQ/ is synchronized by BCLK/ and is supported only by the IPC.
CCLK/	<i>Constant Clock.</i> An asynchronous clock signal available for general use by modules on the bus. On the IPB/IPC, CCLK/ is synchronous with BCLK/ and has the same period and duty cycle.
DAT0/-DATF/	<i>Data.</i> These 16 bidirectional lines transfer data to or from the addressed memory location or I/O port. DAT0/ is the least-significant bit. The IPB/IPC, as a bus master, has a 8-bit wide data path and only uses DAT0/-DAT7/. The IPC memory can use all 16 data lines (DAT0/-DATF/).
INH1/	<i>Inhibit RAM.</i> Prevents RAM devices from responding to the memory address on the address lines. INH1/ effectively allows ROM devices to override RAM devices when ROM and RAM are assigned the same address space. INH1/ may also be used to allow memory mapped I/O devices to override RAM.
INH2/	<i>Inhibit ROM.</i> Prevents ROM devices from responding to the memory address on the address lines. INH2/ effectively allows start-up software such as ROM-based bootstrap programs to override another ROM device when the two ROMs are assigned the same address space. INH2/ may also be used to allow memory mapped I/O devices to override ROM.
INIT/	<i>Initialize.</i> This signal resets the entire system to a known internal state. INIT/ is activated by the front panel RESET switch. When activated by another bus master, INIT/ will not initialize the IPB/IPC.

DESCRIPCION DE SEÑALES DEL MULTIBUS

Signal Mnemonic	Functional Description
INT0/-INT7/	<i>Interrupt.</i> A set of eight, multilevel interrupt request lines accepted by the parallel interrupt resolving logic of the IPB/IPC (INT0/ is the highest-priority interrupt). In response to an interrupt, the IPB/IPC executes an interrupt acknowledge sequence to call the associated user interrupt service routine.
INTA/	<i>Interrupt Acknowledge.</i> A signal generated by a master processor in response to an interrupt. Since the IPB/IPC incorporates the interrupt resolution logic, the INTA/ signal is used internally and is not output on the bus.
IORC/	<i>I/O Read Command.</i> This signal indicates that the address of an input port has been placed on the address lines and that the port is to place the requested data byte on the data lines.
IOWC/	<i>I/O Write Command.</i> This signal indicates that the address of an output port has been placed on the address lines and that the data byte on the data lines is to be accepted by the port.
MRDC/	<i>Memory Read Command.</i> Indicates that the address of a memory location has been placed on the address lines and that the requested data byte is to be placed on the data lines.
MWTC/	<i>Memory Write Command.</i> Indicates that the address of a memory location has been placed on the address lines and that the data byte to be written into the addressed location has been placed on the data lines.
XACK/	<i>Transfer Acknowledge.</i> This signal is the response from memory or an I/O port that indicates that the specified read/write operation has been completed. That is, data has been placed on, or accepted from, the data lines.

DESCRIPCION DE SEÑALES DEL MULTIBUS (Cont.)

- Especificaciones

Las especificaciones para los tres modelos básicos de sistema de desarrollo de la Serie II de Inteltec, se listan en la siguiente tabla:

IPB Master Processor (Models 220, 221, 230, and 231) Microprocessor: RAM: ROM: Bus:	8080A-2 operating at 2.6 MHz. 32k (see note). 4k (2k in monitor, 2k in boot/diagnostic). Multibus architecture; bus clock rate is 9.8304 MHz.
IPC Master Processor (Models 225 and 226) Microprocessor: RAM: ROM: Bus:	8085A-2 operating at 4.0 MHz. 64k. 4k (2k in monitor, 2k in boot/diagnostic). Multibus architecture; bus clock rate is 9.8304 MHz.
Interrupts:	8-level, maskable, nested priority interrupt network initiated from front panel or user-selected devices.

ESPECIFICACIONES DEL SISTEMA DE DESARROLLO

I/O Interfaces Serial:	Two RS232 channels at 110-9600 baud (asynchronous) or 150-56,000 baud (synchronous); programmable baud rates and serial formats. Serial Channel 1 provided with 20 mA current loop.
Parallel:	Interface provided for paper tape punch, paper tape reader, line printer, and UPP-103 Universal PROM Programmer.
Direct Memory Access (DMA):	Standard capability of Multibus architecture; implemented for user-selected DMA devices through optional DMA module. Maximum transfer rate of 2 MHz.
Diskette Subsystem (Models 220, 221, 225 and 226) Number of Drives: Storage Capacity: Transfer Rate: Access Time: Average Random Positioning: Rotational Speed: Average Rotational Latency: Recording Mode:	One, single density. 250k bytes (formatted). 250k bits/second. Track-to-track: 10 ms. Head settling time: 10 ms. 260 ms. 360 rpm. 83 ms. FM.
Diskette Subsystem (Models 230 and 231) Number of Drives: Storage Capacity: Transfer Rate: Access Time: Average Random Positioning: Rotational Speed: Average Rotational Latency: Recording Mode:	Two, double density. 1M bytes (formatted) total. 500k bits/second. Track-to-track: 10 ms. Head settling time: 10 ms. 260 ms. 360 rpm. 83 ms. M ² FM.
AC Requirements Input Voltage: Input Current:	100/120/220/240 Vac \pm 10%, 47-63 Hz, single phase. Model 220 and 225: 5.9A Model 221 and 226: 3.1A Model 230: 5.4A Model 231: 2.7A
Electrical Characteristics:	See table 1-5.
Environmental Characteristics Operating Temperature: Relative Humidity:	32° to 95°F (0° to 35°C). To 90% without condensation.
Physical Characteristics Models 220, 221, 225 and 226 Main Chassis:	Width: 17.37 in. (44.12 cm). Height: 15.81 in. (40.16 cm). Depth: 19.13 in. (48.59 cm). Weight: 86 lb (39 kg).
Models 230 and 231 Main Chassis:	Width: 17.37 in. (44.12 cm). Height: 15.81 in. (40.16 cm). Depth: 19.13 in. (48.59 cm). Weight: 73 lb (33 kg).
Dual Drive Chassis (Models 230 and 231):	Width: 17.6 in. (44.7 cm). Height: 5.7 in. (14.7 cm). Depth: 19.4 in. (49.3 cm). Weight: 43.0 lb (19.5 kg).
Keyboard:	Width: 17.37 in. (44.12 cm). Height: 3.0 in. (7.62 cm). Depth: 9.0 in. (22.86 cm). Weight: 6.0 lb (2.72 kg).

2.2 PLACA DEL PROCESADOR INTEGRADO (IPB)

La placa del procesador integrado es un simple sistema computador basado en el microprocesador 8080A desarrollada para el sistema de desarrollo Intellec Series II. El IPB incluye, aparte del microprocesador y su contro lógico asociado, 4K bytes de ROM y dos canales series de E/S así como los interfaces al Multibus, el controlador de E/S (IOC) y el subsistema de E/S paralelo (PIO)

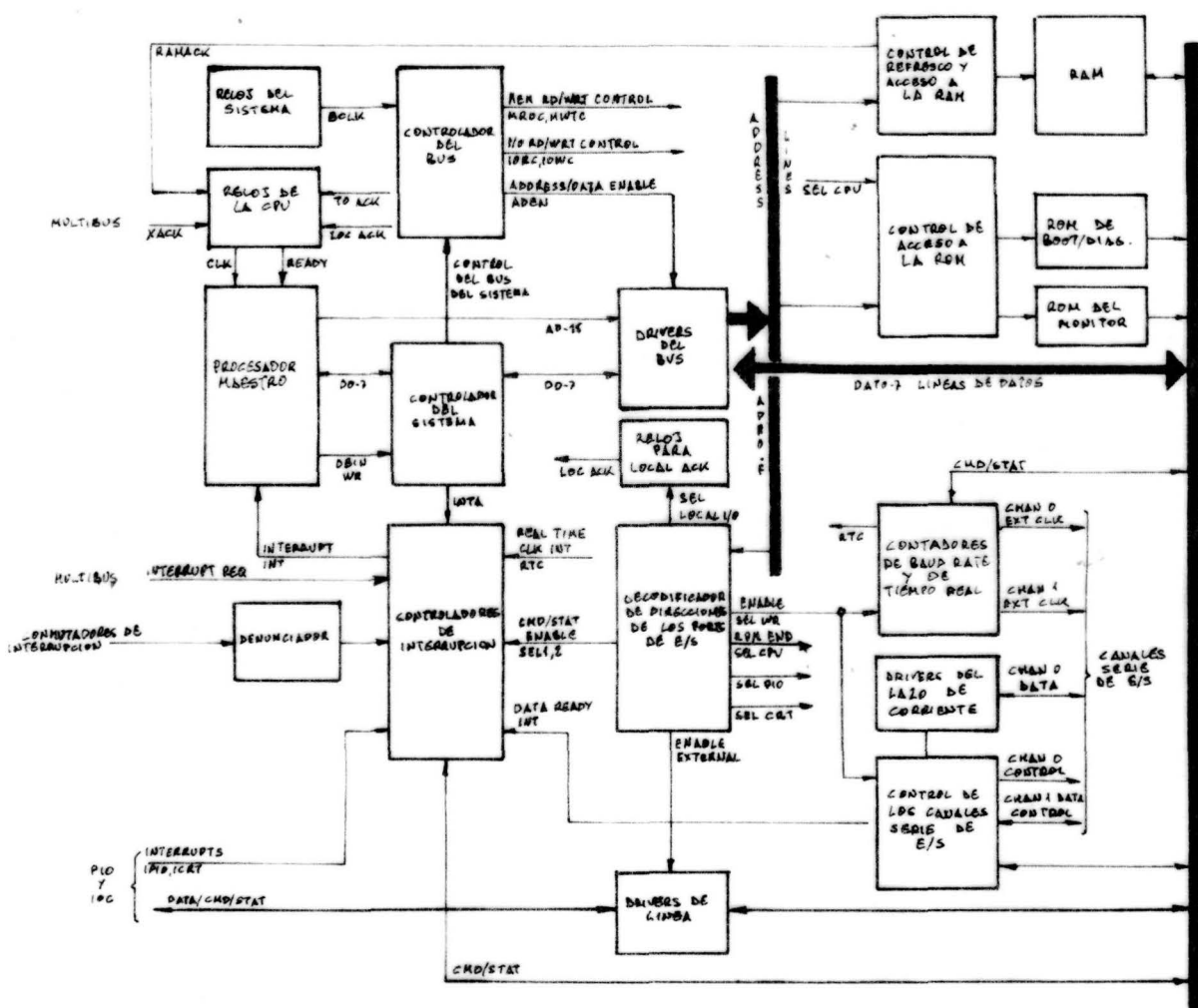


DIAGRAMA DE BLOQUES DEL IPB

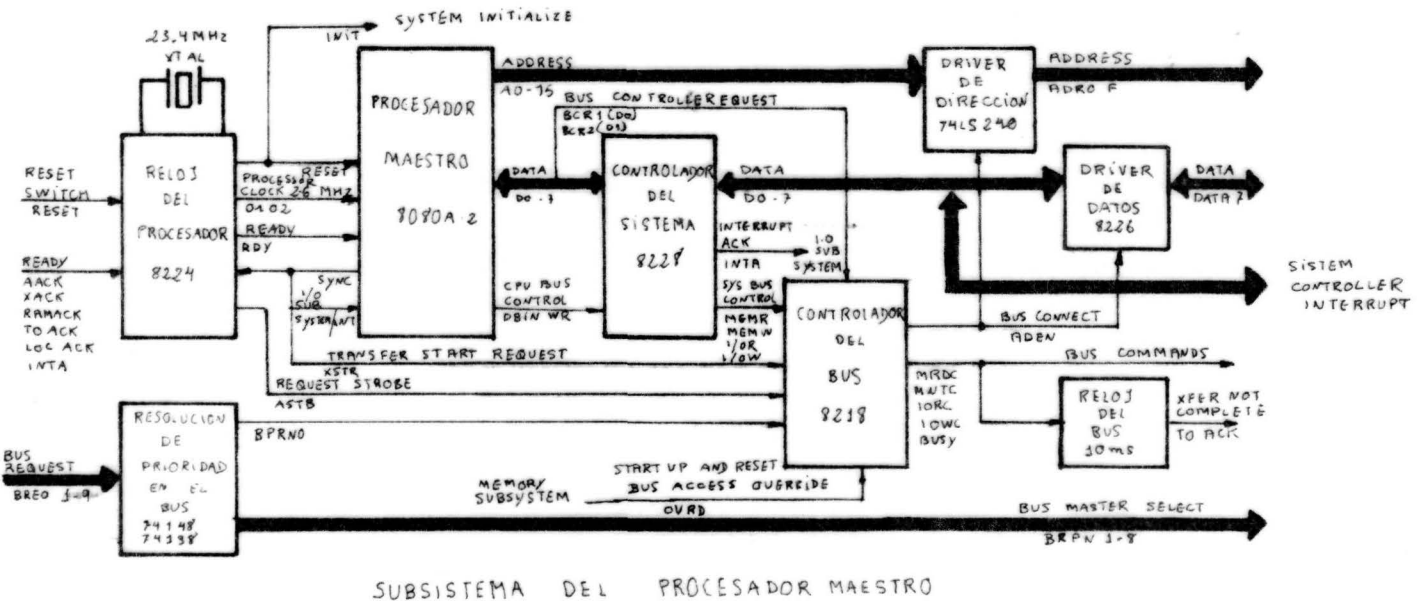
Las comunicaciones entre el IPB y los dispositivos periféricos, se desarrollan a través de los buses del PIO y del IOC, o bien a través del Multibus. Los buses del PIO y del IOC pueden ser controlados directamente por el microprocesador del IPB o indirectamente por cualquier otro procesador a través del IPB. Las transferencias de datos pueden ser iniciadas bien mediante interrupciones hardware o bien mediante el "análisis" software de requerimientos de servicios a través de puertos dedicados de E/S. Cuando se añaden nuevos dispositivos al sistema es necesario añadir asimismo nuevos puertos de E/S o nuevas interrupciones.

La organización del Multibus es tal que el microprocesador del IPB tiene la menor prioridad de los diez potenciales procesadores que pueden conectarse a él.

Los recursos del IPB son accesibles a través del Multibus a cualquier otro procesador para lo cual debe usar los puertos de E/S definidos por el IPB. La ROM que contiene los programas de testeo y diagnóstico no es accesible a otro procesador desde que ésta es activada a través de un puerto de control accesible solamente al IPB.

2.2.1 SUBSISTEMA PROCESADOR MAESTRO

El subsistema procesador maestro del IPB es el centro computacional y de control de éste. Los principales elementos de este subsistema son: el microprocesador 8080A-2, el generador de reloj 8224, el driver de datos 8226, el controlador de sistema 8228 y el controlador del bus 8218.



SUBSISTEMA DEL PROCESADOR MAESTRO

2.2.1.1. PROCESADOR MAESTRO Y GENERADOR DE RELOJ

El procesador maestro 8080A-2 y su asociado generador de reloj 8224 son empleados de una forma convencional, con la excepción de que entre cada ciclo máquina de acceso a memoria se inserta un estado de espera. Para compensar estos estados de espera, el procesador maestro usa un reloj de 2.6 MHz., con lo que el promedio de tiempo empleado por una instrucción es aproximadamente el mismo que utilizaría con un reloj de 2.0 MHz. sin estado de espera

Para insertar el estado de espera, el acceso lógico a la RAM provee señal de reconocimiento (RAM AACK) a la entrada "READY" del generador de reloj. La salida sincronizada RDY del generador de reloj es llevada asimismo a la entrada RDY del microprocesador 8080. El tiempo de RAM AACK es tal que cuando el microprocesador muestra primero su entrada RDY (durante el estado T_2), RDY es inactiva, y se inserta entonces un estado de espera entre cada ciclo de máquina. Cuando la entrada RDY es muestreada de nuevo (durante T_{wait} : estado de espera), es activa, y el microprocesador sale del estado de espera y entra en el estado T_3 . Las otras señales de reconocimientos a la entrada RDY del generador de reloj (XACK/, AACK/, LOC ACK y TO ACK) son usadas para insertar uno o más estados de espera entre cada ciclo de máquina cuando sea necesario para transferencias de E/S internas y externas y operaciones de acceso a memoria. LOC ACK (reconocimiento local) proviene del reloj de reconocimiento local del IPB y se usa con las transferencias locales de E/S (transferencias entre los canales series de E/S, los dispositivos programables del IPB, el IOC o el PIO). TO ACK (reconocimiento de tiempo de salida) se usa para permitir al 8080 completar un ciclo de máquina cuando no se recibe el esperado reconocimiento del comando dentro de aproximadamente 10 mseg. Una señal adicional en la entrada ready es INTA/ (reconocimiento de interrupciones). Esta señal provee la indicación ready requerida por el 8085 durante los ciclos de máquina de reconocimiento de interrupción.

2.2.1.2. CONTROLADOR DEL SISTEMA

El controlador del sistema sirve también como **driver** bidireccional para el bus de datos del procesador maestro (DO-D7). Desde que el controlador de interrupciones del sistema se conecta al bus de datos del procesador maestro, y no al bus de datos del sistema, otro procesador no puede acceder al controlador de interrupciones del sistema. También, durante una operación de lectura de E/S del controlador de interrupciones del sistema, los transeptores del bus de datos del sistema (8226) son forzados en su modo de salida (write) a desconectar el bus de datos del sistema del procesador maestro.

El controlador del sistema (8228), al principio de cada ciclo de máquina, decodifica la salida de estados del procesador maestro en señales de control individuales. Las señales de control de lectura/escritura (MEMR/, MEMW/, I/OR/ y IOWR/) son usadas por el controlador del bus (8218) para controlar las comunicaciones internas y externas del IPB.

La señal de control de reconocimiento de interrupción (INTA/) es usada por el controlador de interrupciones del sistema (8259) (para iniciar la secuencia del rector de interrupciones) y por el generador de reloj (para habilitar la entrada RDY del procesador maestro durante los ciclos de máquina de reconocimiento de interrupciones).

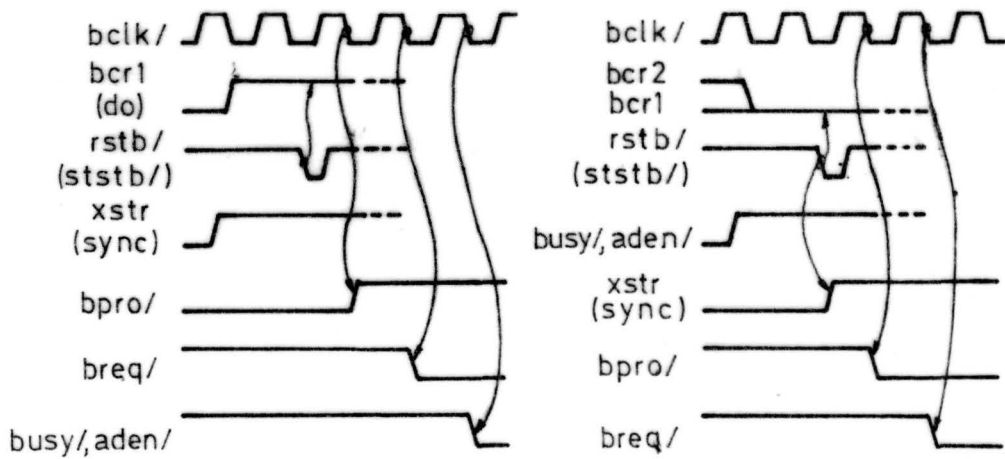
2.2.1.3. CONTROLADOR DEL BUS

El controlador del bus 8218 provee todas las funciones de adquisición, temporización y control del bus para el IPB.

2.2.1.3.1. ADQUISICION DEL BUS

Cuando el procesador maestro necesita usar el bus del sistema el controlador del bus asegura el acceso a éste, y mantiene el acceso al bus hasta que otro procesador solicita el bus (el procesador maestro tiene la menor prioridad en el bus) o hasta que el procesador maestro insta al controlador del bus a liberar el bus mediante una instrucción Halt.

Para iniciar un ciclo de adquisición del bus, el controlador del bus necesita un nivel activo en la entrada RSTB/ y un nivel activo en una o en ambas de las dos entradas BCR1 y BCR2 (solicitud de control del bus 1 y 2). Recordando el cronograma de tiempos de los ciclos de máquina del 8080, al principio de cada ciclo, STSTB/ (que es una salida del generador de reloj, y que en este caso está conectada a RSTB/) pasa a estado activo y una "palabra de estado" (definiendo el tipo de ciclo de máquina a ser ejecutado) es sacada por el bus de datos del procesador maestro. Si ningún otro procesador está solicitando el bus, la entrada BPRN/ al controlador del bus será activa. En la pendiente de caída del segundo o tercer (dependiendo del ciclo de máquina) ciclo de reloj del bus (BCLK/), si el bus no está en uso (indicado por un nivel inactivo en BUSY/) y si no se ha recibido un requerimiento del bus por parte de otro procesador el controlador del bus activa BUSY/ y ADEN/ (habilitación de direcciones y datos). BUSY/ es una señal bidireccional del bus que, cuando está activa, indica a todos los otros procesadores que el bus se está usando. ADEN/ es usada localmente por el IPB para habilitar los latches (cerrojos) de direcciones, y los transceptores de datos. Una vez el IPB ha adquirido el bus, el controlador del bus mantiene el acceso al bus (manteniendo BUSY/ activa) hasta que otro procesador solicite el bus o se ejecute una instrucción Halt.



ADQUISICION DEL BUS

2.2.1.3.2. LIBERACION DEL BUS

Cuando otro procesador solicita el bus, el control del bus completa su actual ciclo de control del bus y libera el bus (inactiva BUSY/) en el siguiente ciclo de reloj. Cuando se ejecuta un ciclo de máquina de reconocimiento de parada (Halt), el controlador del bus libera el bus en el tercer ciclo de reloj del bus.

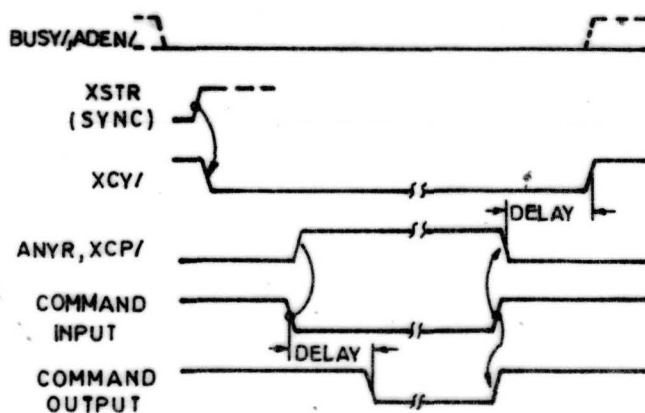
2.2.1.3.3. SOBRECARGA DEL BUS

El controlador del bus tiene una entrada OVRD (override: sobrecarga) que, cuando es activa, permite al procesador maestro mantener el acceso al bus entre ciclos del bus previniendo el reconocimiento de una solicitud del bus desde otro procesador. La entrada OVRD es tomada del bit 6 del latch de habilitación de la TOM y es activado a desactivado a través del puerto de control del IPB. OVRD es activada automáticamente cuando el sistema es inicializado para prevenir que cualquier otro procesador acceda al bus durante el procedimiento de inicialización del procesador maestro (después de la inicialización, el programa monitor inactiva OVRD).

2.2.1.3.4. CICLO DE CONTROL DEL BUS

El controlador del bus necesita la recepción de una solicitud de comenzar la transferencia (XSTR) y la adquisición del bus (BUSY/ y ADEN/ activas) antes de que pueda comenzar un ciclo de control del bus. La entrada XSTR al controlador del bus es la salida SYNC del procesador maestro. Después durante cualquier ciclo de máquina, SYNC llega a ser activa antes de que S TSTB/ sea activa (la señal que inicia el ciclo de adquisición

del bus), si el controlador del bus no tiene acceso al bus cuando empieza un ciclo de máquina, el ciclo de control del bus es iniciada cuando el bus sea adquirido (cuando BUSY/ y ADEN/ sean activas). Ppor otro lado, si el controlador del bus ya tiene el bus, el ciclo de control es iniciado por XSTR.



CICLO DE CONTROL DEL BUS

2.2.1.4. LOGICA DE RESOLUCION DE PRIORIDADES EN EL BUS

La lógica de resolución de prioridad en el bus incluye un codificador de prioridades de 8 a 3 (74148) y un decodificador de 3 a 8 (74138). Las entradas primarias al codificador son las señales de demanda o solicitud del bus BREQ1/ hasta BREQ8/. Si hay más de una activa, el codificador sólo genera el código binario para la solicitud de prioridad más alta (la de número más alto). El decodificador usa el código binario del codificador para activar la apropiada señal de prioridad de entrada en el bus (BRPNx/) al procesador que lo solicita.

Los procesadores de más alta y de más baja (IPB) prioridad están excluidos de la lógica de resolución de prioridad. La solicitud de más alta prioridad del bus (BREQ9/) es usada para bloquear todas las otras demandas del bus y no requiere la señal BRPN/, porque asume el control del bus tan pronto como haya sido completada la actual transferencia del bus.

2.2.2. SUBSISTEMA DE MEMORIA DEL IPB

El subsistema de memoria del IPB controla hasta 64k de RAM (32k en el IPB y 32k en placa opcional) y 4k de ROM. Como el bus de direcciones del procesador maestro es de 16 bits y sólo puede direccionar un máximo de 64k posiciones, los dos tipos de memorias (ROM y RAM) comparten el mismo espacio de

direcciones. La técnica que permite direccionar espacio compartido es conocida como de "ROM ensombrecida" en donde la ROM es seleccionada preferentemente a menos que sea deshabilitada.

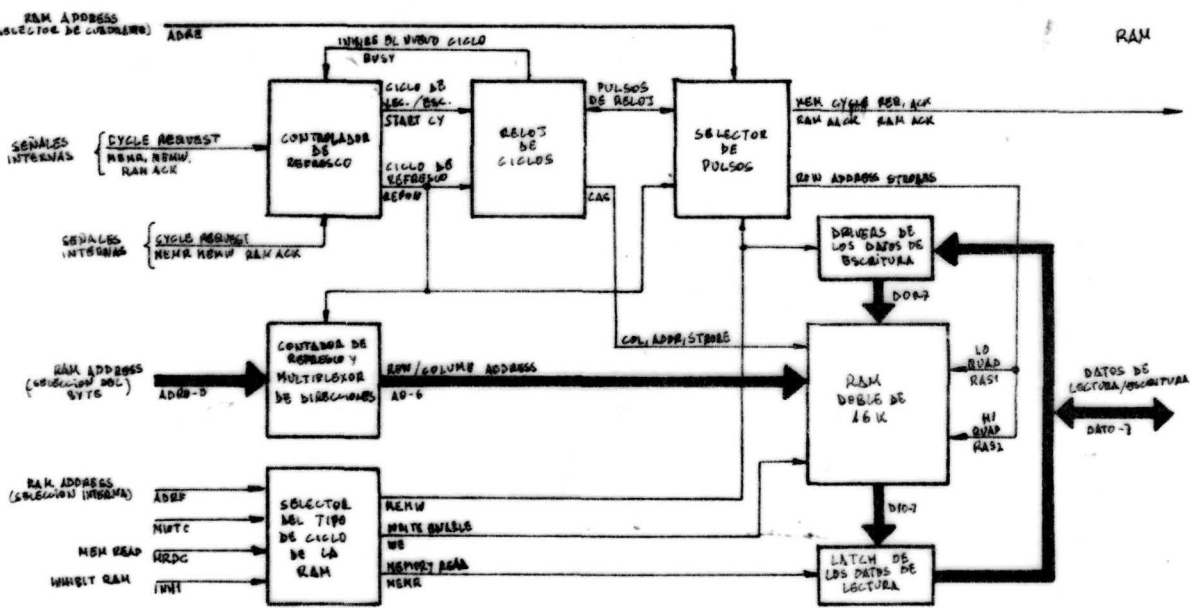
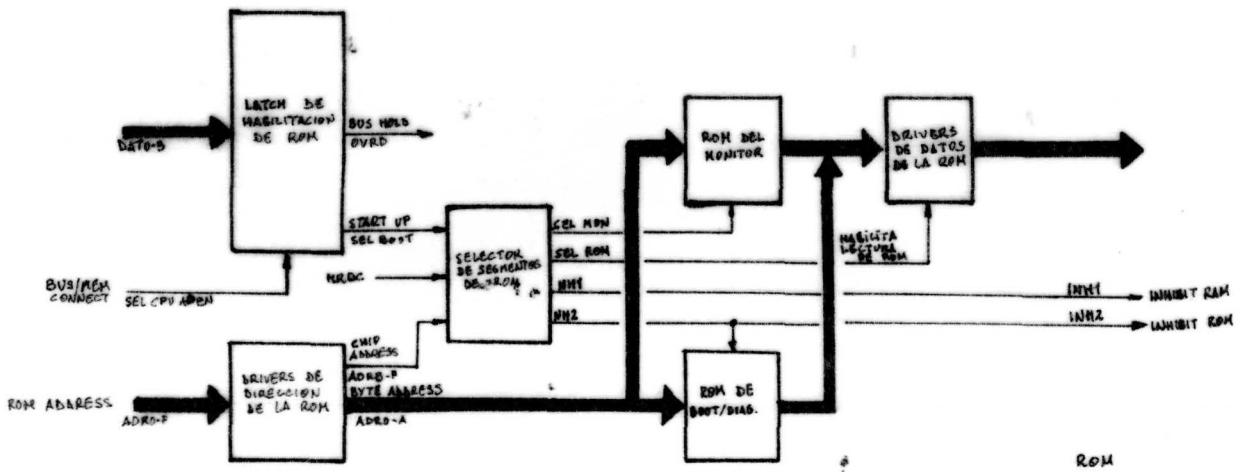
Los principales elementos que incluye el subsistema de memoria son: 16k bits de RAM dinámica, 2 de 2k bytes de ROM, un controlador de refresco de la RAM (3222), un contador 3242, una ROM de alta velocidad 3601 y una gran variedad de circuitos TTL.

2.2.2.1. ROM DEL IPB

La memoria de sólo lectura, ROM, disponible en el IPB, está dividida en dos segmentos. Un segmento está formado por un chip de 2k que contiene el programa monitor, y el otro está formado también por un chip de 2k y contiene el programa de chequeo o test y de diagnóstico.

El programa monitor es accesible a todos los procesadores conectados al multibus. El segmento de 2k de ROM del monitor comienza en la dirección F800H. Este segmento no puede ser "ensombrecido", por lo que los 2k últimos (desde F800H hasta FFFFH) de la RAM opcional de 32k no pueden ser usados.

El segmento de test/diagnóstico solamente es accesible al procesador maestro. El puerto de control del procesador maestro (port FFH) se usa para activar o desactivar bits individuales dentro del latch de habilitación de la ROM con objeto de mover, habilitar o inhibir el segmento de ROM de test/diagnóstico. Cuando el procesador maestro es inicializado, el latch de habilitación de la ROM es limpiado o reseteado (todas las salidas activas). La combinación de START UP/ y SEL BOOT/ en el selector de segmentos de ROM, establecen la posición inicial del segmento de ROM en la posición 0H, así como en la posición E800H. El programa de chequeo o test salta inmediatamente a la posición E800H e inactiva la salida START UP/ del latch de habilitación de la ROM (escribiendo 02H en el puerto de control). Después de la ejecución del programa de test/diagnóstico, el control es transferido al programa monitor en la posición F800H. El programa monitor lo primero que hace es inactivar la salida SEL BOOT/ del latch de habilitación de ROM (escribiendo 04H en el puerto de control).



SUBSISTEMA DE MEMORIA DEL IPB

2.2.2.2. RAM DEL IPB

Con la configuración normal, el IPB puede acceder a 32k o a 64k de RAM, en donde los primeros 32k están localizados en el IPB, y los segundos en una placa opcional de 32k de RAM. Recordemos que los 2k últimos del segmento más alto de la RAM (desde F800H hasta FFFFH) no pueden ser usados por compartir su espacio con la ROM

La RAM interna de 32k está implementada como dos memorias duales de 16k para permitir el uso de chips standards Intel tales como el 2116 o el 2117 que son RAM dinámicas de 16k por 1 bit. Ocho de estos chips están conectados en paralelo para establecer un "banco" básico de 16k bytes, y dos de estos bancos comprenden los 32k de RAM internos. La selección de un

banco de 16k se efectúa mediante el segundo bit más significativo del bus de direcciones (ADRE). Los restantes 14 bits son multiplexados a través de un multiplexor de direcciones Intel 3242 y un contador de refresco, en la forma de una fila de 7 bits. En otras palabras, cada chip de la RAM es tratado como una matriz de 128 por 128, proveyendo almacenamiento para una posición de bit específica de 16k bytes distintos.

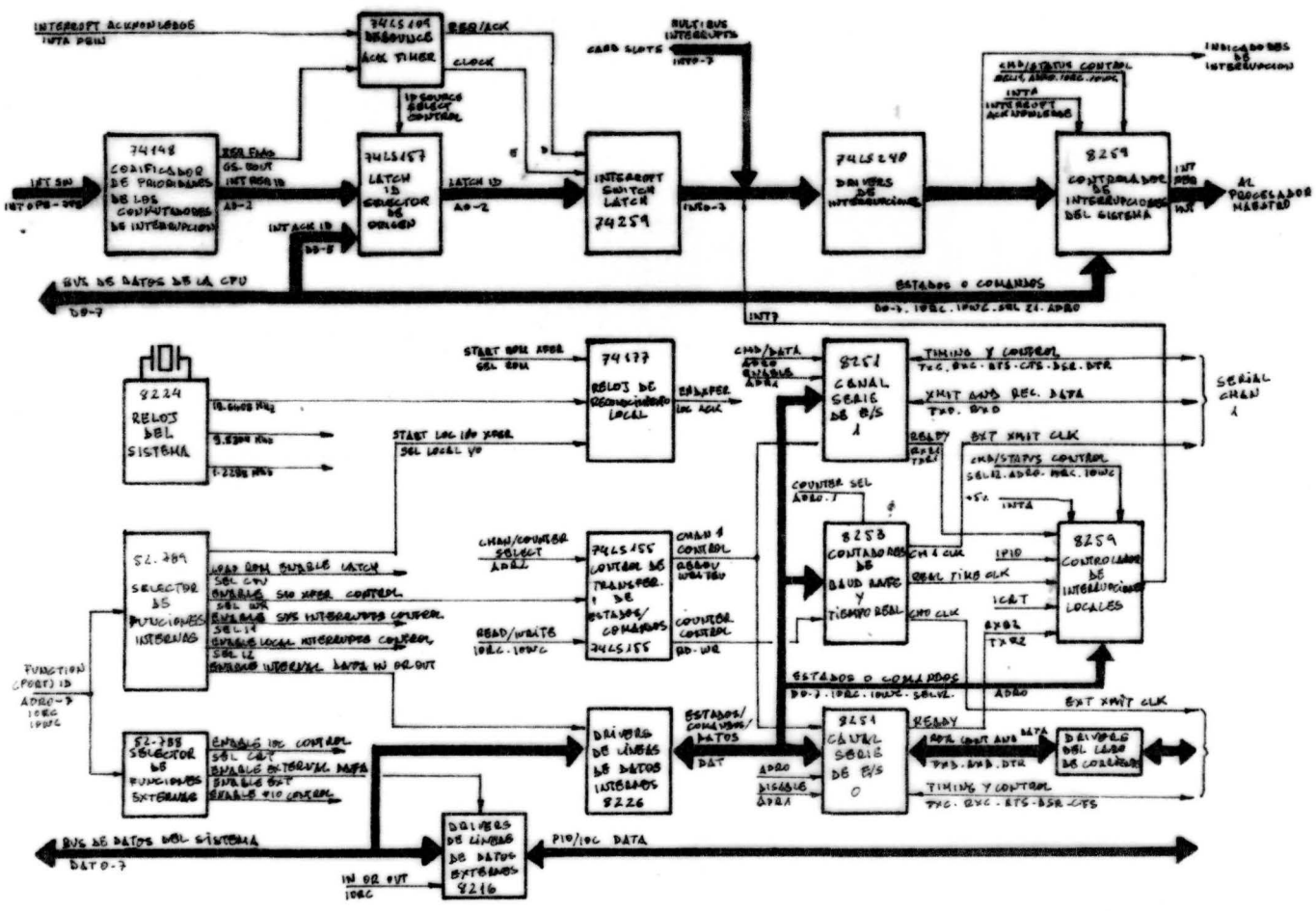
2.2.3. SUBSISTEMA DE E/S DEL IPB

El subsistema de E/S del IPB posee todo lo necesario para mantener las comunicaciones de todos los sistemas de desarrollo de la serie, con el procesador maestro. Estas comunicaciones incluyen: transferencias de E/S del sistema a o desde otras placas opcionales del Multibus y transferencias de E/S locales.

Las transferencias de E/S del sistema se efectúan a través del interface del Multibus. Las transferencias locales de E/S usan el Multibus, pero se efectúan a través de ports de E/S del procesador maestro. Las transferencias locales de E/S pueden dividirse en externas (IOC y PIO) e internas. Todas las transferencias internas son entre el procesador maestro y todos los recursos programables del IPB. En el caso de transferencias externas, el subsistema de E/S decodifica las direcciones de los ports de E/S y habilita las líneas de datos del IOC y del PIO. Las direcciones de los ports también hacen que el subsistema de E/S genere una señal de selección para el PIO o el IOC. Las señales de selección inician las comunicaciones entre el procesador maestro y el microprocesador 8041 del PIO o el microprocesador 8080A-2 del IOC.

Los circuitos de transferencias internas de E/S decodifican las direcciones de los ports de E/S del procesador maestro. Cada dirección de port. decodificada produce la generación de una señal de selección que se usa para habilitar una transferencia de datos entre el procesador maestro y el chip programable. Los tres tipos de transferencias de datos que pueden ocurrir son:

- Salida de un comando al chip seleccionado
- Entrada de un byte de estados del chip seleccionado
- Entrada o salida de un byte de datos a o desde el chip seleccionado



SUBSISTEMA DE E/S DEL IPB

Dentro del subsistema de E/S se emplean un total de cinco chips programables. Las funciones de cada chip son:

- Canal de E/S serie 0 (8251).- Está formado por transmisor-receptor universal sincrono-asincrono (USART) que establece un lazo de corriente 20 mA, y un interface RS-232 entre un dispositivo serie externo y el procesador maestro del IPB. Cuando el sistema es inicializado, el programa test activa el canal (o programa el canal) para: operación asíncrona con dos bits de stop, longitud de carácter de ocho bits y un factor de "baud rate" de 16k (la señal del reloj externo del canal o del contador de baud rate es de 1.76 kHz. con lo que dividido por 16, provee un baud rate de 110).
- Canal de E/S serie 1 (8251).- Idéntico al canal 0 con la excepción de que no está disponible para establecer el lazo de corriente. La señal de reloj externo del canal 1 es de 38.4 kHz. y provee una velocidad de 2400 baudios.

- Contador de tiempo real/Baud rate (8253).- Consiste en un temporizador programable formado por tres contadores independientes. Dos de estos contadores establecen los "baud rates" para el canal serie de E/S. El tercer contador es usado para generar una señal de reloj de tiempo real de 1 mseg. El reloj de tiempo real no es usado para operaciones básicas del sistema de desarrollo; está reservado para aplicaciones del usuario. Cuando el sistema es inicializado, todos los tres contadores son programados para operar en el modo 3 (generador de onda cuadrada) y usan un registro contador de 2 bytes.

- Controlador de interrupciones del sistema (8259).- Es un controlador de interrupciones programable que establece prioridades y enmascaramiento para ocho líneas de solicitud de interrupciones (INT0-7). La más alta prioridad está asignada a INT0. Cada línea puede llevar una solicitud de interrupción desde los interruptores del panel frontal o desde algún usuario del Multibus.

La interrupción de menor prioridad (INT7) es usada por el controlador de interrupciones locales del IPB. INT0 y INT1 están reservadas para el monitor y el ISIS respectivamente.

- Controlador de interrupciones locales (8259).- El controlador de interrupciones locales procesa 7 solicitudes de interrupción: una solicitud de interrupción para datos recibidos y datos transmitidos en cada canal serie de E/S, una solicitud de interrupción del PIO, una solicitud de interrupción del IOC y una solicitud de interrupción del reloj de tiempo real.

Las asignaciones de interrupción para el controlador de interrupciones locales son:

INT0 Datos de entrada listos en el canal serie de E/S 0

INT1 Datos de salida listos en el canal serie de E/S 0

INT2 Datos de entrada listos en el canal serie de E/S 1

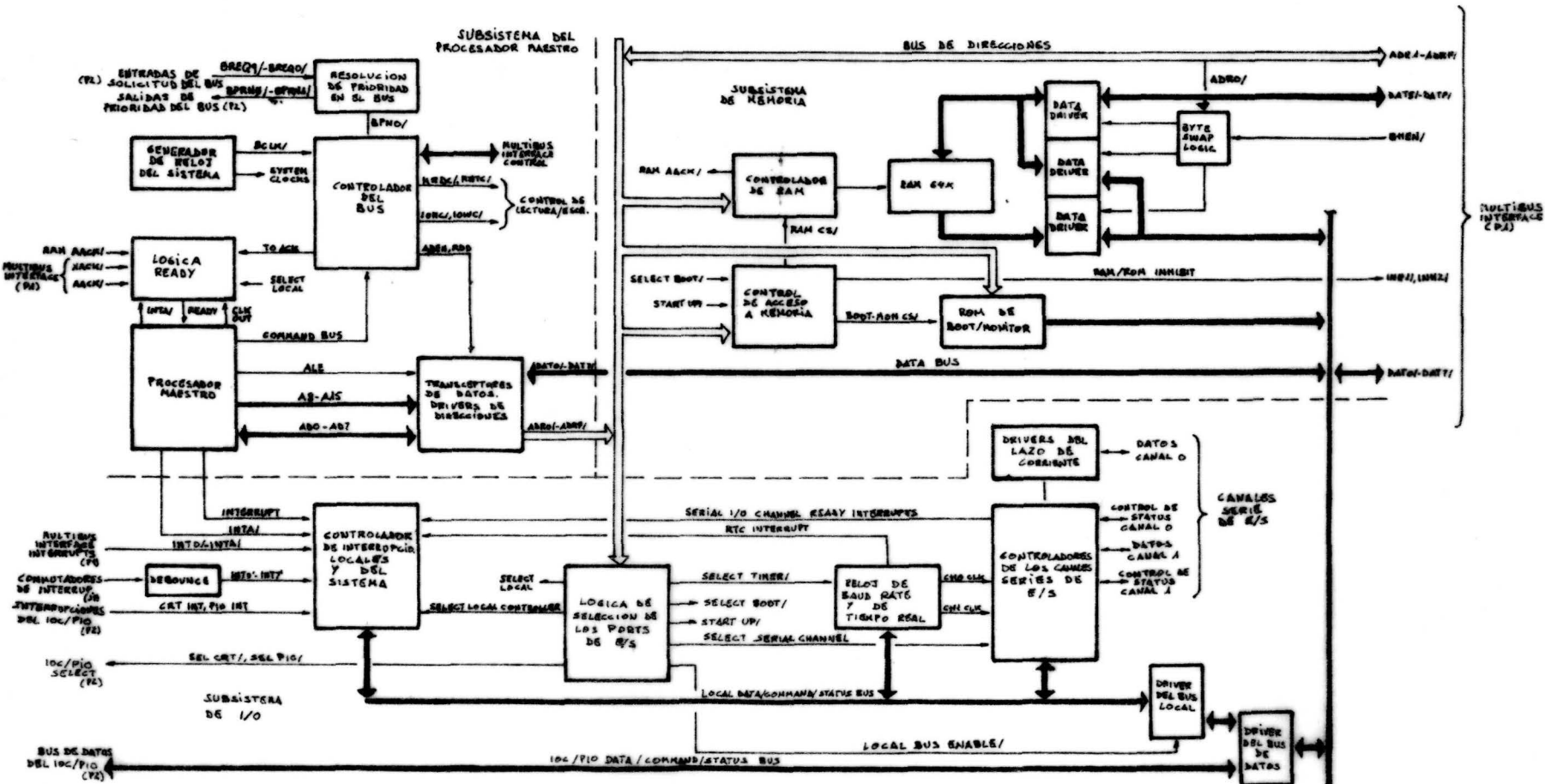
INT3 Datos de salida listos en el canal serie de E/S 1

INT4 Reloj de tiempo real
INT5 Operación completa en el PIO
INT6 Operación completa en el IOC
INT7 No se usa

2.3. TARJETA DEL PROCESADOR INTEGRADO (IPC)

La tarjeta del procesador integrado es un simple sistema computador basado en el microprocesador 8085A desarrollada para el sistema de desarrollo Inteltec Series II. El IPC incluye 64k bytes de RAM, 4k bytes de ROM que contienen los programas de test, diagnóstico y monitor, 2 canales series de E/S así como el interface al multibus, el controlador de E/S (IOC), y el subsistema de E/S paralelo (PIO).

DIAGRAMA DE BLOQUES DEL IPC

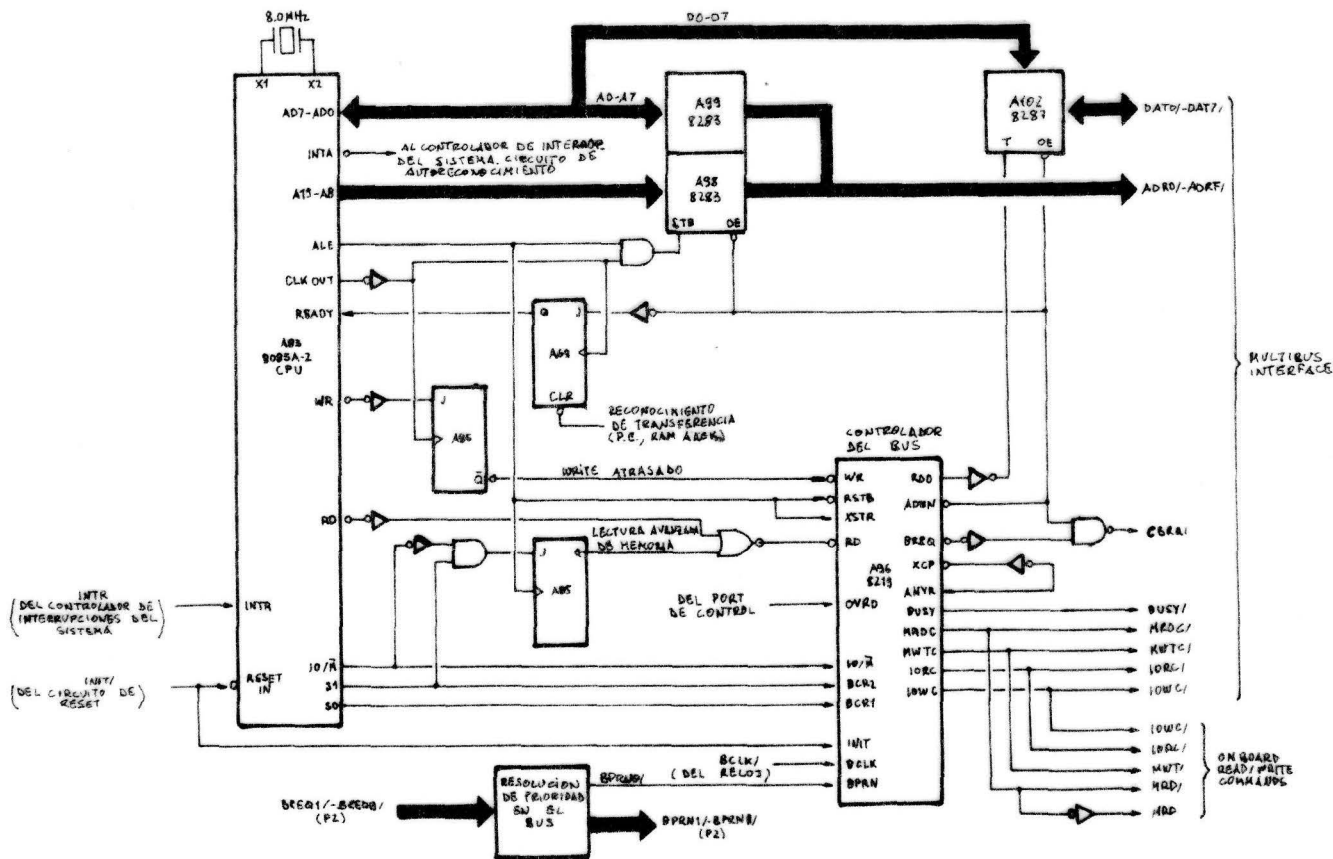


Las transferencias de datos entre el IPC y los periféricos asociados con el IOC/PIO, los canales series de E/S o el multibus, son iniciados mediante interrupciones hardware o como el resultado del análisis del IPC de los estados de los dispositivos a través de puertos dedicados de E/S.

La operación del multibus está ordenada de forma que el IPC tiene la menor prioridad en el bus de los 10 procesadores potenciales que pueden ser conectados al multibus. El uso de los recursos del IPC (p. ej.: canales series de E/S, CRT, programador de PROM, etc.) por otro procesador sólo está restringido a que éste último debe usar los prots de E/S definidos por el IPC. La ROM que contiene los programas de chequeo y diagnóstico no es accesible directamente a otro procesador.

2.3.1. SUBSISTEMA DEL PROCESADOR MAESTRO

El subsistema del procesador maestro es el centro de cómputo y de control del IPC. Los principales componentes del procesador maestro son: microprocesador 8085A-2, controlador de bus 8219, 2 latches octales 8283 y un transceptor octal 8287.



SUBSISTEMA PROCESADOR MAESTRO

2.3.1.1. PROCESADOR MAESTRO

El procesador maestro del IPC es un microprocesador 8085A-2 que opera a 4.0 MHz. Este microprocesador es usado de una manera convencional, con las siguientes excepciones: las funciones HOLD, RST y TRAP no son usadas, un estado de espera es insertado entre cada ciclo de lectura de memoria (o instrucción "fetch") y entre cada ciclo de escritura de memoria son insertados dos estados de espera.

Para establecer los estados de espera, la entrada READY del 8085 es controlada por el flip-flop A69. Al principio de cada ciclo de instrucción, el flip-flop es mantenido en su estado "reset" (READY inactiva) mediante la ausencia de cualquiera de las señales de reconocimiento (RAM ACK/, reconocimiento local, TO ACK, etc.). Durante un ciclo de lectura o de escritura de memoria, la señal de "reconocimiento avanzado de RAM" (RAM AACK/) hace que el flip-flop salga de su estado reset. El flip-flop es por tanto activado en la siguiente pendiente de caída de CLK OUT/ (la señal de salida de reloj de 8085) haciendo que la salida ADEN/ del controlador del bus sea activa.

La operación del controlador del bus es iniciada en la pendiente de bajada de ALE durante el estado T_1 del ciclo de máquina y ADEN/ pasa a activa cuando el controlador del bus adquiere el bus, y permanece activa hasta que otro procesador maestro solicite el bus, o sea ejecutada una instrucción Halt.

Las señales de lectura de memoria (MRD/) y de escritura de memoria (MWT/) del controlador del bus (que hacen que el controlador de RAM inicie un ciclo de acceso a RAM) son desplazadas un ciclo de reloj de la CPU.

La entrada WR al controlador del bus se produce medio ciclo de reloj más tarde, en el estado T_2 , debido a la acción del flip-flop de retardo de escritura, mientras que la entrada RD se produce medio ciclo de reloj antes en la pendiente de subida de ALE en el estado T_1 (ALE activa el flip-flop de avance de escritura para el ciclo "fetch", y para los ciclos de máquina de lectura-escritura).

La remporización de la señal RAM AACK/ del controlador de RAM y el ciclo de diferencia entre las señales MRD/ y MWT/ producen un estado de espera entre cada ciclo de lectura de memoria (en el ciclo "fetch") y dos estados de espera entre cada ciclo de escritura de memoria. Las otras señales de reconocimiento llevadas a la entrada "clear" (CLR) del flip-flop A69 (la salida del reloj de reconocimiento local, TO ACK y RAM XACK/) son usadas para insertar estados de espera cuando sean requeridos

por transferencias internas de E/S, transferencias externas de E/S u operaciones de acceso a memoria.

La señal TO ACK (reconocimiento de salida de reloj) permite al 8085 completar un ciclo de máquina cuando el reconocimiento del comando esperado no es recibido dentro de aproximadamente 10 mseg. RAM XACK/ es la señal de reconocimiento normal (no avanzada) del controlador de RAM. Como el IPC usa la señal RAM AACK/, la señal RAM XACK/ no es requerida por el IPC. RAM AACK/ y RAM XACK/ sin embargo, se producen simultáneamente cuando el controlador de RAM está en el modo "SACK retrasado" (cuando un ciclo de acceso a RAM es solicitado durante un ciclo de refresco de memoria). La salida del reloj de reconocimiento local TO ACK o RAM XACK/ generan la señal de reconocimiento de transferencia al multibus (XACK/) cuando otro procesador maestro usa los recursos del IPC.

Otra señal adicional que es llevada a la entrada "clear" del flip-flop A69 es INTA/ (reconocimiento de interrupciones). Esta señal provee la indicación "ready" requerida por el 8085 durante los ciclos de máquina de reconocimiento de interrupción.

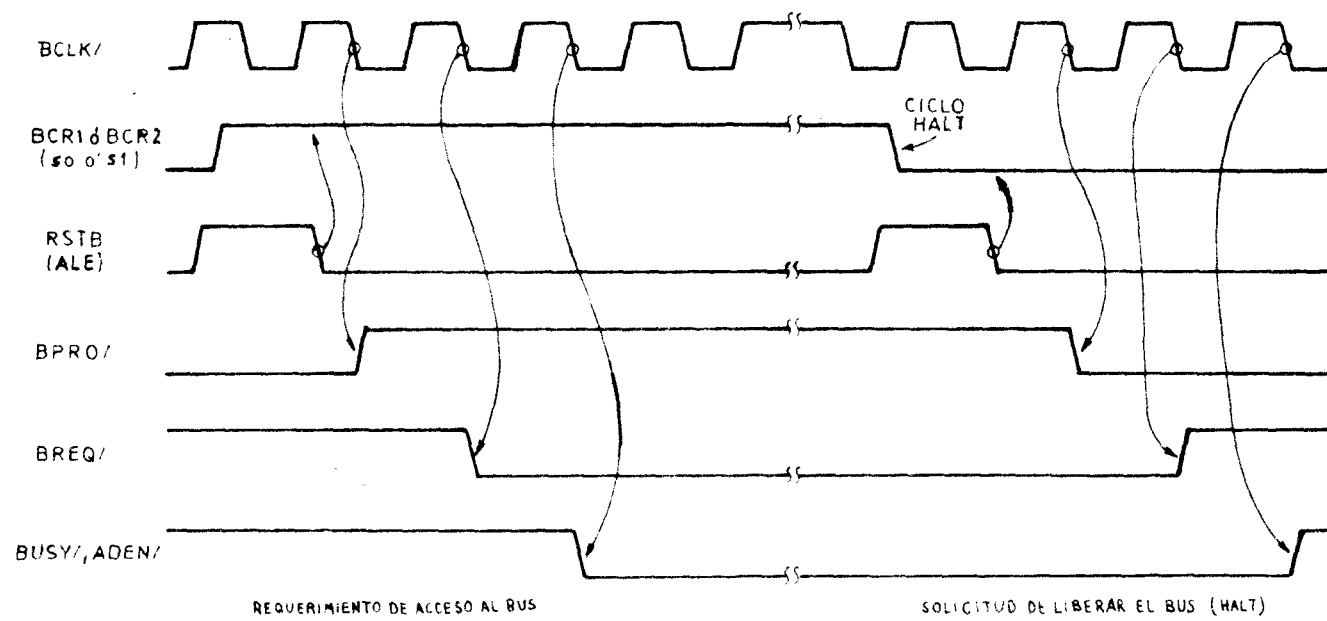
2.3.1.2. CONTROLADOR DEL BUS

El controlador de bus 8219 provee las funciones de adquisición, temporización y control del bus para el IPC.

2.3.1.2.1. ADQUISICION DEL BUS

Cuando el procesador maestro del IPC requiere el uso del bus del sistema, el controlador del bus asegura el acceso al bus y mantiene éste hasta que otro procesador maestro solicita el bus (el IPC tiene la menor prioridad en el bus) o hasta que el procesador maestro libera el bus mediante una instrucción Halt.

Para iniciar un ciclo de adquisición del bus, el controlador del bus necesita un nivel activo en las entradas BCR1, BCR2 o en ambas y un nivel activo en la entrada RSTB/. Las entradas BCR1 y BCR2 son las señales de estado S0 y S1 del 8085. Recordando la temporización de los ciclos de máquina del 8085, las salidas de estados y ALE pasan a ser activas al principio del estado T_1 de cada ciclo de máquina, y ALE vuelve a un nivel inactivo (generando RSTB/) poco después pero estando aún en el estado T_1 .



ADQUISICION DEL BUS

Si otro procesador maestro no está solicitando el bus, la entrada BPRNO/ (prioridad 0 de entrada al bus) del circuito lógico de prioridad en el bus será activa.

En el tercer ciclo de reloj del bus (BCLK/), si el bus no está en uso (indicado por un nivel inactivo en BUSY/) y si no ha sido recibida una solicitud de uso del bus de otro procesador maestro, el controlador del bus activa BUSY/ y ADEN/ (habilitación de direcciones y datos). BUSY/ es una señal bidireccional del bus que cuando está activa indica a todos los demás procesadores que el bus se está usando. ADEN/ es usada localmente por el IPC para habilitar los "latches" de direcciones (A98 y A99), los transceptores de datos (A102) y el flip-flop A69. Una vez el IPC ha adquirido el bus, el controlador del bus mantiene el acceso a éste (manteniendo BUSY/ activa) hasta que otro procesador solicite el bus o se ejecute una instrucción Halt.

Si el bus está en uso (si BUSY/ está a nivel bajo) o si un requerimiento del bus está pendiente (si BPRNO/ está a nivel alto) cuando el 8085 solicita el bus, el controlador del bus retrasa la generación de BUSY/ y ADEN/ hasta un ciclo de BCLK/ después de que el bus llegue a estar disponible y no hayan solicitudes pendientes.

2.3.1.2.2. LIBERACION DEL BUS

Una vez que el procesador maestro ha adquirido el bus, sólo lo abandona cuando otro procesador solicite el bus o cuando el procesador maestro ejecute una instrucción Halt. Cuando otro procesador solicita el bus, el controlador del bus completa su actual ciclo de control del bus y libera ésta (inactiva BUSY/) en el siguiente ciclo de reloj del bus. Cuando se ejecuta una instrucción Halt, el controlador del bus libera éste en el tercer ciclo de reloj del bus.

2.3.1.2.3. SOBRECARGA DEL BUS

El controlador del bus tiene una entrada OVRD (sobrecarga) que cuando está activa permite al procesador maestro mantener el acceso al bus entre ciclos del bus previniendo el reconocimiento de una solicitud del bus desde otro procesador. La entrada OVRD es tomada del bit 6 del puerto de control del IPC. OVRD es activada automáticamente cuando el sistema es inicializado para prevenir que otro procesador acceda al bus durante el procedimiento de inicialización del procesador maestro (después de la inicialización, el programa monitor inactiva OVRD).

2.3.1.2.4. CICLO DE CONTROL DEL BUS

El controlador del bus necesita la recepción de una solicitud de comienzo de transferencia (XSTR) y la adquisición del bus (BUSY/ y ADEN/ activas) antes de que pueda empezar un ciclo de control del bus. La entrada XSTR al controlador del bus es la señal ALE del 8085 (XSTR es activa en la pendiente de subida de ALE, mientras que RSTB es activa en la pendiente de caída). Por tanto, si el controlador del bus ya tiene acceso al bus, el ciclo de control es iniciado por XSTR; si el controlador del bus no tiene acceso al bus, el ciclo de control es iniciado cuando el bus es adquirido (cuando BUSY/ y ADEN/ sean activas).

Para todos los ciclos de máquina que requieren acceso al bus, el 8085 usa IO/M y RD o WR para definir el comando del controlador del bus a ser generado para el ciclo de control. Cuando las entradas del controlador del bus RD o WR pasan a ser activas, el controlador del bus activa ANYR (cualquier requerimiento). Si se especifica una operación de lectura (RD activa), el controlador del bus activa adicionalmente RDD. ANYR es conectada externamente a la entrada XCP del controlador del bus (transferencia completa), mientras que RDD es usada para controlar la dirección de los transceptores del bus de datos (A102).

2.3.1.4. LOGICAL DE RESOLUCION DE PRIORIDAD EN EL BUS

El "logical" de resolución de prioridad en el bus incluye un codificador de prioridades de 8 a 3 (74148) y un decodificador de 3 a 8 (74138). Las entradas primarias al codificador son las señales de solicitud del bus BREQ1/ hasta BREQ8/. Si hay más de una de estas señales activa, el codificador sólo genera el código binario para la solicitud de mayor prioridad (la de más alta numeración). El decodificador usa el código binario del codificador para activar la señal de prioridad de entrada al bus (BRPNx/) apropiada, al procesador que lo solicita.

Los "masters" del bus de mayor y menos prioridad (el IPC) están excluidos del logical de resolución de prioridad. La solicitud de mayor prioridad (BREQ9/) se usa para bloquear todas las otras solicitudes, y no requiere la señal BPRN/ porque asume un control total del bus tan pronto como haya sido completado el último ciclo de transferencia del bus. El IPC solicita continuamente el uso del bus, pero es inhibido por BPRNO/ si otro maestro del bus está solicitando éste.

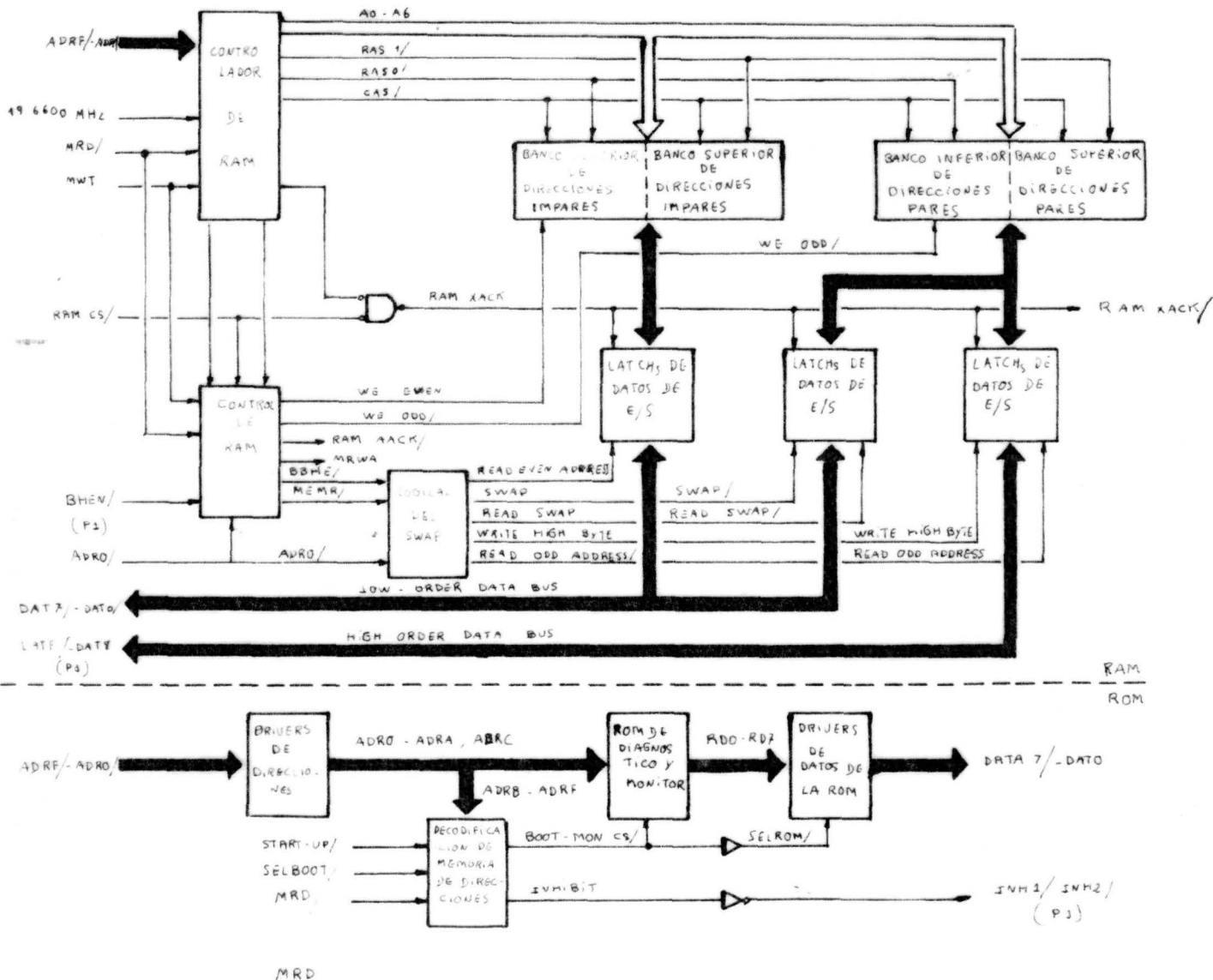
Las líneas BREQ/ y BPRN/ están implementadas de forma que

la prioridad de las placas insertadas en el sistema de desarrollo se incrementa hacia la parte más baja de éste. Es por eso que el IPC tiene la menor prioridad, ya que solamente puede ser conectado en la parte más alta del sistema de desarrollo.

2.3.2. SUBSISTEMA DE MEMORIA

El subsistema de memoria del IPC controla 64k bytes de RAM y 4k bytes de ROM. Como el procesador maestro sólo tiene 16 líneas para direcciones (solamente puede direccionar 64k bytes de memoria), los dos tipos de memoria, RAM Y ROM, comparten el mismo espacio de direcciones. La técnica que permite direccionar espacios compartidos es conocida como de "ROM ensombrecida", en donde la ROM es seleccionada preferentemente a menos que sea deshabilitada. Desde un punto de vista del hardware, la RAM y la ROM son simplemente dos esclavos del procesador maestro.

Los principales elementos del subsistema de memoria son: 32 RAMs dinámicas de 16k bits, una ROM de 4k bytes, un controlador de RAM 8202, una ROM 3625A (para decodificación de las direcciones de memoria), y una serie de circuitos TTC.



2.3.2.1. ROM DEL IPC

La memoria ROM está formada por dos segmentos de programa separados contenidos en un simple chip de 4k de ROM. Un segmento contiene el programa monitor, y el otro el programa de diagnóstico y testeo. El programa monitor es accesible a todos los maestros del bus y empieza en la dirección F800H. Este segmento no puede ser "ensombrecido", y los 2k últimos (desde F800H hasta FFFFH) de RAM no pueden ser usados.

El segmento de ROM de diagnóstico y testeo solamente es accesible al procesador maestro del IPC. Para habilitar o deshabilitar este segmento de ROM se usan bits individuales dentro del port de control del procesador maestro (port FF). Cuando el procesador maestro es inicializado, el port de control (A84) es reseteado (SEL BOOT/, START UP/ y OVRD activas y las interrupciones deshabilitadas). La combinación de START UP/ y SEL BOOT/, en el decodificador de direcciones de memoria, establecen la posición de comienzo del segmento de ROM de diagnóstico y testeo en la posición 0H, así como en la posición E800H. El programa de testeo salta inmediatamente a la posición E800H del segmento de ROM e inactiva la salida START UP/ del port de control (escribiendo 02H en el port FF). En el decodificador de direcciones de memoria, el nivel inactivo de START UP/ deshabilita el segmento de ROM que empieza en la posición 0H, y el segmento de diagnóstico y testeo es habilitado sólo para las direcciones desde E800H hasta EFFFH. Deshabilitando el segmento de ROM de la posición 0H redefinimos las posiciones antes de la E800H como RAM. Después de la ejecución del programa de diagnóstico y testeo, el control es transferido al programa monitor en la posición F800H. El programa monitor, cuando empieza su ejecución, inactiva la salida SEL BOOT/ del port de control (escribiendo 04H en el port FF).

En el decodificador de direcciones de memoria, el nivel inactivo de SEL BOOT/ deshabilita la entrada de selección del chip de la ROM para las direcciones entre E800H y EFFFH (el segmento de ROM de testeo y diagnóstico desaparece del espacio de direcciones) y todas las direcciones antes de la F800H son redefinidas como RAM.

Hay que hacer notar, sin embargo, que en el programa monitor se incluye la posibilidad de rehabilitar la porción de diagnóstico de la ROM. Entrando el comando del monitor Z\$ se transfiere el control al programa de diagnóstico (este comando reactiva la salida SEL BOOT/ escribiendo 0CH en el port FF).

El port de control también incluye un bit de habilitación de interrupciones. Este bit es limpiado (las interrupciones deshabilitadas) cuando el IPC es inicializado, y posteriormente activado cuando se entra en monitor. Las rutinas del monitor que controlan el IOC y el PIO limpian automáticamente este bit cuando son llamadas para prevenir que sean servidas otras interrupciones.

2.3.3.2. RAM DEL IPC

Los 64k bytes de RAM del IPC están implementados con 32 chips de RAM dinámica de 16k x 1 bit que están divididos en 4 bancos de 16k bytes. Los dos bancos más bajos contienen las direcciones de las posiciones pares e impares desde la 0H a la 7FFFH, y los dos bancos más altos contienen desde la 8000H hasta FFFFH. La selección de un banco individual está determinada por el estado de los bits de direcciones de más bajo y más alto orden (ADRO selecciona si el banco es par o impar y ADRF selecciona el banco de orden bajo o alto). Los 14 bits de direcciones restantes (ADRI-ADRE) son multiplexados por el controlador de RAM para formar una matriz de 7 pilas (7 bits) y 7 columnas (7 bits). En otras palabras, cada uno de los 8 chips de RAM en el banco forman una matriz de 128 por 128 bits que pueden ser direccionados mediante la multiplexación de una matriz de 7 por 7 bits.

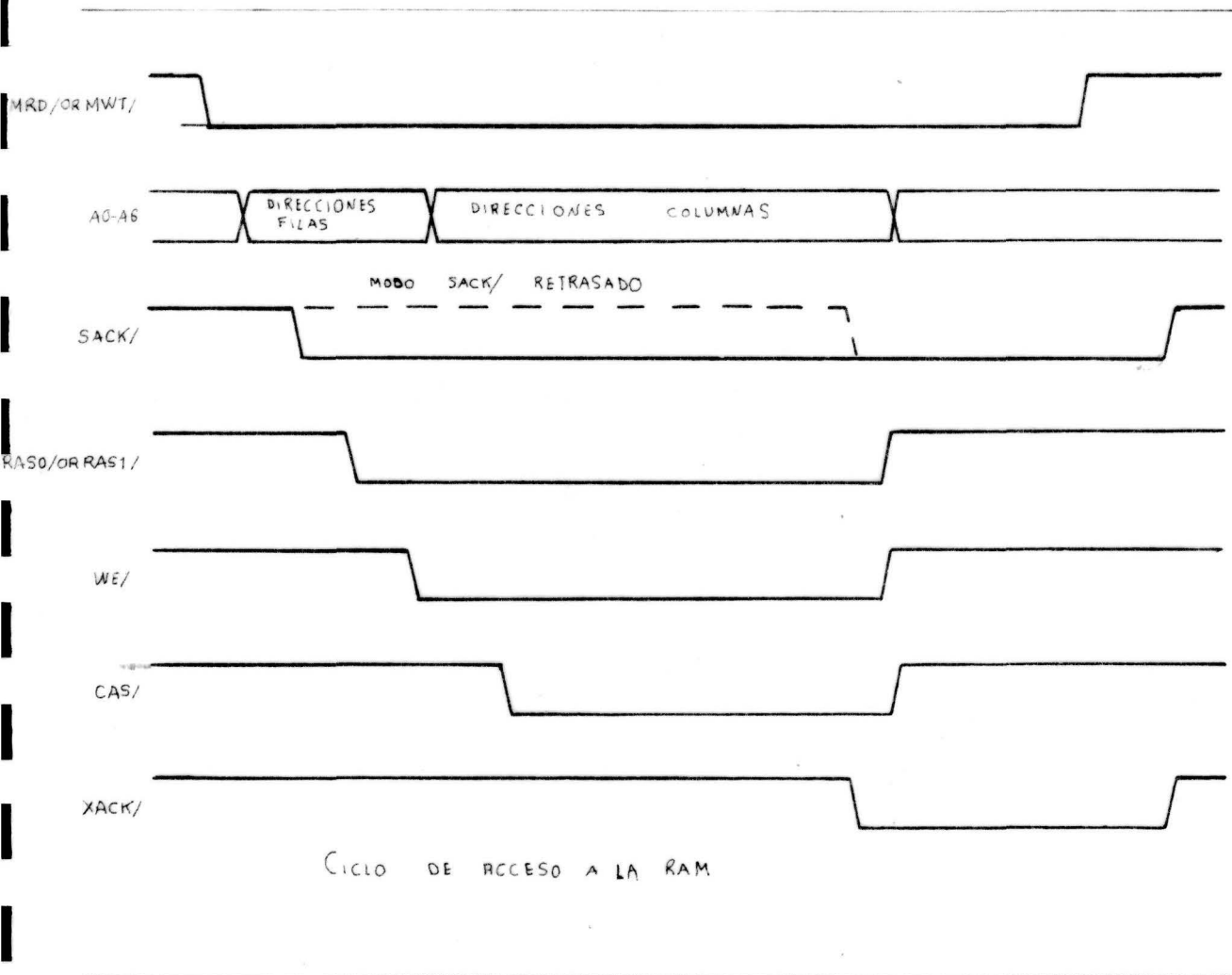
Todos los ciclos de acceso a la RAM, bien cuando sean iniciados por el procesador maestro del IPC o bien por otro master del bus, son controlados por un controlador de RAM 8202 de Intel. El controlador de RAM es seleccionado para un ciclo de acceso por la señal RAM CS/ del decodificador de direcciones de memoria.

Esta señal es activa cuando la dirección que hay en el bus está dentro del rango 0H hasta F7FFFH

El ciclo de acceso a lectura o a escritura es iniciado por MRD/ o MWT/ respectivamente. El controlador de RAM cuando detecta un nivel activo en su entrada RD o en su entrada WR, comienza el ciclo de acceso multiplexando los 7 bits de direcciones a las entradas de los bancos ALO hasta AL6 como una fila de 7 bits activando SACK/ (reconocimiento del sistema) y RASO/ o RAS1/ para validar la fila de direcciones en los bancos bajos (RASO/ activa) o en los bancos altos (RAS1/ activa). Si se especifica una operación de escritura (WR activa), el controlador de RAM activa la señal WE/ (habilitación de escritura). Esta señal es usada por el control lógico de la RAM para generar WE EVEN/ o WE ODD/ según sea el estado del bit de direcciones

más bajo (ADRO). El controlador de RAM multiplexa entonces los 7 bits de direcciones a las entradas de los bancos AHO hasta AH6 como una columna de 7 bits, y activa CAS/ (validación de columna de direcciones) para acceder al byte direccionado. Cuando finaliza la señal CAS/ (aproximadamente después de 250 ns.), el controlador de RAM activa la señal XACK/ (reconocimiento de transferencia). Esta señal se usa para guardar el byte de datos en el latch de entrada seleccionado (en el caso de la lectura).

En caso de que se solicite un ciclo de acceso a la RAM durante un ciclo de refresco, SACK/ es retrasada y coincide con la señal XACK/.



El controlador de RAM usa un reloj interno para iniciar un ciclo de refresco cada 14.6 microseg. Durante el ciclo de refresco, el controlador de RAM activa las señales RAS0/ y RAS1/ (sin activar CAS/) para refrescar una fila completa (128 bytes) dentro de cada banco. Un contador de filas interno de módulo

128 es incrementado con cada ciclo de refresco, de forma que 64k bytes son refrescados en aproximadamente 2 mseg.

El control lógico de la RAM usa el bit de direcciones más bajo (ADRO) para determinar la selección de un banco par o impar, y usa ADRO y la entrada BHEN/ (habilitación de byte alto) del multibus para determinar la ruta del byte de datos entre la RAM y los buses de datos alto y bajo.

La siguiente tabla muestra la operación del control lógico para los accesos de lectura y escritura a la RAM:

Control Signal Levels		Banks Selected*	Data Lines Selected	Control Outputs Active	
ADRO	BHEN/			Read Access	Write Access
Low	High	Even	DAT0/-DAT7/	READ EVEN ADDRESS/	WE EVEN/
High	High	Odd	DAT0/-DAT7/	SWAP/ READ SWAP/	WE ODD/ SWAP/
Low	Low	Both	DAT0/-DATF/	READ EVEN ADDRESS/ READ ODD ADDRESS/ (WRITE HIGH BYTE/)	WE EVEN/ WE ODD/ WRITE HIGH BYTE/
High	Low	Odd	DAT8/-DATF/	READ ODD ADDRESS/	WE ODD/ WRITE HIGH BYTE/

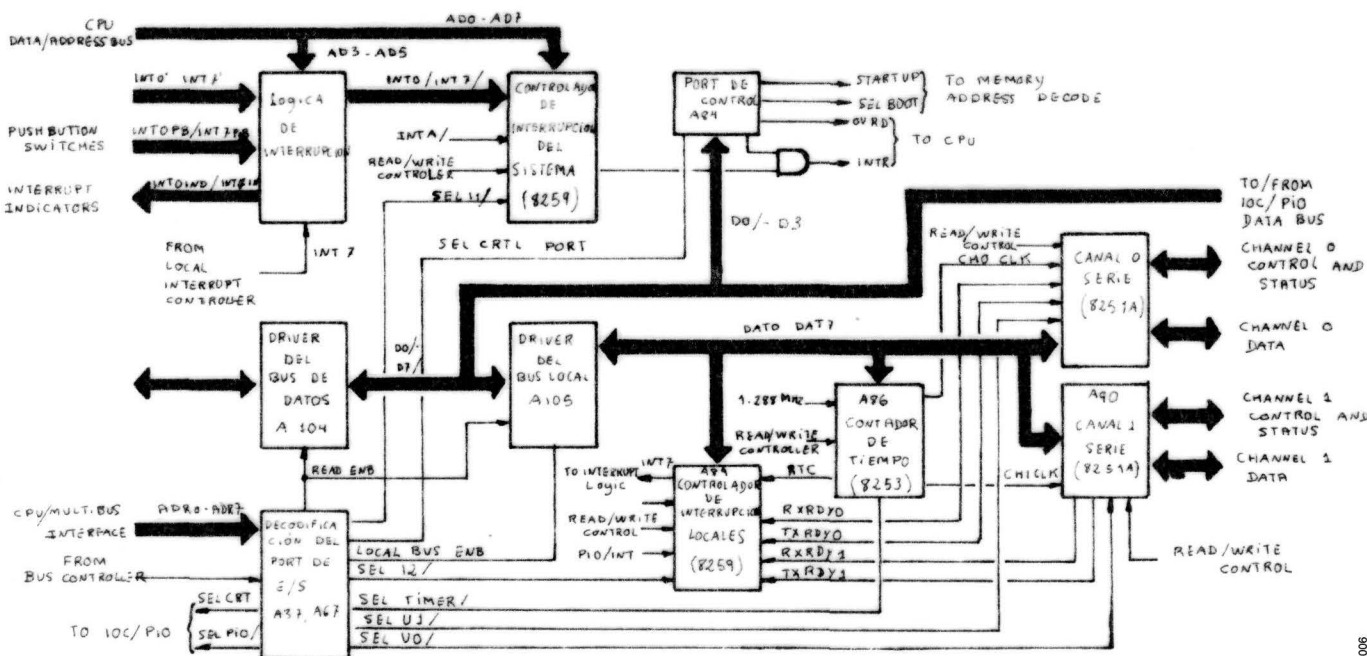
OPERACION DE CONTROL LOGICO DE LA RAM

2.3.3. SUBSISTEMA DE ENTRADA-SALIDA DEL IPC

El subsistema de E/S provee el control de todas las comunicaciones del sistema de desarrollo con el procesador maestro del IPC. Tales comunicaciones comprenden: transferencias de E/S del "sistema" o desde otras placas opcionales del multibus, y transferencias de E/S "locales".

Las transferencias de E/S del "sistema" se llevan a cabo a través del interface del multibus.

Las transferencias de E/S "locales" usan el multibus, pero se desarrollan a través de ports de E/S del procesador maestro. Las transferencias de E/S locales se pueden dividir en transferencias externas (PIO y IOC) y transferencias internas. Todas las transferencias de E/S internas son entre el procesador maestro del IPC y todos los circuitos directamente programables por el IPC.



SUBSISTEMA DEL E/S DEL IPC

La decodificación de las direcciones de los ports de E/S internos y externos se realiza mediante un decodificador 16-bit. Este decodifica diecinueve direcciones de ports distintas para producir diez señales de control.

En la siguiente table se muestran las direcciones de los ports y las señales asociadas de selección de chip para el control de E/S interno y externo:

Port Address	Device Selected	Port Function	Output Signal Active
F0		Channel 0 Baud Rate	SEL TIMER
F1	8253 Timer	Channel 1 Baud Rate	LOCAL BUS ENB/
F2		Real Time Clock	SEL LOCAL
F3		Counter Mode Select	
F4		8251 Serial I/O	Data
F5	Channel 0	Command/Status	LOCAL BUS ENB/
F6	8251 Serial I/O	Data	SEL LOCAL
F7		Channel 1	Command/Status
F8	PIO Subsystem	Data	
F9		Command/Status	
FA	8259 Local	Command/Status	
FB	Interrupt Controller	Command/Status	
FC	8259 System	Command/Status	
FD	Interrupt Controller	Command/Status	
FE	Reserved		
FF	Control Port	Acces Control	
CO		Data	
C1	Input/Output Controller	Command/Status	
C2		Reserved	
C3		Reserved	
		Reserved	

2.3.3.1. TRANSFERENCIAS DE E/S EXTERNAS (PIO/IOC)

El subsistema de E/S del IPC controla la ruta de las transferencias externas de E/S hasta y desde el PIO y el IOC. En el caso de transferencias externas, el subsistema de E/S decodifica las direcciones de los ports de E/S y habilita las líneas de datos necesarias del PIO y del IOC. El subsistema de E/S también usa las direcciones de los ports para generar una señal que selecciona el PIO o el IOC.

Más adelante, volveremos a entrar en más detalles con respecto al PIO y al IOC y a sus ports de control.

2.3.3.2. TRANSFERENCIAS INTERNAS DE E/S

Los circuitos de control lógico de E/S internos pueden establecer comunicaciones entre el procesador maestro y los chips programables del subsistema de E/S del IPC. El proceso desarrollado en la comunicación puede ser explicado mediante la siguiente secuencia:

- Selección del chip mediante la dirección del port de E/S
- Selección del modo de transferencia: entrada o salida de comando y entrada o salida de datos
- Habilitación de las líneas de datos entre el chip y el procesador maestro

- Activar el reloj de reconocimiento local de 2 microseg. de retraso para permitir al procesador maestro completar el ciclo de transferencia (entrada READY del microprocesador activa).
- Transferencia del comando, estado o byte de datos a través del bus de datos.

Dentro del subsistema de E/S se emplean un total de cinco chips programables:

- Canal serie 0 de E/S (8251).- Está formado por un transmisor-receptor universal sincrónico/asincrónico (USART) que establece un interfaz RS-232 o un lazo de corriente de 20 mA. entre un dispositivo serie externo y el procesador maestro del IPC. Mediante comandos se puede establecer comunicación asincrónica "full duplex" desde 110 hasta 1200 baudios para el interfaz del lazo de corriente usando 7 bits más paridad para compatibilidad con teletipo. Mediante comandos se puede establecer también comunicación sincrónica "full duplex" desde 150 hasta 56k baudios para compatibilidad con RS-232 o terminales bisíncronos, o modems. El canal serie puede usar reloj interno o externo según la colocación del "Jumper". Los bytes de estados retornados en comandos indican el estado de la transferencia que se esté realizando en ese momento y la existencia de algún error de transmisión. Cuando el sistema es inicializado, el programa de "test" prepara el canal para operar en modo asincrónico con 2 bits de stop, 8 bits de longitud carácter, y un factor de baud rate, de 16X (la señal del reloj externo del canal 0 es de 1.76 kHz., que dividida por 16, provee un "baud rate" de 110).
- Canal serie 1 de E/S (8251).- Es igual al canal 0, con la excepción de que no dispone de los circuitos necesarios para establecer el lazo de corriente de 20 mA. El canal serie 1, cuando el sistema es inicializado, es programado igual que el canal serie 0. La señal del reloj externo del canal 1 es de 38.4 kHz. proveyendo un "baud rate" de 2400.
- Contador de Baud rate/tiempo real (8253).- Está formado por un reloj programable constituido por 3 contadores independientes. Dos de estos contadores establecen los "baud rates" para los canales series

de E/S. El tercer contador se usa para generar una señal de reloj en tiempo real de 1 mseg. Mediante comandos se puede establecer la frecuencia y el modo de operación del reloj/contador. El contenido de cada contador puede ser llevado al procesador maestro mediante comandos. El reloj de tiempo real no es usado por el sistema de desarrollo; está reservado para aplicaciones del usuario. Cuando el sistema es inicializado, los tres contadores son programados para el modo tres (generador de onda cuadrada) y usan un registro contador de 2 bytes. Los valores del registro contador y las funciones de cada contador son:

Counter	Function	Counter Value
0	Channel 0 Clock	698
1	Channel 1 Clock	32
2	1 ms Real Time Clock	1229

- Controlador de interrupciones del sistema (8259).-

Está formado por un controlador de interrupciones programable que establece prioridades y máscaras para 8 líneas de solicitud de interrupciones (INT0-INT7). La prioridad más alta es asignada a INTO. Cada línea puede llevar una solicitud de interrupción, bien de los interruptores del panel frontal, bien de un usuario del multibus. Algunos dispositivos del multibus, tales como el módulo ICE, pueden emplear una o más interrupciones reservadas para habilitar la identificación por software, pero no existe distinción alguna entre una solicitud de interrupción desde el multibus y su equivalente del panel frontal. La interrupción de menor prioridad (INT7) es usada por el controlador local de interrupciones del IPC. Este controlador opera de forma que si una solicitud de interrupción está siendo servida, puede ser interrumpida por otra de mayor prioridad. Los estados retornados mediante comandos incluyen la identidad de la interrupción que está siendo servida, las identidades de todas las interrupciones que quedan pendientes, y el contenido del registro de máscara interno.

De las 8 interrupciones, INTO y INT1 son reservadas por el monitor y el ISIS, e INT7 es usada por el controlador de interrupciones locales.

- Controlador de interrupciones locales (8259).- El controlador de interrupciones locales procesa siete solicitudes de interrupción: una solicitud de interrupciones debido a la recepción o transmisión de datos de cada canal serie de E/S, una solicitud de interrupción del PIO, una solicitud de interrupción del IOC, y una solicitud de interrupción del reloj de tiempo real. Después de que se produzca INT7, el programa del procesador maestro analiza el controlador de interrupciones locales para determinar el origen de la solicitud de interrupción. Un código que identifica la interrupción de más alta prioridad que está siendo servida es retornado al procesador maestro a través de las líneas de datos. En todo lo demás, el controlador de interrupciones locales funciona igual que el controlador de interrupciones del sistema.

La asignación de interrupciones para el controlador de interrupciones locales es la siguiente:

INT0 Datos de entrada listos en el canal serie de E/S 0
INT1 Datos de salida listos en el canal serie de E/S 0
INT2 Datos de entrada listos en el canal serie de E/S 1
INT3 Datos de salida listos en el canal serie de E/S 1
INT4 Reloj de tiempo real
INT5 Operación completa del PIO
INT6 Operación completa del IOC
INT7 No se usa

La dirección del port de E/S se usa con el comando de E/S de lectura o de escritura para controlar la transferencia de un comando, un estado, o un byte de datos entre el chip programable y el procesador maestro. Las señales IORC/ y IOWC/ solamente establecen la dirección del flujo de datos. Las salidas SELI1/ y SELI2/ del decodificador lógico de ports son usadas para seleccionar el controlador de interrupciones locales 8259 y el controlador de interrupciones del sistema 8259, respectivamente. Ambos controladores usan IORC/ para habilitar la

lectura de estados y IOWC/ para habilitar la escritura de comandos, y ADRO para aumentar los comandos que está apareciendo en la línea de datos.

Una diferencia entre la implementación de los 2 controladores es que el controlador de interrupciones del sistema está conectado directamente al bus de datos de la CPU (ADO-AD7), mientras que el controlador de interrupciones locales está conectado al bus de datos del sistema (DATO/-DAT7/) a través de los drivers del bus de datos local. Es importante destacar que la señal de "selección de chip" del controlador de interrupciones está conectada a ADEN/, y por tanto, sólo puede ser direccionado por el procesador maestro del IPC.

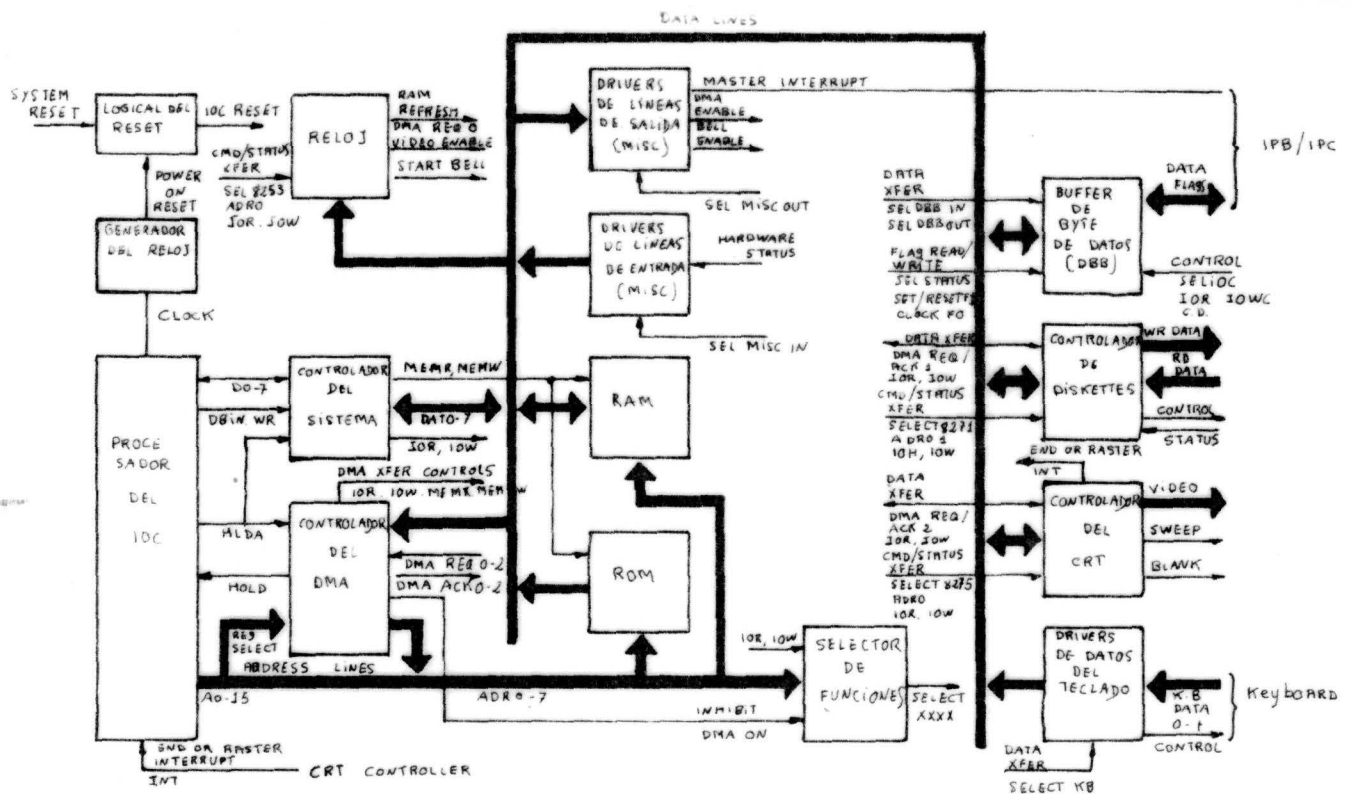
2.4. CONTROLADOR DE ENTRADA-SALIDA (IOC)

El controlador de E/S está formado por todos los elementos hardware necesario para establecer la comunicación entre el IPB/IPC y el diskette integral o la pantalla.

El IOC y el PIO ejecutan un programa de ROM residente para controlar los dispositivos, mientras que los datos para o desde los dispositivos son procesados solamente por el IPB/IPC. La similitud funcional del IOC y del PIO permiten el uso de interfaces del IPB/IPC que son virtualmente idénticos en cuanto a la secuencia de señales hardware y al formato de los comandos y registros de estado empleados.

Este interface común del IOC y del PIO con el IPB/IPC es conocido como "buffer de byte de dato" (DBB). En el PIO, el DBB está implementado dentro del microprocesador 8041. En el IOC, el DBB está implementado por elementos TTC discretos. En ambos casos el DBB es un buffer bidireccional que provee almacenamiento para un byte de entrada, y un byte de salida. Hay 4 "flags" asociados con el DBB:

- Buffer de entrada lleno (IBF): indica la presencia de un nuevo byte de entrada.
- Buffer de salida lleno (DBF): indica la presencia de un nuevo byte de salida.
- Comando/dato (C/D): identifica el byte como información de control (comandos o estados) o datos.
- Flag de ocupado (BUSY) (FO): indica la imposibilidad de aceptar un nuevo comando del IPB/IPC (ya que el comando actual está siendo procesado).



DIAGRAMAS DE BLOQUES DEL IOC

Los flags del DBB son usados para alertar al IOC cuando un comando o un byte de datos es entrado por el IPB/IPC, y para indicar cuando es aceptado un estado o un byte de datos de salida por el IPB/IPC. Los flag también son usados por el IPB/IPC para determinar el estado de la operación del IOC. La secuencia operacional básica del IOC es:

- Emisión de un comando por el IPB/IPC
- Aceptación del comando y los asociados bytes de datos de entrada por el IOC
- Análisis de los flags del DBB por el IPB/IPC
- Aceptación de los datos de salida o bytes de estado del IOC por el IPB/IPC.

Todas las transferencias de datos, comandos y bytes de estados entre el IOC y el IPB/IPC son controladas por el IPB/IPC, y cada transferencia de datos o de estados es en respuesta a un comando específico; el IOC no puede iniciar una transferencia a o desde el IPB/IPC. Sin embargo, una serie de comandos del IPB/IPC pueden iniciar una secuencia de acontecimientos relativamente complejos dentro del IOC. Por ejemplo, los comandos del IPB/IPC pueden establecer los parámetros para la transferencia de bloques desde el diskette y entonces controla la transferencia de datos desde el IOC al IPB/IPC. El IOC usa los parámetros

del bloque para efectuar transferencias del bloque mediante el DMA desde el IOC al IPB/IPC. Las transferencias de datos son hechas entonces de la RAM del IOC al IPB/IPC. La escritura al diskette es efectuada en la misma forma excepto que la transferencia del bloque mediante el DMA ocurre después de que todas las transferencias de bytes desde el IPB/IPC a la RAM han sido completadas.

La otra transferencia de datos a través del DMA es con el CRT. En este caso, las transferencias de bloques a través del DMA desde la RAM son usadas para realizar el refresco de la pantalla CRT. La transferencia de bytes desde el IPB/IPC al segmento del display de la RAM del IOC apenas modifican los datos que van a ser visualizados. Los bloques del DMA para el CRT son de tamaño fijo (suficiente para llenar la pantalla), y las áreas en blanco de la pantalla están formadas por el carácter "espacio".

En contraste al procesador maestro del IPB/IPC, el procesador del IOC sólo hace uso de una interrupción hardware. Esta simple interrupción es generada por el controlador del CRT (8275) al final del "raster" de visualización del CRT. La interrupción lleva al procesador del IOC a determinar las direcciones de la RAM que van a ser accedidas en el siguiente "raster" del display. Los nuevos parámetros (dirección de comienzo y fin de cuenta) para la primera parte de la pantalla son sacados al canal 2 del DMA, mientras que los nuevos parámetros para la segunda parte de la pantalla son sacados al canal 3 del DMA.

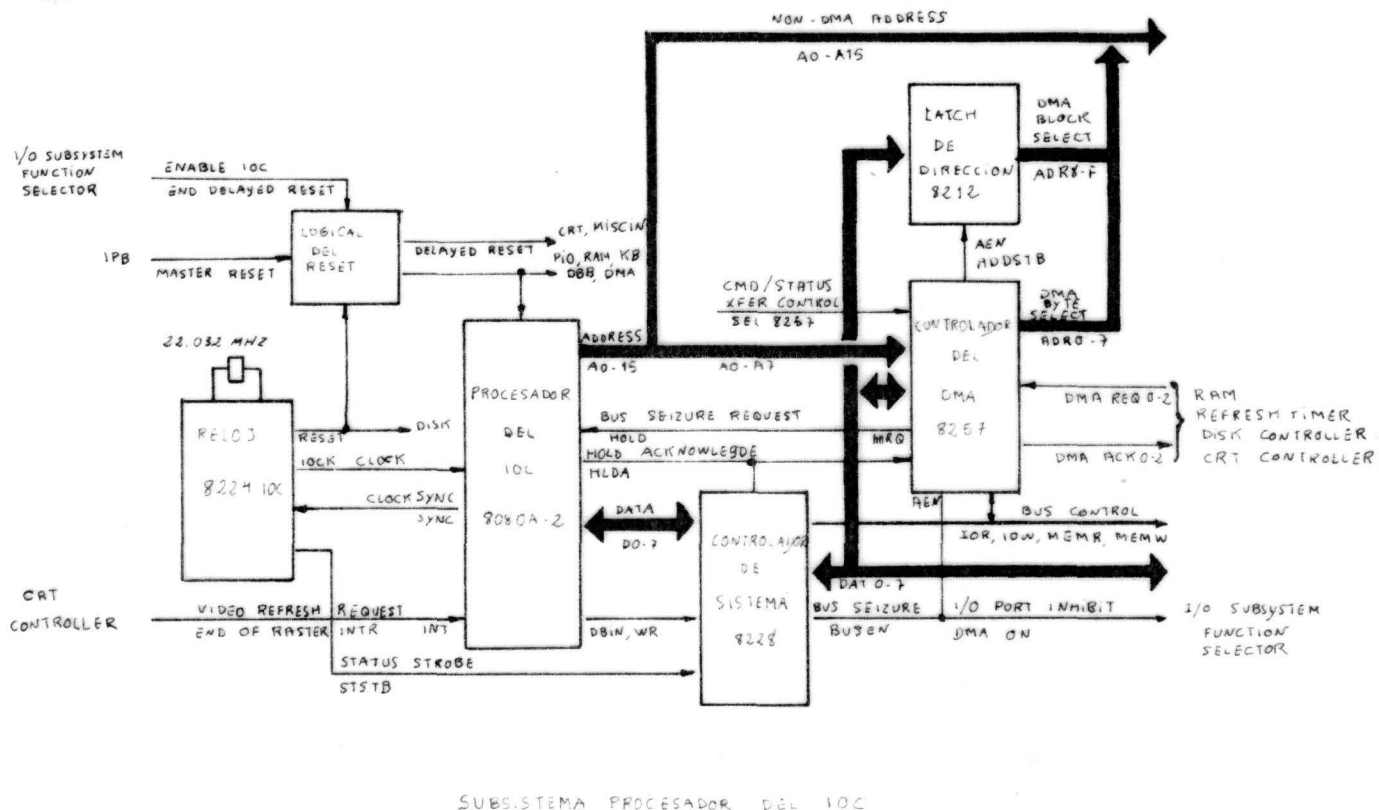
El canal 2 del DMA provee los datos de la pantalla, y cuando el canal 2 alcanza su final de cuenta, los parámetros previamente sacados al canal 3 son transferidos automáticamente al canal 2.

La parte más baja de la pantalla, que contiene las entradas más recientes, aparece entonces en la pantalla. Los canales 0 (refresco de la RAM) y 1 (transferencia de datos con el diskette) del DMA no usan interrupciones hardware.

Además de los cuatro canales de DMA, todas las comunicaciones con el procesador del IOC son efectuadas a través de los ports de E/S. Los ports de E/S son usados para la comunicación con el procesador del IPB/IPC a través del DBB, para la entrada de datos y estados del procesador del teclado, para la comunicación con los chips programables del IOC, y para controlar de otra forma las operaciones del IOC. Los chips programables incluyen un controlador de DMA (8257), un reloj de "intervalos" (8253), un controlador de diskette 8271 y un controlador de CRT (8275).

2.4.1. SUBSISTEMA PROCESADOR DEL IOC

El subsistema procesador del IOC incluye un microprocesador 8080A-2, un generador de reloj 8224, un controlador de sistema 8228, un controlador de DMA 8257, y un registro de 8 bits bidireccional (8212) que mantiene los 8 bits más significativos del controlador de DMA que direccionan la RAM del IOC.



SUBSISTEMA PROCESADOR DEL IOC

2.4.1.1. PROCESADOR DEL IOC Y GENERADOR DE RELOJ

El generador de reloj emplea un cristal de 22.032 MHz. para producir una señal de reloj de 2.448 MHz. que es usado en todas las partes del IOC.

El procesador del IOC tiene una conexión externa entre sus pins WAIT y READY. Con este arreglo, el procesador provee un simple estado de espera para cada acceso a memoria o a entrada/salida. No es necesario una señal de reconocimiento porque todos los elementos de memoria y de E/S del IOC están capacitados para aceptar (leer) o proveer (escribir) el byte accedido sin el uso de estados de espera adicionales.

Para efectuar el acceso directo a memoria (DMA), el procesador del IOC recibe la señal HOLD del controlador de DMA (HRQ) y responde generando la señal de reconocimiento de Hold (HLDA). La señal HRQ es llevada externamente a una puerta AND junto con la señal controlada por software DMA ENABLE para permitir la deshabilitación del DMA durante el encendido o reseteado, cuando todos los chips programables están siendo inicializados por el procesador del IOC. Todas las otras veces, la DMA es habilitada.

El uso de la señal HOLD para retrasar el procesamiento durante los ciclos de DMA es una forma de "robo de ciclo". La DMA es usada para el refresco de la RAM y para las transferencias entre la RAM y el controlador de CRT o el controlador de diskette. Las transferencias con el diskette siempre ocurren en respuesta a un comando del IPB/IPC en cuyo caso el robo de ciclo no retrasa el proceso "normal". El refresco del CRT y de la RAM producen un alargamiento de todos los procesos del IOC. La RAM es refrescada a intervalos de 15.5 microseg., y el CRT requiere una transferencia de la DMA cada 8.33 microseg. aproximadamente en las peores condiciones. Las dos operaciones de refresco usan el 38% del tiempo de proceso disponible.

Las transferencias con el diskette a través del DMA se producen cada 32 microseg. cuando son requeridas por un comando del IPB/IPC y usan un 7% adicional del tiempo de proceso. Por lo tanto, en las peores condiciones, todos los otros procesos se realizan dentro del 55% del tiempo restante disponible.

2.4.1.2. LOGICA DEL RESET

La lógica de reset del IOC provee dos tipos de "resets" hardware. El primer tipo es generado cuando el generador de reloj detecta la aplicación de potencia a la placa del IOC. El segundo tipo se produce cuando el interruptor RESET del panel frontal es presionado, causando la generación de la señal MASTER RESET/. Ambos tipos producen la generación de la señal INITLZ/ que es aplicada a muchos elementos de la placa del IOC, incluyendo el procesador del PIO. El reset producido cuando se enciende el sistema de desarrollo generará dos señales adicionales (RESET y RESET/) para resetear todos los elementos incluyendo el controlador de CRT 8275 y el reloj 8253 (que no son reseteados por la señal MASTER RESET/).

La razón por la que el 8275 y el 8253 no son reseteados por la señal MASTER RESET/ es que esta acción podría causar un colapso en el "raster" del CRT (como ocurre cuando se enciende o

se apaga el sistema de desarrollo). El interruptor RESET, de esta forma, desaparece los caracteres de la pantalla sin que desaparezcan las señales de barrido horizontal y vertical (HORIZ DRIVE y VERTICAL DRIVE/).

2.4.1.3. CONTROLADOR DE DMA

El controlador de DMA 8275 es el único chip del IOC, aparte del procesador del IOC, que puede asumir un control total del bus del IOC. Las líneas de direcciones del procesador y las líneas de datos del procesador y del controlador de sistema 8228 del IOC son flotadas mientras el controlador de DMA genera las direcciones de la RAM y las señales de control requeridas para efectuar una transferencia de datos entre la RAM y el dispositivo requerido. Después de tomar el control del bus, un byte de datos simple es transferido durante 4 ciclos de reloj después de lo cual, el control del bus es retornado al procesador, a menos que esté pendiente otra solicitud de DMA.

Durante la operación normal (sin DMA), el controlador de sistema decodifica los comandos que aparecen en las líneas de datos del bus (D0-D7) del procesador, durante el estado T_2 para producir las señales de control de lectura/escritura a memoria (MEMR/ y MEMW/) y las señales de control de lectura/escritura en entrada/salida (IOR/ y IOW/). La señal de control del bus a ser generada es determinada durante el tiempo de validación de estado (STSTB). Si se especifica una operación de lectura, el controlador de sistema espera por el procesador para producir WR/ antes de activar IOW/ o MEMW/. Durante este tiempo, las líneas de direcciones (A0-A15) contienen una dirección de memoria (si MEMR/ o MEMW/ está activa) o una dirección de port (si IOR/ o IOW/ está activa) .

Durante el encendido o el reseteo del sistema, todos los canales del DMA son deshabilitados y consecuentemente inicializados a través de los ports de E/S FO-F8 del procesador del IOC.

En la inicialización se establecen la dirección de comienzo de la RAM y el fin de cuenta (p. ej. el tamaño del bloque) para cada uno de los canales de la DMA que va a acceder a la RAM. Después de la inicialización, se envía un comando de activación de modo al controlador de DMA para establecer los modos de operación de todos los canales, y selectivamente habilitar cada canal. En el IOC, el modo de operación seleccionado usa la "opción de autocarga", en donde la dirección de comienzo y el final de cuenta, suplido al canal 3 de la DMA, son transferidos automáticamente al canal 2 cuando el "fin de cuenta" del canal 2

ha sido decrementado a cero (final de transferencia de bloque) de la DMA). El modo de autocarga es establecido por el 8257 y no puede afectar a, o ser duplicado por, los canales 0 y 1 del DMA.

Cuando los canales son inicializados, los parámetros de los bloques son llevados a los canales 1, 2 y 3 del DMA. El canal 0 no necesita parámetros de los bloques puesto que sólo "roba" ciclos de máquina para efectuar ciclos de refresco de la RAM.

El canal 0 de la DMA efectúa un ciclo de refresco de la RAM cada 15.5 microseg. para lo cual usa el "timer" 8253. La salida del "timer"; DMA REQ0, es la solicitud de mayor prioridad de la DMA. El controlador de la DMA genera entonces la señal de solicitud de "hold" (HRQ) para acceder al bus del IOC. Cuando el procesador del IOC ha completado el actual ciclo de instrucción genera la señal de reconocimiento de "hold" (HLDA), y se desconecta de las líneas de datos (D0-D7) y de direcciones A0-A15). El controlador del sistema, asimismo, acepta HLDA y flota sus líneas de datos (D0-DAT7).

El controlador de la DMA, desde que "siente" HLDA, genera las señales DMA ACK 0/ y MEMR/ para producir un ciclo de refresco de la RAM. Durante este ciclo de refresco, el controlador de DMA es el "master" del bus del IOC, y efectúa un ciclo normal de transferencia de datos. Sin embargo, una vez se ha empezado el ciclo de refresco, la RAM ignora las direcciones de la DMA. El contenido de las líneas de datos también es ignorado ya que AEN (DMA ON) deshabilita la decodificación de las direcciones de los ports, y porque con DMA ACK 0/ no se habilitan otros usuarios de la DMA.

Después de que el controlador de DMA ha sido inicializado para operar con el diskette, se inicializa el controlador de diskette 8271. Este generalmente consiste en especificar exactamente que área del diskette va a ser accedida. Después de que el controlador de diskette ha localizado el área especificada, genera DMA REQ1. Entonces, el controlador de DMA genera HRQ, y se convierte en el master del bus del IOC (cuando el procesador responde con HLDA). El controlador de DMA coloca, entonces, los 8 bits más significativos de la dirección de comienzo en las líneas de datos y genera ADDSTB para que el latch 8212 almacene esta mitad de la dirección de la RAM. A continuación, el controlador de DMA activa DMA ACK 1/ para informar al controlador de diskette que ha sido seleccionado para el ciclo de DMA.

El controlador de DMA genera IOR/ y MEMW/ (para lectura del disco) o IOW/ y MEMR/ (para escritura en el disco) basándose en los dos bits más significativos de la palabra del contador de fin previamente recibida. La RAM, entonces, accede a la dirección seleccionada, y se produce una transferencia de un byte entre la RAM y el controlador de diskette.

El controlador de DMA incrementa entonces la dirección del canal 1 de la DMA y decrementa el contador de final del canal 1 para prepararse para el próximo DMA REQ 1.

Cuando el contador de final llega a cero, el registro de estado del canal 1 del controlador de DMA informa al procesador del IOC que el bloque completo ha sido transferido a o desde el diskette.

Los canales 2 y 3 de la DMA operan juntos para refrescar la pantalla CRT a una velocidad de 50 o 60 hertzios (dependiendo de la frecuencia de la red). El canal 2 es responsable directamente de la transferencia de caracteres de la RAM al controlador de CRT. El canal 3 se usa para almacenar los nuevos parámetros del bloque (dirección de comienzo, y contador de fin) para el canal 2. Los parámetros del bloque del canal 3 son cargados automáticamente en el canal 2 cuando el contador de fin para el canal 2 decrementa hasta cero.

La principal ventaja de la carga automática es que el "scrolling" de la pantalla puede ser efectuado sin volver a escribir toda el área de RAM que contiene los datos del CRT. (El "scrolling" hace que todas las líneas de la pantalla se muevan hacia arriba cuando se escribe una nueva línea al final de ésta). Con el método de carga automática, una dirección dada de la RAM permanece asociada con un carácter específico de la pantalla, aún cuando el carácter se mueva como resultado de un scrolling.

El controlador de DMA utiliza 2 bloques de direcciones de la RAM. El primer bloque define el área de RAM desde la línea más vieja del texto (la más alta de la pantalla) hasta el final del buffer de la RAM usado para la pantalla. El segundo bloque define el área de RAM desde el principio del buffer de la RAM hasta la última línea del texto.

La posición relativa de los datos de la pantalla en el buffer de la RAM puede ser la siguiente:

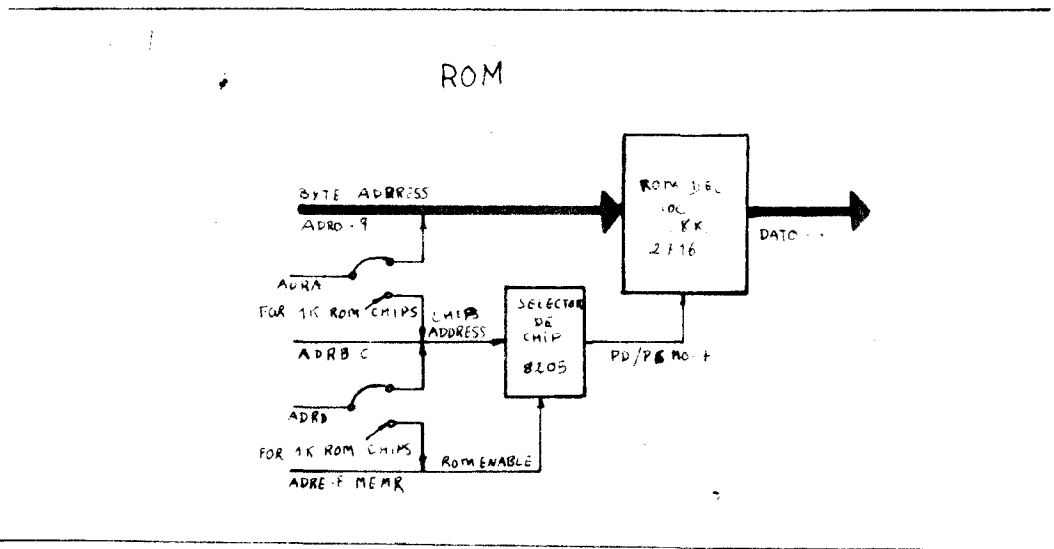
Cuando se añaden nuevas líneas al texto, la dirección de comienzo del primer bloque y la dirección de final del segundo bloque son cambiadas para reposicionar los datos en la pantalla. Cuando la pantalla está completamente llena, cada línea nueva se sobrescribe sobre la última línea.

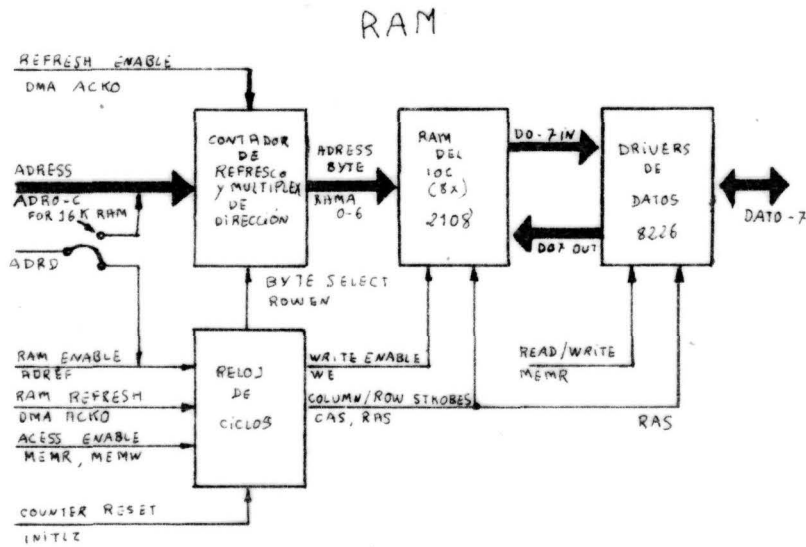
Todas las transferencias de datos de la DMA al controlador de CRT se hacen a través del canal 2 de la DMA. El canal 3 de la DMA no recibe solicitud de DMA ni usa solicitudes de hold o de reconocimiento de DMA.

2.4.2. SUBSISTEMA DE MEMORIA DEL IOC

El subsistema de memoria del IOC es bastante simple debido al uso de direcciones no solapadas para la RAM y la ROM. El circuito de RAM está formado por ocho chips de RAM 2108/2109 proveyendo 8k de RAM, dos 8226s que proveen buffer para una línea de datos, un multiplexor de direcciones y contador de refresco 3242 y un timer TTL que genera las señales de control de la RAM. La ROM está formada por 4 chips de 2k bytes de ROM y un decodificador binario 8205 que selecciona los chips de ROM.

La RAM y la ROM pueden ser ampliadas a 16k bytes mediante la modificación de la placa del IOC. La ampliación de la RAM puede efectuarse reemplazando los chips 2108/2109, y cambiando los "jumpers" (puentes) que cambian la función del bit de dirección ADDR. Los circuitos de la ROM están puenteados normalmente para chips de 2k de ROM 2316/2716, y la expansión puede efectuarse insertando 4 chips adicionales de 2k de ROM cuyos sócalos ya están previstos en la placa.





SUBSISTEMA DE MEMORIA DEL IOC

2.4.2.1. ROM DEL IOC

Con la serie normal de 4 chips de 2k de ROM, las direcciones asignadas a la ROM son desde 0000 hasta 1FFF. En este caso, los 5 bits más significativos de las líneas de direcciones (ADRB hasta ADRF) son decodificados para producir la señal MEMR/ que selecciona un chip de la ROM, y los restantes bits (ADRO-ADRA) son aplicados directamente a los chips de ROM para seleccionar el byte especificado.

2.4.2.2. RAM DEL IOC

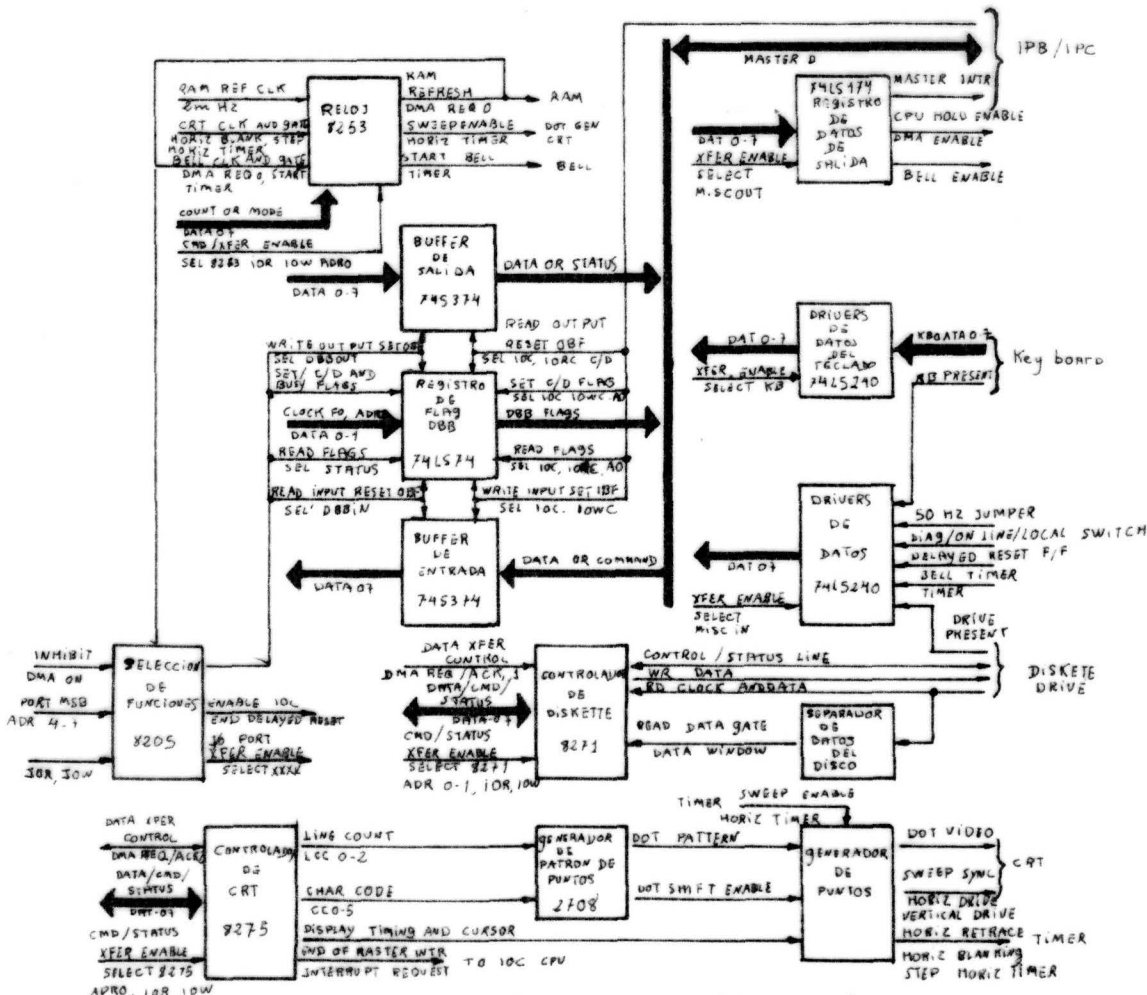
Los circuitos de la RAM del IOC han sido diseñados para la simplificación del trabajo, mediante el empleo del modo "refresco de ciclo de lectura" en el cual el tiempo del ciclo de refresco es igual al de un ciclo de lectura normal. Cada 15.5 microseg. se produce un ciclo de refresco. Durante cada ciclo, son refrescadas las 128 posiciones de RAM de una fila. Los 64 ciclos nece-

sarios para refrescar los 8k de RAM se completan en menos de 1 mseg.

Los ciclos de refresco son iniciados cuando el timer del subsistema de E/S genera DMA REQ 0. Cuando el controlador de DMA genera DMA ACK 0/, la RAM del IOC empieza un ciclo de refresco. Durante el ciclo de refresco, el bus del IOC está en estado "desocupado" porque el controlador de DMA inhibe al procesador del IOC (HOLD) y a los ports de E/S (DMA ON). Entre ciclos de refresco, la RAM del IOC responde a los comandos del bus de control de memoria (MEMR/ o MEMW/) sólo si la dirección está en el rango entre 4000 hasta 5FFF (ADRE verdadera y ADDR y ADRF falsas).

2.4.3. SUBSISTEMA DE E/S DEL IOC

El subsistema de E/S del IOC establece vias de comunicación entre el procesador del IOC y los circuitos externos al IOC. Hay 4 vias primarias : Una via comprende el buffer bidireccional DBB. La segunda está formada por receptores de línea TTL para las entradas desde el procesador del teclado 8041. La tercera es una via bidireccional con el diskette integral. La cuarta es con el CRT.



SUBSISTEMA DE E/S DEL IOC

2.4.3.1. DECODIFICACION DE LOS PORTS DEL IOC

El decodificador lógico de los ports del IOC decodifica los 4 bits más significativos del bus de direcciones para controlar el hardware del IOC. En la siguiente tabla se muestran las salidas del decodificador de ports y las direcciones de los ports asociados. En algunos casos (ports 1x-5x), las direcciones de los ports son usadas para activar o resetear los flip-flop o para activar el timer de alarma. En estos casos, las líneas de datos del bus del IOC no se usan, y los bits menos significativos sólo se usan para diferenciar entre la activación y reseteo de los flip-flops (solamente el port 2x). En todos los otros casos, la dirección del port se usa para habilitar la lectura o escritura de un registro.

El decodificador lógico de ports del IOC está formado por 2 decodificadores binarios 8205 los cuales son deshabilitados durante una transferencia de DMA (DMA ON). Uno de los decodificadores (A60) es habilitado sólo durante las operaciones de escritura de E/S (IOW/), mientras que el otro (A59) es habilitado durante las operaciones de lectura (IOR/) y escritura (IOW/). Esta artimaña inhibe el uso de comandos de lectura en puertos de E/S que están reservados exclusivamente para la escritura de datos o control de las funciones del hardware. La habilitación de los decodificadores binarios es controlada por el bit más significativo del bus de direcciones del port (ADR7). Este bit habilita el chip decodificador de lectura/escritura cuando es "verdadero", y el de sólo escritura cuando es "falso".

I/O Port address	Mnemonic	Function
0x	SELECT DBBOUT/	Write to DBB output buffer, set/reset OBF flag
1x		Not used
2x	CLOCK FO/	Set/reset IOC busy flag
3x	SET F1/	Set Command/Data flag (status to IPB/IPC)
4x	RESET F1/	Reset Command/Data flag (data to IPB/IPC)
5x	START TIMER/	Enable 100ms timer (bell)
6x	SELECT MISCOUT/	Write to miscellaneous output register
7x		Not Used
8x	SELECT MISCIN/	Read miscellaneous data
9x	SELECT KB/	Read keyboard data
Ax	SELECT STATUS/	Read DBB status (flags)
Bx	SELECT DBBIN/	Read DBB input buffer, reset IBF flag
Cx	SELECT 8271/	Read/write diskette controller
Dx	SELECT 8275/	Read/write CRT controller
Ex	SELECT 8253/	Read/write interval timer
Fx	SELECT 8257/	Read/write DMA controller

2.4.3.2. BUFFER DE BYTE DE DATOS (DBB)

El DBB está formado por dos registros de 8 bits (el buffer de entrada y el de salida), 4 flip-flops (que indican el estado de la transferencia entre el IOC y el IPB/IPC) y 2 drivers de línea de 4 bits.

El procesador del IOC examina los flags (antes de usar el DBB) para detectar la presencia de un nuevo comando y, si es necesario, testear los flags para detectar la presencia de datos en el buffer de entrada. Por otro lado el IPB/IPC examina los flags y aguarda la indicación de que el comando ha sido totalmente procesado.

Dos de los flags, el de buffer de salida lleno (DBF) y el de buffer de entrada lleno (IBF) indican la presencia de datos válidos en el DBB. El flag IBF es activado por el procesador maestro del IPB/IPC cuando envía un dato o un comando al DBB (SEL IOC/, IOWC/). Este flag es reseteado por el IOC cuando el byte es accedido a través de la dirección de port Bx (SEL DBBIN/).

El flag DBF es activado por el IOC cuando un byte de datos o estados es llevado al buffer de salida a través de la dirección de port 00 (SEL DBBOUT/). El flag DBF es reseteado por el IPB/IPC cuando el byte es accedido (SEL IOC/, IORC/, y AO todos bajos).

El tercer flag, el flag de dato/comando (F1) es activado o reseteado por los dos procesadores. Para entradas desde el IPB/IPC (SEL IOC/, IOWC), este flag es controlado por el estado de AO. Para salidas, el software del procesador del IOC usa las direcciones de los ports 3x y 4x para generar SETF1/ y RESET F1/ para activar o resetear respectivamente el flag C/D.

El cuarto flag asociado con el DBB es el flag de "ocupado" del IOC (FO). Este flag es activado por el IOC cuando se acepta un comando, y reseteado por el IOC cuando el comando está totalmente procesado. Las direcciones 00 y 01 de los ports activan y resetean el flag respectivamente.

2.4.3.3. CIRCUITOS DE ENTRADA DEL TECLADO

Los circuitos de entrada del teclado están formados por los drivers de línea que controlan la entrada de datos desde el procesador del teclado 8041. La habilitación de los drivers de línea se producen cuando el procesador del IOC genera las direcciones de los ports de E/S 90 y 92. Los bits más significativos de la dirección de los ports son decodificados para generar SELECT KB/. Esta señal habilita los drivers de línea y selecciona el DBB del procesador del teclado para una operación

de lectura de datos (READ RB/). Uno de los bits menos significativos (ADRL) es enviado directamente al procesador del teclado en donde selecciona un byte de carácter (cuando es falso) o un byte de estados (cuando es verdadero).

2.4.3.4. CIRCUITOS DE CONTROL DEL CRT

Los circuitos de control del CRT están formados por un controlador de CRT 8275, una ROM 2708/2308 que es usada para proveer el patrón de puntos de los caracteres visualizados, y una variedad de elementos TTL.

Los circuitos de barrido están diseñados para la generación de un "raster", en donde se producen 260 o más barridos horizontales por cada barrido vertical.

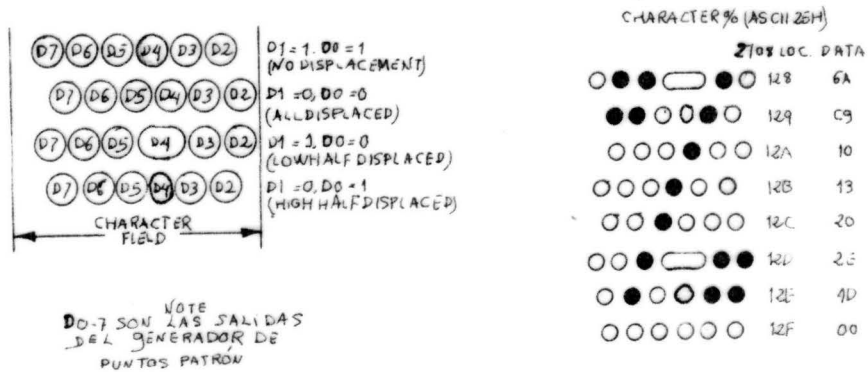
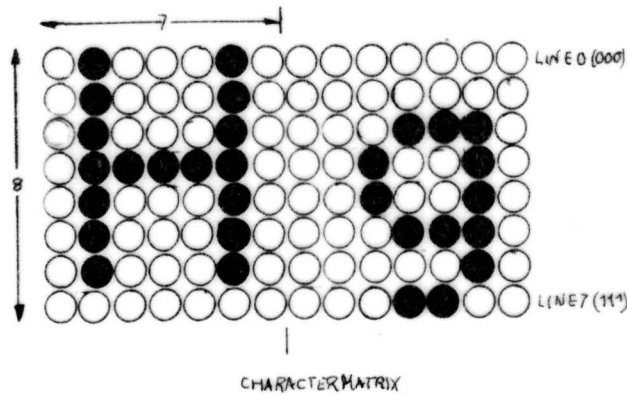
El método standard para la visualización de caracteres en la pantalla es producir puntos que intensifiquen los segmentos de los caracteres que coinciden con los barridos horizontales específicos. Cada carácter es visualizado dentro de una matriz de 7 por 8 puntos, y un carácter simple está formado normalmente por 5 puntos de ancho y 7 de alto. La pantalla completa está formada por 80 caracteres de ancho por 25 de alto.

Los caracteres que van a ser visualizados son entrados a través del DMA cuando el controlador de CRT genera DMA REQ 2/. La secuencia es tal que los 80 caracteres que comprenden una fila son entrados y procesados como una entidad. Durante el procesamiento de una fila, la entrada a la DMA continúa, ya que el controlador de CRT tiene almacenamiento para dos filas completas de caracteres.

Cuando está procesando una fila, el controlador de CRT genera los contadores de línea (LCO-LC2) y los códigos del carácter (CCO-CC6). Los códigos del carácter para cada carácter dentro de una fila son sacados, en secuencia, para cada una de las 8 líneas de la fila.

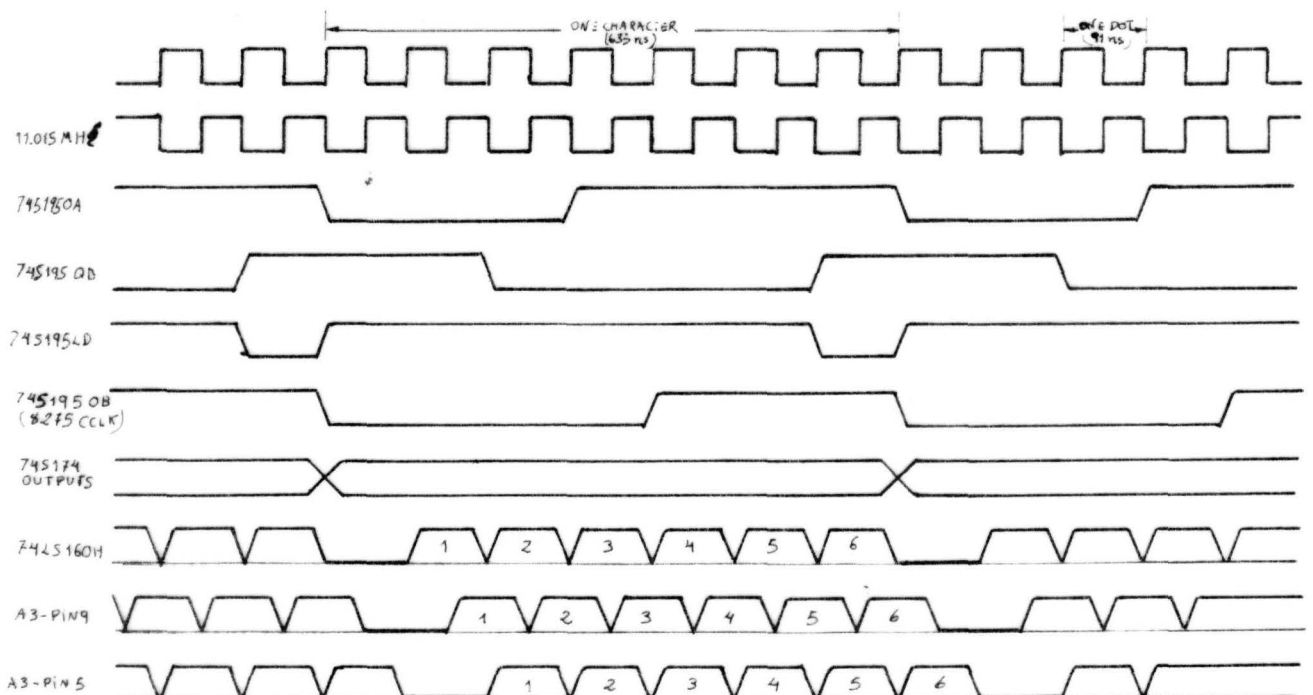
La ROM usa el contador de línea y el código del carácter como una dirección para acceder a un patrón de puntos específico.

Para cada línea de un carácter, la salida de la ROM son 6 bits para los puntos más 2 bits que controlan la inserción de un pequeño (medio punto) retraso en la visualización de otros 3 medios bits del patrón de puntos. Este desplazamiento selectivo de los puntos permite la readaptación del carácter.



MATRIZ DE VISUALIZACION DE CARACTERES

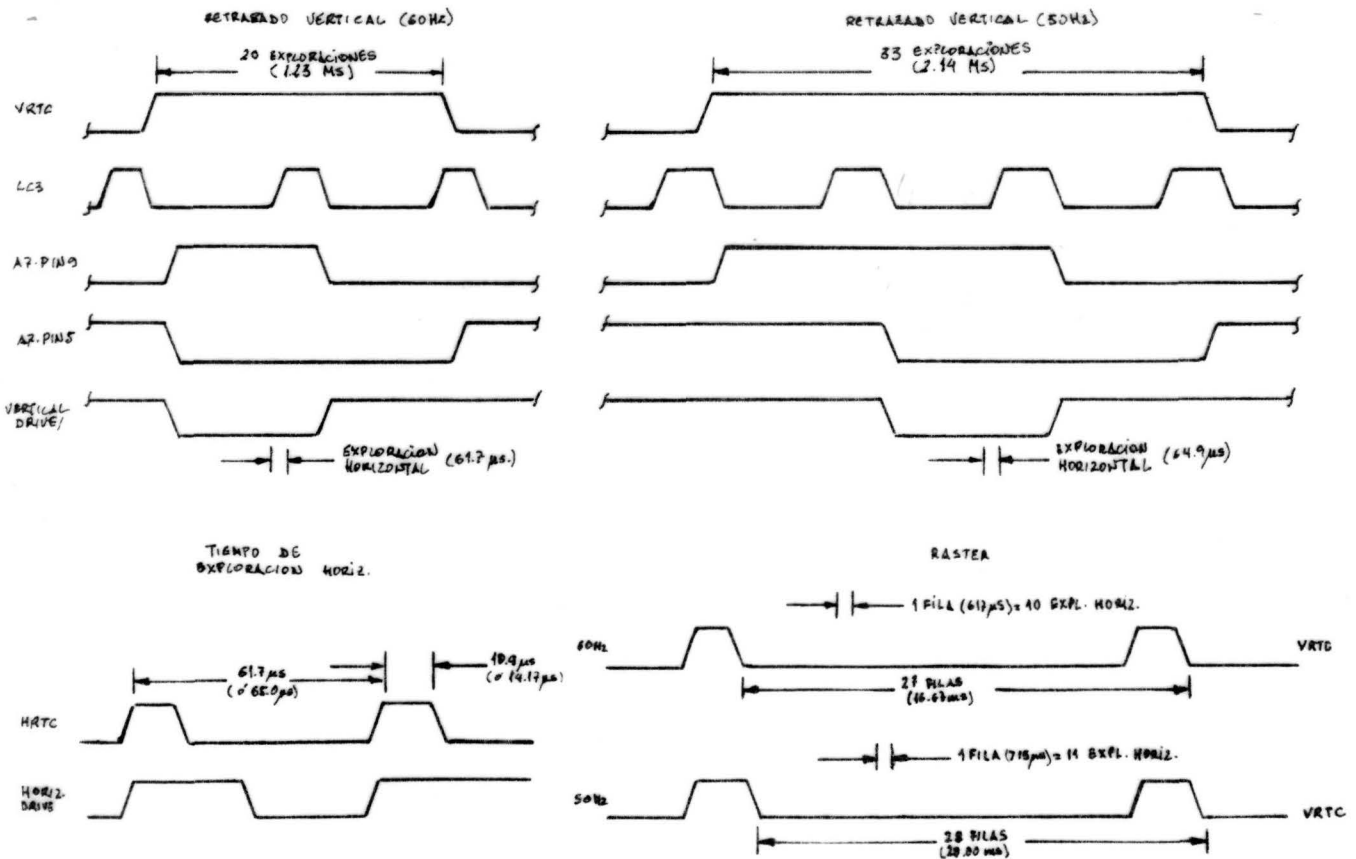
Los 6 bits del patrón de puntos y los 2 bits de desplazamiento salen de la ROM como un byte completo. El generador de puntos del CRT señala la cadena de patrones de puntos e inserta los retrasos especificados por los bits de desplazamiento. El flujo serie de puntos resultante está sincronizado al barrido horizontal y es "iluminado" durante el tiempo de retrazo y durante el barrido que separa las filas de caracteres.



TEMPORIZACION DEL PATRON DE PUNTOS

El flujo serie de puntos es aplicado a dos flip-flops. Uno de estos flip-flops usa el reloj de pulsos de 11.016 MHz. y suministra los puntos normales (no desplazados). El segundo flip-flop usa el complemento del reloj de 11.016 MHz. y suministra los puntos desplazados. Estos dos flip-flops se usan para producir todos los patrones de puntos. Sus salidas proveen la combinación deseada de puntos normales y desplazados para el multiplexor (A5) 74LS151.

El barrido vertical de la pantalla está sincronizado a la frecuencia de la red (50 Hz. o 60 Hz.) para prevenir el parpadeo que puede producirse en la pantalla por estar desfasada frente a la luz ambiente. Sin embargo, el tiempo de visualización de cada carácter, el número de caracteres por fila, y el número de filas de caracteres permanece independiente de la velocidad del barrido vertical. Para compensar la operación a 50 Hz., se incrementa el número de barridos horizontales y se extiende el intervalo de retrazado, horizontal y vertical. Estas últimas variables son establecidas por los valores enviados al controlador de CRT durante la inicialización. Los circuitos asociados a la temporización del hardware controlan el disparo del barrido durante los intervalos de retrazado. Estos circuitos incluyen el timer de intervalos 8253 y el registro 74S195 del generador de puntos de la pantalla. Las variables controladas tanto por software como por hardware son determinadas por la presencia o ausencia del "Jumper" (puente) de 50 Hz. (W8).



TIEMPOS DE BARRIDO DEL CRT

Durante el encendido o el reseteado, el software del IOC lee el byte de entrada que contiene el bit del jumper de 50 Hz. Si el jumper está quitado (60 Hz.), los parámetros enviados por el software del IOC al controlador de CRT incluyen un contador de fila de retrazado vertical de 2, un valor de líneas por fila de 10, y un contador de retrazado horizontal de 20. Si el jumper es instalado (50 Hz.), el contador de fila de retrazado vertical es de 3, el valor de líneas por fila es de 11, y el contador de retrazado horizontal es 26. Estas dos combinaciones de los parámetros de inicialización proveen la compensación requerida para la operación de 50 Hz. y de 60 Hz.

2.4.3.5. CIRCUITOS DE CONTROL DEL DISKETTE

Los circuitos de control del diskette integral están formados por un controlador de diskette 8271 más los elementos TTL que forman el separador de datos del diskette. El controlador de diskette recibe comandos a través de los ports de E/S del procesador del IOC, transfiere datos de lectura/escritura a través del canal 1 de la DMA, y establece el interface con el "driver del diskette integral" del sistema de desarrollo.

El controlador de diskette permanece en estado inactivo hasta que recibe un comando del IOC a través del port de E/S. Se usan tres ports de E/S, cuyas direcciones son: port C0 (para comandos y estados), port C1 (para parámetros y resultados), y port C2 (para resetear el controlador de diskette).

Los bits más significativos de la dirección del port (ADR4-ADR7) son decodificados externamente por el selector de funciones (A59) para seleccionar el 8271. La salida del selector de funciones, combinada con IOW/, hace que el controlador de diskette acepte los comandos o parámetros que aparecen en las líneas de datos (DAT0-DAT7). El bit menos significativo de la dirección del port (ADRO) se usa para distinguir entre comandos y parámetros. Cuando el 8271 es seleccionado e IOR/ es activa, el procesador del IOC está solicitando un byte de estado o de resultado, dependiendo de el estado de ADRO. El segundo bit menos significativo de la dirección del port (ADRI) sólo es activo cuando el procesador del IOC usa el port C2 para resetear el controlador de diskette.

El byte de comando define la operación a ser desarrollada mientras que los bytes de parámetros siguientes suministran la información requerida para efectuar la operación (para una operación de lectura/escritura del diskette, los bytes de parámetros proveen las direcciones de la pista y del sector).

Cuando un comando requiere el reposicionamiento de la cabeza de lectura/escritura y/o la transferencia de datos a o desde el diskette, el controlador de diskette entra en la fase de ejecución después de recibir el último byte de parámetros.

Cuando el controlador entra en la fase de ejecución, primero chequea el drive para ver si está listo, y entonces, si es necesario, pasa la cabeza a la pista direccionada.

Si se especifica una operación de lectura, el controlador de diskette permanece en el modo lectura. Después de localizada la marca de dirección de los datos del sector direccionado, el controlador empieza el ensamblado de los bit de datos serie leídos del disco en bytes de 8 bits. Cada vez que un byte es ensamblado, el controlador de diskette genera una solicitud de DMA (DMA REQ1) al controlador de DMA. El controlador de DMA, en respuesta, genera DMA ACK1/ al controlador de diskette y escribe entonces el byte ensamblado en la posición de la RAM del IOC direccionada. Cuando el sector completo (128 bytes) ha sido transferido, el procesador del IPB/IPC lee el registro de resultado del controlador de diskette para determinar si la operación ha sido terminada con éxito y entonces lee el bloque de datos de la RAM del IOC a través del DBB.

Si se especifica una operación de escritura, el controlador de diskette conmuta del modo lectura al modo escritura y genera una solicitud de DMA para acceder al primer byte de datos de la RAM del IOC que va a ser escrito. (El procesador del IPB/IPC debe escribir primero el bloque de datos en la RAM del IOC antes de iniciar la operación de escritura del diskette). El controlador de diskette vuelve a escribir los bytes de sincronismo y la marca de dirección de los datos, y a continuación comienza la escritura de los bytes de datos accedidos como un flujo serie de pulsos de reloj y datos. Cuando el primer byte está siendo escrito, el siguiente byte se está leyendo ya desde la RAM. Cuando se ha escrito el bloque de datos completo (1024 bits), el controlador añade un carácter CRC de 2 bytes al final del bloque de datos.

La técnica empleada en la grabación del diskette es la de frecuencia doble (FM) y está basada en una célula de bit de 4 microseg. Cada célula de bit empieza con un pulso de reloj de 250 ns. (los pulsos de reloj proveen la sincronización durante la operación de lectura del diskette). Cuando se va a escribir un "1" lógico, se genera un segundo pulso de 250 ns aproximadamente 2 microseg. después del principio de la célula del bit (si se va a escribir un "0" lógico, el segundo pulso es omitido).

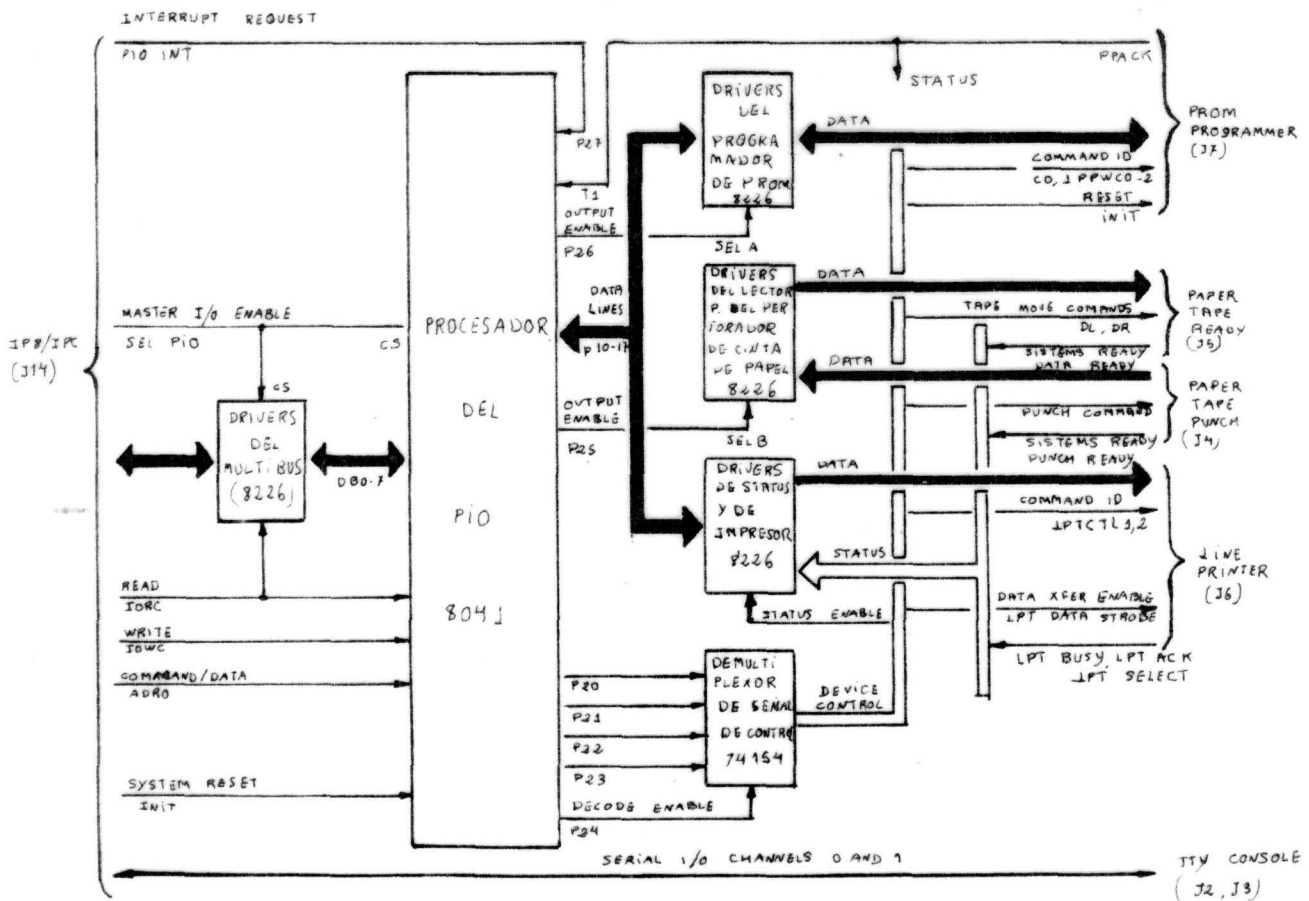
Los pulsos de 250ns., la velocidad de los bits (4 microseg.), y la velocidad de transferencia de la DMA (32 microseg.) se derivan todas del reloj de 4 MHz. usado por el controlador de diskette.

2.5. SUBSISTEMA DE ENTRADA/SALIDA PARALELO (PIO)

El subsistema de E/S paralelo (PIO) está formado por todos los elementos hardware necesarios para establecer la comunicación entre el IPB/IPC y los 4 periféricos satandard del sistema. Estos periféricos son: lectura de cinta de papel perforado, perforador de cinta de papel, impresora de línea, y programador universal de PROMs de INTEL.

El subsistema PIO está implementado en la misma placa del IOC.

Los elementos que forman el PIO son: un microprocesador 8041, ocho drivers de línea bidireccionales 8226, y un demultiplexor TTL 74154. El procesador del PIO contiene la "inteligencia" necesaria para el control y dirección de los cuatro dispositivos periféricos; la función de los demás elementos solamente es de aislar al procesador o demultiplexor sus salidas. La misión del procesador del IOC es convertir las órdenes que le llegan del IPB/IPC en secuencias detalladas de señales requeridas por los dispositivos externos.



DIAGRAMAS DE BLOQUES DEL PIO

2.5.1. PROCESADOR DEL PIO

El microprocesador 8041 es un procesador de propósito general que requiere un mínimo de circuitos externos. En realidad puede ser considerado como un interface universal de periféricos debido a su capacidad de actuar como interface para cualquier dispositivo periférico a la vez que sirve de esclavo para un microprocesador maestro 8080/8085. El 8041 tiene un timer interno, 1k byte de ROM y 64 bytes de RAM.

Como el 8041 emplea memoria interna no tiene pins reservados para direccionamiento. En su lugar existen tres canales de E/S paralelo de 1 byte. El primero de estos canales (D0-D7) es muy similar al bus bidireccional del 8080. El procesador del PIO usa este bus para la comunicación con el procesador maestro del IPB/IPC. Los otros dos canales son ports de E/S de 8 líneas cada uno. El primer port de E/S (pins P10-P17) es usado para transferir datos a o desde los 4 dispositivos externos. El segundo port de E/S (pins P20-P27) se usa para proveer las señales que controlan la información que se transfiere por el primer port. Las señales de los 2 ports son conocidas colectivamente como el bus del PIO.

Cinco de los ocho pins del segundo port controlan el demultiplexor 74154 para generar 13 señales de control, que junto con los tres pins restantes del port forman un bus de 16 señales de control.

De estas 16 señales, tres son usadas dentro del PIO para controlar la ruta de los datos; una es una solicitud de interrupción al IPB/IPC y las 12 restantes son usadas directamente por los dispositivos externos.

2.5.2. INTERFACE PIO-IPB/IPC

Todas las comunicaciones entre el PIO y el IPB/IPC son iniciadas por el procesador maestro del IPB/IPC, en donde el procesador del PIO actúa como esclavo.

El protocolo de la comunicación está basado según los requerimientos del DBB, el cual es accesible del 8041. El DBB provee almacenamiento para los datos y para 4 flags que son accesibles independientemente al PIO y al procesador maestro del IPB/IPC. Estos flags son el IBF (buffer de entrada lleno), el OBF (buffer de salida lleno), el FO (8041 ocupado) y el C/D (que identifica los datos de entrada al PIO como comandos o como byte de datos, y los datos de salida del PIO como estados o byte de datos).

El procesador maestro del IPB/IPC lee los flag para determinar cuando el PIO ha aceptado un byte de entrada, cuando se está retornando un dato o un estado, y cuando se ha completado la operación del PIO. El procesador del PIO lee los flags para determinar cuando se ha recibido un comando o un dato de entrada, y cuando ha sido aceptado por el IPB/IPC un byte de estados o de datos. Ambos procesadores pueden activar y resetear los flags. Los drivers de los ports de E/S del IPB/IPC y el DBB son habilitados por SEL PIO/. Esta señal deriva del bus de direcciones de los ports.

La dirección de la transferencia es transmitida al DBB por MASTER RD/ (salida del PIO) y por MASTER WR/ (entrada al PIO), y el tipo del byte de entrada al PIO (comando o dato) es definido por el estado de MASTER AO. Los drivers de los ports de E/S usan sólo MASTER RD/ para determinar la dirección de la transferencia. El PIO no puede habilitar estos drivers, y por tanto, no puede iniciar una transferencia.

Las otras señales usadas por el interface del IPB/IPC-PIO son PARALLEL INT (interrupción del PIO) y INITLZ/ (reset del sistema). La interrupción del PIO se deriva del bit más significativo del bus del PIO (P27). INITLZ/ se produce cuando se enciende el sistema o se pulsa el interruptor externo RESET, produciendo la reinicialización del software y cancelando las últimas operaciones a realizar.

2.5.3. INTERFACE CON LOS DISPOSITIVOS PERIFERICOS

Como ya hemos dicho el PIO usa dos ports que actúan como interface con los periféricos. El primero de estos ports se usa para la transferencia de datos bidireccional, y el segundo para controlar esta transferencia. El port 1 (P10-P17) utiliza tres pares de drivers bidireccionales 8226 que actúan como interface a los periféricos. Un par de drivers (A78-A79) se usan para dos modos de intercambios con el programador de PROM. Un segundo par (A73-A74) suministra una salida para datos con el perforador de cintas de papel, y una entrada de datos del lector de cintas de papel. El tercer par (A75-A76) provee una salida de datos hacia la impresora o retorna un byte de estados de todos los periféricos asociados al PIO.

Los 4 bits menos significativos del port dos (P20-P23) son decodificados por el decodificador 74154 (A77) para generar una de las trece señales de control. El bit P24 del port 2 es usado para habilitar el decodificador, y los bits P25 y P26 para controlar la dirección de los pares de drivers del lector/perforador y del programador de PROM, respectivamente.

Para asegurar que las líneas de la señal de control decodificada (P20-P23) son estables cuando el decodificador es habilitado, el 8041 ejecuta una secuencia de 2 instrucciones. La 1ª instrucción define la señal de control y la dirección del driver, pero no habilita el decodificador (P24 permanece alta). La 2ª instrucción, entonces, activa P24 sin alterar el estado de los otros bits del port 2 para generar la señal de control requerida. En la siguiente tabla se define la decodificación de la señal de control:

Port Bits (P23) (P22) (P21) (P20)				Control Signal Active	Control Signal Function
0	0	0	0	PPWC0/	PROM programmer write strobe 0
0	0	0	1	PPWC2/	PROM programmer write strobe 2
0	0	1	0	PPWC1/	PROM programmer write strobe 1
0	0	1	1	PPRC0/	PROM programmer read request 0 (data)
0	1	0	0	PPRC1/	PROM programmer read request 1 (status)
0	1	0	1	INIT/	PROM programmer reset
0	1	1	0	PUNCH COMMAND/	Paper tape punch operation
0	1	1	1	DL/	Paper tape reader left tape advance
1	0	0	0	DR/	Paper tape reader right tape advance
1	0	0	1	LPT DATA STROBE/	Line printer data strobe
1	0	1	0	LPT CTL 1/	Line printer control 1
1	0	1	1	LPT CTL 2/	Line printer control 2
1	1	0	0	STATUS ENABLE/	Device status input enable
1	1	0	1	none	Unused
1	1	1	0	none	Unused
1	1	1	1	none	Unused

DECODIFICACION DE LAS DIRECCIONES PARA LAS SEÑALES DE CONTROL DE LOS DISPOSITIVOS PERIFERICOS

Para sacar un byte de datos, el 8041 desarrolla la siguiente secuencia:

1. El byte de datos es colocado en las líneas de datos del port 1 (P10-P17)
2. El primer byte de control es sacado por el port 2. Este byte define la señal de control del periférico a ser generada (P20-P23) y asegura que los pares de drivers estén activados para la salida de datos. (P25 y P26 en estado alto).
3. El siguiente byte de control habilita el decodificador (P24 en estado bajo) para activar la señal de control decodificada. Hay que hacer notar que si bien el byte de datos de salida está disponible

en los 3 pares de drivers, solamente será aceptado por el periférico que reciba la señal de control.

4. El siguiente byte de control es sacado del port 2 para deshabilitar el decodificador (P24 alta) con el fin de retornar la señal de control del periférico seleccionado a un nivel inactivo.

La operación del 8041 durante la entrada de datos depende del periférico del que provenga. Cuando el byte de datos provenga del programador de PROMs, el 8041 define la señal de control (PPRC0/ o PPRC1/) y habilita el par de drivers, que hacen de interfaz con el programador, para entrada de datos (P26 baja) con el primer byte de control, y entonces genera la señal de control. El 8041 acepta el byte de entrada por el port 1, y saca a continuación un byte de control para terminar la señal de control.

Cuando el byte de datos proviene del lector de cinta de papel perforada, el 8041 saca un byte de control para habilitar el par de drivers del lector/perforador para entrada de datos (P25 baja), y entonces lee el byte de datos por el port 1. (sólo se saca un byte de control ya que el lector no necesita una señal de control de lectura).

Para leer en el byte de estado de los periféricos, el 8041 define la señal de control (STATUS ENABLE/) con el primer byte de control y genera la señal con el segundo byte de control. STATUS ENABLE/ activa el par de drivers de la impresora de líneas y de estados para entrada de datos y el 8041 acepta el byte de estados por el port 1. A continuación un byte de control termina la señal de control.

El byte de estado de los periféricos refleja el estado general de todos los periféricos asociados al PIO. Este byte es usado por el 8041 para informar al procesador maestro del estado de los periféricos mediante comandos. La siguiente tabla define los bits individuales del byte de estados:

CAPITULO II

REDES DE ORDENADORES. ESTUDIO DE LAS
COMUNICACIONES SERIES

1. EVOLUCION DE LAS REDES DE ORDENADORES

Es dramático ver como cada generación de ordenadores desbancó a la anterior, y como un pequeño chip manufacturado hoy tiene la misma potencia y capacidad que las grandes máquinas que hace relativamente pocos años que ocupaban habitaciones y hasta edificios enteros. Los computadores son cada vez, y más rápidamente, más pequeños y económicos, más potentes y funcionales, haciéndose más accesibles al gran público.

Hasta hace pocos años los computadores eran para la gran mayoría de las personas máquinas grandes, caras y complicadas que requerían habitaciones aisladas, y que sólo podían ser operadas por expertos.

En la década de los 50, los usuarios de estos computadores debían llevar sus trabajos a las máquinas para ser ejecutados por lotes (mientras se estaba ejecutando un programa los usuarios no podían interactuar directamente con él). Por tanto, no había una comunicación directa entre el usuario y el ordenador.

El primer acceso interactivo a los computadores llegó en la década de los 60 con los sistemas de tiempo compartido. Este fue el resultado de la combinación de los teletipos y las tecnologías de transmisión de datos, gracias a lo cual se llegó al terminal interactivo. Conectando estos terminales al sistema central con líneas de baja velocidad, los usuarios estaban capacitados para interactuar directamente con el computador y podían compartir sus recursos de información y procesos. Aquí nacieron las necesidades de las redes.

Durante la década de los 70, debido a que los computadores se adaptaron rápidamente a las aplicaciones específicas que demandaban los usuarios (fábricas, bancos, oficinas, etc.), el uso de minicomputadores versátiles y relativamente económicos se extendió rápidamente. Estos computadores permitían ficheros, programas, dispositivos periféricos y de almacenamiento compartidos.

Pero estos usuarios necesitaban intercambiar datos entre departamentos de una misma fábrica (comunicaciones sobre pequeñas distancias), o entre fábricas de una misma compañía (comunicaciones a larga distancia), por ejemplo. De aquí nació una nueva y gran necesidad para las comunicaciones (esta vez no ya entre el usuario y el ordenador sino entre ordenadores). Estas comunicaciones requerían mayor velocidad y capacidad que con los antiguos terminales.

Las redes se diseñaron de forma que los minicomputadores se colocaban en donde se hacía necesario su trabajo, y cada procesador servía a un objetivo claro y definido. Los procesadores vecinos se comunicaban a través de la red. La red podía ser expandida y reconfigurada, fácil y rápidamente, según las necesidades. La naturaleza modular de la red era tal que un fallo en una parte de la red tenía un efecto limitado, o ninguno, en la operación del resto.

En la década de los 80 las ventajas de los sistemas distribuidos fueron reconocidos por fábricas, laboratorios y oficinas. Es normal ahora que los sistemas sean localizados en su lugar de aplicación, para procesamiento de datos, dirección y utilización de bases de datos, control de procesos, procesamiento de palabras, correo electrónico, etc., y todos estos sistemas distribuidos comunicados sobre redes de pequeñas y grandes distancias basadas primariamente en las tecnologías de las comunicaciones sobre largas distancias.

1.1. SISTEMAS DISTRIBUIDOS. CLASIFICACION

Después de hacer una breve historia de la evolución de los sistemas informáticos, y por tanto, de las redes que los interconectan, vamos a hacer una pequeña introducción a los sistemas distribuidos.

El término de sistema distribuido se ha empleado para denominar indistintamente a diferentes clases de sistemas informáticos, en los que la capacidad de tratamiento de la información está repartida en el espacio (sus elementos están alejados unos de otros), y que han aparecido como una alternativa a aquellos otros sistemas en los que la potencia de tratamiento de la información se encuentra concentrada en un único elemento, un único computador.

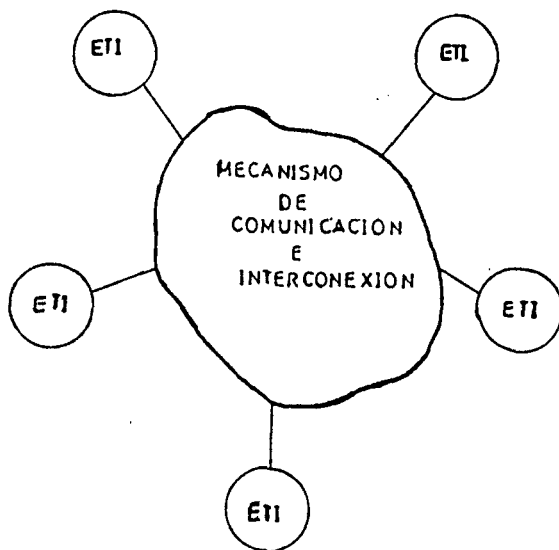


Fig. -2.1-

En la figura 2.1. se representa un esquema general de un sistema distribuido en el que aparece un conjunto de elementos de tratamiento de la información (ETI) interconectados mediante lo que llamamos "Mecanismo de Comunicación e Interconexión" (MCI).

En un intento de hacer una clasificación de los distintos tipos de soluciones que pueden encontrarse bajo esta denominación (ETI-MCI), en la figura 2.2 se ha representado sobre una escala de distancias entre los elementos de tratamiento de la información, el área geográfica a que aproximadamente corresponden, y algunas de las denominaciones utilizadas para identificar las soluciones desarrolladas.

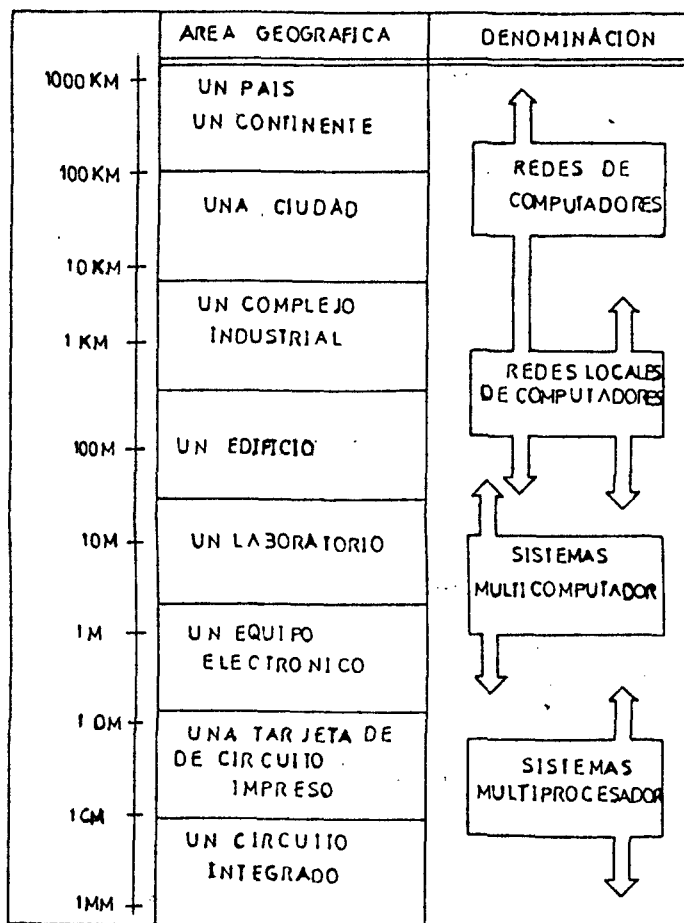


Fig. -2.2-

Por lo que se refiere a las denominaciones utilizadas, nos hemos limitado únicamente a 4 tipos de sistemas:

- a) Redes de computadores
- b) Redes locales de computadores
- c) Sistemas multicomputadores
- d) Sistemas multiprocesadores

1.1.1. REDES DE COMPUTADORES

Las redes de computadores surgen históricamente a finales de los años 60 como una solución para la interconexión de computadores situados en lugares remotos con el objetivo fundamental de permitir a cualquier usuario de cualquier computador acceder y utilizar los recursos, ya sean hardware o software, del conjunto de las máquinas que constituyen la red.

Los trabajos en el campo de las redes de computadores partieron de máquinas existentes, y el gran esfuerzo se realizó en la resolución del problema de la interconexión eficiente de dichas máquinas situadas en muchos casos a centenares de kilómetros de distancia, utilizando en un principio medios de comunicación preexistentes: la red telefónica.

Durante la última década, los fabricantes de equipos informáticos han ido desarrollando arquitecturas para la realización de redes de computadores, ya sea proporcionando todos los elementos para construir la red, ya sea permitiendo la utilización de servicios públicos de transmisión de datos, para la comunicación entre computadores, con lo que las redes de computadores son actualmente un hecho.

1.1.2. REDES LOCALES

La experiencia alcanzada en el campo de las redes de computadores tuvo su influencia decisiva en el desarrollo de las denominadas redes locales de computadores.

El proyecto Aloha, desarrollado en la Universidad de Hawai, fue un hito importante en este campo. Otro hecho importante lo ha constituido el sistema Ethernet desarrollado por Xerox, del que nos ocuparemos ampliamente en otro lugar.

El tema de las redes locales de computadores es actualmente un campo de conocimientos y realizaciones en completa evolución sin haber alcanzado la sedimentación necesaria a la que estamos habituados en otros campos de la informática, es decir, la disponibilidad de una serie de productos comercializados como ocurre en el campo de las redes de computadores.

1.1.3. SISTEMAS MULTICOMPUTADORES

Ya durante la década de los 60 los fabricantes de equipos informáticos fueron paulatinamente desarrollando y aplicando la idea de la descentralización de funciones en un computador. Así aparecieron, en las máquinas clásicas, unidades especializadas, bien en la manipulación de periféricos, o bien en la gestión de comunicaciones, por citar algunos casos más conocidos.

En el campo de los microcomputadores existen desde hace algunos años especificaciones del mecanismo de comunicaciones (buses de comunicación), componentes y sistemas que permiten la realización de sistemas organizados como multimicrocomputadores.

1.1.4. SISTEMAS MULTIPROCESADORES

La realización de máquinas potentes, para el tratamiento de la información, basadas en la cooperación sistemática y ordenada de elementos de menor potencia, funcionando en paralelo, ha sido constantemente una tentación del arquitecto de computadores desde mucho antes de la aparición del microprocesador. Sin embargo, en muchos casos, estos esfuerzos se han visto superados por el propio avance de la tecnología que posibilitaba computadores de un solo procesador potentes y rápidos, y no justificaba el interés por los sistemas multiprocesadores. Por lo tanto, el uso de sistemas multiprocesadores se ha visto reducido a casos específicos en los que la velocidad y características del problema lo justifican, como podría ser el de sistemas de tratamiento de señales en tiempo real.

2. OBJETIVOS DE LAS REDES DE ORDENADORES

Con el fin de proveer un fácil intercambio de la información y de compartir los recursos, las redes de ordenadores deben ser diseñadas para ajustarse a la naturaleza particular de la comunicación.

Los objetivos que debe cubrir la red son, por tanto:

- Proveer la compartición de recursos tales como información (bases de datos) o procesadores.
- Proveer comunicaciones, entre usuarios y/o procesadores.
- Mejorar la seguridad de la red de forma que si un procesador se avería, otro procesador pueda tomar su lugar. Igualmente, si falla un enlace en la ruta, otra ruta puede llegar a estar disponible.
- Proveer la capacidad de distribución de las funciones del procesamiento. Por ejemplo, que una transacción sea traducida en un nodo, procesado en otro, y la respuesta formateada en un tercer nodo.
- Suministrar control centralizado para un sistema distribuido geográficamente.
- Proveer alta velocidad y ancho de banda. Los procedimientos de control de la red y de canalización de la información deben introducir el menor retraso posible en la transmisión de datos.
- Bajo costo. El costo de los dispositivos (terminales, procesadores, periféricos, etc.) está disminuyendo, a la vez que el número de usuarios, y por tanto de nodos, que necesitan participar de la red se está incrementando.

Por lo tanto, la red debe estar diseñada de forma que sea fácil de ampliar al mínimo costo.

- Compatibilidad. La red debe de estar diseñada de forma que sea capaz de mantener la comunicación entre una gran variedad de dispositivos. A su vez debe estar capacitada para conectarse a otras redes ya sean locales o de largo alcance.
- La red debe ser diseñada para permitir la adición de nuevos dispositivos, y la reordenación de la configuración de la propia red. Estos cambios deben producir mínimas o nulas molestias en el resto de la red.
- La red debe ser simple de configurar, de conectarse a ella y de usar. Los usuarios deben poder utilizar

todas las ventajas y recursos de la red con un mínimo de conocimientos técnicos.

3. COMPONENTES DE LA RED

Una red de comunicación de computadores está formada por una serie de nodos con recursos de computación, y computadores de "conmutación nodal" que establecen la comunicación a través de una serie de enlaces de transmisión.

Un usuario accede a la red a través de un terminal o una consola de sistema.

El transporte de los mensajes hasta el terminal receptor o el nodo que los procesa se efectúa a través de una "red de comunicación".

De este modo, las redes de comunicación entre computadores pueden ser divididas convenientemente en dos subredes: "la subred del usuario" y "la subred de comunicación".

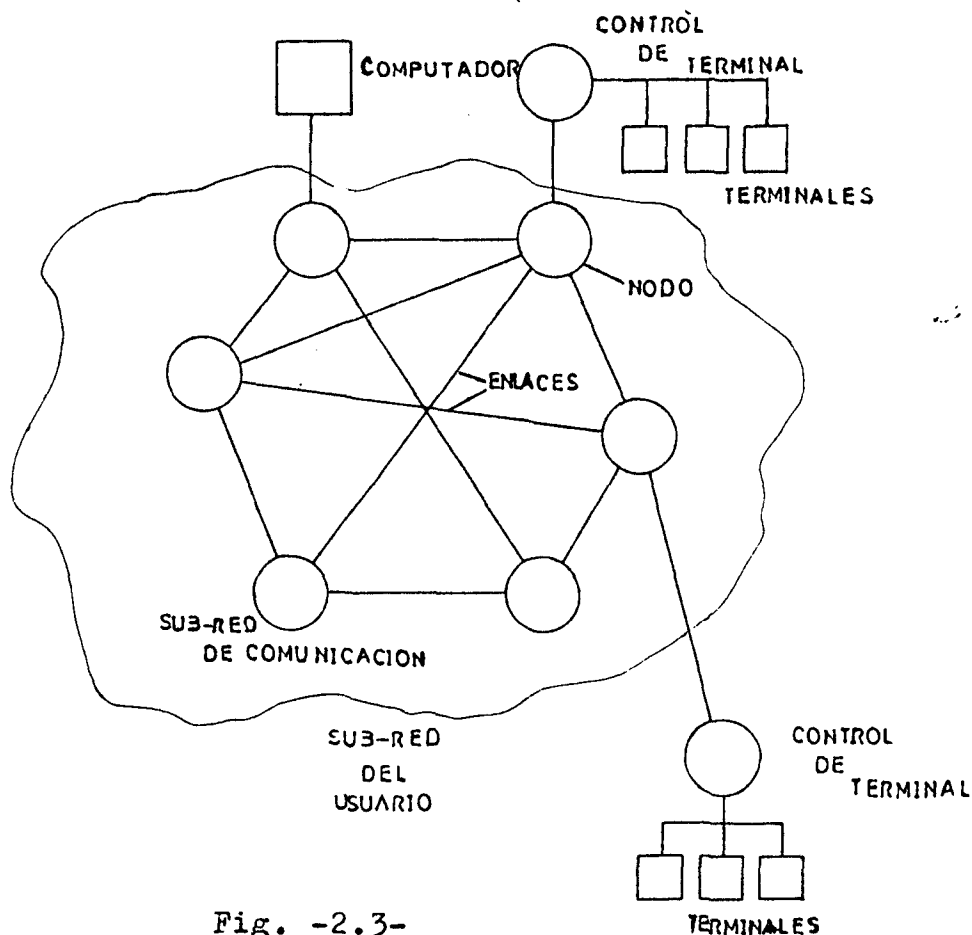


Fig. -2.3-

3.1. SUBRED DEL USUARIO

Esta subred suministra al usuario la capacidad de acceder a la red. Está formada por tres tipos de componentes: computadores "Host", controladores de terminales y terminales.

Los computadores "Host" proveen las bases de datos y programas de aplicación que pueden ser utilizados por los usuarios de los terminales. El computador "Host" se conecta a un nodo de la subred de comunicación a través de un canal multiplexor de alta

velocidad o un enlace de comunicación.

Un usuario de terminal, generalmente accede a la red a través de un controlador de terminales. Los controladores de terminales pueden proveer varios controles para un grupo de terminales, reduciendo por tanto la complejidad y los costos de estos. El controlador de terminales puede ser conectado a un nodo de la red directamente o a través de un procesador "Host".

Alternativamente, la función del controlador de terminales puede ser incluida en un nodo de la red. En este caso, el terminal se conecta directamente al nodo de la red.

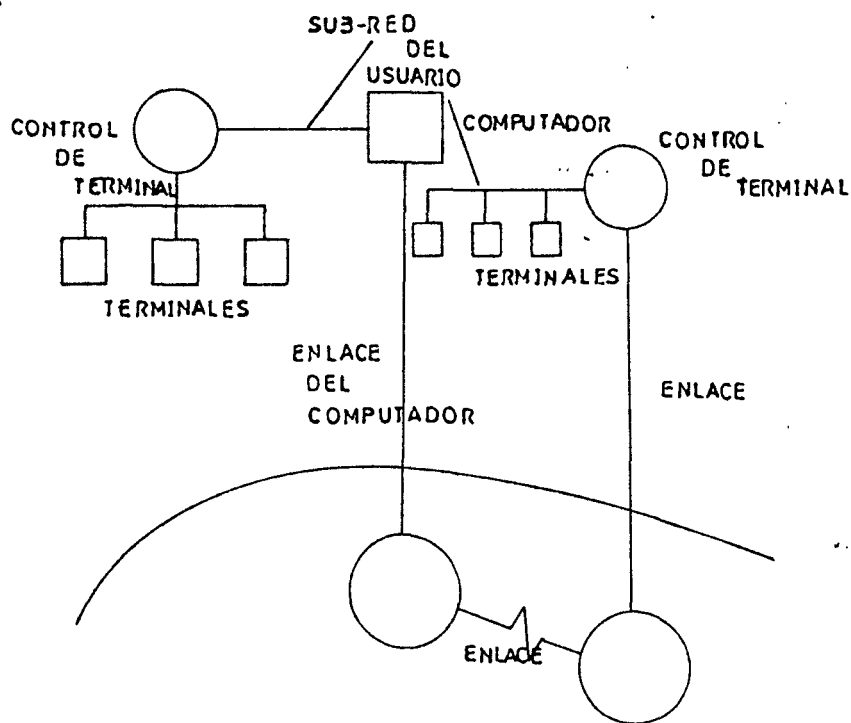


Fig. -2.4-

3.2. SUBRED DE COMUNICACION

Esta subred está formada por una serie de nodos en los cuales residen los procesos que establecen la comunicación entre ellos (a través de una serie de enlaces). Un ejemplo de una subred de comunicación puede verse en la siguiente figura. Está formada por nodos, enlaces de transmisión y equipos conversores de señal.

Los enlaces de transmisión son los canales de comunicación usados para transmitir datos. En los últimos años, los canales de satélite y las comunicaciones por radio han sido empleados también como enlaces de transmisión.

En algunos casos se emplea más de un enlace entre un par de nodos adyacentes, para proveer mayor ancho de banda y seguridad (mediante la redundancia).

Los convertidores de señal realizan la transformación entre las señales digitales aceptadas por los nodos de la red y las señales analógicas usadas para la transmisión en los enlaces.

Los nodos realizan una función doble: de interface para la subred del usuario y de almacenamiento y retransmisión (store and forward) para otros nodos de la red.

Como interface provee funciones tales como la recepción y la liberación de mensajes, y la "monitorización" del estado de transmisión del mensaje.

Como nodo de almacenamiento y retransmisión, provee las funciones de conmutación, para establecer la ruta de los mensajes en la red hasta el nodo de destino apropiado.

El software de un nodo de la subred de comunicación debe estar formado por varios componentes. Primero debe observar los protocolos de la red, o las reglas que determinan los contenidos y temporización relativas para intercambiar mensajes e información de control con otros nodos. Asimismo, debe proveer interfaces apropiados para procesadores "Host", terminales y controladores de terminales.

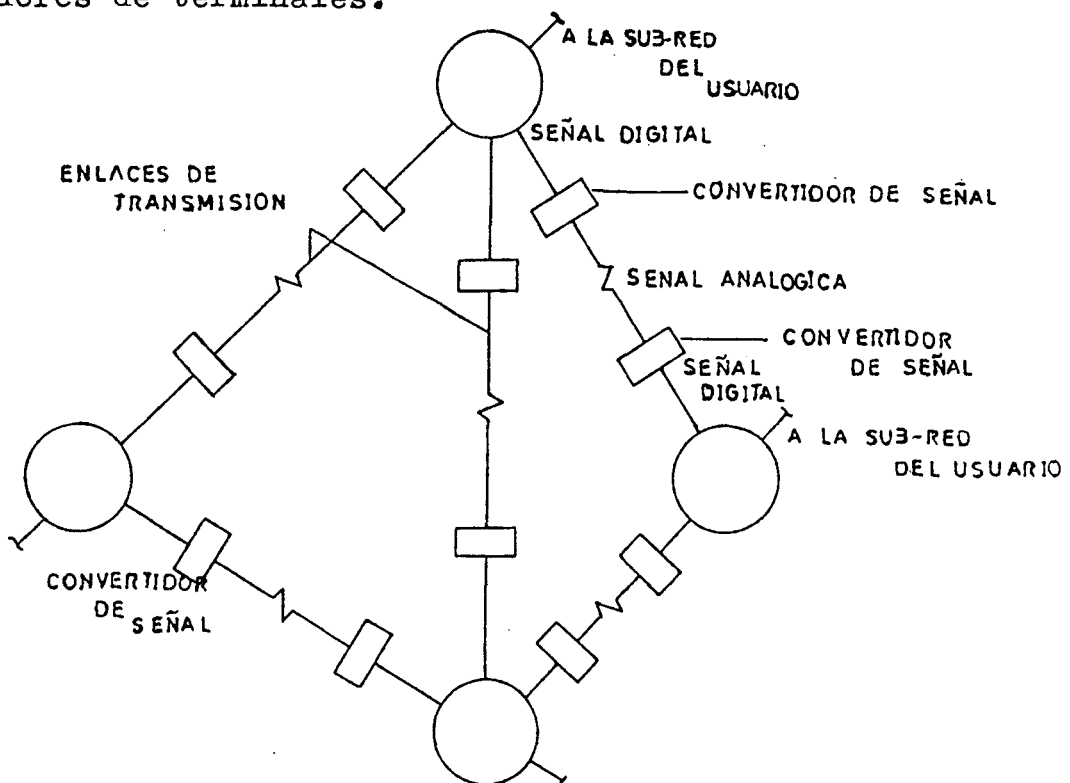


Fig. -2.5-

4. CONTROL Y TOPOLOGIA DE LA RED

La topología de la red es la disposición geométrica de los enlaces y nodos que componen la red. Un enlace (también llamado línea, canal o circuito) es la vía de comunicación entre dos nodos.

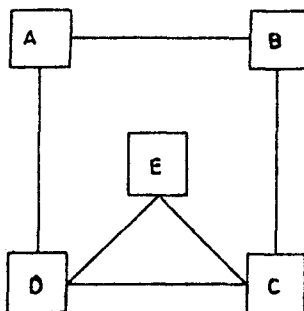
4.1. CAMINOS FISICOS Y LOGICOS

La realización de un enlace implica la utilización de una vía de comunicación que aquí designaremos de forma genérica con el nombre de camino lógico.

Por otra parte, en un determinado sistema, un camino lógico deberá materializarse utilizando los medios físicos de que se disponga, es decir, utilizando los caminos físicos existentes en el sistema. Un camino físico será, pues, una vía de comunicación realizada sobre un soporte material capaz de permitir la transmisión de la información mediante la utilización de algunos de los parámetros físicos de dicho medio.

Por lo tanto, caminos (o conexiones) físicos son aquellos elementos materiales que permiten la comunicación directa entre dos nodos. Y un camino lógico implica que dos nodos pueden comunicarse ya sea a través de una conexión directa o no.

Para entenderlo mejor veamos el siguiente ejemplo:



Vemos que entre A-B, A-D, C-D, C-B, E-C y E-D existen caminos físicos. Sin embargo, entre A y E no existe camino físico por ejemplo, pero si existe un camino lógico, puesto que A puede comunicarse con E a través de D.

4.2. ENLACES PUNTO A PUNTO Y MULTIPUNTO

Hay dos clases de enlaces que sirven como bloque de construcción de las topologías de redes.

Un enlace "punto a punto" es un circuito que conecta 2 y solo 2 nodos sin pasar a través de un nodo intermedio.

Como cada nodo sólo puede comunicarse con aquellos otros nodos a los que está ligado físicamente, la realización de una red en la que todos los nodos estén enlazados entre sí (es decir, que cada nodo pueda comunicarse directamente con todos

los otros nodos de la red), puede resultar bastante cara y compleja (según el número de nodos que compongan la red).

La estructura de red punto a punto puede ser simplificada, y la comunicación entre nodos facilitada dándole a algunos o a todos los nodos la capacidad de "retransmitir" mensajes (creando enlaces lógicos y eliminando la necesidad de enlaces físicos). Esto, por supuesto, tiene como contraposición la desventaja de que el diseño de los nodos se vuelve más complejo, y la comunicación resulta más lenta ya que los nodos "retransmisores" tienen que invertir un cierto tiempo en comprobar si la transmisión es con él o es con el nodo siguiente.

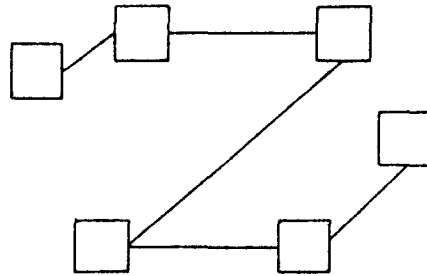


Fig. -2.6-

Un enlace "multipunto" es una simple línea compartida por más de dos nodos.

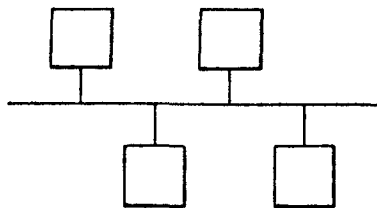


Fig. -2.7-

Este tipo de enlace, también llamado "distribuido" reduce el número de líneas, y por tanto, el coste de la red.

Asimismo, los nodos de una línea multipunto son generalmente más complejos que los nodos de un enlace punto a punto. Ellos deben, al igual que los nodos con capacidad de retransmitir, manejar mensajes basados en sus direcciones.

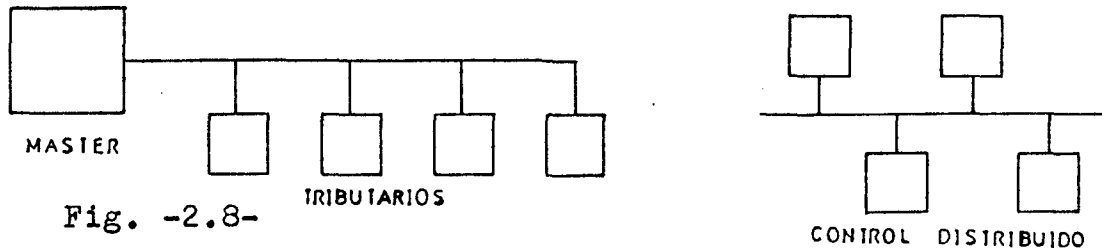
Y además, deben implementar algunos recursos para controlar el acceso a la línea (multipunto), previniendo posibles conflictos, ya que la línea es compartida por varios nodos.

4.3. TOPOLOGIA Y CONTROL DE LOS ENLACES FISICOS. TIPOS DE REDES

La estrategia de control de una red se decide fácilmente en los etapas de diseño, cuando se especifica el número y clase de los nodos participantes, así como la naturaleza de su interacción. La configuración de los nodos y enlaces de la red -la topología- es la principal manera de implementar la estrategia de control elegida.

En la fase de diseño de la red se debe decidir si el control va a ser centralizado o distribuido. Con control centralizado, el acceso a la red y la repartición del canal son controladas

por un solo nodo. Cuando el control es distribuido, los nodos tienen la capacidad de establecer conexiones y acceder al canal independientemente teniendo en cuenta, claro está, una serie de reglas.



Un ejemplo de dos estrategias de control típicas puede verse en las configuraciones multipunto. El control de la línea multipunto puede ser centralizado en uno de los nodos, llamado el "master" (maestro). Este nodo controla el uso que hacen de la línea los otros nodos, llamados "tributarios". En este caso, todos los mensajes de la línea, para, desde y entre los nodos tributarios pasan a través del master. Alternativamente, el control puede ser distribuido tal que cada nodo tiene la capacidad de pugnar por la línea para transmitir, cuando ésta está libre de mensajes, Este mecanismo de "pugna" es coordinado mediante una serie de reglas implementadas en cada nodo.

4.3.1. TOPOLOGIAS NO RESTRINGIDAS

Las configuraciones de las redes no restringidas (también llamadas híbridas) no son específicas. Las conexiones son determinadas normalmente por la economía de la red. Cuando el costo del medio de transmisión es alto, y el número de nodos es limitado, se gana eficiencia eligiendo sólo las conexiones necesarias.

Las topologías no restringidas pueden ser hechas de combinaciones de enlaces punto a punto y multipunto usando nodos retransmisores y no retransmisores.

La particular flexibilidad de la configuración y la eficiencia ofrecida por esta topología hace que sea usada a menudo por redes de conmutación de paquetes de largo alcance.

Sin embargo, las características de las redes no restringidas están en contradicción con los objetivos de diseño que como ya vimos, debían tener las redes:

- Los nodos que pueden tomar decisiones para la retransmisión de mensajes debe ejecutar a menudo más funciones relacionadas con la red de las que son deseables. Esto introduce retrasos en la comunicación.
- La eficiencia y economía ganadas haciendo sólo las conexiones necesarias no son aplicables a todas las redes. Particularmente en casos donde muchos nodos

comparten un medio de transmisión de bajo coste (tal como cable coaxial) sobre distancias limitadas.

Muchas redes están basadas en topologías simples y estructuradas, tales como las redes en anillo, tipo bus, y en estrella. Estas topologías permiten una implementación más efectiva y uniforme de las estrategias de control de la red, cumpliendo además los objetivos de diseño de éstas.

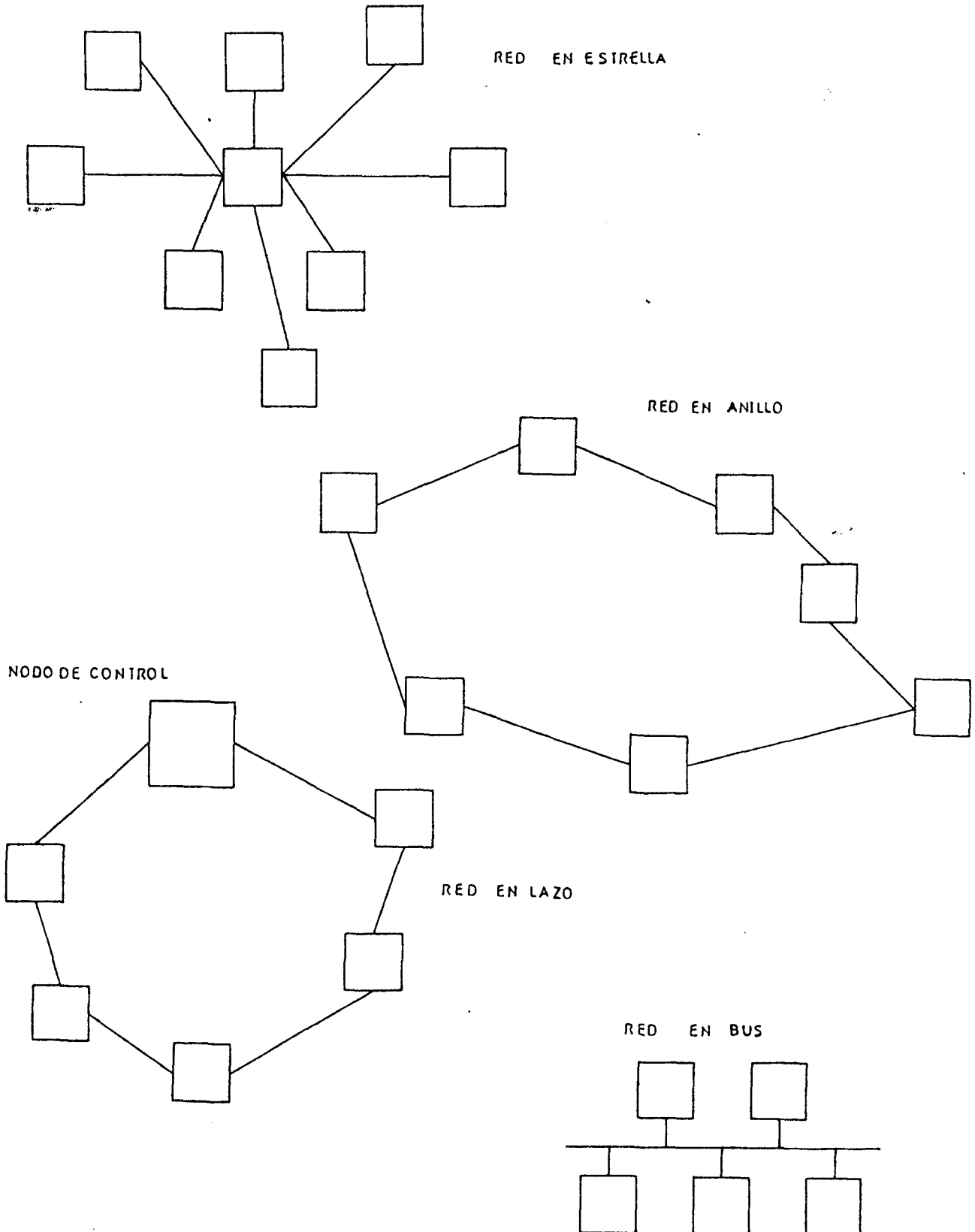


Fig. -2.9-

4.3.2. REDES EN ESTRELLA

La característica principal de las redes en estrella, o radiales, es que todos los nodos están unidos a un simple punto. Las funciones de esta red dependen de la forma en que se ejerza el control. Este se realiza normalmente según una de las siguientes tres formas.

Primero, las configuraciones en estrella son usadas frecuentemente para redes en las que el control está localizado en el nodo central o conmutador. En este caso, todas las retransmisiones de los mensajes -del nodo central a los periféricos, entre nodos periféricos, y de todos los nodos a puntos remotos- se efectúan por el nodo central. Esto tiene la ventaja de que descarga a los nodos periféricos de las funciones de control.

Los nodos periféricos se conectan al nodo central mediante líneas punto a punto, eliminando la necesidad de enlaces complejos y de estrategias de control de otras topologías, y permitiendo una conexión a la red simple y económica a través del nodo central. Por cada nodo conectado debe añadirse una línea punto a punto.

Las redes en estrella que son controladas por un nodo central son eficaces cuando el volumen de comunicación es entre el nodo central y los periféricos. Si el tráfico entre los nodos periféricos es alto, en lugar del nodo central se coloca un conmutador (inoportuno)

Segundo, una red en estrella podría ser construida de forma que el control de las comunicaciones pueda ser ejercido por un nodo periférico, o tercero, pueda ser distribuido generalmente a todos los nodos. En ambos casos, la función de control del nodo central puede ser minimizada. Este nodo podría actuar como un simple conmutador para establecer la conexión entre los nodos periféricos.

En todas las redes en estrella el nodo central es el punto más importante en caso de fallos. Si se estropea éste, dejará de funcionar toda la red. En el caso de control central, el tamaño y capacidad de la red es función directa de la potencia del nodo central.

La topología de estrella se usa a menudo en aplicaciones de tiempo compartido, en donde el nodo central sirve como controlador del tiempo. También se usa en redes telefónicas PBX (Private Branch Exchange) y en pequeñas redes agrupadas tales como las de procesamiento de palabras.

4.3.3. REDES EN ANILLO

La característica principal de las redes en anillo es que los nodos, que están conectados mediante enlaces punto a punto, son colocados formando una configuración circular ininterrumpida. Los mensajes transmitidos viajan de nodo a nodo a lo largo del anillo. Cada nodo debe ser capaz de reconocer su propia dirección para poder aceptar mensajes. Además cada nodo sirve como "repetidor activo", retransmitiendo mensajes direccionados a otros nodos.

Cuando las redes en anillo se usan con control distribuido, deben usarse métodos de acceso a ésta que prevengan posibles conflictos debidos a demandas simultáneas del canal compartido. Una forma de hacer esto es haciendo circular un bit patrón, llamado la "señal", alrededor del anillo.

Un nodo gana el acceso exclusivo al canal cuando "agarra la señal", y pasa el derecho de acceder al canal (por ejemplo la señal) a otro nodo cuando finaliza su transmisión.

Cuando el control es distribuido, cada nodo puede comunicar directamente con todos los otros nodos (conexión lógica) bajo su propia iniciativa.

Las redes en anillo con control centralizado son conocidas a menudo como "lazos". Uno de los nodos (el nodo de control) controla el acceso, y las comunicaciones efectuadas en el canal por los otros nodos. Una vez un nodo es autorizado por el nodo de control a enviar un mensaje, éste puede viajar a través del anillo a su destino sin la intervención adicional del nodo de control.

Los "lazos" pueden ser diseñados también de forma que todos los mensajes deben pasar a través del nodo de control. En éste caso, el lazo funciona como una red en estrella con un nodo de control localizado centralmente.

Los fallos en un nodo, en un componente activo (tal como un repetidor), o cualquier otra rotura en la configuración de anillo producirá siempre una parada en toda la red.

4.3.4. REDES TIPO BUS

La topología tipo bus funciona en la misma forma que la línea multipunto (como una simple línea compartida por varios nodos).

En contraste con las topologías en estrella y en anillo, los nodos en la red tipo bus comparten un mismo canal físico.

La topología tipo bus, como la de anillo, ha sido usada a menudo para el control distribuido de procesos. Los mensajes o

colocados en el bus son "radiados" a todos los nodos. Los nodos deben ser capaces de reconocer su propia dirección para poder recibir los mensajes transmitidos. Sin embargo, a diferencia de la red en anillo, no tienen que repetir los mensajes destinados a otros nodos. Como consecuencia de esto se evitan los retrasos asociados a la retransmisión de los mensajes, y los nodos son liberados de la responsabilidad de controlar el canal a éste nivel.

Sin embargo, las topologías tipo bus son posibles pero no comunes. Lo normal es que haya un nodo "master" con varios nodos "tributarios". Ahora bien, como en todos los casos de control centralizado, si el nodo "master" falla, deja de funcionar toda la red, negando de ésta forma una de las características más atractivas de la topología tipo bus.

5. TECNOLOGIA DE TRANSPORTE DE LOS MENSAJES

Las redes pueden ser clasificadas convenientemente según la técnica empleada para el transporte de los mensajes entre nodos. Existen básicamente tres tipos: redes de conmutación de circuitos, de mensajes y de paquetes. A continuación haremos un estudio detallado de cada una.

5.1. REDES DE CONMUTACION DE CIRCUITOS

Se define a la conmutación de circuitos como el procedimiento que enlaza a voluntad dos o más equipos terminales de datos y que permite la utilización exclusiva de un circuito de datos durante la comunicación. Este es el método predominantemente utilizado en las redes telefónicas.

En la conmutación de circuitos, cada vez que se origina una llamada debe establecerse la vía eléctrica o el circuito que provea la interconexión entre los dos nodos. Esto debe producirse antes de que comience la transmisión de datos en sí. La pequeña pausa que hay desde que se termina de marcar el número (en la red telefónica) hasta que comienza a sonar el pitido es el tiempo necesario para establecer el circuito o vía de enlace y se conoce como "tiempo de establecimiento de la llamada".

Después de que se establece la conexión, el uso de éste enlace es exclusivo y continuado hasta que termina la conversación; después, se desconecta y queda preparada para otra conexión.

El tiempo requerido para el establecimiento de la llamada no representa un problema en comunicaciones de larga duración. Sin embargo, en comunicaciones mucho más cortas (del orden de microsegundos) que requieren que toda la capacidad del canal sea dispuesta rápidamente, la conmutación de circuitos es lenta, relativamente cara e ineficiente.

5.2. REDES DE CONMUTACION DE MENSAJES

Una red de conmutación de mensajes transmite un mensaje entre los nodos moviendo dicho mensaje a través de varios buffers y enlaces de transmisión. El mensaje sale del nodo fuente y llega hasta el siguiente nodo, que lo almacena y retransmite hasta el siguiente; así continúa hasta llegar al nodo "destinatario". La transmisión de un mensaje desde un nodo no empieza hasta que se haya dispuesto un buffer para él en el siguiente nodo. La vía para la transmisión del mensaje puede ser fija, o puede ser determinada dinámicamente mientras el mensaje avanza hacia su nodo de destino.

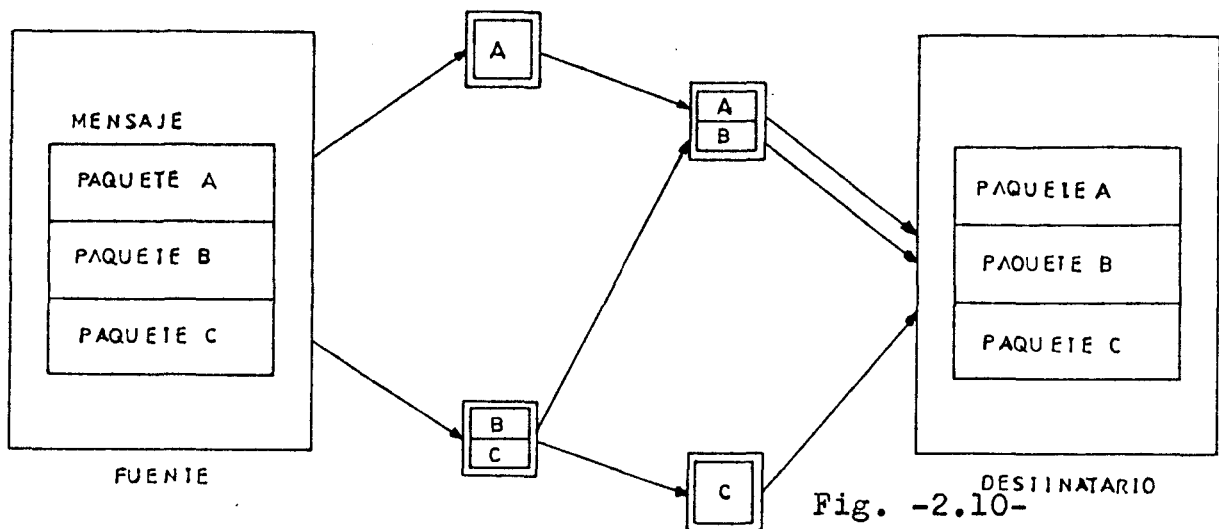
Esta clase de redes son llamadas también redes de "almacenamiento y retransmisión", ya que los mensajes son almacenados en cada nodo y luego retransmitidos hasta el siguiente nodo de su vía. Un importante atributo de las redes de conmutación de mensajes es la gran variación en los retrasos de los mensajes a través de la red.

5.3. REDES DE CONMUTACION DE PAQUETES

Las redes de conmutación de paquetes se desarrollaron como la solución para el intercambio de mensajes entre computadores.

La ISO define la conmutación de paquetes como un procedimiento de transferencia de datos mediante paquetes provistos de direcciones, en el que la vía de comunicación se ocupa solamente durante el tiempo de transmisión de un paquete, quedando a continuación la vía disponible para la transmisión de otros paquetes.

En realidad, las redes de conmutación de paquetes sólo difieren de las redes de conmutación de mensajes en que los mensajes largos son descompuestos primero en segmentos de tamaño fijo llamados "paquetes". Estos paquetes pueden viajar independientemente a través de la red hasta que alcanzan el nodo deseado, en donde son reensamblados de nuevo para recomponer el mensaje. De este modo, muchos paquetes del mismo mensaje pueden ser transmitidos simultáneamente.



Los paquetes que componen un mensaje particular son almacenados en los buffers de memoria del nodo de destino hasta que lleguen todos los paquetes necesarios para reensamblar el mensaje completo. Por lo tanto, los nodos de una red de conmutación de paquetes, aparte de los buffers de almacenamiento y retransmisión necesitan buffers reensambladores.

En ocasiones, a los nodos de un sistema de este tipo se les denomina "conmutadores de paquetes" debido a las funciones que realizan.

6. NODOS DE TRANSMISION

Existen tres formas básicas de transmitir o transportar los datos entre dos nodos de la red: Transmisión Simplex, Half dúplex y Full dúplex.

6.1. TRANSMISION SIMPLEX

En este modo, la transmisión se realiza solamente en un sentido, sin posibilidad de hacerlo en el opuesto.

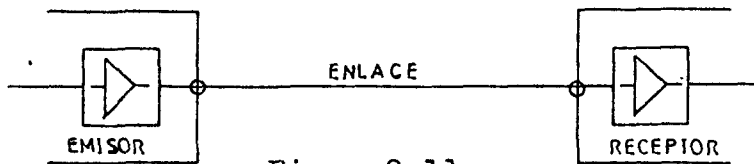


Fig. -2.11-

Este sistema tiene muy escaso uso en transmisión de datos, con la excepción de telemetría.

6.2. TRANSMISION HALF-DUPLEX

En este modo, la transmisión puede llevarse a cabo en los dos sentidos, pero no simultáneamente. Es decir, que en un momento dado, la transmisión sólo se puede estar realizando en un sentido.

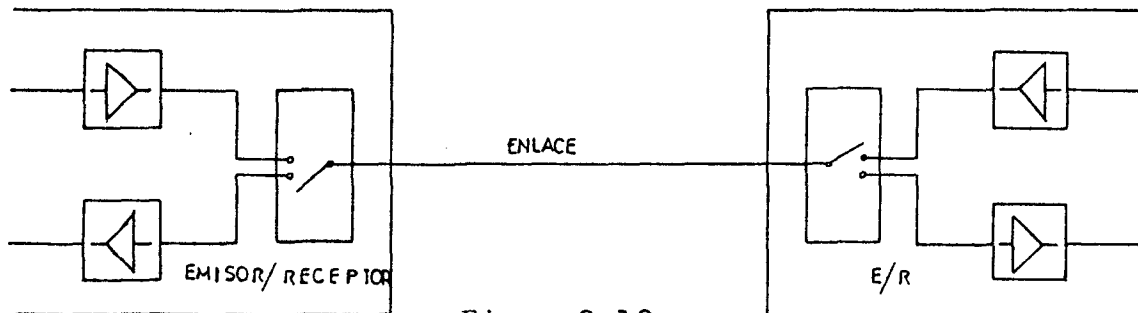


Fig. -2.12-

Este tipo de transmisión es el más corrientemente usado en la transmisión de datos, incluso sobre circuitos que permiten el modo dúplex.

6.3. TRANSMISION FULL-DUPLEX

En el modo de transmisión Dúplex o Full-dúplex, los datos pueden fluir simultáneamente e independientemente en los dos sentidos.

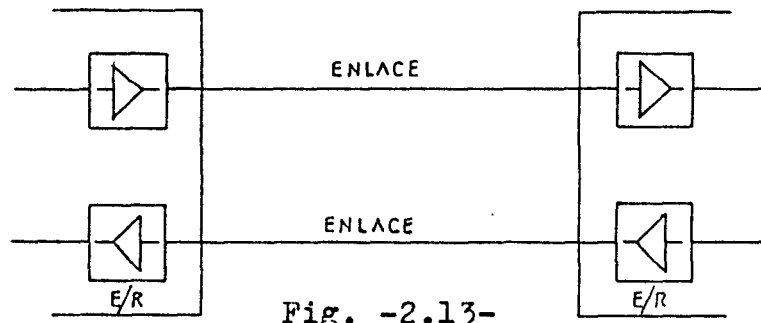


Fig. -2.13-

Este sistema de transmisión ofrece la máxima funcionalidad y efectividad para una transmisión de datos. Sin embargo, debido a la complejidad que deben tener los interfaces de los equipos terminales de datos, y los modems, este sistema se utiliza muchísimo menos que la transmisión Half-dúplex.

7. PROTOCOLOS DE LAS REDES

Una red está formada por una serie de nodos interconectados que permiten el intercambio de unidades de información, o datos, entre ellos. Para que dicho intercambio sea ordenado, se necesita que cada nodo se ajuste a una serie de convenios o reglas pre-establecidas. Estas reglas especifican los formatos y la temporización relativa de los mensajes que van a ser intercambiados entre los nodos. Los protocolos de las redes están formados esencialmente por tres elementos:

- 1) La sintáxis, o lo que es lo mismo, la estructura de los datos y de los mensajes de control;
- 2) La semántica, o la serie de mensajes de control que van a ser usados, las acciones a ser desarrolladas, y las respuestas que van a ser devueltas; y
- 3) La temporización, o especificación del orden de ejecución de los eventos.

Una arquitectura de red especifica las funciones precisas que podrían efectuar una red y sus componentes, a la vez que provee el armazón en el que pueden participar nodos con posibles estructuras diferentes de software y hardware.

7.1. CONCEPTOS EN DISEÑOS DE PROTOCOLOS

El principal propósito de un protocolo es establecer ordenadamente intercambio de información entre procesos y dirigir eficientemente los recursos de la red. De este modo, un protocolo puede ser concebido también como una serie de convenios entre dos procesos en comunicación.

Consideremos el intercambio de mensajes entre dos procesos A y B.

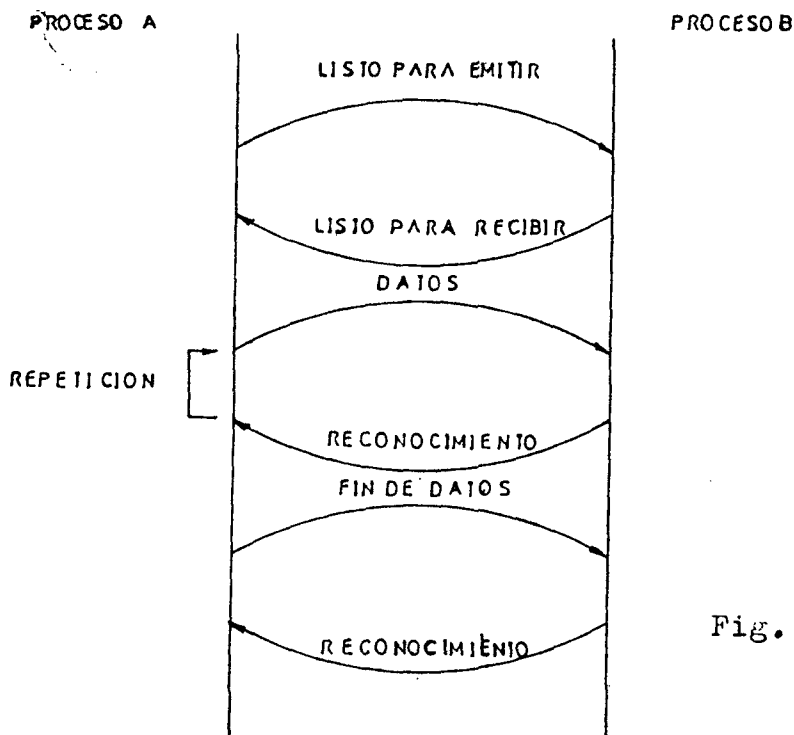


Fig. -2.14-

Para comenzar la transmisión desde A hasta B, el proceso A envía el mensaje de control "listo para enviar" (Ready to send) al proceso B. El proceso B responde transmitiendo el mensaje de control "listo para recibir" al proceso A. A partir de aquí se inicia la transmisión de datos desde el proceso A al proceso B. Los datos son enviados por el proceso A y reconocidos por el proceso B. Para finalizar el intercambio, el proceso A envía el mensaje de control "fin de datos" (End of data) al proceso B, y éste contesta con un reconocimiento al proceso A.

El protocolo para éste simple intercambio de datos debe especificar también el formato de cada mensaje de control, el encajamiento de los datos (esto ya se verá más adelante), y la temporización relativa.

La necesidad de establecer protocolos puede apreciarse mejor cuando hay varios tipos de nodos en la red. Consideraremos la siguiente red:

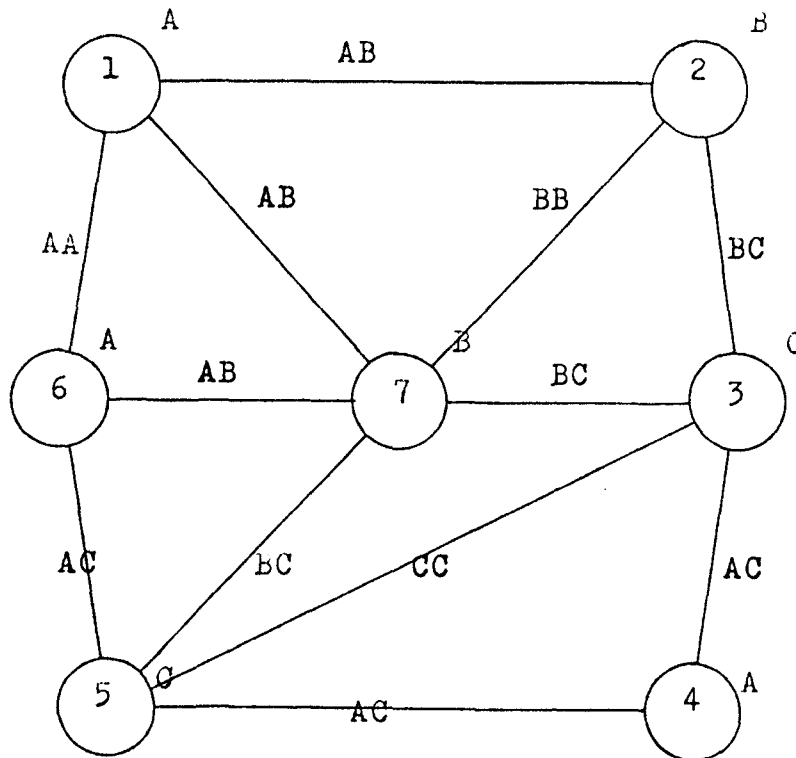


Fig. -2.15-

La red está formada por 7 nodos. Hay tres tipos de nodos (A, B y C) diferentes en función de las reglas que utilizan para comunicarse con los demás. Para comunicar estos nodos hay 6 formas únicas de hacerlo. Estas seis combinaciones son AA, AB, AC, BB, BC y CC. Ahora bien, éste problema puede ser simplificado estableciendo un protocolo común que debe ser seguido por cada tipo de nodo.

Un protocolo de red está constituido por tres componentes lógicos. El primero es una entidad tal como los programas de aplicación (residentes en un sistema computador), o un operador de terminal. El segundo componente es alguna forma de emparejamiento o ligazón para estas entidades que proveen una vía de comunicación lógica entre las dos entidades. Este ligazón puede ser permanente o existir sólo durante el tiempo en que las dos entidades desean comunicarse. Por último, existe la necesidad de que haya un mecanismo para transportar la información entre las dos entidades emparejadas.

Una vez conocidas estas tres entidades, el diseñador de protocolos debe establecer varias reglas para el uso y gestión de la red de transporte. Estas reglas no deben estar expuestas a los usuarios de la red, o sea, a los operadores de los terminales. Tales reglas deberían residir dentro de la red de transporte y ser transparentes a las entidades en comunicación. Las reglas para la red de transporte pueden definir las rutas que van a seguir los mensajes, la segmentación de los mensajes en paquetes de tamaño fijo, o los procedimientos para prevenir la congestión en la red.

Esta descripción que hemos hecho nos lleva al concepto de las "capas" (layers).

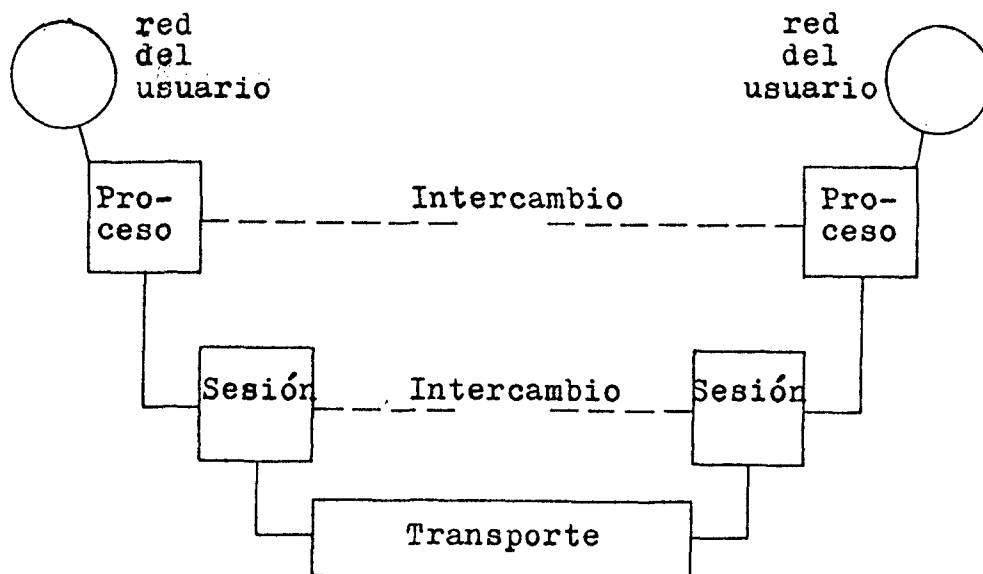


Fig. -2.16-

Como se ve, el usuario de la red accede a ella a través de alguna entidad lógica que llamaremos "proceso". La capa o el nivel "sesión" recibe un mensaje del nivel "proceso" y realiza la acción necesaria, tal como el establecimiento o gestión del diálogo entre los usuarios de la red. El nivel "sesión" envía mensajes al siguiente nivel más bajo, o sistema de transporte,

que entrega los mensajes al nivel "sesión" del nodo de destino. El principio es que un nivel dado de un nodo intercambia mensajes lógicamente con su correspondiente nivel de otro nodo, y que los procesos de los otros niveles son transparentes a él. Hay protocolos par a par para la comunicación entre los niveles correspondientes de diferentes nodos.

Un nivel dado también se comunica con sus niveles adyacentes superior o inferior a través de un protocolo de interface.

Este concepto de división en "niveles" se sigue prácticamente en todos los diseños de protocolos porque tiene varias ventajas:

- Permite la interacción entre los niveles emparejados funcionalmente de diferentes nodos.
- Hace que el software del protocolo sea simple de describir, desarrollar, y testear.
- Permite cambios o modificaciones en un nivel dado sin afectar a los otros niveles.
- Provee una gestión sencilla de los protocolos.

7.2. FUNCIONES DE LOS PROTOCOLOS

Los nodos de la red intercambian dos tipos de mensajes: controles y datos. Los datos son unidades de información intercambiadas entre usuarios de la red, tales como operadores o programas de aplicación. Los mensajes de control son usados para intercambiar información entre los niveles de diferentes nodos para facilitar la transmisión de los datos. Ahora bien, estos mensajes de control no son conocidos por los usuarios de la red, y se transmiten dentro de ésta.

A continuación veamos algunas de las numerosas funciones que realizan los protocolos de las redes:

1. Intercambio ordenado de datos.
2. Gestión de prioridades de las entradas a la red y de los niveles de transmisión.
3. Sincronización del proceso.
4. Establecimiento de la sesión entre usuarios de la red.
5. Terminación de la sesión entre usuarios de la red.
6. Establecimiento y asignación de las rutas de los mensajes.
7. Control del flujo y prevención de congestiones.
8. Secuenciamiento: transmisión en secuencia de los datos.
9. Direccionamiento de los componentes y usuarios de la red.

10. Utilización eficiente de los recursos de la red.
11. Gestión, monitorización y protección de los recursos.
12. Transparencia de las niveles entre los usuarios y nodos de la red.
13. Transmisión fiable de los mensajes, incluyendo el control de errores.
14. Testeo de los recursos de la red, tales como los enlaces y las rutas.
15. Seguridad.

Cada nivel de un nodo soporta su propio protocolo para comunicarse con el nivel correspondiente de otro nodo.

7.3. ORGANISMOS DE NORMALIZACION

Ya hemos visto los criterios de diseño y las funciones que debe cumplir un protocolo.

A continuación vamos a presentar de una manera sucinta los trabajos de algunas de las principales organizaciones que se ocupan de actividades de normalización en el campo de los sistemas distribuidos y en el de la transmisión a través de redes públicas de datos.

Hay que decir que la problemática en torno a las normalizaciones es ciertamente compleja; en primer lugar aparecen las soluciones y a continuación se toma como base una de ellas -la que más presiona, generalmente-, un comité la corrige y modifica convenientemente y finalmente elabora una norma; posteriormente se adopta, pero no exactamente como ha sido emitida. A pesar de todo, las normas suelen ser una valiosa fuente de información.

7.3.1. LA ISO

La Organización Internacional de Normalización ISO (de la que España es país miembro), es una federación de organismos nacionales de normalización y se ocupa de la elaboración de recomendaciones internacionales a partir de propuestas de los países miembros y otros organismos profesionales. Sus trabajos se organizan en comités técnicos (TC) por grandes áreas de trabajo y éstos a su vez se subdividen en subcomités (SC) para el estudio de temas específicos.

7.3.2. EL CCITT

Por lo que se refiere al Comité Consultivo Internacional Telegráfico y Telefónico (CCITT) diremos que sus miembros son las organizaciones nacionales de Correos y Telecomunicaciones

(PTT), así como las compañías privadas que ofrecen servicios públicos de comunicaciones y está organizado en comisiones de estudios encargadas de elaborar propuestas que son presentadas a la Asamblea Plenaria para su aprobación.

7.3.3. EL IFIP, LA ECMA Y OTROS

A nivel de organizaciones profesionales cabe citar también las actividades de la Federación Internacional para el Tratamiento de la Información (IFIP) dentro de su comité técnico: comunicaciones de datos y los correspondientes grupos de trabajo (WG). Hay que decir que éste organismo no emite recomendaciones; sus comités técnicos elaboran propuestas que transmiten fundamentalmente a ISO para su posterior estudio y aprobación si procede.

En esta enumeración es necesario mencionar también a la Asociación Europea de Fabricantes de Computadores (ECMA) que quizás más que elaborar, adapta y publica normas confeccionadas por otros organismos. Está organizada en comités técnicos (TC) de los cuales el número 9 (TC9) se ocupa de los temas referentes a la transmisión de datos. Hay que añadir que sus documentos son de una claridad sorprendente.

Citaremos también al IRANOR, Instituto de Racionalización y Normalización, que es el organismo español encargado de emitir normas nacionales, adoptar las internacionales y participar en los trabajos de los organismos internacionales de normalización.

Y finalmente citaremos algunas de las organizaciones americanas que se encargan también de éste tipo de trabajos: la American National Standard Institute (ANSI), la Electronic Industries Association (EIA) y la Federal Telecommunications Standards Committee (FTSC).

Por último, en la siguiente figura damos la relación de los organismos de normalización más importantes y de los distintos comités técnicos y subcomités encargados de elaborar recomendaciones sobre los sistemas distribuidos y la transmisión a través de redes públicas.

ORGANISMO	AREA DE TRABAJO
ISO -Organización Internacional de Normalización	- Comité Técnico 97, Subcomité 16 (ISO/TC 97/SC 16) -Interconexión de sistemas abiertos (Open Systems)
CCITT -Comité Consultivo de Telegrafía y Telefonía	- Comisión de Estudio XVII - Recomendaciones serie V (Transmisión de datos a través de Telefónica y Telex) - Comisión de Estudio VII - Recomendaciones serie X (Transmisión de datos a través de red pública para transmisión de datos)
ECMA -Asociación Europea de Fabricantes de Ordenadores	- Comité Técnico nº 9 (TC 9) -Transmisión de datos
IFIP -Federación Internacional para el Tratamiento de la Información	- Grupo de trabajo nº 6 (WG 6) -Comunicaciones de datos
IRANOR -Instituto de Racionalización y Normalización	-Participación española en entes internacionales de normalización -Elaboración de normas españolas

Fig. -2.17- ORGANISMOS DE NORMALIZACION

7.4. MODELO DE REFERENCIA PARA LA INTERCONEXION DE SISTEMAS ABIERTOS. RECOMENDACION X.25

Como ya hemos visto anteriormente, en el análisis de un sistema de interconexión se utiliza habitualmente la metodología consistente en una estructuración según una jerarquía de niveles o estratos, técnica por otro lado ya habitual en informática. La ISO ha adoptado esta solución en su modelo de referencia. En el siguiente aparecen los elementos constitutivos de dicha jerarquía, cuyas definiciones damos a continuación:

- El sistema de interconexión está formado por un conjunto de "entes" situados a diferentes "niveles" estructurales, denominados "estratos".
- Los entes de un determinado nivel "n" cooperan entre sí de acuerdo con un determinado protocolo "n".
- Los entes de un nivel "n" utilizan los servicios (n-1) proporcionados por los entes de los niveles

inferiores mediante un "acceso" a ellos. La estructura de estos niveles inferiores es desconocida para el nivel "n", el cual nuevamente tiene en cuenta los servicios proporcionados, por lo que se ha denominado bloque "n-1".

- Los entes de un nivel "n" realizan unas determinadas funciones "n", utilizando los servicios de los entes del nivel "n-1" y proporcionando a su vez servicio a los entes del nivel "n+1".

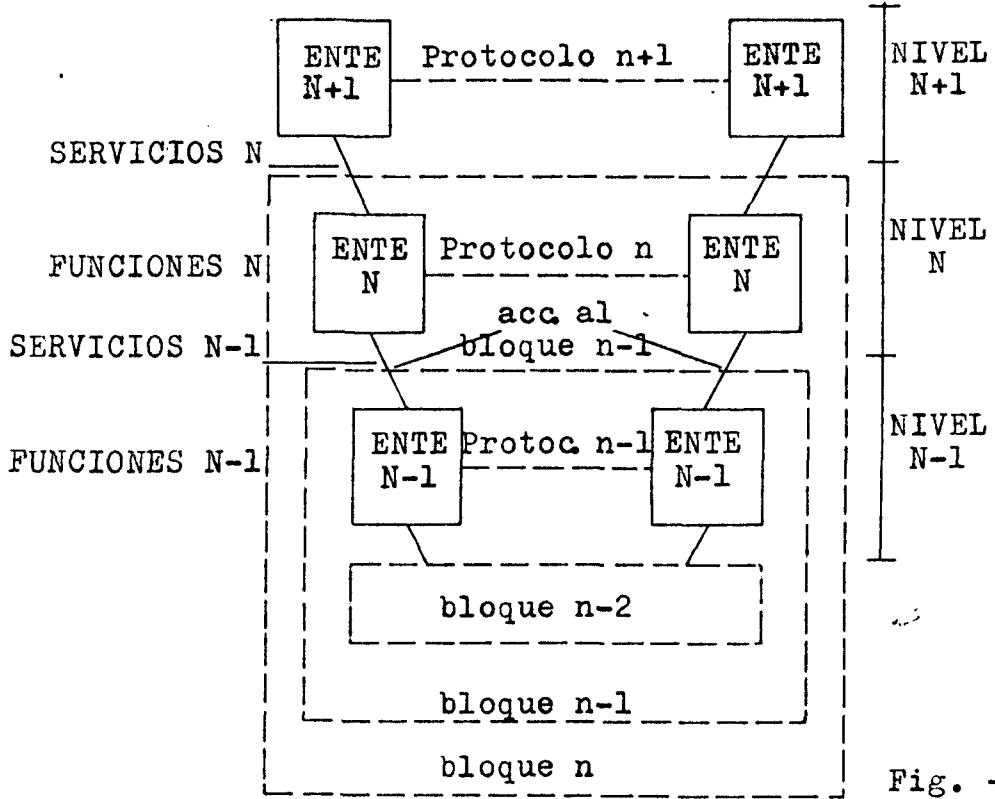


Fig. -2.18-

Según ISO, el modelo al que hemos hecho referencia es suficiente para representar configuraciones simples como sería el caso de sistemas interconectados a través de una línea dedicada.

Basado en dicho modelo se elaboró, pues, el que aparece en la siguiente figura tratando de cubrir configuraciones más complejas como sería el caso de interconexiones a través de una red pública de transmisión de datos.

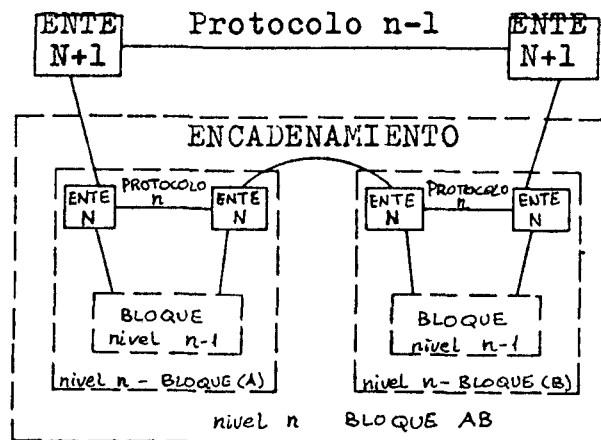


Fig. -2.19-

En este caso se han permitido encadenamientos entre bloques, lo que a su vez permite, como más adelante veremos, soslayar los problemas específicos e internos a la red de transmisión.

Como resultado de todo lo visto, tras una discusión de las propuestas presentadas por los países miembros, ISO elaboró un modelo de referencia para la interconexión de sistemas abiertos, estructurado en siete niveles o estratos.

Antes de adentrarnos en el estudio detallado de las características del modelo de referencia, creemos necesario hacer un comentario general tanto sobre el alcance como sobre sus objetivos.

Vamos, pues, a considerar el problema a través de tres grandes aspectos:

- a) El punto de vista del usuario,
- b) El hecho de que el sistema puede estar formado por máquinas físicamente alejadas, y
- c) El hecho de que para la interconexión pueda utilizarse una red pública de transmisión de datos.

Desde el punto de vista del usuario, un sistema distribuido continuará viéndose como cualquier otro sistema informático. Es decir, formado por un conjunto de "elementos" que aquí se denominará "procesos de aplicación" y entre los cuales podrá establecerse un conjunto de "relaciones" denominadas aquí conexiones. Este punto constituye un aspecto importante del modelo. El aspecto "distribuido" del sistema debe, en principio, ser transparente al usuario. Las funciones que pueda ser capaz de realizar deben, pues, ser similares a las que se ejecutan en un sistema basado en una máquina única.

El hecho de que el sistema esté formado por máquinas físicamente alejadas implica fundamentalmente que la información deba ser transportada entre ellas, ya que en definitiva constituyen "elementos finales" del sistema. En el modelo de ISO aparece claramente reflejada la problemática del transporte de la información.

Por último, el transporte de la información implica la utilización de un medio de transmisión de datos, generalmente una red pública. Por éste motivo se diferencia claramente esta problemática de "transmisión" de la información como una parte de las funciones que constituyen el transporte.

En las figuras 2.20 y 2.21 se representan esquemáticamente las ideas que se acaban de exponer y que servirán de pauta para el estudio de los elementos que constituyen el modelo de referencia, cuya estructura general aparece en la figura 2.22 y podemos esquematizar como se muestra en la figura 2.23.

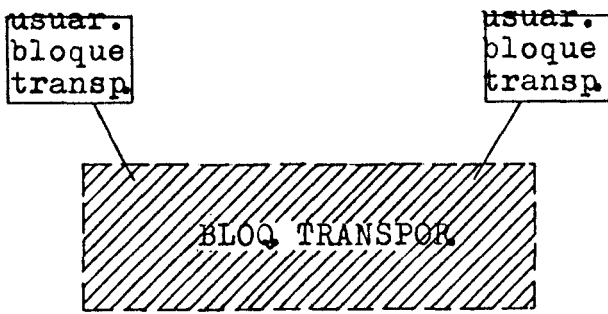


Fig. -2.20-

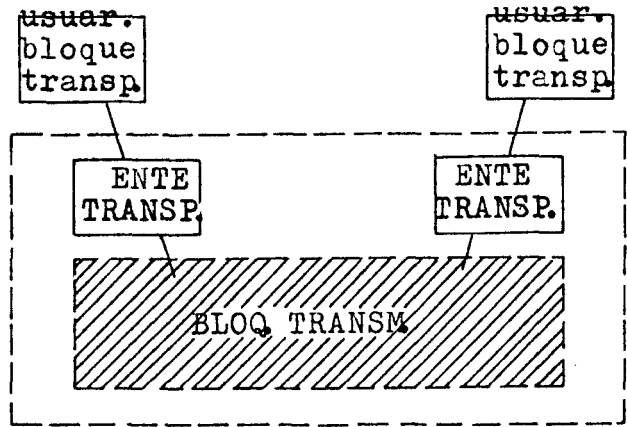


Fig. -2.21-

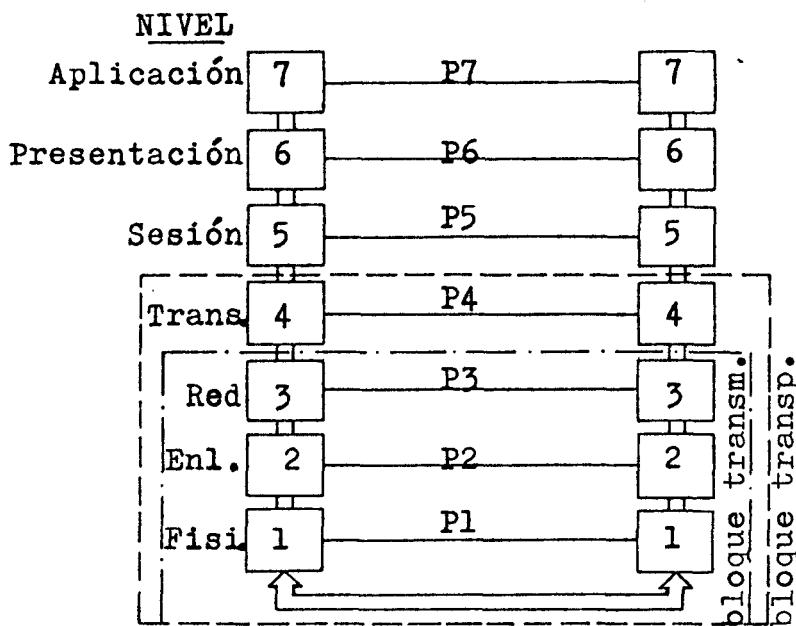


Fig. -2.22-

USUARIOS DEL BLOQUE DE TRANSPORTE

nivel 7 - Aplicación	Orientado a la aplicación
nivel 6 - Presentación	Orientados al Sistema
nivel 5 - Sesión	
BLOQUE DE TRANSPORTE	
nivel 4 - Transporte	Orientados a las comunicaciones
BLOQUE DE TRANSMISION	
nivel 3 - Red	
nivel 2 - Enlace	
nivel 1 - Físico	

Fig. -2.23-

El modelo de ISO está formado por siete niveles. Cada nivel, con la excepción del nivel físico, usa los servicios del nivel inmediatamente inferior. Los entes al alcance de ciertos niveles se definen para representar un "punto final" y son usados para proveer servicios al siguiente nivel más alto. Una "conexión" en un nivel dado es una asociación establecida para la comunicación entre dos o más entes del nivel más alto.

7.4.1. LOS USUARIOS DEL BLOQUE DE TRANSPORTE

- Nivel 7 - Aplicación

Se trata del nivel superior del modelo de referencia y en él se llevan a cabo las funciones específicas de comunicación entre los diferentes procesos de aplicación que constituyen el sistema, tales como operadores de terminal o programas de aplicación.

La comunicación entre los procesos se realiza mediante un determinado protocolo. En las especificaciones provisionales de ISO se mencionan cinco grupos de posibles protocolos todos ellos gestionados por los elementos que constituyen el nivel aplicación. Los cinco grupos de protocolos mencionados son los siguientes:

Grupo 1 - Protocolos de gestión del sistema, orientados a la realización de las funciones de gestión del propio sistema de interconexión.

Grupo 2 - Protocolos de gestión de la aplicación, orientados al control de las funciones de gestión de la ejecución de los procesos de aplicación tales como gestión de acceso a determinadas partes del sistema, resolución del interbloqueo (deadlock), contabilidad y facturación de la utilización (accounting), etc.

Grupo 3 - Protocolos del sistema para la materialización de las comunicaciones entre procesos de aplicación, como por ejemplo, acceso a ficheros, comunicación entre tareas, activación remota de procesos, activación remota del sistema.

Grupos 4 y 5 - Protocolos específicos para aplicaciones, ya sea industriales, de cálculo, de manejo de información, bancarias, líneas aéreas, etc.

- Nivel 6 - Presentación

El objetivo de los elementos situados a éste nivel es proporcionar un conjunto de servicios a los entes que constituyen el nivel superior. Dichos servicios están fundamentalmente orienta-

dos a la interpretación de la estructura de las informaciones intercambiadas por los procesos de aplicación.

Como ejemplo del tipo de funciones que es posible encomendar a los entes que constituyen este nivel, ISO menciona:

- En lo que se refiere a los protocolos de terminales virtuales:
 - La selección del tipo de terminal
 - La gestión de los formatos de presentación de los datos
- En lo que se refiere a los protocolos de manipulación de ficheros virtuales:
 - Ordenes de manejo y formateado de los ficheros:
 - Conversiones de códigos de los datos
- En lo que se refiere a la transferencia de información y a la manipulación de tareas:
 - Formateado de los datos y órdenes de control
 - Control de la forma de transferir informaciones

Realmente, las funciones asignadas a los niveles aplicación y presentación son de la misma naturaleza y en cierto modo complementarias. Podría decirse que la diferencia entre dichas funciones es similar a la que existe entre significado y presentación de la información, entre semántica y sintaxis de los datos que constituyen la comunicación entre procesos de aplicación.

En el nivel presentación se han concentrado, pues, todas aquellas funciones necesarias para permitir una heterogeneidad en la forma en que intercambian información los procesos de aplicación que dialogan (en el caso de que exista dicha heterogeneidad). Realmente el nivel presentación contribuye a asegurar el carácter abierto del sistema.

- Nivel 5 - Sesión

El objetivo de los elementos situados en este nivel es proporcionar un soporte a la comunicación entre los entes del nivel presentación.

El nivel sesión proporciona dos clases de funciones:

- 1) La unión y desunión de dos entes del nivel presentación para el intercambio de información, y
- 2) Controlar el intercambio de datos, delimitar y sincronizar las operaciones entre los dos entes del nivel presentación.

De este modo, este nivel proporciona la iniciación y finalización de la sesión, delimitación de los datos, y control del diálogo.

Cada vez que se desea establecer una comunicación entre dos elementos de sistemas distintos, se establece una sesión entre los correspondientes entes de presentación afectados. La sesión regula el diálogo entre ellos y deja de existir cuando éste finaliza.

Al igual que en el nivel presentación, también aquí pueden existir tantos entes como sea necesario, uno por cada uno de los del nivel superior. Cada ente del nivel sesión se identificará mediante una dirección, asociada a un elemento capaz de almacenar la información que se intercambia (latch).

Así pues, en el establecimiento de una sesión intervienen dos etapas bien diferenciadas:

- Orden de establecimiento de la sesión dirigida a un "latch" específico situado en el sistema informático.
- Una vez establecida la sesión se procede al intercambio tanto de datos como de información de control.

Tenemos que añadir que una sesión puede establecerse bien para permitir una comunicación bidireccional, bien únicamente unidireccional.

7.4.2. EL BLOQUE DE TRANSPORTE

El objetivo del bloque de transporte es hacer posible el establecimiento de sesiones entre sistemas distintos, eso es, como su nombre indica transportar la información a través del mecanismo de comunicación e interconexión.

- Nivel 4 - Transporte

El objetivo de los elementos que componen este nivel consiste en proporcionar un servicio de transporte de la información a través del sistema. Este servicio deberá ser transparente para los usuarios (elementos del nivel sesión) liberándolos de ese modo de todo lo referente a la forma de llevar a cabo dicho transporte.

El nivel transporte proporciona fundamentalmente tres tipos de servicios:

- Servicios orientados hacia el establecimiento de una conexión
- Servicios orientados hacia la realización de transacciones

- Servicios orientados hacia la difusión de información a múltiples destinatarios.

Una de las razones que justifican la existencia de este nivel es la optimización de los recursos de comunicaciones con objeto de minimizar el coste de dichos intercambios de información.

A los entes de este nivel se les denominan estaciones de transporte o puntos finales del bloque de transporte.

Las operaciones de intercambio de información entre estaciones de transporte se realizan mediante protocolos denominados de transporte entre puntos finales (end-to-end transport protocols) actualmente en fase de discusión.

7.4.3. EL BLOQUE DE TRANSMISION

- Nivel 3 - Red

El objetivo de este nivel es proporcionar los elementos necesarios para intercambiar información entre los entes del nivel transporte a través de una red de transmisión de datos.

Las funciones asignadas a los entes del nivel red cobran pleno sentido cuando en la comunicación se utiliza una red de transmisión de datos.

Para los intercambios de información con las redes públicas de paquetes, el CCITT ha definido un protocolo de red dentro de la Recomendación X.25 del que nos ocuparemos más adelante y cuyo análisis constituye una buena ilustración de las funciones asignadas a éste nivel.

- Nivel 2 - Enlace

El objetivo de este nivel es proporcionar los elementos necesarios para establecer, mantener y terminar interconexiones de enlace de datos entre entes del nivel red.

Un enlace de datos se establece siempre entre dos puntos físicos de conexión del sistema. En todos los casos se considera que un enlace es siempre bidireccional.

Existen en la práctica diferentes tipos de protocolos de enlace utilizados en el intercambio de información entre sistemas informáticos. Al igual que en el nivel anterior, el CCITT ha seleccionado uno de ellos como protocolo de enlace de la Recomendación X.25.

- Nivel 1 - Físico

En este nivel se definen y materializan las características mecánicas, eléctricas, funcionales y de procedimiento para establecer, mantener y terminar la interconexión física entre un equipo terminal de datos (DTE) y un equipo terminal del circuito

de datos (DCE). Las Recomendaciones seleccionadas por el CCITT para este nivel son la V.24 para redes telefónicas y la X.21 para redes de datos..

7.5. LA RECOMENDACION X.25

El intercambio de información entre un Equipo Terminal de Datos, funcionando en el modo de transmisión de paquetes, y el Equipo Terminal del Circuito de Datos de una red pública de conmutación de paquetes está regulado por la Recomendación X.25 promulgada por el CCITT.

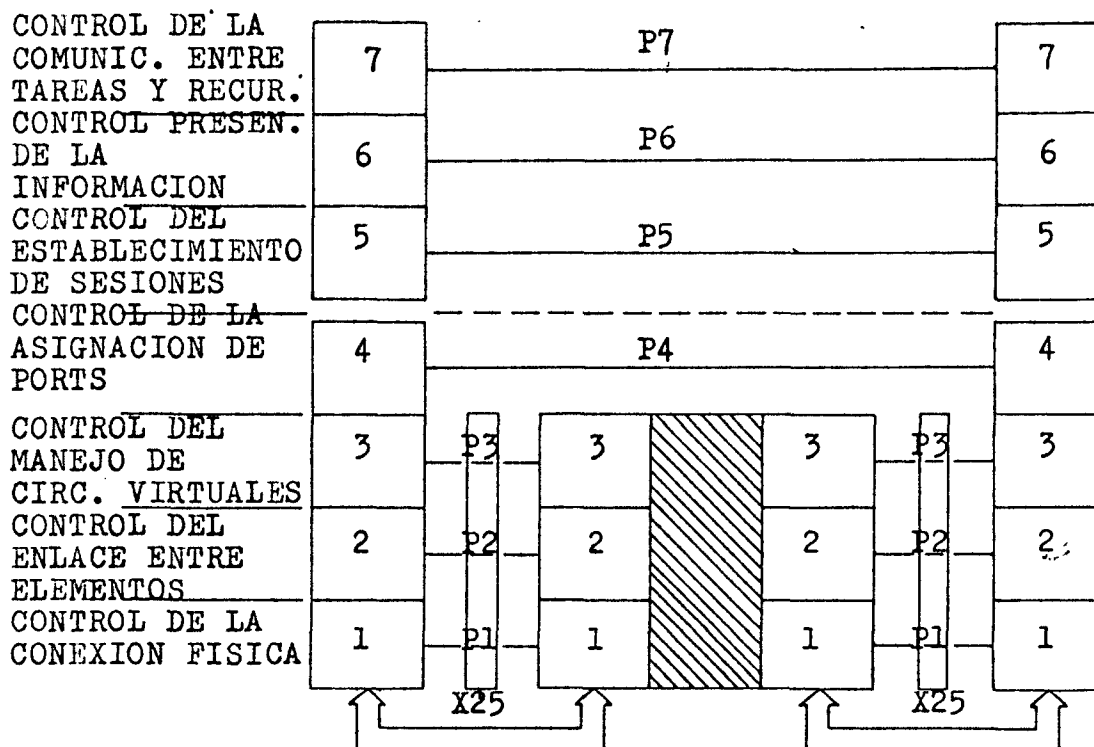


Fig. -2.24-

En la norma X.25 se definen pues las características físicas y funcionales de los elementos que constituyen los niveles 1, 2 y 3 de la interconexión que hemos definido en el apartado anterior.

En este apartado solamente pretendemos hacer una descripción somera de dicha norma.

- X.25 Nivel 1 - Físico

En este nivel se definen las características físicas, eléctricas, funcionales y de procedimiento para establecer, mantener y desconectar el enlace físico entre el DTE y el DCE.

Los elementos que constituyen este nivel están contemplados en la norma X.21. Esta recomendación define el interface de aplicación general entre el DTE y DCE para funcionamiento síncrono en redes públicas de datos.

La definición de los circuitos de enlace entre un DTE y un DCE corresponden a la recomendación X.24 y la definición de las

características eléctricas de dichos circuitos corresponde a las recomendaciones X.26 o X.27 en función del modo eléctrico de transmisión, lo que condiciona la velocidad máxima de transmisión serie permitida; estas recomendaciones son, respectivamente, equivalentes a las RS-422 y RS-423 de la EIA, y a la V.10 y V.11 del propio CCITT.

Paralelamente a esto, está contemplada dentro de X.25 la posibilidad de interconexión conforme a la recomendación X.21-bis que define la utilización en las redes públicas de datos de equipos terminales de datos DTE concebidos para su conexión con modems síncronos de la serie V.

En esta norma X.21-bis, las características eléctricas de los circuitos de enlace de lado del DCE se ajustan a la recomendación V.28 y la asignación de patillas en el conector a la norma ISO-DIS 2110-2; asimismo, del lado del DTE, dichas características podrán ajustarse, bien a la recomendación V.28, bien a la recomendación X.26 como en el caso anterior.

Las definiciones de los circuitos de enlace utilizados en este caso, se ajustan a la recomendación V.24. Estas dos recomendaciones V.24 y V.28 son en gran parte equivalentes a la conocida RS-232-C de la EIA.

En cualquier caso, de una forma u otra de las previstas por el CCITT, el objetivo de este nivel consistirá en establecer, mantener y desconectar el enlace físico entre un DTE y un DCE para de esa forma permitir el acceso a la red pública de transporte de paquetes; y por parte del usuario, el problema continuará consistiendo en conectar un equipo facilitado por el fabricante con un modem suministrado por la compañía telefónica, nada más.

- X.25 Nivel 2 - Enlace

En este nivel se define el procedimiento de acceso al enlace para el intercambio de datos por el enlace entre el DTE y el DCE.

Este procedimiento utiliza el principio y la terminología del "procedimiento de control para enlaces de datos de alto nivel" (HDCL) especificado por ISO.

Hay que decir que, en realidad, el procedimiento utilizado en este nivel 2 corresponde a un subconjunto de las posibilidades del procedimiento HDCL correspondiente al modo de funcionamiento denominado equilibrado o LAPB (balanced-modelink-acces-procedure) a la que se llegó tras introducir modificaciones en las propuestas iniciales.

Brevemente diremos que el procedimiento HDCL permite intercambiar paquetes entre un DTE y un DCE, corregir los errores que hubieran podido detectarse en la retransmisión de los paquetes afectados, controlar el flujo de paquetes en el enlace DTE-DCE, así como confirmar la recepción correcta de los paquetes transmitidos.

-X.25 Nivel 3 - Red

En este nivel se definen los procedimientos de control para el intercambio de paquetes que contengan información de control y datos del usuario entre el DTE y el DCE.

El intercambio de paquetes entre el DTE y el DCE se realiza a través de lo que se ha denominado un canal lógico, de forma que entre un DTE y un DCE pueden existir uno o más canales lógicos con la posibilidad de ser utilizados independientemente unos de otros.

Un intercambio de información entre dos DTE implica la realización de una llamada virtual entre ambos, o lo que es lo mismo, el establecimiento de un circuito virtual entre los dos DTE implicados. Es evidente, pues, que la creación de un circuito virtual exige la utilización de dos canales lógicos, uno por el que accede a la red el DTE que ha originado la llamada y otro por el que recibe la llamada el DTE destinatario.

Así pues, la competencia del nivel 3 quedará restringida a las operaciones de establecimiento, mantenimiento y desconexión de circuitos virtuales. La realización de estas operaciones está regulada en el denominado protocolo de nivel 3.

8. ARQUITECTURAS DE LAS REDES MAS IMPORTANTES

Después de analizar rápidamente los conceptos en diseño de protocolos, y el modelo de referencia de ISO, vamos a tratar de explicar de manera rápida y sencilla las arquitecturas (y protocolos) de las redes más importantes.

8.1. ARQUITECTURA SNA

El SNA (System Network Architecture) como arquitectura, identifica y define los posibles elementos dialogantes de una red y describe los protocolos que deben regir su diálogo.

Esta arquitectura es absolutamente abstracta -es decir, independiente de productos y arquitecturas hard/software-, y abierta, en continua evolución, a fin de adaptarse a lo que se va anticipando como factible y necesario dentro del binomio formado por el estado de la tecnología y las necesidades del usuario.

8.1.1. TOPOLOGIA DE UNA RED SNA

8.1.1.1. COMPONENTES BASICOS

La topología básica de una red SNA consta de dos niveles de red. Un primer nivel, que llamaremos dominio, consiste en un conjunto de nodos tributarios (terminales, clusters) de un nodo principal (host processor), al que están conectados por medio de diversos sistemas de comunicaciones (enlaces TP o conexión a canal) y a través de nodos auxiliares (controladores de comunicaciones locales o remotos). Este sistema dispone en sus nodos principal y tributarios de puertas de acceso para lo que se llama, según la nomenclatura SNA, Usuarios Finales (end users: EU), es decir, programas de aplicación u operadores de terminales. Tales puertas de acceso reciben el nombre de Unidades lógicas (logical unit: LU) y constituyen el elemento básico dialogante de la red.

Las LU vienen a ser una especie de nodos lógicos, que pueden estar agrupados dentro de los productos hardware y software -nodos físicos-, que constituyen la red.

Cada uno de tales nodos físicos dispone de un nodo adicional de control, con capacidad de gestionar sus propios recursos (tales como línea/s de comunicaciones, por ejemplo), ayudar a sus LU en el establecimiento de nuevas sesiones y restaurarlas en caso de que sea necesario. Se le denomina Unidad Física (Physical Unit: PU) y se diferencia fundamentalmente de la LU por la inexistencia de interface con el usuario final.

Todos los nodos lógicos (PU y LU) de un dominio están controlados por un único nodo lógico de control, denominado "System Services Control Point" (SSCP) residente en el nodo físico principal del mismo ("host"). Este control lo ejerce mediante una serie de comandos arquitecturados a tal fin, y que el SSCP intercambia con los PU y LU mediante sendas sesiones, que a tal fin mantiene con ellos de modo continuo, y que son prerequisites para el establecimiento de sesiones entre LU.

De esta forma vemos que un nodo físico de la red puede llegar a contener hasta tres tipos distintos de nodos lógicos (fig. 2.25):

- PU: uno por cada nodo físico
- SSCP: uno en el nodo principal de cada dominio
- LU: un número variable en cada nodo físico con interface a usuarios finales

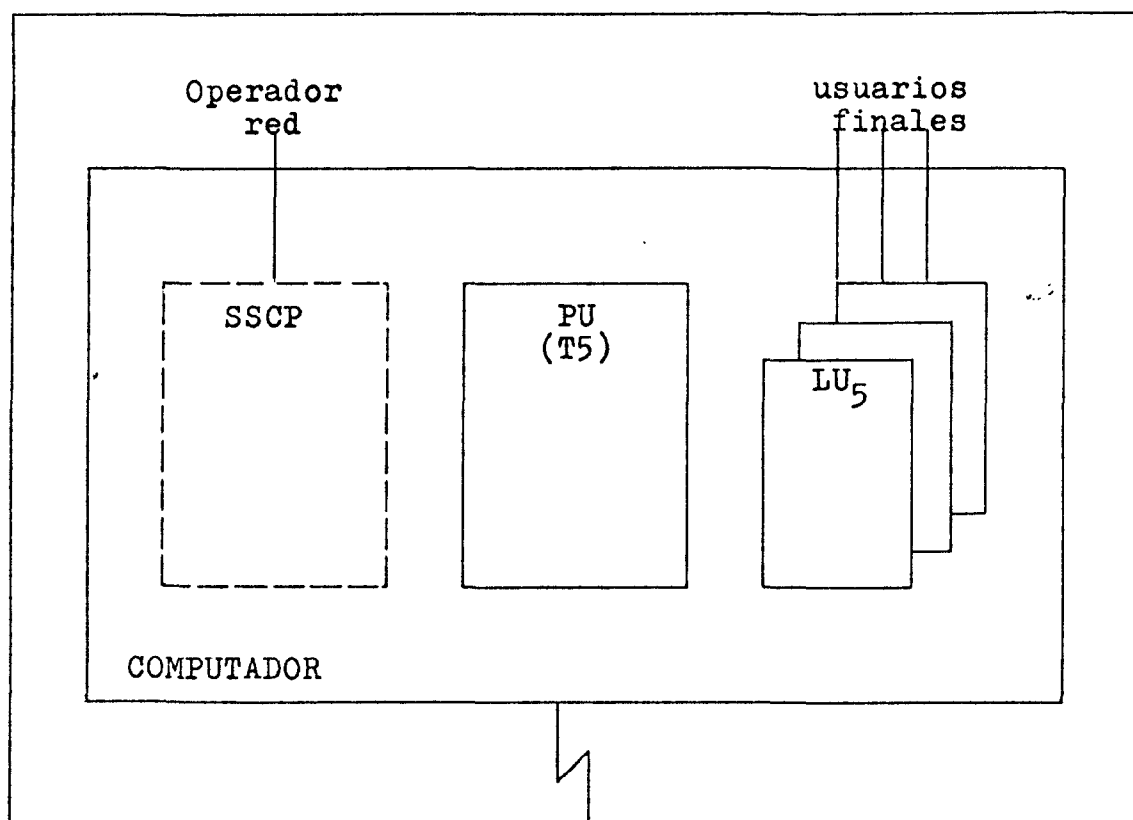


Fig. -2.25-

Las distintas agrupaciones de estos dan lugar a los siguientes tipos de nodos físicos, según el actual estado de definición (fig. 2.26):

- a) Host (PU tipo 5): contiene un SSCP, una PU y un número variable de LU, generalmente asociados a otros tantos programas de aplicación.
- b) Controlador de comunicaciones (PU tipo 4): constituye un nodo intermedio de la red con misión principal de control simultáneo de múltiples recursos de comunicaciones. Para ello sólo requiere una PU.

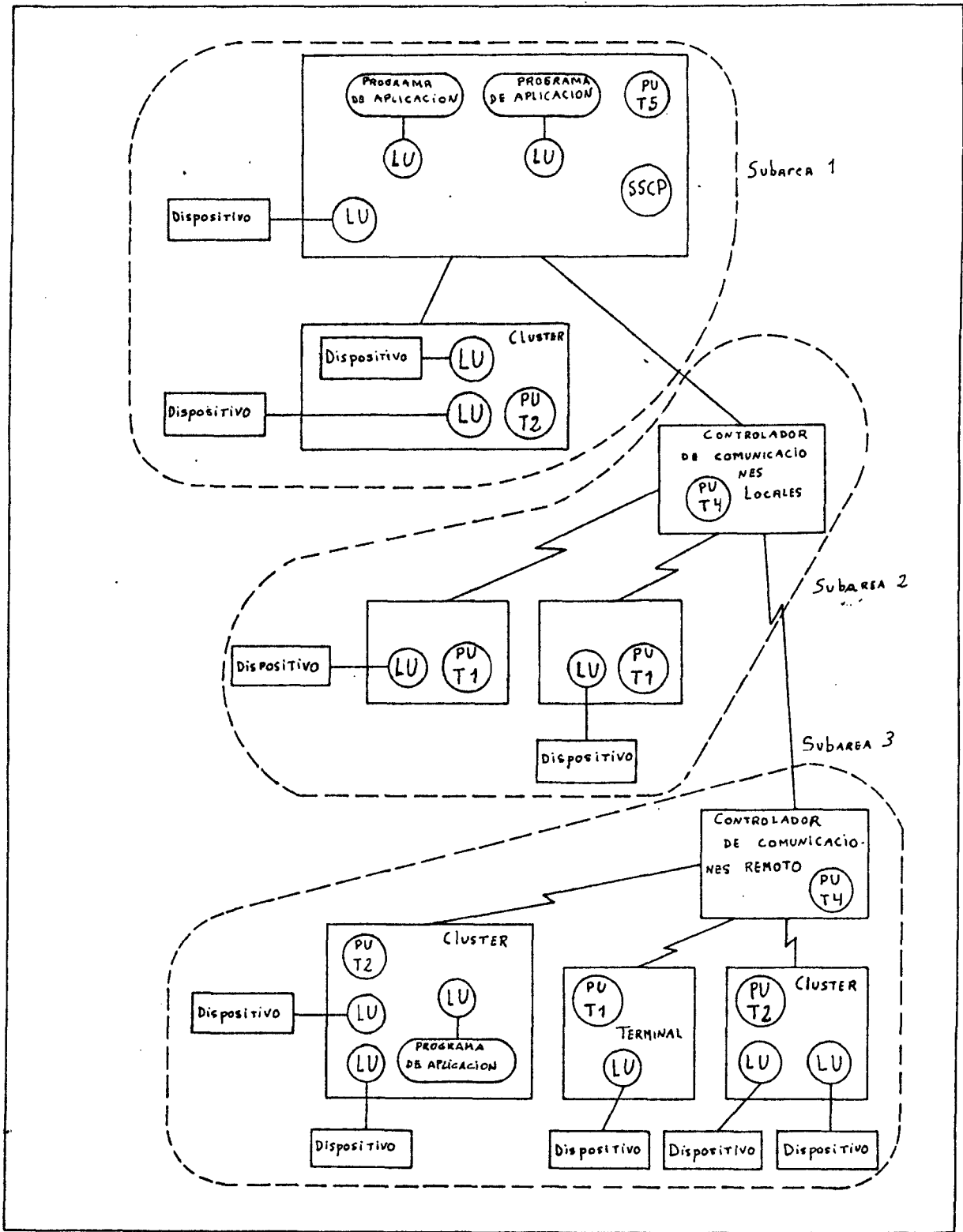


Fig. -2.26-

- c) Cluster de terminales (PU tipo 2): contiene una única PU y una LU para cada usuario final que lo comparte. Si el cluster es de los que suelen llamarse "inteligentes", estos usuarios finales pueden ser programas de aplicación que, a su vez, pueden controlar dispositivos (discos, teclados, etc.), de forma externa a la arquitectura.
- d) Terminal (PU tipo 1): generalmente asimilado a un terminal no inteligente monoestación. En tal caso contiene una PU y una LU.

8.1.1.2. DIRECCIONAMIENTO

Tales nodos físicos tienen su correspondiente sistema de direccionamiento, utilizado únicamente para la transferencia de datos entre cada dos nodos adyacentes, según el protocolo de comunicaciones aplicable a cada caso -conexión a canal en máquinas locales o SDLC para enlaces remotos-.

Desde el punto de vista del "routing" de datos, los algoritmos del SNA tienen en cuenta otro tipo de direccionamiento, a nivel de nodo lógico, es decir, SSCP, PU y LU, estando provisto cada uno de ellos de una dirección única dentro de la red. Por esto, en la nomenclatura SNA se les reconoce de forma global como "unidades direccionables de la red" (Network Addressable Unit: NAU).

Toda PU tipo 4 o 5 puede tener conectadas directamente PU del tipo 1 o 2, que, de hecho, constituyen los elementos terminales de la red. El mecanismo de direccionamiento consiste en la división de un dominio en subáreas (fig. 2.26), coincidentes con todas y cada una de las PU tipo 4 y 5, identificadas con una dirección de 1 a 8 bits dentro de la red y asignando a cada una de las PU tipo 1 y 2 que les tributan y a sus correspondientes LU, una dirección de elemento dentro de su propia subárea, de 8 a 15 bits. Con todo ello, cualquier NAU dentro de una red está identificada por una dirección única del tipo subárea-elemento de 16 bits de longitud en total.

Esta dirección es utilizada para identificar el origen y el destinatario de cada elemento de información intercambiada entre dos interlocutores de una sesión, y, las de ambos combinadas, para identificar un elemento de información como perteneciente a una sesión determinada.

8.1.1.3. SISTEMAS MULTIDOMINIO - NETWORKING

Hemos visto como se constituye una estructura jerárquica controlada por un único SSCP, generalmente coincidente con un

único "host". Un número determinado de estos dominios puede interconectarse entre sí por medio de sus PU tipo 4, de forma semejante a la representada en el ejemplo de la figura 2.27. Cada uno de los SSCP así interconectados aporta el conjunto de las NAU que controla a la red global que pasan a constituir.

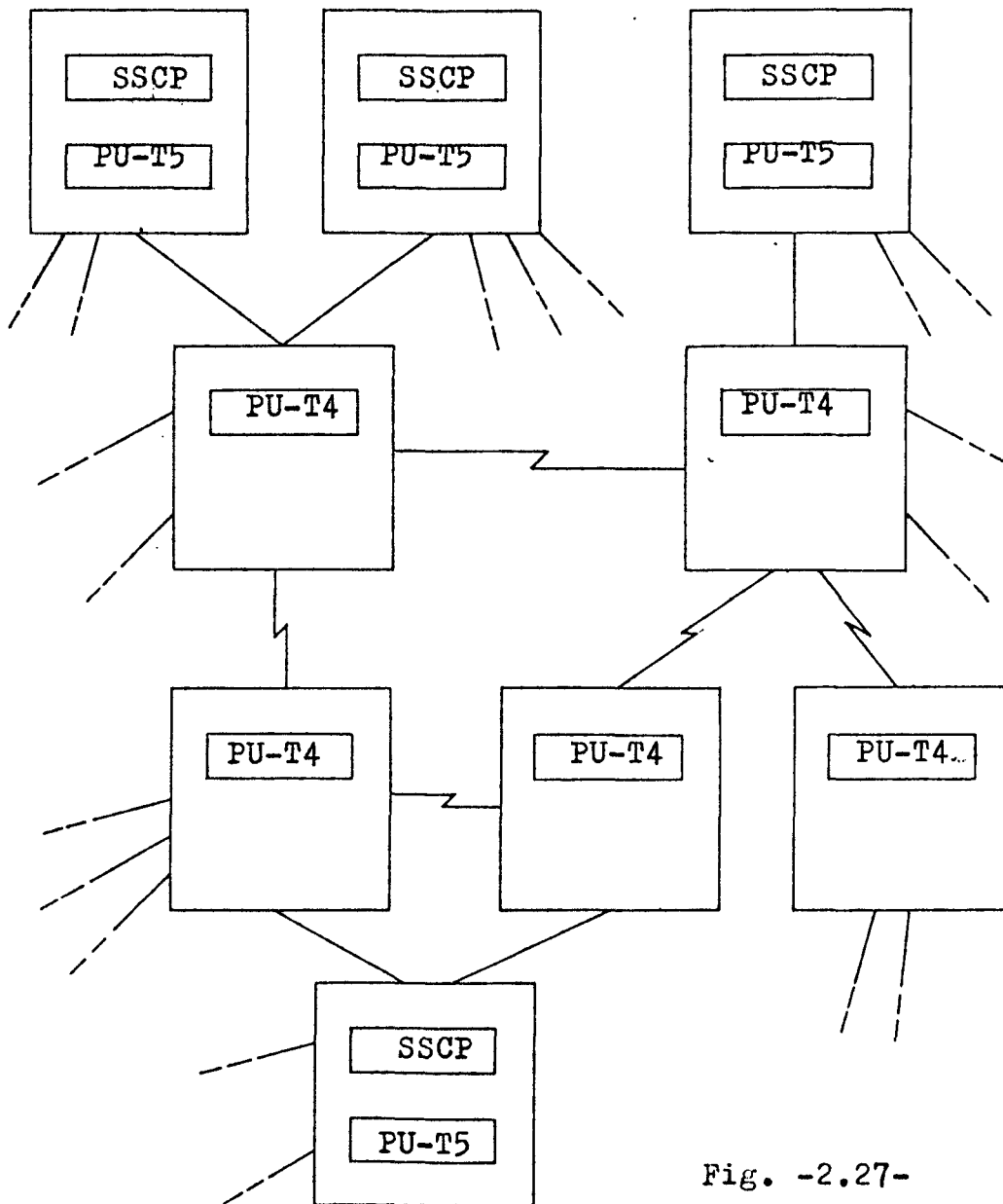


Fig. -2.27-

De esta forma es posible establecer sesión entre LU pertenecientes a dominios distintos, siempre que exista previamente sesión entre los correspondiente SSCP.

8.1.1.4. RED COMUN

Todas las NAU de un nodo físico comparten una función común de "routing" que podríamos denominar control de ruta (traducción aproximada de Path Control: PC). Cuando esta función recibe una unidad de información, basándose en su origen/destino, localiza el camino más inmediato para transferirla a su origen final. Por tal motivo, el PC hace interface directamente con los distintos enlaces y sus modalidades de control de transmisión, con lo cual,

observando una red SNA a nivel más detallado, tal como la representada en la figura 2.28, veríamos que el SNA nos presenta una estructura cortical con dos niveles claramente diferenciados.

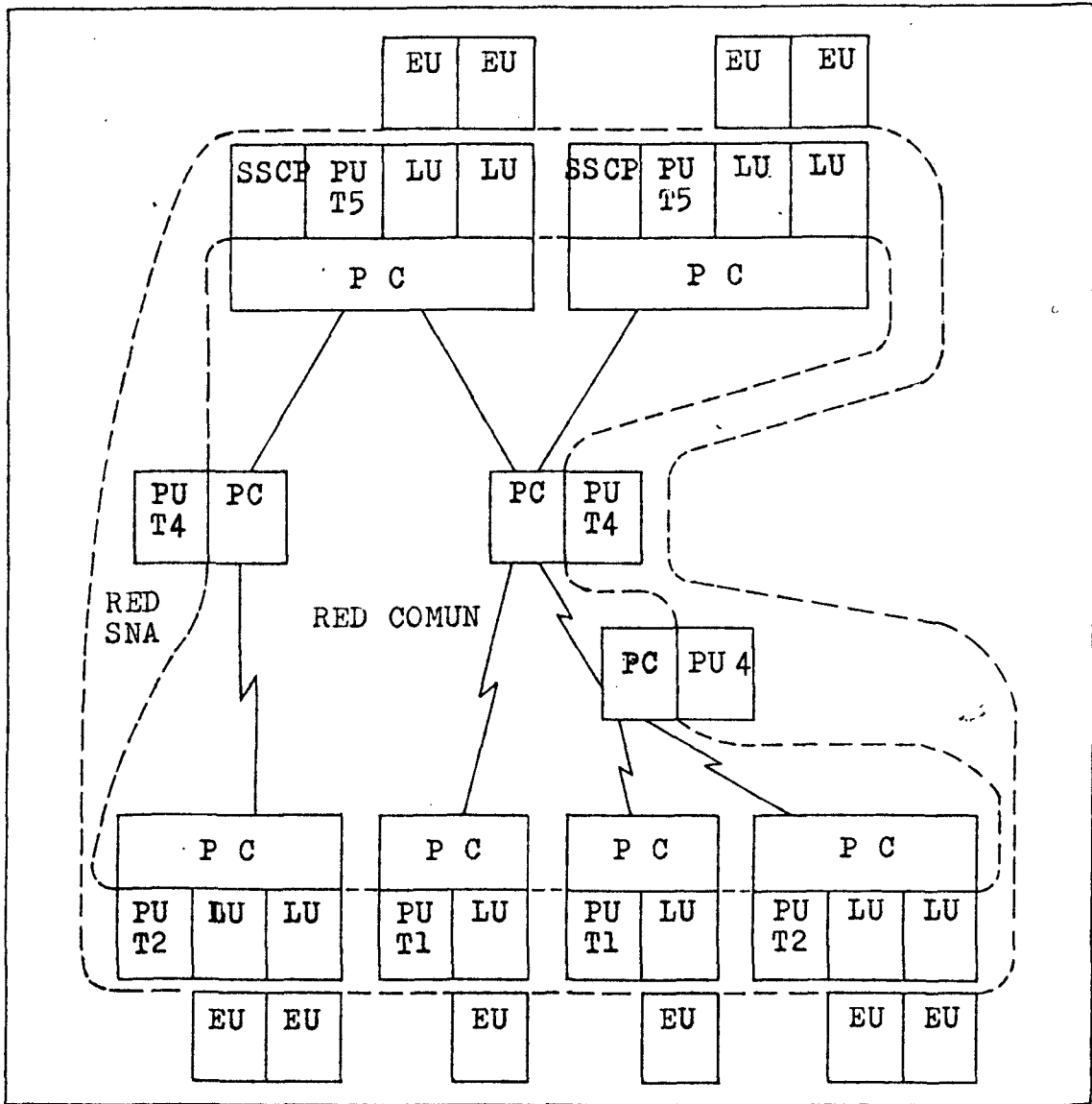


Fig. -2.28-

El más externo es aquél al cual hacen interface los distintos usuarios finales de la red. El interno, resultante de la interconexión, por distintos procedimientos, y con una cierta estructura predefinida, de todos los PC de los distintos nodos físicos, constituye lo que se denominará red común.

A la red común se conectan todas las NAU que integran la red, con lo cual inmediatamente se aprecia que tal red común constituye una especie de red de transporte de la información perteneciente a las distintas sesiones establecidas entre las NAU. Adicionalmente ejerce misión de capa homogeneizadora de interface entre las NAU, absolutamente estandarizadas, y los distintos tipos de enlace que puede existir entre los nodos físicos. De esta forma, las gestiones distintas de enlace y limitaciones físicas propias de la realización de los nodos y calidad de las líneas (tamaño de buffers, bloque óptimo de transmisión, etc.), se resuelven de modo local entre cada par de PC adyacentes, y se mantienen transparentes para la NAU, especializados en su problemática de protocolos "end-to-end" y en el tratamiento de los datos transportados por la red común.

8.2. ARQUITECTURA DECNET

DECNET es una familia de productos de software que permiten a dos o más ordenadores formar una red. A esta red pueden enlazarse ordenadores de la misma o diferente arquitectura (16 bits y 32 bits) y con el mismo o con diferentes sistemas operativos.

En la figura 2.29 representamos una red de 6 nudos, teniendo cada uno de ellos un sistema operativo diferente (representado por XXX en DECNET-XXX) y, en el caso de DECNET-VAX y DECNET-20, diferente arquitectura hardware de ordenador, respecto a lo restante. El DECNET en cada nudo actúa como un interface entre el sistema operativo del nudo y la red (en cada caso, se formatea la información de acuerdo con unas reglas comunes dadas por DECNET (fig. 2.30). De este modo, en cada nudo, la correspondiente versión del DECNET es capaz de reconocer los formatos de los datos que viajan por la red y convertir en formatos reconocibles por su propio sistema operativo.

Una característica general del proceso distribuido es que se usa del orden del 80% de los recursos de ordenador para proceso de aplicaciones locales, y sólo del orden del 20% se consume en trabajo relacionado directamente con funciones de red. Por ello, dentro de la red cada nudo debe tener el sistema adecuado para las necesidades locales. En caso contrario, se desperdiciará una cantidad considerable de recursos de ordenador.

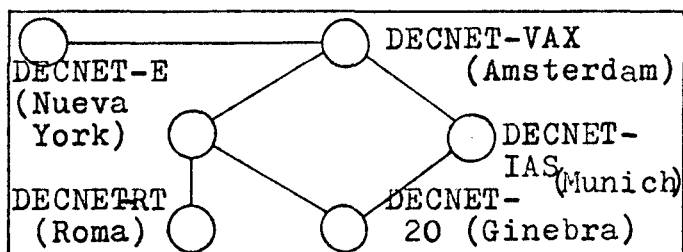


Fig. -2.29-

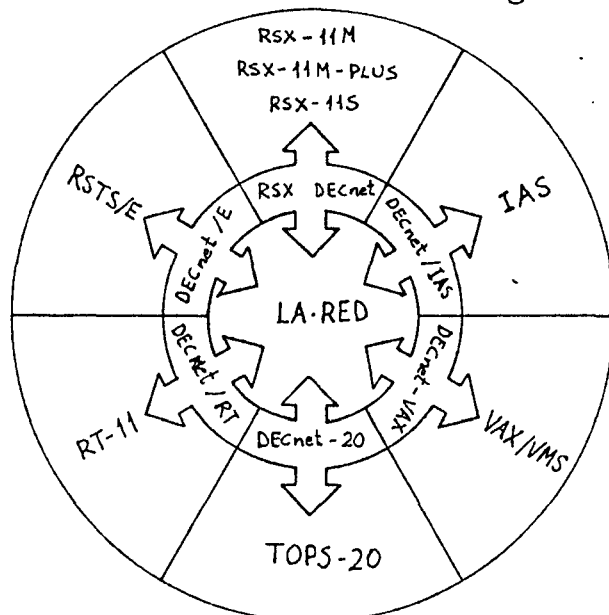


Fig. -2.30-

Pensando en esto, DECNET aporta una gran flexibilidad al permitir configurar en diferentes nodos ordenadores de diferentes arquitecturas con diferentes sistemas operativos diseñados para adaptarse en el mayor grado posible a las posibles aplicaciones, en cada caso.

Entre las funciones de red DECNET más importantes destacaremos las siguientes:

- Comunicaciones entre tareas. Intercambio de datos entre dos programas, en el mismo o en diferentes nodos, a través de un camino lógico establecido entre ambos en base a consentimiento mutuo (establecimiento de sesiones)
- Acceso a recursos remotos. Tanto programas como terminales de usuarios pueden acceder a ficheros y/o dispositivos en nodos remotos. Entre las operaciones que se pueden realizar están las siguientes:
 - a) Transferencia de ficheros entre dos nodos
 - b) Manipulación de ficheros residentes en nodos

remotos (por ejemplo: abrir, borrar o añadir datos a ficheros remotos).

- c) Utilización compartida de periféricos conectados a cualquier otro nudo remoto de la red.
- d) Envío de ficheros conteniendo comandos a sistemas operativos remotos (por ejemplo: procesamientos secuenciales desatendidos) con ejecución, a su recepción, en el nudo remoto.

- Comunicación entre terminales de la red. Intercambio de mensajes entre terminales de usuario mediante una rutina de utilidad DECNET.
- Terminales remotos. Un terminal local de un nudo puede ser usado como local de un nudo remoto, estableciendo una conexión lógica entre este terminal y el nudo remoto, ejecutando este nudo remoto los comandos que se tecleen en dicho terminal.
- "Management" de la red. DECNET proporciona las herramientas adecuadas para planificar una generación, generar, definir, monitorizar y controlar nudos de la red.
- Carga a través de la línea (down-line loading). En nudos satélites, con sistema operativo "solo memoria" es posible cargar a éstos, tanto el sistema operativo como las aplicaciones de forma remota, a través de la línea, desde otro nudo con recursos de almacenamiento propios, produciendo el arranque automático de las aplicaciones en el nudo satélite. Asimismo, es posible enviar volcados de memoria a través de la línea, desde estos nudos satélites, a otros con capacidad de almacenamiento y análisis de estos volcados.
- Tests de Loopback. Son herramientas incluidas en DECNET para probar hardware y software de comunicaciones y aislar problemas de red.

8.2.1. ARQUITECTURA DE REDES DIGITALES

El diseño de todo el software DECNET está basado en la arquitectura de redes DIGITAL (Digital Network Architecture), DNA, la cual es una estructura lógica, que proporciona un modelo para todas las realizaciones DECNET.

Consta de una serie de estratos, cada uno de los cuales define un conjunto diferente de funciones de red y un conjunto de reglas para efectuar esas funciones. La estructura DNA establece la forma en que estas funciones se relacionan entre sí.

8.2.1.1. ESTRATOS DNA

La figura 2.31 muestra los estratos funcionales de DNA (digital network architecture).

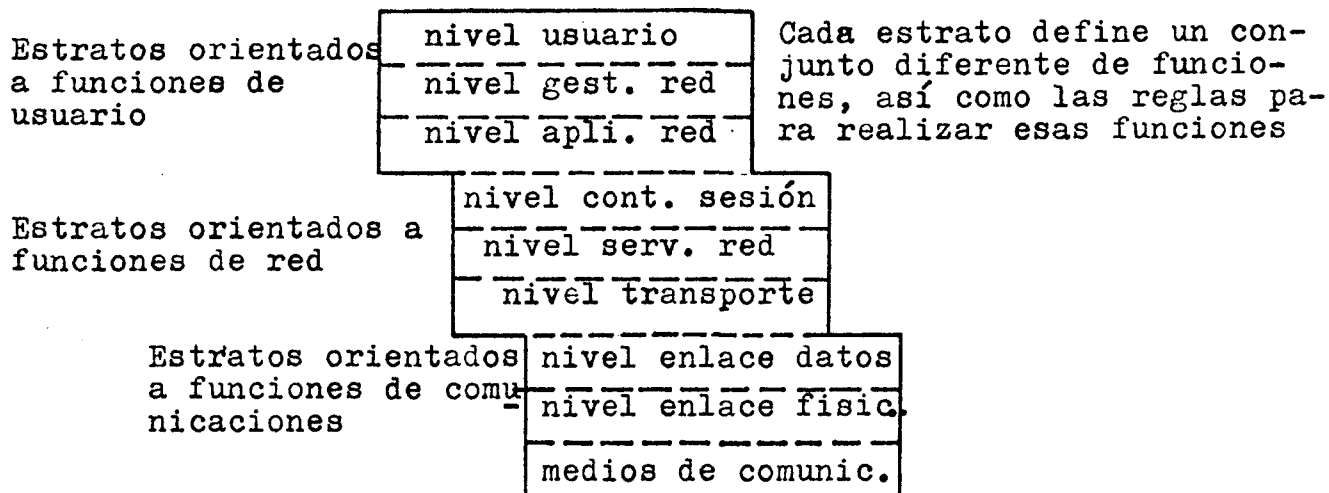


Fig. -2.31-

A continuación describiremos la función de cada LAYER (estrato).

- Nivel usuario (user layer). Conjunta los programas escritos por el usuario con los servicios de acceso a la red. Es el estrato de más alto nivel en la red.
- Nivel gestión de la red (network management layer). Define las funciones usadas por operadores y programas, para planificar, controlar y mantener la operación de redes DECNET.
- Nivel aplicación de la red (network application layer). Define las funciones de red usadas por los dos estratos de más alto nivel. Las más importantes son: acceso remoto a ficheros, transferencia de ficheros y uso de terminales remotos de órdenes (terminales que, conectados a un nudo local, pueden pasar a funcionar como si estuvieran conectados localmente a un nudo remoto, pasando a través de los controles de DECNET).
- Nivel control de sesión y nivel servicios de la red (session control layer and network services layer). El conjunto de estos dos estratos define un mecanismo el cual permite que se comuniquen entre sí programas situados en el mismo o en diferentes nudos de la red (sin importar la situación de estos nudos en la red), en base a un consentimiento mutuo.

Entre ambos programas, independientemente del número de nudos que los separen, se crea un camino lógico (Logical Link) que será la vía lógica de intercambio de información entre esos dos programas.

- Nivel transporte (transport layer). El estrato de transporte define los mecanismos utilizados para transportar una unidad de datos de un nudo a otro, cualquiera que sea su posición en la red.
- Nivel enlace de datos (data link layer). El estrato de enlace de datos define los mecanismos que se utilizan para establecer la comunicación entre dos nudos adyacentes, libre de errores. Este estrato es independiente de las características físicas de los dispositivos de comunicaciones.
- Nivel enlace físico (physical link layer). El estrato de enlaces físicos conjunta el manejador software (driver) de cada dispositivo de comunicaciones, con el propio hardware de comunicaciones. El hardware incluye interfaces, modems y líneas de comunicaciones.

9.1 COMUNICACIONE SERIE FRENTE A COMUNICACION EN PARALELO

Las transferencias de datos externas que se producen entre dispositivos que no estan en el mismo chasis que la CPU, o no estan próximos a ésta, y la propia CPU pueden ser, o son normalmente, de dos formas: serie ó paralelo.

Veamos ahora las diferencias relativas de estos dos tipos de transferencias en base a los siguientes criterios:

DISTANCIA La distancia de una transferencia de datos paralelo normalmente es menor de 100 pies, mientras que las distancias que recorren las transferencias series pueden ir desde unos pocos metros hasta miles de kilómetros.

VELOCIDAD Las velocidades típicas de las transferencias paralelo varían desde 0 hasta varios millones de bits por segundo, mientras que en las transferencias series varían desde 0 hasta aproximadamente 2 millones de bps. Esto se debe, por supuesto, a que en la transmisión paralelo se transmite un mayor número de bits simultáneamente. En ambos casos, la velocidad de la transferencia es inversamente proporcional a la distancia a recorrer. Esto se muestra en la figura 2.32.

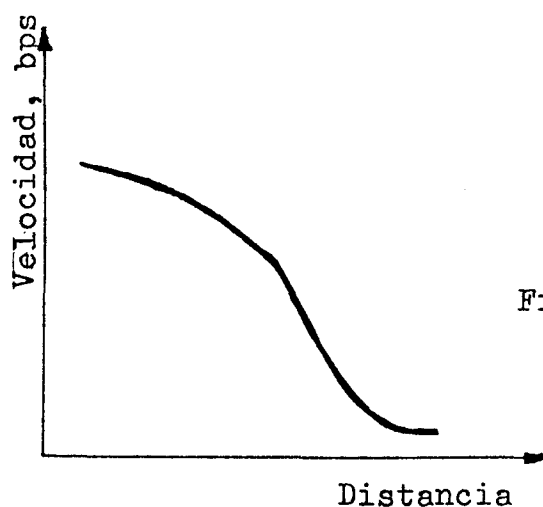


Fig. -2.32-

NIVELES DE SEÑAL Las "interfaces" paralelas normalmente emplean señales o niveles TTL (0-5 V). Las señales TTL también se usan en interfaces series. Sin embargo, es mucho más común encontrar dispositivos operando a niveles de señal RS-232C (+12,-12V), o a niveles del Lazo de corriente de 20 mA.

PERDIDA DE SEÑAL Y AMPLIFICACION Como para transmitir las señales tenemos que usar cables, siempre tendremos problemas de pérdida de señal en la recepción. Estas pérdidas estan

relacionadas directamente con la longitud del cable. Para com pensar éstas pérdidas, son necesarios transmisores de gran po tencia, o amplificadores de señal. La amplificación de una se ñal serie es significativamente menos compleja que la amplifi cación de señales paralelo. Los problemas de fases y tiempos en la amplificación de muchas señales paralelo pueden llegar a ser un significativo factor de costo.

COSTES Sobre distancias por encima de los 15 metros el coste de líneas de datos múltiples llega a ser prohibitivo. El "logical" necesario para transformar los datos de paralelo a serie en la transmisión, y viceversa en la recepción, es me nos caro que el hardware necesario para efectuar la transfere ncia paralelo sobre largas distancias.

TELECOMUNICACIONES La red telefónica es el medio más co munmente usado para comunicaciones sobre largas distancias. La información digital debe ser, por tanto, serializada y con vertida a una señal de audio.

Después de comparar la transmisión de datos paralelo con la transmisión de datos serie podemos concluir que ambas son rentables según sea el tipo de transmisión. Para transferencias de datos a alta velocidad y sobre pequeñas distancias es preferible la transmisión de datos en paralelo. Pero para comuni caciones a baja velocidad y sobre largas distancias, la trans misión serie es a menudo la única alternativa.

9.2 TRANSMISION SERIE

Hay dos problemas principales que deben ser resueltos por los dispositivos que desean comunicarse sobre un enlace serie. El primer problema es la serialización de los datos paralelos en la transmisión y viceversa en la recepción de los datos. El segundo problema es la sincronización mutua. El transmisor y el receptor deben coordinar sus operaciones de forma que los bits enviados sean muestreados a la misma frecuencia con que fueron enviados.

9.2.1 CONVERSION PARALELO A SERIE

Antes de la existencia de los dispositivos de mediana y lar ga escala de integración (MSI y LSI), la implementación de un canal serie de E/S era significativamente más compleja que la de una canal paralelo.

La función o la misión de convertir los datos de paralelo a serie y viceversa era realizada por registros de desplazamiento.

Ahora hay circuitos especializados que no sólo incorporan registros de desplazamiento, sino que además poseen características tales como la conexión directa al bus del sistema y opciones programables para la velocidad, paridad, y el formato de los caracteres. Estos circuitos reciben varios nombres :

- + UART: universal asynchronous receiver-transmitter
- + USART: universal synchronous/asynchronous receiver-transmitter
- + SIO: serial input/output circuit

Todos estos dispositivos especializados usan una combinación de lógica de detección y resincronización (para reconocer el principio de la transmisión, de un carácter o de un bloque) y de registros de desplazamiento directamente conectados al bus de datos del sistema y al canal serie.

La llegada de los circuitos UARTs, USARTs, y SIOs ha reducido en gran medida el tiempo de desarrollo, y consiguientemente el costo de la implementación de un canal serie de E/S.

9.2.2 SINCRONIZACION DE LOS DISPOSITIVOS

Dos dispositivos interconectados por un enlace de comunicación están equipados sólo para discriminar entre los estados lógicos 0 y 1 en la línea. Por tanto, un importante prerrequisito para que haya una transferencia de datos útil, es establecer un esquema de codificación que permita representar información empleando para ello los niveles lógicos 0 y 1.

En aplicaciones tales como comunicaciones de datos, éste esquema representa normalmente caracteres tales como letras, números y signos de puntuación como cadenas de bits de tamaño fijo. En sistemas microcomputadores, el método o esquema más frecuentemente usado es el código ASCII.

Bien, supongamos que ambos dispositivos están preparados para trabajar con el mismo esquema de codificación de la información. Ahora bien, todavía quedan por resolver varios puntos antes de que pueda haber un intercambio útil de información:

- + Lógica positiva ó negativa: ¿Cuál de los dos niveles físicos representa un estado lógico 0, y cuál un estado lógico 1?. Para algunos enlaces de comunicación estos niveles

les físicos están representados por voltajes. Para otros, están representados por corrientes, dos frecuencias de audio, etc.

- + Orden de transmisión de los bits: ¿Qué bit del carácter se envía primero?
- + Temporización: ¿Cómo podemos saber cuando finaliza un bit y cuando empieza el siguiente? ¿Y cuando termina un carácter y comienza el siguiente?

El estudio de los niveles lógicos y físicos, y de toda la lógica asociada a ellos lo realizaremos más adelante cuando estudiemos todos los protocolos de comunicaciones series. En cuanto al orden de transmisión de los bits, en la mayoría de los casos se envía primero el bit menos significativo y a continuación todos los demás. Y de todo lo referente a la temporización hablaremos a continuación.

9.2.2.1 Temporización

La primera cuestión a resolver respecto a la temporización es fijar o establecer un convenio en cuanto a la duración de un bit simple. Por ejemplo, si dos dispositivos en comunicación establecen el convenio de que cada bit dura 1 segundo; en caso de que el dispositivo A quiera enviar un 0 durante 1 seg. un 1 durante 2 seg. y un 0 durante 3 seg., el dispositivo B podrá leer la siguiente secuencia: 111001, ya que el bit menos significativo es enviado primero. Ahora bien, en caso de que el dispositivo B entienda que un bit dura $\frac{1}{2}$ seg., leerá entonces la siguiente secuencia: 111111000011. Por tanto, ambos dispositivos deben establecer claramente cuál es la velocidad de transferencia, y además, ambos deben poseer un reloj con el que poder muestrear las secuencias de bits.

Un problema que puede ocurrir igualmente son las pequeñas diferencias que pueden existir entre el reloj de transmisión y el recepción. Debido a estas diferencias, el muestreo de los bits de la secuencia llegará a ser erróneo.

Para compensar estas incompatibilidades de los relojes se realiza una resincronización periódica. Esta resincronización se puede realizar utilizando varias técnicas dependiendo de cada protocolo de transmisión específico. Además, la resincronización es una de las principales características que diferencian a unos protocolos de otros. En las comunicaciones entre sistemas computadores se usan principalmente dos protocolos: protocolos síncronos y asíncronos. El asíncrono es, sin embargo el más usado en los sistemas microcomputadores.

9.2.2.2 Transmisión Síncrona-Transmisión Asíncrona

En las transmisiones series asíncronas, la resincronización se produce carácter a carácter, y está coordinada por el uso de bits de "start" y de "stop".

El bit de start representa una señal mediante la cual el dispositivo transmisor dice al receptor: "A éste bit le sigue un byte de datos".

Por otro lado, en la transmisión síncrona se usa una señal de reloj separada asociada a los datos. Para transmisiones locales, ésta señal de reloj puede ir sobre una línea física distinta a la de los datos. Ahora bien, para comunicaciones de larga distancia, la señal de reloj se codifica con los datos para la transmisión y se decodifica en la recepción para establecer la oportuna coordinación. Asimismo, en los protocolos síncronos la resincronización se produce bloque por bloque de datos. Un bloque de datos puede ser el equivalente a muchos bytes de datos o caracteres. De hecho, la gestión de los datos en este protocolo puede ser orientada sobre los bits o sobre los caracteres.

En ambos casos deben añadirse algunos bits de control a la secuencia de bits de los datos para poder identificar el principio y el final de los caracteres en la transmisión asíncrona y el principio y el final de los bloques de datos en el caso de transmisión síncrona. Estas diferencias se ilustran en las siguientes figuras:

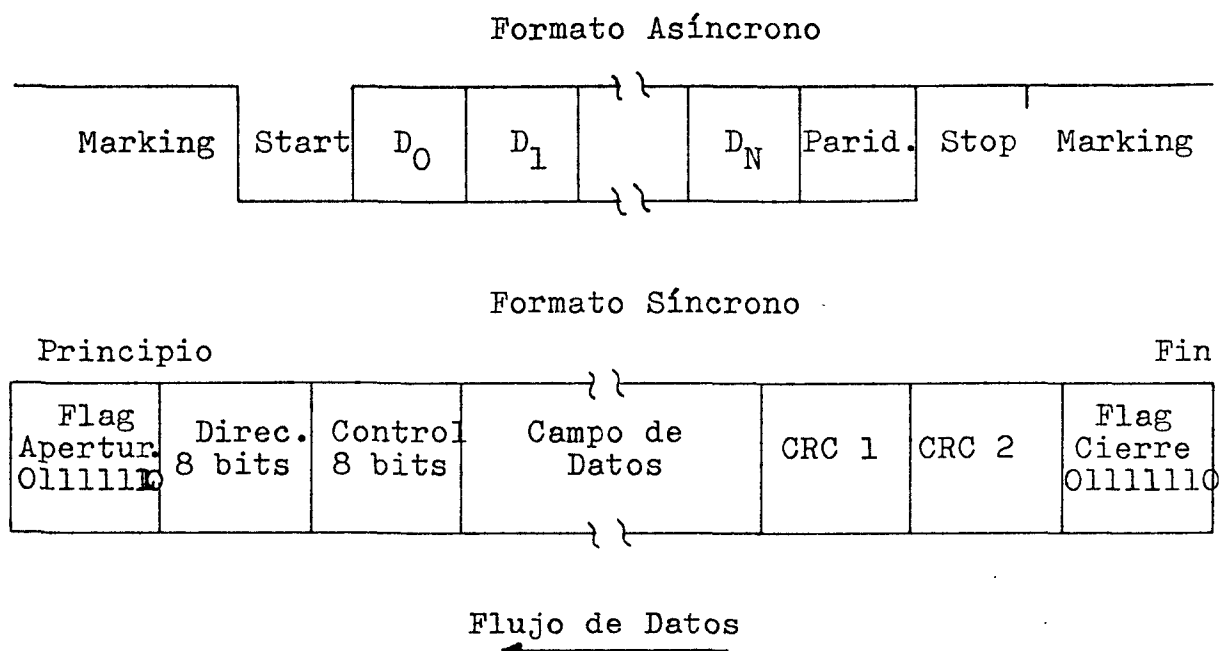


Fig. -2.33-

9.2.3 PROTOCOLO SERIE ASINCRONO

Como ya dijimos anteriormente, en la transmisión serie asíncrona los caracteres son transmitidos a intervalos irregulares y la resincronización se realiza carácter a carácter. Los bits especiales de control llamados bits de start y de stop son el mecanismo por el que se produce esta resincronización.

A continuación vamos a analizar cada uno de los bits que componen una transmisión serie asíncrona.

9.2.3.1 Bit de Start

En el protocolo serie asíncrono, cuando el canal permanece en estado de no transmisión está al nivel lógico 1, ó como se suele llamar en estado "marking". Cuando un transmisor quiere enviar un caracter, primero envía el bit de start colocando la línea en estado lógico 0 durante el tiempo que dura 1 bit. Un problema potencial que puede ocurrir son los ruidos que pueden producirse en la línea y que pueden hacer pensar al receptor que se trata de un bit de start. Esto puede resolverse si el receptor muestrea el canal en reposo 2, 4, 16 ó más veces durante el tiempo que dura 1 bit. (Actualmente todos los dispositivos del mercado que se encargan de la transmisión serie estan preparados para realizar ésta validación).

9.2.3.2 Bits de Datos

Una vez el receptor ha validado el bit de start, activa su registro de desplazamiento para comenzar a aceptar bits de datos desde la línea. Mientras que el número de bits de datos puede ser 5, 6, 7, ó 8 normalmente se transmiten 7 ó 8 bits. El número de caracteres posibles (ó símbolos) que pueden ser representados con 5, 6, 7, ó 8 bits son respectivamente 32, 64, 128, y 256.

Por tanto, podemos ver que pueden existir problemas usando 5 ó 6 bits ya que en el alfabeto (inglés), ya que todos los códigos usados en sistemas computadores usan el alfabeto inglés) existen 26 letras mayúsculas, 26 letras minúsculas, y un mínimo de 10 caracteres especiales que son los más comunmente usados tales como: puntos, dos puntos, punto y coma, paréntesis, etc.

De éste modo vemos que es necesario un código de 7 bits para poder trabajar como mínimo con tratamientos de textos y procesamientos de otras naturalezas.

El código ASCII de 7 bits es el código más universalmente utilizado en sistemas microcomputadores, y hasta en sistemas de otras tallas. El código ASCII que proviene de "American Standard Code for Information Interchange" sólo tiene como rival en cuanto a su extensión universal al código EBCDIC de 8 bits de IBM (extended binary-coded-decimal interchange code).

9.2.3.3 Bit de paridad

Después de enviarse los datos, normalmente se envía en las transmisiones asíncronas el bit de paridad (entiéndase después de enviarse los bits de datos que componen el carácter). Este bit de paridad es un mecanismo para implementar un esquema limitado de detección de errores. El dispositivo fuente sabe que ha enviado los datos al dispositivo destinatario, pero debería haber algún mecanismo para determinar si estos datos han sido bien recibidos por el receptor. Estos errores están producidos por ruidos e interferencias en las líneas, y constituyen uno de los más complejos y a la par interesantes problemas a resolver en el mundo de las comunicaciones de datos.

Pueden implementarse varios sistemas de detección de errores que reduzcan la probabilidad de éstos. Uno de ellos es que el dispositivo fuente envíe información redundante al dispositivo destinatario. Esto es, que el dispositivo fuente envíe la misma información al dispositivo destinatario de dos formas diferentes. Entonces el destinatario puede comparar las dos piezas de información. Si son iguales puede asumir que las dos piezas se han recibido correctamente. Si no se puede concluir que ha habido error en el envío de los datos. Si bien este sistema es bastante seguro y fiable, como habrán podido comprobar es bastante rudimentario y despilfarrador puesto que desperdicia bastante capacidad del canal de transmisión. Nada menos que se reduce la velocidad de transmisión del canal a la mitad.

Sin embargo, existe un método más económico de transmitir información redundante. Los datos en sí mismos no son repetidos, sino una característica de ellos. La Paridad es una característica de los datos determinada por el número de "unos" lógicos que hay en los bits de datos más el bit de paridad en sí mismo. Hay dos elecciones. El bit de paridad puede ser determinado según que el número de los lógicos sea par ó impar. Si el bit de paridad se determina en base a un número par de bits de datos más el de paridad al esquema se le llama Paridad par.

Si el bit de paridad se determina en base a un número impar de bits de datos más el de paridad, al esquema se le llama Paridad impar. Quizás se vea más claro con el siguiente ejemplo:

Byte de datos	Paridad par	Paridad impar
0000 0000	0	1
0000 0010	1	0
1111 1111	0	1
1010 0110	0	1

Como podemos ver, éste bit de paridad sólo sirve para detectar errores en un sólo bit. Más aún sólo sirve para detectar un número impar de bits de error.

Este simple bit de paridad es el esquema de detección de errores usado en el protocolo serie asíncrono. El cálculo del valor de éste bit de paridad a ser transmitido puede ser hecho mediante hardware o software. Todos los interfaces series asíncronos que se encuentran en el mercado desarrollan estas funciones.

9.2.3.4 Bits de Stop

Después de la transmisión de los bits de datos y del bit de paridad, si se especifica por el usuario, el transmisor envía bien 1, 1.5 ó 2 bits de stop. Estos bits de stop son simplemente "unos" lógicos que duran 1, 1.5 y 2 bits de duración. Estos bits de stop fuerzan a la línea a asumir el estado "marking" durante al menos el tiempo que dura 1 bit antes de enviar el próximo carácter. Esto hace que el bit de start comience con una transición desde el nivel lógico 1 al nivel lógico 0. Este bit de stop se usaba para propósitos de retardo en los viejos teletipos. Sin embargo, con las velocidades que se usan actualmente en las transferencias no es necesario este bit de stop. Ahora bien, se hace necesario al menos 1 bit de stop para asegurar que cada carácter comienza con una transición desde el nivel lógico 1 al nivel lógico 0.

9.2.4 PROTOCOLO SERIE SINCRONO

En contraste con el esquema de sincronización (carácter a carácter) usado por el protocolo serie asíncrono, el esquema de sincronización síncrono usa esencialmente dos técnicas de sincronización. Primero, se usa un reloj de muestreo para definir los bits individuales cuando van llegando desde la línea.

Este reloj de sincronización dice al receptor exactamente cuando muestrear la línea de datos para detectar cada bit. Existen dos formas principales en los interfaces síncronos de proveer el reloj de datos: "al lado" ó "dentro" de los datos.

El esquema de datos "al lado" usa una línea física separada de los datos para llevar el reloj desde el transmisor al receptor. Este esquema es bastante aceptable para comunicaciones locales, pero es inaceptable sobre transmisiones de largo alcance.

Por el otro lado, el esquema "dentro de" codifica la señal de reloj como parte de la señal de datos. El dispositivo que efectúa ésta codificación del reloj y/o los datos se llama "modem síncrono". Este modem síncrono usa una técnica de modulación de la señal que permite al receptor decodificar el reloj de los datos recibidos desde el canal telefónico.

El segundo esquema de sincronización usado por el protocolo síncrono está diseñado para identificar bloques individuales de datos. Un bloque de datos puede estar compuesto por cualquier número de bits. La identificación de los grupos de bits como caracteres sólo puede hacerse mediante software descomponiendo e interpretando los bits en el bloque de datos recibido. El protocolo que no identifica los grupos de bits con caracteres es llamado protocolo "orientado a los bits", y el que soporta esquemas de codificación de caracteres es llamado protocolo "orientado al carácter".

Los protocolos síncronos identifican los bloques de datos a través de caracteres especiales ó secuencias de bits llamados "caracteres de sincronismo" en el caso de protocolos orientados a los bits, y "flag de comienzo" en el caso de protocolos orientados a los bytes. Al principio de la transmisión, o cuando empieza un bloque de resincronización, el receptor entra en el modo de "búsqueda" (hunt) para localizar en el flujo de bits uno ó más caracteres de sincronismo. Después de esto el receptor puede asumir que está sincronizado con el transmisor.

Los protocolos serie síncronos más comunmente usados son:

BISYNC (IBM) Binary Synchronous Communications

SDLC (IBM) Synchronous Data-Link Control

HDLC (ISO) High-Level Data-Link Control

ADCCP (ANSI) Advanced Digital-Communications-Control Protocol

X.25 (CCITT) Recommendation from the X.25 Committee
DDCMP (DEC) Digital Data-Communications Message Protocol
UDLC (UNIVAC) Univac Data-Link Control
BDLC (Burroughs) Burroughs Data-Link Control

9.2.5 DETECCION DE ERRORES

Una función muy importante que desarrollan todos los circuitos serie de E/S es la detección de errores. Los tres errores que pueden ocurrir son:

Error de Formato (Framing)

Error de paridad

Error de desbordamiento del receptor

Veamos a continuación cuál es el significado de cada uno de estos errores.

9.2.5.1 Error de Formato

Los errores de formato son detectados cuando el receptor espera un bit de stop y en su lugar recibe un 0 lógico. Los bits de stop son los únicos bits que el receptor puede anticipar. Por tanto, el error de formato puede ser definido como la ausencia de un bit de stop esperado. Este error puede deberse a un fallo en el receptor, a un fallo en el transmisor, o a ruidos en la línea.

9.2.5.2 Error de Paridad

El receptor efectúa el cálculo de la paridad en base al esquema acordado previamente entre ambos dispositivos (par, impar ó sin paridad), y compara éste valor calculado con el recibido desde el transmisor. Si son distintos se produce un error de paridad.

9.2.5.3 Error de desbordamiento del receptor

Los receptores de los interfaces series asíncronos típicos son capaces de almacenar sólo unos pocos caracteres. El SIO del Z80 es capaz de almacenar tres caracteres en el lado receptor. Este "buffer" está organizado como una pila FIFO (primero en entrar, primero en salir). El "desbordamiento" del buffer se produce por lo tanto, cuando se han recibido cuatro caracteres sin que ninguno de ellos haya sido leído por la CPU.

Tipicamente, la representación de los datos es binaria, en donde un 1 lógico se representa mediante +5 V y un 0 lógico se representa mediante 0 V. Estos niveles se conocen como lógica TTL, y es la tecnología standard en las comunicaciones entre componentes sobre pequeñas distancias. Ahora bien, es impráctico definir una norma en la que los niveles esten definidos por voltajes exactos, así, en la actualidad la lógica TTL asocia rangos de voltaje con los dos niveles. De este modo, un transmisor debe proporcionar un voltaje de al menos 2.4 V para transmitir un nivel lógico 1. Por otro lado, el receptor necesita solamente un voltaje de 2.0 V para entender que es un 1 lógico. Esta diferencia permite algunas pérdidas de voltaje entre el transmisor y el receptor. A esto se le llama margen de ruido, que en este caso es de 0.4 V.

Para transferencias de datos internas al computador, los niveles de señal TTL son ideales por varias razones. Primero, las necesidades de potencia y la disipación de calor es menor. Segundo, estos niveles son compatibles directamente con un gran número de circuitos integrados evitando el coste adicional de "drivers" y circuitos receptores. Por último, estos niveles permiten trabajar a las altas velocidades que se requieren en las transferencias de datos internas al computador.

Desafortunadamente, existen varios problemas serios relacionados con los niveles TTL en comunicaciones a distancias mayores de unos pocos metros. Primero, los niveles TTL son muy sensibles al ruido externo. Y segundo, la escasa diferencia de voltaje que existen entre un nivel lógico y otro es muy sensible a las pérdidas de voltaje en la línea.

Es por esto que a partir de 1940, a medida que se iba extendiendo el área de acción de los computadores, se desarrollaron sistemas para comunicar dispositivos a larga distancia sobre la línea telefónica.

El problema de convertir las señales binarias ON/OFF a una forma que pueda ser propagada efectivamente a través de la línea telefónica se resolvió usando el "modem". Sin embargo, se creó un nuevo problema, era necesario establecer una normalización para controlar y especificar la comunicación entre el dispositivo digital y el modem.

En 1969, la "Electronics Industry Association" (EIA) publicó una normalización al respecto que se usa prácticamente en todos los sistemas microcomputadores actualmente. Esta normalización se conoce normalmente como protocolo RS-232C.

La necesidad es la madre de la invención, y el protocolo RS-232C nació debido a la creciente utilización de la red telefónica para las comunicaciones de datos.

El protocolo RS-232C establece normalización respecto a la comunicación entre un "Equipo terminal de datos" (DTE) y un "Equipo de comunicación de datos" (DCE). Veamos a continuación la definición de estos términos dada por John McNamara (Digital Press, 1977):

DCE: Es el equipo que proporciona las funciones necesarias para establecer, mantener, y terminar una conexión; para efectuar la conversión y codificación de la señal necesaria para la comunicación entre el equipo terminal de datos y el circuito de datos. El DCE puede ser o no una parte integral del computador.

DTE: Es el equipo que compone el emisor de datos, el receptor, y todo lo referente al almacenamiento de estos. Este equipo normalmente comprende las siguientes unidades funcionales: control lógico, almacenamiento, y uno o más dispositivos de entrada o salida. También puede contener control de errores, sincronización, y capacidad de identificación de la estación.

En la figura 2.34 se muestran DTEs y DCEs en un circuito de comunicación ilustrando lo explicado anteriormente.

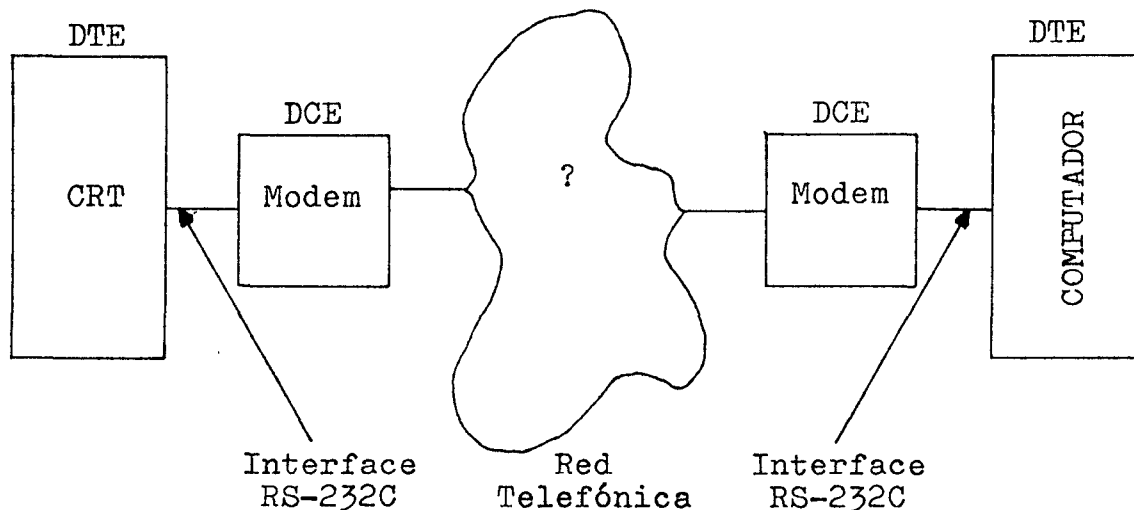


Fig. -2.34-

Esencialmente, el DTE representa la fuente y/o el destino de los datos. El DCE facilita la transmisión de los datos desde la fuente hasta el destinatario. Un modem es, por ejemplo, un dispositivo DCE.

9.3.1 CARACTERISTICAS ELECTRICAS DE LAS SEÑALES

En el protocolo RS-232C la velocidad de transmisión en la comunicación entre DTEs y DCEs está en el rango entre 0 y 20.000 bps. De este modo, la velocidad más alta comunmente usada es de 19.2 Kbps.

Asimismo, se establece una limitación en cuanto al largo del cable de 50 pies (15 m.). Este limite puede ser excedido si se cumplen ciertas condiciones que se detallaran más adelante.

A continuación se detallan las características que deben reunir las señales eléctricas del RS-232C utilizando para ello los nombres representados en la figura 2.35.

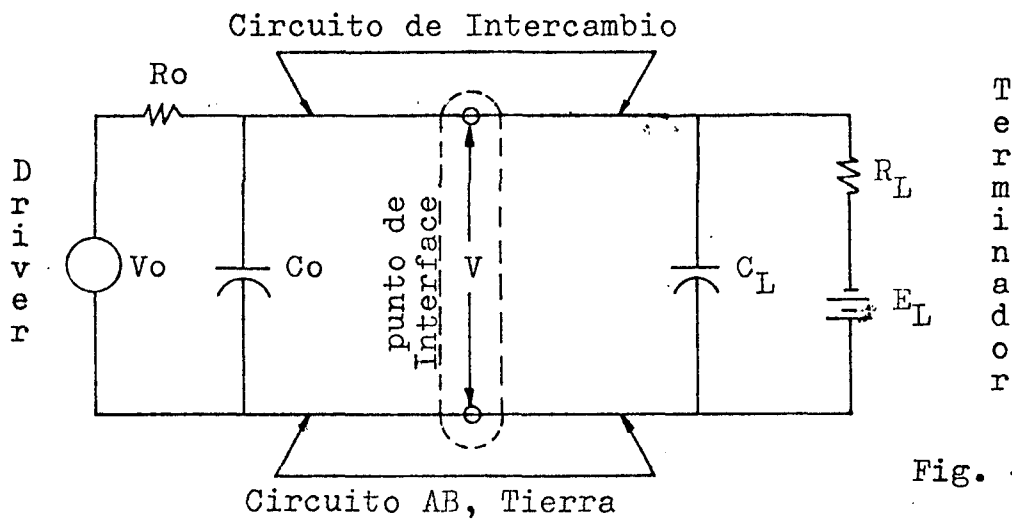


Fig. -2.35-

- Una señal de cualquier pin del conector RS-232C tiene un estado asociado a ella. Este estado puede ser uno de los pares que se muestran a continuación:

Estados	Voltajes de la señal			
	-25V	V	-3V	3V V 25V
Lógica binaria		1		0
Condición de la señal		MARK		SPACE
Función		OFF		ON

Es importante destacar que el protocolo RS-232C emplea lógica negativa, y que una condición ON está asociada con el estado lógico 0, mientras que una condición OFF está asociada con el estado lógico 1. El voltaje de la señal V está medido con respecto al circuito de tierra del que hablaremos más adelante. El rango desde -3V hasta +3V es la región de transición en la que no está definido ningún estado.

2. Para representar un estado lógico 1 o una condición MARK, el generador debe aplicar un voltaje entre -5 y -15 V. Y para representar un estado lógico 0 o una condición SPACE debe aplicar un voltaje entre +5 y +15 V.

Creo que es importante hacer notar que esta regla en conjunción con la regla 1, implica que hay un margen de 2 V para insensibilizarlo frente al ruido.

3. La capacidad paralelo C_L del lado terminador no debe exceder de 2500 pF incluyendo la capacidad del cable.

Esta regla contribuye significativamente a la limitación de longitud del cable de 50 pies.

4. El voltaje en circuito abierto o sin carga no puede exceder de 25 V. Esto quiere decir que no pueden haber voltajes el protocolo RS-232C que excedan de 25 V.
5. Un generador RS-232C debe ser capaz de poder estar conectado a cualquier otro generador o receptor del bus sin sufrir ningún tipo de daño.

9.3.2 CARACTERISTICAS MECANICAS DEL INTERFACE

En el bus RS-232C se utiliza normalmente el conector DB-25, que es compatible con el conector standard ISO 2113 promulgado por la Organización Internacional de Normalización (ISO).

En la figura 2.36 se detallan las características mecánicas de éste conector.

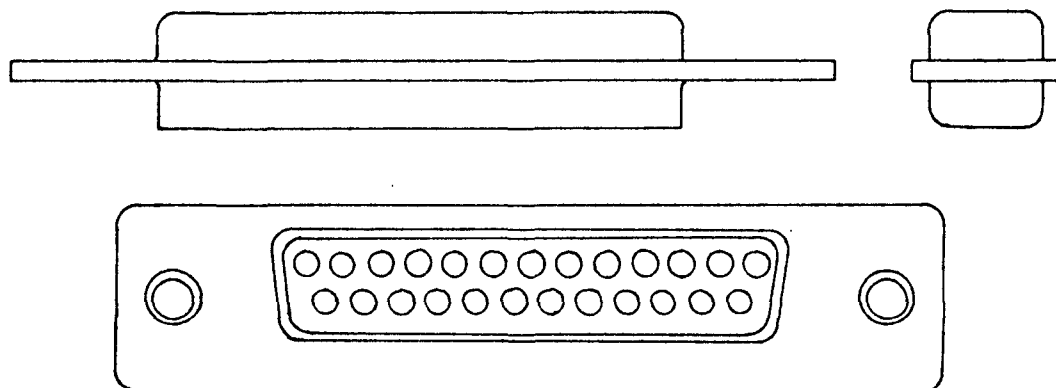


Fig. -2.36- CONECTOR DB-25

9.3.3 CARACTERISTICAS FUNCIONALES DE LOS CIRCUITOS

En la figura 2.37 se relaciona la asignación de pines en el conector del interface standard RS-232C. Como vemos, los circuitos mostrados pueden ser divididos en cinco categorías:

- Tierra o Retorno Común (A)
- Circuitos de Datos (B)
- Circuitos de Control (C)
- Circuitos de Reloj (D)
- Circuitos del Canal Secundario (S)

La letra en paréntesis después de cada categoría se utiliza para definir cada uno de los circuitos en función de las categorías a las que esté referido.

A continuación vamos a definir cada uno de estos circuitos:

9.3.3.1 Circuito AA: Tierra de Protección

Este circuito esta vinculado o unido eléctricamente a la estructura o el chasis del equipo. Como este circuito no se utiliza como tierra de referencia para el resto de los circuitos es común no encontrarlo en muchos interfaces. Por lo tanto, este circuito es en cierto modo opcional.

9.3.3.2 Circuito AB: Tierra de Señal o de Retorno Común

Este conductor se utiliza como referencia para el resto de los circuitos del interface, excepto para la tierra de protección (Circuito AA). Este es el único circuito que es imprescindible en cualquier aplicación que se le dé al interface.

9.3.3.3 Circuito BA: Transmisión de Datos

Las señales de éste circuito son transmitidas desde el DTE al DCE. El DTE mantiene esta señal en el estado lógico 1 (condición MARKING) mientras no se estan transmitiendo datos.

En todos los sistemas que utilicen el interface RS-232C, el DTE no debe transmitir datos hasta que todos los circuitos que se detallan a continuación no esten al nivel lógico 0 (condición ON):

1. Request to Send (circuito CA)
2. Clear to Send (circutio CB)

Categorías	Asignación de pines	V.24 del CCITT	Descripción	Gnd	Datos		Control		Temporización	
					Desde el DCE	al DCE	Desde el DCE	al DCE	Desde el DCE	al DCE
	1		Tierra de protección	x						
AB	7	102	Tierra de señaliz. o retorno común	x						
BA	2	103	Transmisión de datos			x				
BB	3	104	Recepción de datos		x					
CA	4	105	Petición de emisión						x	
CB	5	106	Preparado para transmitir				x			
CC	6	107	Modem preparado				x			
CD	20	108	Terminal de datos preparado						x	
CE	22	125	Detector de señal de llamada				x			
CF	8	109	Detector de portadora en línea				x			
CG	21	110	Detector calidad señales en línea				x			
CH	23	111	Selector de velocidad binaria						x	
CI	23	112	Selector de velocidad binaria				x			
DA	24	113	Sincronismo en transmisión por ETD							x
DS	15	114	Sincronismo en transmisión por ETCD						x	
DD	17	115	Sincronismo en recepción						x	
SBA	14		Transmisión de datos secundaria				x			
SBB	16		Recepción de datos secundaria		x					
SCA	19		Petición de emisión secundaria						x	
SCB	13		Prep. para transmitir secundario				x			
SCF	12		Detec. de port. en línea secundario				x			

Figura 2.37

CIRCUITOS DEL RS-232 C

3. Data Set Ready (circuito CC)

4. Data Terminal Ready (circuito CD)

Para un sistema microcomputador con un terminal local conectado a él a través de un interface RS-232C, si el terminal actúa como un DTE (que es lo más normal), entonces el computador debe actuar como un DCE. De este modo, el terminal (DTE) habla por la línea "Transmisión de datos", y el computador (DCE) escucha por este circuito. Por tanto, vemos que el término "transmisión" se refiere al DTE y no al DCE.

En caso de que usted tenga algún problema en una comunicación con un interface RS-232C, es bastante probable que sea por alguna de las dos siguientes razones:

- (1) Una de las señales expuestas anteriormente está en estado OFF.
- (2) ó el Terminal está transmitiendo en la misma línea que el computador, por lo que ninguno está escuchando al otro.

9.3.3.4 Circuito BB: Recepción de datos

Las señales en este circuito son transmitidas desde el DCE al DTE. Durante los intervalos en que no se están transmitiendo datos, y siempre que no se estén transmitiendo datos, este circuito se mantiene en el estado lógico 1 (condición MARKING).

En un canal half-duplex, ésta línea se mantiene en la condición OFF cuando Request to Send (circuito CA) está en la condición ON.

9.3.3.5 Circuito CA: Request to Send

Este circuito lleva una solicitud de transmisión desde el DTE al DCE. Esta señal (RTS) junto con la señal llamada Clear to Send (CTS ó circuito CB) coordinan la transmisión de datos entre el DTE y el DCE.

Para canales simplex y full-duplex, un nivel lógico 0 en esta línea mantiene al DCE local en el modo de transmisión, mientras que un nivel lógico 1 mantiene al DCE local en el modo de no transmisión, es decir, que no pasará los datos que ha recibido desde el DTE a la red de comunicaciones.

Para canales de comunicación half-duplex, una condición ON en Request to Send mantiene al local DCE en el modo de transmisión, y una condición OFF mantiene al DCE local en el modo de recepción. Por supuesto, el modo de recepción quiere decir que el DCE aceptará datos desde la red de comunicación y los pasará a su DTE local.

Todo lo visto hasta este momento se ilustra mejor en la figura 2.38, en la que podemos ver que una transición de RTS desde la condición ON a la condición OFF ordena al DCE local completar la transmisión de todos los datos que han cruzado previamente el punto de interface en el circuito de Transmisión de Datos, y ponerse en el modo de no transmisión (full-duplex ó simplex) ó en el modo de recepción (half-duplex). El DCE local responde a ésta señal poniendo a OFF la señal Clear to Send.

Como se muestra en la figura 2.38, una vez Request to Send ha sido puesta a OFF, no puede ser activada a ON hasta que Clear to Send haya sido puesta a OFF por el DCE local. Esta regla previene contra los errores de desbordamiento debidos a que el DTE continúe la transmisión sin que el DCE haya completado la transmisión anterior.

Dependiendo de la aplicación particular del DTE y el DCE, el protocolo entre Request to Send y Clear to Send descrito anteriormente puede producirse carácter a carácter, bloque a bloque, o sencillamente no producirse, es decir, que ambas están continuamente activadas durante el tiempo que dura la conexión.

9.3.3.6 Circuito CB: Clear to Send

Esta es una señal de control que se transmite desde el DCE hasta el DTE para indicar que el DCE está listo para recibir datos desde el DTE por el circuito de Transmisión de datos. Cuando esta señal está a ON, y las señales Request to Send, Data Set Ready, y Data Terminal Ready están todas a ON, constituye una indicación al DTE de que los datos transmitidos serán enviados por el DCE al canal de comunicación. Cuando Clear to Send está a OFF, indica que el DCE no está listo, y por lo tanto, el DTE no debe intentar transmitir datos.

Clear to Send es puesta a ON como respuesta a una condición ON simultánea de Request to Send y Data Set Ready.

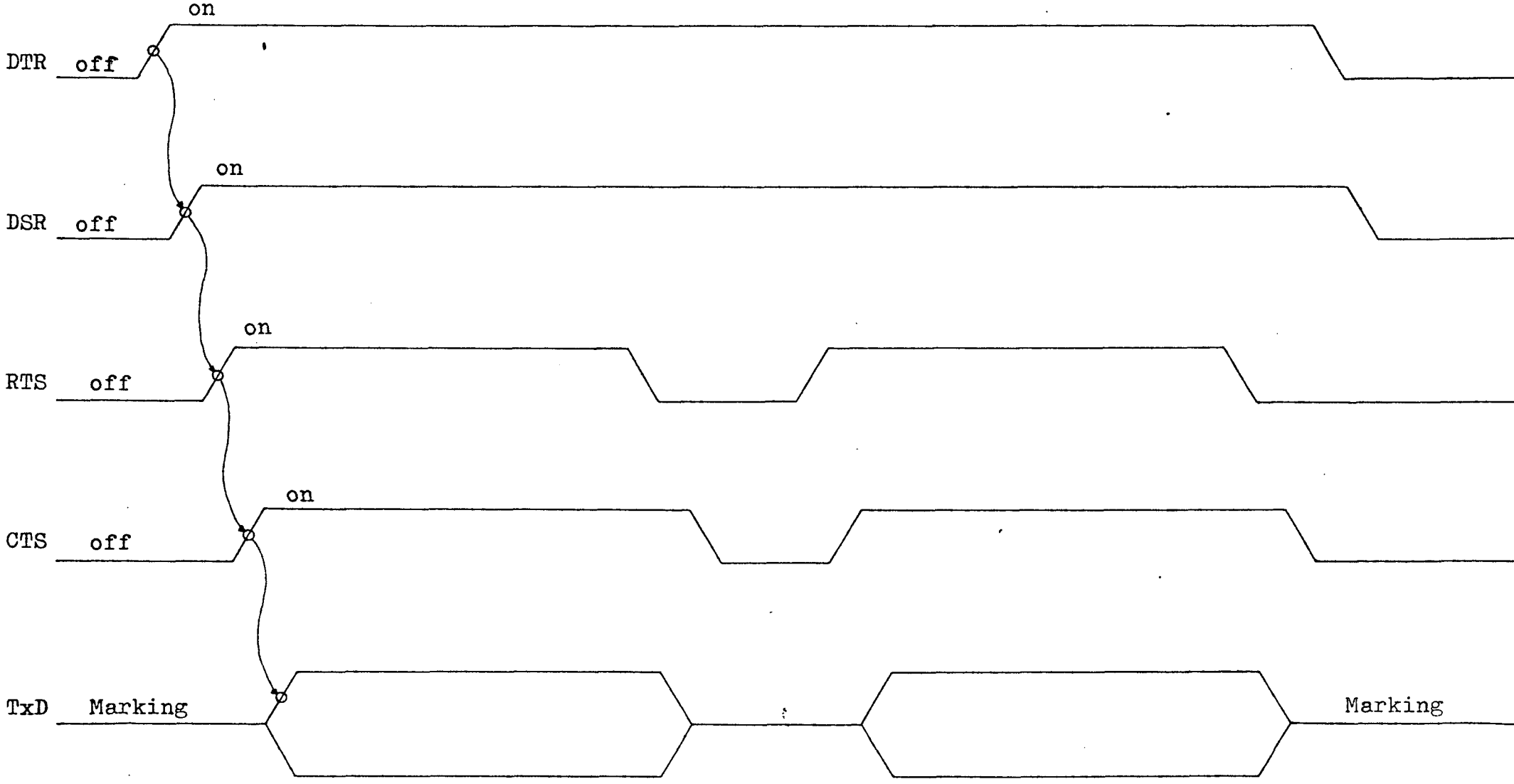


Fig. -2.38.

Si Request to Send no se implementa, se asumirá que está constantemente en la condición, y Clear to Send responderá a la activación y desactivación de Data Set Ready.

En interfaces RS-232C entre DTEs y DCEs usando la red pública de telefonos, son necesarios los siguientes circuitos:

Circuito CC	Data Set Ready
Circuito CD	Data terminal Ready
Circuito CE	Ring Indicator
Circuito CF	Received Line Signal Detector

Como todas estas funciones son bastante específicas para el uso de la red telefónica, a menudo estos circuitos son omitidos en la mayoría de las aplicaciones locales entre sistemas microcomputadores.

9.3.3.7 Circuito CC: Data Set Ready

La dirección de ésta señal de control es desde el DCE al DTE. Si la señal Data Set Ready está a ON quiere decir que el DCE está conectado al canal de comunicación. En situaciones de llamada automática esto quiere decir que el DCE ha marcado el número, completado el establecimiento de la llamada, y está en el modo de transmisión de datos.

9.3.3.8 Circuito CD: Data Terminal Ready

La dirección de ésta señal de control es desde el DTE al DCE. Data Terminal Ready debe estar a ON antes de que el DCE pueda poner a ON Data Set Ready, indicando que ha sido conectado al canal de comunicación. Una vez el DCE está conectado y los datos estan siendo transmitidos, una transición de Data Terminal Ready desde la condición ON a la condición OFF causa la desconexión del DCE del canal de comunicación.

Esencialmente, Data Terminal Ready y Data Set Ready implementa una versión más estática del protocolo entre Clear to Send y Request to Send. Por más estática se entiende que, mientras que el protocolo entre Clear to Send y Request to Send se refiere a la disposición del canal, el protocolo entre Data Terminal Ready y Data Set Ready se refiere a la disposición de los equipos. La disposición del equipo, normalmente es menos volátil que la disposición del canal.

9.3.3.9 Circuito CE: Ring Indicator

La dirección de esta señal de control es desde el DCE al DTE. Cuando esta señal está a ON indica que el DCE está recibiendo una señal de llamada (ring). Esta señal se mantiene en la condición OFF entre "llamadas", y siempre que no se estén recibiendo señales de llamada. La principal aplicación de esta señal de control es en configuraciones con modems de contestación automática.

9.3.3.10 Circuito CF: Received Line Signal Detector

El DCE envía una condición ON al DTE por este circuito cuando está recibiendo una señal portadora desde el DCE remoto. Un nombre muy usado también para esta señal es Data Carrier Detect (DCD). En los modems esta señal está conectada normalmente a un indicador LED etiquetado como Carrier, ó también como 109.

Los restantes circuitos Signal Quality Detector, Data Signal Rate Selector, etc. no se tratan aquí debido a que su aplicación es realmente escasa en la mayoría de los sistemas microcomputadores. Si se desea profundizar sobre el significado y aplicación de estas señales utilizar la bibliografía dada al final de este capítulo.

9.3.4 CONFIGURACIONES MAS COMUNES (NO STANDARDS)

En la figura 2.39 se ilustran los diagramas esquemáticos para varios cables RS-232C especiales (no standards) que se utilizan muy a menudo en sistemas microcomputadores.

La figura (a) es el esquema de un cable standard full-duplex. Las figuras (b) hasta (e) muestran algunas variaciones del standard. Las flechas al final de las líneas indican la dirección de la señal asociada del RS-232C asociada a ella. A continuación vamos a tratar más detalladamente cada uno de estos cables.

9.3.4.5 Modelo económico de tres cables

La figura 2.39b representa un cable con sólo tres conductores. Esta configuración se utiliza mucho en muchos sistemas microcomputadores. Este cable provee el número mínimo de circuitos necesarios para la comunicación full-duplex.

El problema más común utilizando este cable es que la

mayoría de los sistemas microcomputadores utilizan los circuitos Request to Send y Clear to Send. Desafortunadamente, estos equipos no transmitiran datos hasta que reciban una condición ON en Clear to Send.

El cable implementado en la figura 2.39c tiene una serie de adiciones respecto al modelo económico para aplicaciones con USARTs basadas en ports de E/S.

9.3.4.2 Modelo de tres cables con puentes de "Lujo"

Como vemos en la figura 2.39c , mediante el puente entre Data Terminal Ready y Data Set Ready se completa la primera fase del protocolo tan pronto como se activa la línea Data Terminal Ready. Al estar Request to Send puenteada a Redeived Line Signal Detector, desde que se activa Request to Send se completa la fase de disposición del canal.

Ciertamente, este cable simplifica la lógica asociada para obtener un intercambio de datos full-duplex RS-232C. Sin embargo, se han omitido algunas características importantes, el puente entre Data Terminal Ready y Data Set Ready omite la secuencia de conexión. El puente entre Request to Send y Data Carrier Detector elimina la función del DCE asociada con la detección de la portadora desde el modem remoto. El puente entre Request to Send y Clear to Send elimina todo el protocolo asociado con Request to Send y Clear to Send; en particular, elimina la prevención que este protocolo hace de los errores de desbordamiento.

En sistemas que no usen la red telefónica, es razonable omitir todos estos circuitos RS-232C que no son estrictamente necesarios.

Las restantes configuraciones son bastante parecidas a ésta con la diferencia de que los dispositivos a conectar son dos DTEs, y por tanto hay que tener cuidado con la dirección de las señales (en particular, con las líneas Transmisión de Datos y Recepción de Datos).

Por lo demás, la justificación de los puentes es la misma que se reseñó más arriba.

Asimismo, todas las omisiones realizadas en la simplificación del protocolo son básicamente las mismas que se explicaron anteriormente.

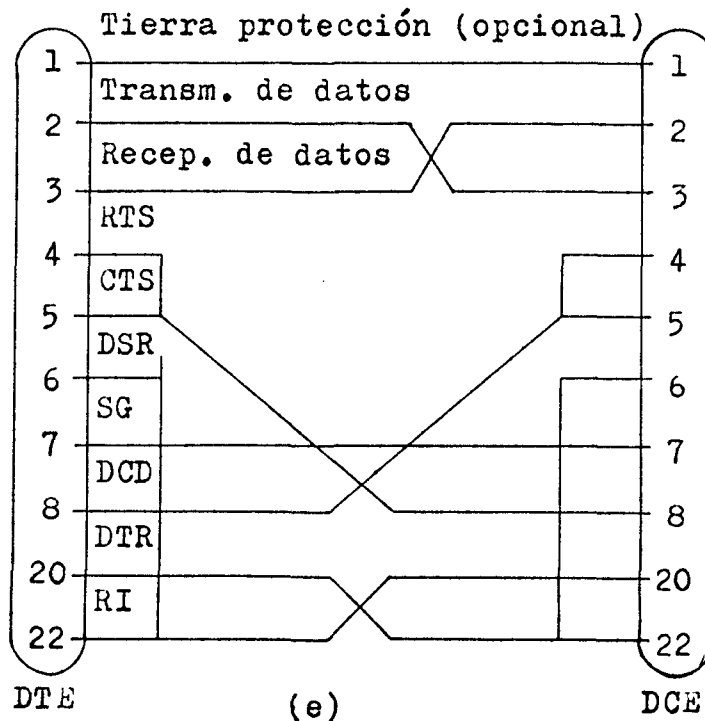
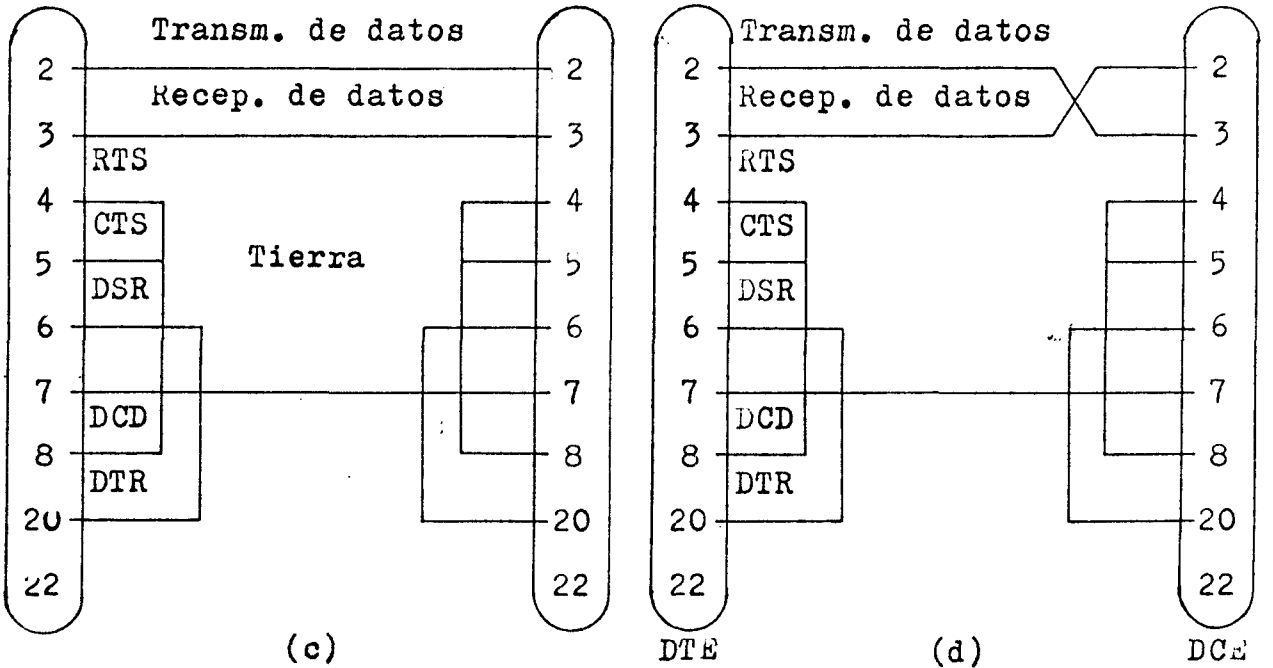
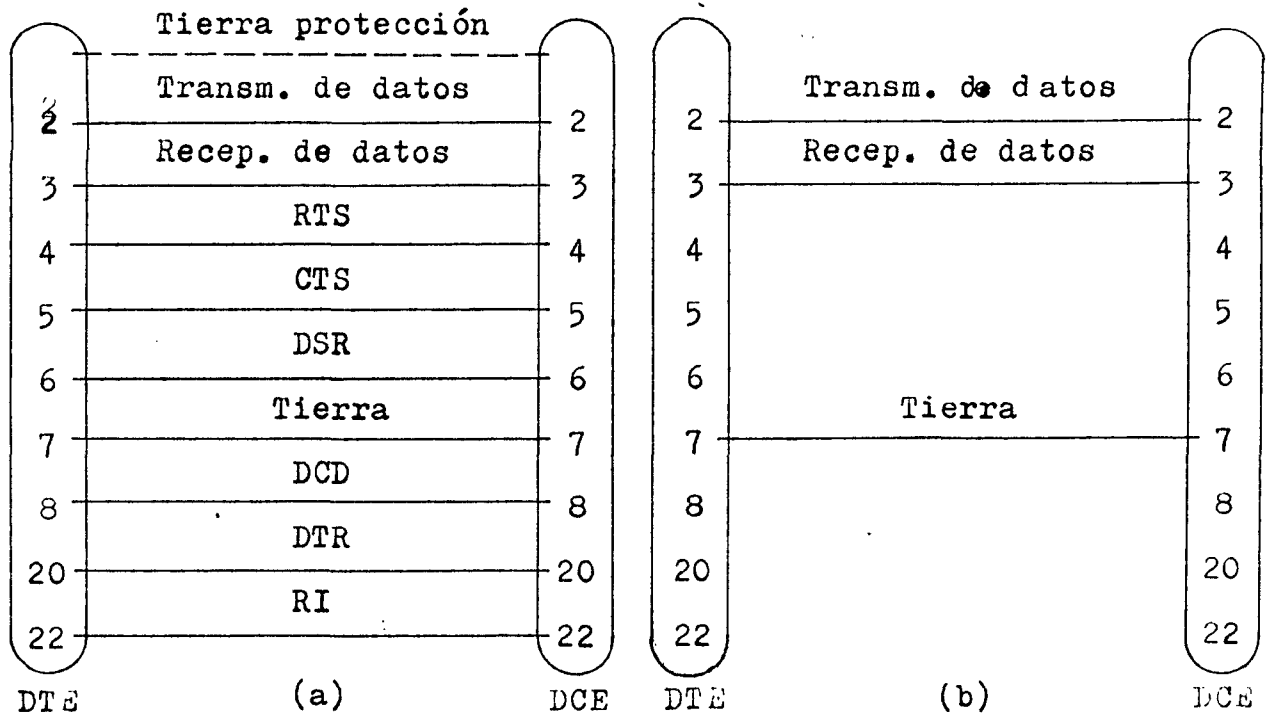


Fig. -2.39-

9.4 INTERFACE DEL LAZO DE CORRIENTE DE 20 MILIAMPERIOS

Hasta la década de los 60 los teletipos militares y de servicios usaban un lazo de corriente de 60 mA para comunicarse sobre largas distancias. El teletipo modelo 32 de la "Teletype Corporation" introdujo ya en 1950 el lazo de corriente de 20 mA, pero no fué hasta la llegada de su sucesor el Modelo 33 cuando el uso del lazo de corriente de 20 mA se hizo extensivo.

Los circuitos del lazo de corriente transfieren los datos de manera serie enviando la corriente en un sentido o en otro para representar niveles lógicos. El lazo de corriente puede ser usado para el intercambio de datos entre dos dispositivos cualesquiera, pero normalmente se utiliza para teletipos o teleimpresoras mecánicas. Como su nombre indica, el lazo de corriente está configurado como un lazo en el que se conectan serialmente uno ó más dispositivos.

La corriente fluye desde el terminal positivo del dispositivo fuente a través de cada uno de los dispositivos conectados, y retorna por el terminal negativo del dispositivo fuente.

Un circuito de lazo típico entre un computador y un terminal está formado por dos líneas simples, una para dirección del flujo de datos. Teóricamente, sin embargo, no hay límite para el nº de dispositivos que se pueden conectar al lazo de corriente. La limitación en el número de dispositivos es función puramente de la potencia disponible en el generador de corriente.

Puesto que cada dispositivo conectado a la línea constituye en realidad una resistencia al flujo de corriente, aplicando la Ley de Ohm podemos saber fácilmente el nº de dispositivos que podemos conectar.

Por lo tanto, para mantener una corriente constante de 20 miliamperios; si la resistencia aumenta como resultado de aumentar el número de dispositivos, debe aumentarse el voltaje en el generador para compensarlo.

De éste modo, los componentes básicos de un lazo de corriente de 20 mA son un generador de corriente, un conmutador de corriente (ó transmisor), y un detector de corriente (ó receptor).

Una cuestión importante a resolver cuando se conectan dos dispositivos mediante un lazo de corriente de 20 mA es la localización del generador de corriente. Si el genrador de corriente está en el transmisor, entonces al transmisor se le llama activo. Igualmente si la fuente de corriente está en el receptor, se le llama receptor activo.

Como ocurre con los DTEs y los DCEs en el interface RS-232C , el lazo de corriente de 20 mA requiere un componente activo y un componente pasivo. De este modo, un transmisor activo debe conectarse a un receptor pasivo, y un transmisor pasivo a un receptor activo.

9.4.1 VENTAJAS Y DESVENTAJAS

Para aplicaciones que impliquen pequeñas distancias, menores de 600 m., y velocidades de transmisión menores de 9600 bps, el lazo de corriente de 20 mA puede sustituir ventajosamente al interface de corriente de 20 mA. Por otro lado, ambos sistemas pueden ser implementados en el mismo conector con un mínimo gasto adicional, ya que el lazo de 20 mA puede utilizar líneas o pines no asignados por el interface standard RS-232C. El interface de 20 mA requiere solamente cuatro líneas:

Transmisión de datos

Retorno de Transmisión

Recepción de datos

Retorno en la recepción

Hay dos problemas principales relacionados con el interface de corriente de 20 mA.

Primero, la ausencia de una normalización mecánica ó elétrica.

El segundo principal problema es el Ruido. Los circuitos eléctricos de los teletipos usan contactos o conmutadores rotativos. La conmutación mecánica de estos contactos produce ruidos eléctricos que deben ser filtrados para asegurar una comunicación fiable. Asimismo, sobre largas distancias los componentes tienden a introducir ruido en los cables cercanos.

En la figura 2.40 se detallan las características más importantes de los protocolos más importantes.

	TTL	RS-232C	RS-422A	RS-423A	20 mA
0 Lógico (generador)	0 a 0.4V	+5 a +15V	+2 a +6V	+4 a +6V	Sin corriente
1 Lógico (generador)	2.4 a 5V	-5 a -15V	-2 a -6V	-4 a -6V	20 mA
Velocidad máxima bps.	$\sim 25 \times 10^6$	2×10^4	10^7	10^5	10^5 900
Distancia máxima, pies	30	50	40	4000	40 4000 Sobre 1000

Figura 2.40

CAPITULO III

INTERCONEXION ENTRE EL SISTEMA DE DESARROLLO MDS 221
Y UN ORDENADOR HP-3000

1. INTRODUCCION

La conexión entre el sistema de desarrollo y el ordenador se efectuó a través de un modem MD-600/1200 bps N. D. de SESA mediante un bus estandar RS-232 C. Para ello se utilizó el canal serie 2 de E/S del sistema de desarrollo.

Como el sistema de desarrollo está equipado con un bus estandar RS-232 C, ya que dispone de una USART 8251 de Intel, y los drivers necesarios para alcanzar los niveles eléctricos requeridos por el protocolo RS-232 C, no fue necesario realizar ninguna modificación en el hardware.

En cuanto al hardware se refiere, el único problema fue, aparte de comprobar que el bus del sistema de desarrollo cumplía los niveles definidos en el protocolo, y por lo tanto, era compatible con el del modem, elegir las líneas mínimas y necesarias definidas en dicho protocolo (ver el tema 9 del capítulo II de este mismo proyecto) para controlar el modem.

El cable diseñado para esta conexión contenía las siguientes líneas del bus RS-232 C (se especifica también el pin que corresponde a cada línea en el conector):

- Pin 2: Transmisión de datos (TXD)
- Pin 3: Recepción de datos (RXD)
- Pin 4: Request to Send (RTS)
- Pin 5: Clear to Send (CTS)
- Pin 6: Data Set Ready (DSR)
- Pin 7: Tierra (GND)
- Pin 20: Data Terminal Ready (DTR)

Quiero destacar, que utilizo la denominación inglesa de la mayoría de las señales, ya que en todos los círculos en que se habla de este protocolo se utiliza la denominación de origen. Por otra parte, sería bastante fácil traducirlas al español, pero particularmente pienso que debe mantenerse su denominación original para que el lector no se cree un confusionismo de señales y nombres en su mente.

En cuanto al software se refiere, se desarrollaron dos programas de los que hablaremos a continuación. Un primer programa llamado "MODEM" cuya misión es hacer que el sistema de desarrollo funcione como un terminal estándar no inteligente, en el que los caracteres tecleados son enviados simplemente al ordenador, y los caracteres devueltos por este son presentados en pantalla.

Y un segundo programa llamado "QUILEY", el el que además de realizarse las funciones efectuados por el programa modem, se realizan el control de la impresora paralelo, y se gestiona la unidad de diskette del sistema de desarrollo.

2. ESTUDIO DEL PROGRAMA MODEM

Este programa fue desarrollado para establecer la comunicación entre un sistema de desarrollo Intel MDS 221 y un ordenador Hewlett-Packard de la serie HP-3000.

De este modo, conseguimos ampliar y mejorar enormemente los recursos disponibles del sistema de desarrollo, puesto que, aparte de poder trabajar como tal, puede funcionar como terminal de un ordenador remoto, conectado a través de un modem, tecleando simplemente el, ahora, comando MODEM, y conectándolo, si no lo está ya, a través del cable RS-232 C del que ya hemos hablado.

La misión de este programa es hacer que el sistema de desarrollo funcione como un terminal estándar no inteligente, en el que los datos tecleados son simplemente enviados al ordenador, y las contestaciones son presentadas en pantalla.

La característica de la comunicación en este caso, limitada al tipo de conexión, es: Comunicación Serie, Asíncrona x64, 1200 baudios, 8 bits de longitud del carácter, sin paridad, y con 2 bits de Stop. Decimos limitada al tipo de conexión, ya que estas características pueden reconfigurarse en el ordenador dependiendo de si el terminal está conectado directamente (en cuyo caso la velocidad de transmisión será más alta, normalmente 2400 baudios), o a través de modem, de la prioridad que tenga dicho terminal en el sistema, y/o de la configuración que se programe en el ordenador. Pero siempre será más fácil y más lógico ajustar la configuración del terminal a la del ordenador, que al revés.

Vamos a explicar un poco que significa todas las características que acabamos de citar: lo de "Comunicación Serie Asíncrona" ya está explicado en la parte teórica de este mismo proyecto (tema 9, capítulo II). Lo de x64 quizás haya que explicarlo un poco más. Esto viene de que la USART 8251 (ver User's Manual de Intel) puede programarse también, independientemente del reloj (Timer), para controlar en cierta manera la velocidad de transmisión.

Programando la USART, podemos conseguir que la velocidad de transmisión sea de $\frac{1}{64}$ la frecuencia del reloj, $\frac{1}{16}$, o bien la misma frecuencia. Por tanto, en este caso, hemos programado la USART para que la velocidad de transmisión sea de $\frac{1}{64}$ la frecuencia del reloj. Es decir, que si programamos el reloj 8253 (ver User's Manual MCS 85 de Intel) para que la frecuencia sea de 76.8 Khz, la velocidad de transmisión será (dividiendo por 64) de 1200 baudios, que es la que utiliza el modem que utilizamos en esta ocasión.

Lo de que la longitud del carácter sea de 8 bits se debe a que la USART puede programarse para trabajar con longitudes de 5, 6, 7, y 8 bits. Sin embargo, a pesar de que el código ASCII, que es el que se utiliza para el intercambio de información entre los dos dispositivos, es de 7 bits, el ordenador utiliza una configuración de 8 bits para los terminales que están conectados a través de modem. Esto se debe a que los modem vienen configurados ya para trabajar con 8 bits, y siempre resultará más fácil ajustar la configuración del ordenador y del terminal (cuyos softwares son más potentes y permiten una operación más versátil y flexible) que la del modem.

Es importante destacar que este octavo bit, que en principio es despreciable, puesto que no contiene ninguna información, ya que el código ASCII sólo utiliza 7 bits, no siempre estaba a cero (que era lo lógico), cuando era enviado desde el ordenador. Tras hacer varias comprobaciones no conseguimos descubrir ninguna ley que definiera el que este bit estuviese unas veces a cero y otras a uno.

En este tipo de transmisión, no se utiliza paridad, ya que como el terminal va a estar controlado por una persona humana, se entiende que si el eco devuelto por el ordenador no coincide con la tecla pulsada, bastará con utilizar la tecla RUBOUT o BACKSPACE para corregirlo.

Por otro lado, este programa está preparado para trabajar con el protocolo software "ENCUIRY-ACKNOWLEDGE". Este protocolo se basa en que cuando comienza la comunicación, y posteriormente cada cierto tiempo, el ordenador envía el carácter ASCII

"Enquiry" (5 Hexadecimal) para preguntarle al terminal: "¿Has sido tú quien me ha enviado el carácter, o ha sido un ruido en la línea y/o algún fallo mío?"; entonces, el terminal contesta automáticamente con el carácter ASCII "Acknowledge" (6 Hexadecimal) para confirmarle al ordenador que ha sido él efectivamente quien ha enviado el anterior carácter. En caso de que el terminal no conteste, el ordenador no entiende que el terminal está solicitando entrar, y por lo tanto, no comenzará la comunicación.

Con respecto a esto, es interesante destacar que debido a que, como dijimos anteriormente, el octavo bit del dato enviado por el ordenador no siempre llegaba a cero, en lugar el carácter 05H llegaba a veces un 85H (que sólo se diferencia del 05H en que el octavo bit está a "1"), con lo cual el programa no contestaba automáticamente con un 06H, y la comunicación no se efectuaba.

Por esta razón, el programa funcionaba algunas veces, y otras no. Fue necesario descubrir que este octavo bit llegaba algunas veces a "1", y que por tanto había que enmascararlo (ponerlo a 0), para que dejasen de haber problemas con la comunicación.

2.1. DESCRIPCION DEL FUNCIONAMIENTO DEL PROGRAMA

El funcionamiento del programa es bastante sencillo. El primer paso a realizar -se entiende que el sistema de desarrollo ya está encendido y cargado con el Sistema Operativo ISIS- es cargar el disco floppy que contiene el programa MODEM, y teclear simplemente el comando MODEM (que no es más que el nombre del programa). Por supuesto, el modem debe estar ya encendido y conectado al sistema de desarrollo. En caso contrario, se presentará en pantalla el mensaje:

CONECTAR EL MODEM

Una vez realizado esto, el programa estará ya en disposición de intercambiar información para el ordenador. A continuación, basta con seguir los pasos necesarios para comenzar la comunicación definidos por el ordenador. En el caso del ordenador HP-

3000, estos pasos son:

1. Pulsar la tecla CARRY RETURN, y en pantalla el ordenador contestará presentando el siguiente carácter: ":"
2. Teclear el comando: HELLO (nombre del usuario). Entonces el ordenador preguntará por la clave de la cuenta presentando el siguiente mensaje:

TECLEE LA CLAVE DE LA CUENTA:

3. A continuación teclear la clave de la cuenta, y estaremos ya dentro del Sistema Operativo del Ordenador. A partir de aquí, ya se puede trabajar con cualquier subsistema de éste (EDITOR, BASIC, PASCAL etc.)

Hay que decir que, puesto que el ordenador utiliza una serie de secuencias de ESCAPES que pueden ser usadas en cualquier programa para controlar la pantalla (Tales como limpiar toda la pantalla, llevar el puntero al principio de la misma, video inverso, etc.), puede que en la pantalla del sistema de desarrollo se vean algunos caracteres extraños en lugar de éstos, y que además no realicen su cometido.

Por otra parte, en este programa no está previsto ningún tipo de teclas especiales para, por ejemplo, abortar la ejecución del mismo y volver al sistema operativo ISIS del sistema de desarrollo, para abortar la ejecución de cualquier comando o programa (tecla BREAK) en el ordenador y devolver el control al sistema operativo de este. Estas teclas sí se utilizan en el programa QUILEY del que hablaremos más adelante, que contiene una versión más depurada del programa MODEM, aparte claro está, de contener todo el control de la impresora y la gestión de fichero. Por tanto, para abortar la ejecución de un programa en el ordenador hay que usar el "CONTROL Y" (si bien no realiza este cometido en todos los subsistemas del mismo). Y para devolver el control al ISIS basta con pulsar la tecla RESET que se encuentra en el panel frontal del sistema de desarrollo.

2.2 EXPLICACION DEL PROGRAMA

El programa MODEM está formado básicamente por cuatro secciones. Una primera sección en la que se programan la USART y el TIMER para las características específicas de esta comunicación. Una segunda sección que llamamos BUCLE en la que se muestrean el teclado y la Usart. Una tercera sección que llamamos KEIB en la que se lee la tecla pulsada y se envía a la Usart. Y por último una cuarta sección que llamamos USAR en la que se lee a través de la Usart el dato llegado desde el ordenador.

2.2.1. SECCION DE PROGRAMACION DE LOS CIRCUITOS

En esta sección, como ya dijimos anteriormente, se programan la Usart y el Timer para comunicación Asíncrona x64, 1200 baudios, 8 bits de longitud del carácter, sin paridad, y 2 bits de Stop.

Para ello utilizamos los siguientes ports (ver capítulo I de este mismo proyecto):

Timer:

- Port F3H: para seleccionar el contador que queremos programar, elegir el modo de la señal de salida, y especificar como se va a leer los bytes que se van a llevar al contador para obtener la frecuencia de salida.
- Port F1H: por este port se envían los bytes a llevar al contador para obtener la frecuencia de salida.

Usart:

- Port F7H: este port se utiliza para el envío de comandos a la Usart del canal serie 2, y para la lectura del "Status" de la misma.
- Port F6H: este port se utiliza para el envío de datos a la Usart, y para la lectura de los datos que llegan a través del canal serie a la misma.

Antes de hablar de la utilización de estos ports en el programa, conviene decir que en éste programa se utilizan las rutinas externas CI, CO, y CSTS, que deberán ser "Linkadas" a nuestro programa una vez ensamblado. Estas rutinas sirven para la lectura del teclado y el envío de datos a la pantalla. En particular, CSTS sirve para comprobar si se ha presionado alguna tecla, en cuyo caso el bit menos significativo del acumulador estará a "1" cuando retorna. CI nos devuelve el código ASCII de la última tecla pulsada en el acumulador. Y CO presenta en pantalla el carácter en código ASCII que se encuentra en el registro C cuando se hace la llamada.

Pues bien, el primer paso que se realiza en esta sección es Resetear la Usart (enviando un "1" en el séptimo bit por el port F7H, con lo cual esta retorna al Modo Programación. El siguiente paso es, por tanto, enviar por el port F7H la instrucción de programación definiendo la configuración de la comunicación (ver User's Manual MCS 85 de Intel).

A continuación programamos el Timer por el port F3H seleccionando el contador 1 (que es el que fija la velocidad de transmisión del canal serie 2), para que funcione en Modo 3 (señal cuadrada), e indicando además que el valor inicial del contador se lo vamos a proporcionar de la siguiente forma: primero se carga el byte menos significativo y a continuación el más significativo.

Por tanto, la siguientes instrucciones lo que hacen es enviar por el port F1H el valor a almacenar en el contador; primero el byte LSB y a continuación el MSB. El valor que se almacena en el contador es 10H, con lo cual la frecuencia de la señal de reloj de salida es 16 veces la de la señal de entrada. Por tanto, como la señal de reloj es de 1.22Mhz., la salida será de 76.8 Khz., que a su vez dividida por 64 en la Usart, dan una velocidad de transmisión de 1200 baudios.

Después de programados estos dos circuitos para el tipo de comunicación que vamos a usar, se entra en lo que es el protocolo (software) RS-232 C en sí.

En este caso particular, el protocolo es bastante sencillo:

- Primero se activa la línea DTR de la Usart enviando el comando 07H por el port F7H. Además, mediante este comando se habilitan los flags de transmisión (TxE) y de recepción (RxE) de ésta, pero esto no forma parte del protocolo en sí.
- A continuación se efectúa un retardo para dar tiempo a que la línea DSR del modem se active en respuesta a DTR.
- Después de efectuado el retardo se muestrea el byte de "Status" de la Usart a través del port F7H para comprobar si está activada la línea DSR (bit más significativo del byte de Status).
- Si la línea DSR no está activada, se presenta en pantalla el mensaje: CONECTAR EL MODEM (Rutina CEM), y se espera a que lo esté (mediante el bucle DOS).
- Una vez activada DSR se completa la primera fase del protocolo (ver el tema 9 del capítulo II de este mismo proyecto), y se activa a continuación la línea RTS de la Usart para solicitar al modem que se prepare para transmitir. Esto se hace enviando el comando 27H a través del port F7H a la Usart.

En realidad, RTS debería activarse cuando se empieza a transmitir realmente un carácter, y desactivarse a continuación después de enviando el mismo. Esto previene contra los errores de desbordamiento (Overrun). Ahora bien, como en este caso particular de conexión RS-232 C, uno de los dos elementos que entran en comunicación es una persona humana, se entiende que esta es capaz de detectar este tipo de errores, observando simplemente si falta algún carácter en la pantalla, y hará lo oportuno para corregirlo. Además, dada la baja velocidad de transmisión, limitada por el modem, es bastante difícil que se pierda algún carácter. Por todo esto, la línea RTS se activa al principio de la comunicación, y se mantiene activada durante el resto de esta.

Después de activada la línea RTS se entra en el bucle principal del programa.

2.2.2 SECCION DEL BUCLE PRINCIPAL (BUCLE)

En esta sección se efectúa un bucle sin fin en el que los pasos a efectuar son bastante simples.

Primero se comprueba si se ha presionado alguna tecla, en cuyo caso se salta a la rutina KEYB para leer la tecla pulsada y enviarla al ordenador. Para realizar esto, se hace una llamada a la rutina CSTS, y si el bit menos significativo del acumulador, retornado en la llamada, está a "1" se salta a la rutina KEYB.

Y a continuación se comprueba si se ha recibido algún dato del ordenador. Si es así, se salta a la rutina USAR para leer el dato recibido y presentarlo en pantalla.

La forma de comprobar si se ha recibido algún dato por la Usart es la siguiente: Se lee en byte de "status" de la Usart a través del port F7H, y se muestrea el segundo bit menos significativo (R_XRDY) de éste. Si este bit está a "1" quiere decir que se ha recibido un dato, y la Usart está lista para enviarlo a la CPU.

2.2.3 SECCION DE LECTURA DEL TECLADO (KEYB)

En ésta sección, que se identifica en el programa con la rutina KEYB, se lee la tecla presionada y se envía al ordenador.

Para leer la tecla pulsada, se hace una llamada a la rutina CI, y ésta nos devolverá en el acumulador el código ASCII de la tecla leída. A continuación se comprueba si la Usart está preparada para transmitir. Para ello, se lee el byte de status de la Usart a través del port F7H, y se muestrea el bit menos significativo (T_XRDY). Si este bit está a "1" indica que la Usart está preparada, y por tanto, el siguiente paso es enviar la tecla pulsada a través del port F6H al ordenador, y volver al bucle principal.

2.2.4 SECCION DE LECTURA DE DATOS RECIBIDOS DESDE EL ORDENADOR (USAR)

En esta sección, que se identifica claramente con la rutina USAR, se lee el dato que llega desde el ordenador y se presenta en pantalla.

Para efectuar esto, leemos el dato a través del port F6H. A continuación enmascaramos el octavo bit poniéndolo a cero mediante la instrucción ANI 7FH, por las razones explicadas anteriormente.

Después de hecho esto, se comprueba si el dato enviado por el ordenador es un "Enquiry" (05 Hex.), en cuyo caso se salta a la etiqueta CINCO y se contesta automáticamente con un "Acknowledge" (06 Hex.) retornando a continuación al bucle principal.

Si el dato no es un Enquiry, se presenta en pantalla (llevándolo al registro C y haciendo una llamada a la rutina externa CO), y se retorna al bucle principal.

Estas son, pues, las 4 secciones básicas en que está dividido el programa, identificándose claramente cada una de ellas. A continuación presentamos el listado del programa MODEM realizado en lenguaje Ensamblador.

LOC	OBJ	LINE	SOURCE STATEMENT
		1	*****
		2	; Este programa mantiene la comunicacion entre un Sistema de Desarro-
		3	; llo MDS 221 de Intel y un Ordenador de la serie HP-3000 a traves de
		4	; Modem. La comunicacion es: Serie, asincrona *64, 1200 baudios,
		5	; 8 bits de largo del caracter, sin paridad, y 2 bits de stop
		6	*****
		7	;
		8	;
		9	EXTRN CI,CO,CSTS
4000		10	ORG 4000H
4000	310046	11	PRINCP: LXI SP,4600H
		12	;
4003	3E40	13	MVI A,40H ; Se resetea la Usart
4005	D3F7	14	OUT 0F7H
4007	3ECF	15	MVI A,0CFH ; Programacion de la Usart del Canal Serie 2
4009	D3F7	16	OUT 0F7H
		17	;
400B	3E76	18	MVI A,076H ; Se programa el Timer para trabajar en Modo
400D	D3F3	19	OUT 0F3H ; 3 con una frecuencia de 76.8 Khz.
		20	;
400F	211000	21	LXI H,010H
4012	7D	22	MOV A,L
4013	D3F1	23	OUT 0F1H
4015	7C	24	MOV A,H
4016	D3F1	25	OUT 0F1H
		26	;
		27	;
4018	3E07	28	MVI A,07H ; Se activa la linea DTR
401A	D3F7	29	OUT 0F7H

LOC	OBJ	LINE	SOURCE STATEMENT
		30 ;	
401C	06FF	31	MVI B,0FFH
401E	05	32	DELAY: DCR B ; Se espera a que se active la linea DSR
401F	C21E40	33	JNZ DELAY ; en respuesta a la linea DTR.
		34 ;	
4022	DBF7	35	IN 0F7H ; Se comprueba si esta activada la linea
4024	E680	36	ANI 80H ; DSR
4026	CC6840	37	CZ CEM
4029	3E37	38	MVI A,37H ; Se activan las lineas DTR, RTS, y los
402B	D3F7	39	OUT 0F7H ; los flags R*E, y T*E.
		40 ;	
402D	CD0000	E 41	BUCLE: CALL CSTS ; Se comprueba si se ha
4030	0F	42	RRC ; presionado alguna tecla
4031	DC5240	43	CC KEYB ;
4034	DBF7	44	IN 0F7H
4036	E602	45	ANI 02H ; Se comprueba si estan llegando
4038	C43E40	46	CNZ USAR ; los datos
403B	C32D40	47	JMP BUCLE
		48 ;	
		49 ;	
403E	DBF6	50	USAR: IN 0F6H ; Se lee el dato llegado a la Usart
4040	E67F	51	ANI 7FH ; Se enmascara el octavo bit
4042	4F	52	MOV C,A ;
4043	FE05	53	CPI 05H ; Se comprueba si el caracter es "Enquiry"
4045	CA4C40	54	JZ CINCO
4048	CD0000	E 55	CALL C0 ; Se presenta el dato por pantalla.
404B	C9	56	RET
404C	0E06	57	CINCO: MVI C,06H ; Se contesta con el caracter "Acknowledge".
404E	CD5D40	58	CALL LAZ02
4051	C9	59	RET

LOC	OBJ	LINE	SOURCE STATEMENT
		60 ;	
4052	CD0000	E 61	KEYB: CALL CI ; Se lee el teclado
4055	4F	62	MOV C,A
4056	FE7F	63	CPI 7FH ; Se compara con la tecla RUBOUT
4058	C25D40	64	JNZ LAZ02
405B	0E08	65	MVI C,08H ; Si era RUBOUT se sustituye por BACKSPACE
405D	DBF7	66	LAZ02: IN 0F7H
405F	E601	67	ANI 01H ; Se comprueba si se puede transmitir por la
4061	CA5D40	68	JZ LAZ02 ; Usart
4064	79	69	MOV A,C
4065	D3F6	70	OUT 0F6H ; Salida del dato por la Usart.
4067	C9	71	RET
		72 ;	
4068	218640	73	CEM: LXI H,MODEM ; Se saca el comentario:
406B	7E	74	UNO: MOV A,M ; CONECTAR EL MODEM
406C	FE2A	75	CPI '*'
406E	CA7940	76	JZ DOS
4071	4F	77	MOV C,A
4072	CD0000	E 78	CALL CO
4075	23	79	INX H
4076	C36B40	80	JMP UNO
4079	DBF7	81	DOS: IN 0F7H
407B	E680	82	ANI 80H
407D	CA7940	83	JZ DOS
4080	0E0A	84	MVI C,0AH
4082	CD0000	E 85	CALL CO
4085	C9	86	RET
		87 ;	
155 4086	434F4E45	88	MODEM: DB 'CONECTAR EL MODEM',0DH,0AH,'*'
408A	43544152		

LOC	OBJ	LINE	SOURCE STATEMENT
408E	20454C20		
4092	4D4F4445		
4096	4D		
4097	0D		
4098	0A		
4099	2A		
4000		89	END PRINCP

PUBLIC SYMBOLS

DELAY A 401E
 USAR A 403E

EXTERNAL SYMBOLS

CI E 0000 CO E 0000 CSTS E 0000

USER SYMBOLS

BUCLE	A 402D	CEM	A 4068	CI	E 0000	CINCO	A 404C	CO	E 0000	CSTS	E 0000
DOS	A 4079	KEYB	A 4052	LAZ02	A 405D	MODEM	A 4086	PRINCP	A 4000	UNO	A 406B

ASSEMBLY COMPLETE, NO ERRORS

3. OBJETIVOS DEL PROGRAMA QUILEY

El objeto de éste programa es establecer todo el protocolo de comunicación entre un Sistema de Desarrollo Intellec MDS 221 y un Ordenador Hewlett-Packard de la serie HP-3000, así como la gestión y control de la impresora y la unidad de diskette del Sistema de Desarrollo.

Concretamente, las funciones que puede realizar son:

1. Transferencias de datos desde el teclado del Sistema de Desarrollo al ordenador.
2. Lectura de datos recibidos desde el ordenador y su transferencia a la pantalla.

Estas dos funciones, en particular, son las realizadas por cualquier terminal standard conectado a un ordenador.

3. Lectura de datos recibidos desde el ordenador y su transferencia a la impresora del Sistema de Desarrollo.
4. Lectura de datos recibidos desde el ordenador y su transferencia a la unidad de diskette del Sistema de Desarrollo.
5. Y por último, transferencia de datos desde un fichero del diskette del Sistema de Desarrollo al ordenador.

El manejo y control de todas estas funciones por parte del usuario se explicará más adelante.

El algoritmo, como puede verse en las figuras, se ha dividido en tres secciones fundamentales para un mejor entendimiento del mismo por parte del usuario.

La explicación de éste algoritmo, así como de sus características más fundamentales se realizará exhaustivamente en el siguiente apartado.

3.1 EXPLICACION DEL ALGORITMO

En el algoritmo se pueden identificar claramente tres secciones:

- Una primera sección en la que (1) se programan los circuitos a utilizar en la comunicación, y se inicializan el puntero del buffer y los flags; y (2), se establece el bucle principal ó central de la comunicación.
- La segunda sección se encarga tanto del almacenamiento en disco del Sistema de Desarrollo, como del control de la impresora.
- La tercera sección se encarga del envío de los datos desde un fichero del disco del Sistema de Desarrollo al ordenador HP-3000.

A continuación vamos a hacer el estudio detallado de cada una de las partes de éste algoritmo.

3.1.1. SECCION DE CONTROL DE LA COMUNICACION Y DEL TECLADO

En ésta sección, el primer paso a realizar es la programación de la Usart y del Timer del Sistema de Desarrollo para el modo de transmisión específico del ordenador.

Después de esto, se resetean los flags, y se inicializa el puntero del buffer. Vamos a explicar esto más detalladamente:

Tanto en el almacenamiento en disco, como con el uso de la impresora, se utiliza un buffer de memoria en el que se almacenan los datos recibidos desde el ordenador. Es el contenido de éste buffer el que, una vez lleno, bien se almacena en disco ó se envía a la impresora dependiendo del estado áctivo ó desactivado del flag de impresora ó del de disco definidos por nosotros en el programa.

Estos flags estan almacenados en una posición de memoria a la que llamamos FLAG en nuestro programa.

Para direccionar la posición dentro del buffer en la que vamos a almacenar cada carácter leído, utilizamos una variable a la que llamamos puntero del buffer (PBUFFER en nuestro programa).

Esta variable se inicializa al principio del programa con la dirección de la primera posición de memoria del buffer.

Después de inicializar el puntero del buffer entramos en el bucle central de la comunicación. En éste bucle se comprueba por un lado, si se ha presionado alguna tecla, y por otro si ha contestado el ordenador.

En caso de haberse presionado alguna tecla, se pueden dar las siguientes posibilidades:

- (1) Si es la tecla HOME, se pone la Usart en la condición BREAK y se vuelve al bucle central. Esta tecla HOME realiza la función de la tecla BREAK de los terminales de la serie HP 2640/5 de Hewlett-Packard, y su misión es abortar el proceso que se está realizando en el ordenador y devolver el control al Sistema Operativo de éste.
- (2) Si la tecla presionada es "Control de Impresora", se activa el flag correspondiente a la impresora, y se retorna al bucle principal.
- (3) Si la tecla es "Control de Disco" se comprueba si está activado el flag de Disco. Si no lo está, se activa, y se abre el fichero en el que queremos almacenar los datos recibidos y volvemos al bucle principal. Si el flag ya estaba activado volvemos al bucle principal.
- (4) Si es la tecla ESCAPE, se retorna al Sistema Operativo ISIS II del Sistema de Desarrollo.
- (5) Si la tecla presionada es el Control de transferir fichero desde el Sistema de Desarrollo al ordenador, se abre el fichero a leer, y se salta a la tercera sección del algoritmo (bifurcación C).
- (6) Si la tecla presionada no es ninguna de las anteriores, se envía a través de la Usart hacia el ordenador.

En caso de que haya contestado el ordenador, se saca por pantalla el dato recibido, y se comprueba si estan activados el flag de disco ó el de impresora. Si ninguno de los dos lo está, se vuelve al bucle principal.

Si alguno de los dos está activado, se almacena el carácter recibido en el buffer.

A continuación se comprueba si éste carácter recibido es uno de los caracteres del código ASCII D1 (11 Hexadecimal) ó D3 (13 Hex.). Estos caracteres especiales del código ASCII enviados por el ordenador los utilizamos para la parada automática de la impresión y/o el almacenamiento en el disco del Sistema de Desarrollo.

En caso de que el carácter leído sea uno de estos dos, saltamos por tanto, a la sección 2 (bifurcación D) del programa.

Si no es ninguno de estos dos caracteres, incrementamos el puntero del buffer y comprobamos si éste apunta a la última posición de memoria del buffer (buffer lleno), en cuyo caso se continua en la sección 2 (bifurcación D) para imprimir ó guardar en disco el contenido de éste buffer. Si no está lleno se vuelve al bucle principal para volver a leer el siguiente carácter.

3.1.2 SECCION DE IMPRESION Y ALMACENAMIENTO EN DISCO

El primer paso a realizar en ésta sección es parar la comunicación para evitar que el ordenador continúe enviando caracteres mientras se está imprimiendo o grabando en disco el buffer.

A continuación, comprobamos si está activado el flag de impresora, en cuyo caso imprimiremos el buffer carácter a carácter, bien hasta que hayamos imprimido la totalidad del contenido del buffer, ó bien hasta que detectemos un D1 ó un D3. Si se ha imprimido todo el buffer vamos a comprobar el flag de disco, y si llega un D1 ó un D3 desactivamos el flag de impresora (parada automática), y vamos a comprobar el flag de disco. Si está activado, se lee el buffer carácter a carácter y se almacena en el disco hasta que se haya grabado la totalidad del buffer, o bien hasta que el carácter leído sea un D1 ó un D3, en cuyo caso se cierra el fichero en el que hemos estado almacenando los datos y desactivamos el flag de disco (parada automática).

Si hemos almacenado todo el buffer, ó si se ha leído un D1 ó un D3 se continua la comunicación y se salta a la bifurcación B para inicializar de nuevo el puntero del buffer y volver al bucle principal.

3.1.3 SECCION DE ENVIO DE UN FICHERO DEL SISTEMA DE DESARROLLO AL ORDENADOR

El primer paso a efectuar es la lectura del primer bloque del fichero (que ha sido abierto para lectura) y pasarlo al buffer.

A continuación se comprueba si se ha presionado alguna tecla; en caso de que no, se comprueba si el ordenador ha contestado.

Si se ha presionado alguna tecla, puede darse alguna de las siguientes posibilidades:

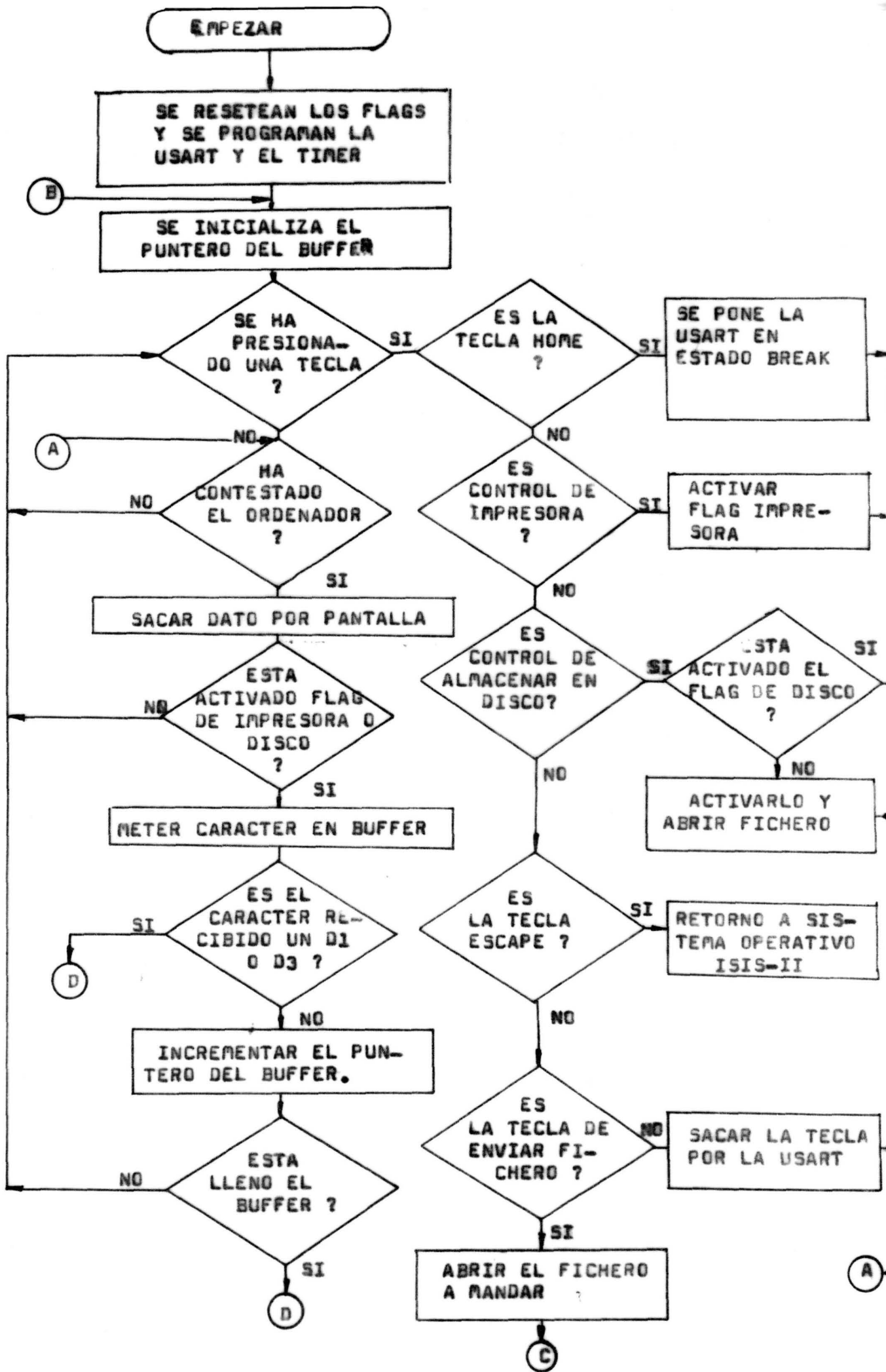
- (1) Si es la tecla ESCAPE se cierra el fichero abierto para la lectura (en el sistema de Desarrollo) y se retorna al ISIS II.
- (2) Si es la tecla HOME se pone la Usart en la condición BREAK, se cierra el fichero abierto para la lectura, y a continuación se salta a la bifurcación B para inicializar el puntero y volver al bucle principal.
- (3) Si es cualquier otra tecla, se ignora, y se vuelve a comprobar si ha contestado el ordenador.

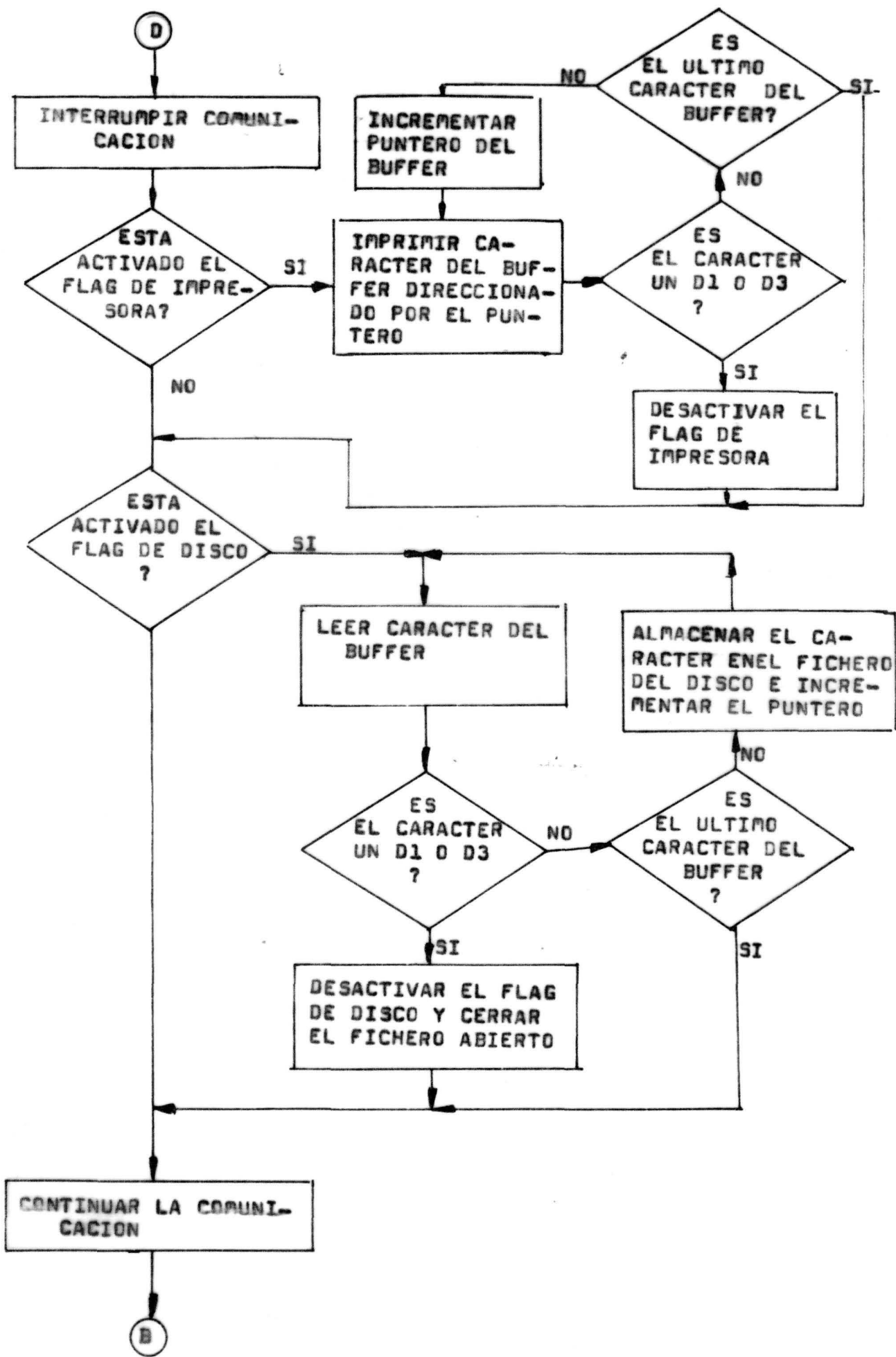
Si el ordenador ha contestado, sacamos el dato recibido por la Usart a la pantalla y volvemos a comprobar si se ha presionado alguna tecla.

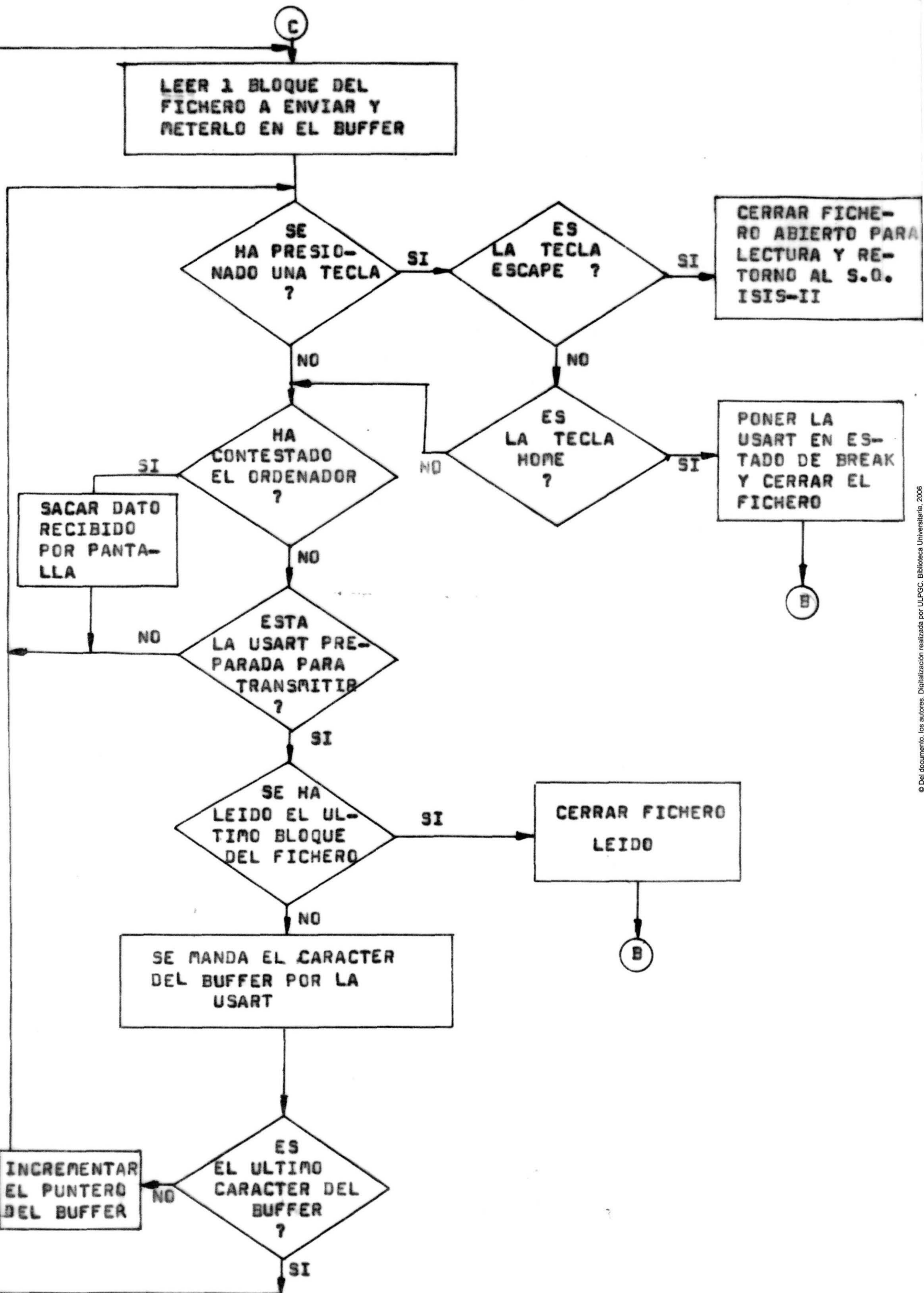
Si el ordenador no ha contestado, se comprueba si la Usart está preparada para transmitir; si no lo está comprobamos si se ha presionado una tecla de nuevo, y si la Usart está preparada miramos si se ha leído ya el último bloque del fichero.

En caso de haber leído el último bloque del fichero, lo cerramos y se salta a la bifurcación B para inicializar el puntero y volver al bucle principal.

Si no se ha leído el último bloque, mandamos el carácter direccionado por el puntero del buffer al ordenador; si éste carácter es el último del buffer leemos un nuevo bloque del fichero (salto a bifurcación C); en caso de que no lo sea, incrementamos el puntero y saltamos a comprobar si se ha presionado alguna tecla de nuevo.







3.2 DESCRIPCION DEL FUNCIONAMIENTO DEL PROGRAMA "QUILEY"

Como hemos visto anteriormente, éste programa realiza las funciones de control y gestión de la comunicación y de ficheros entre el Sistema de Desarrollo MDS 221 y el ordenador HP-3000.

Pasemos ahora a ver una descripción del funcionamiento del programa desde el punto de vista del usuario. Los pasos a seguir son:

- I) Cargar en el driver de discos del Sistema de desarrollo el disco conteniendo el programa QUILEY.
Y bajo control del Sistema Operativo ISIS ejecutar el programa escribiendo el comando:

- QUILEY

- II) El modem debe estar conectado al Sistema de Desarrollo; en caso contrario, en la pantalla se verá el mensaje:

CONECTAR EL MODEM

- III) Realizar la comunicación usual con el ordenador de manera análoga a como se haría utilizando un terminal standard.

- IV) Para el uso de la impresora, proceder siguiendo los siguientes pasos:

- + a) Dar una orden de listado del programa/fichero que deseemos imprimir.

- + b) Antes de pulsar la tecla "RETURN" presionar el "CONTROL I", -de éste modo la impresora queda preparada-; en el caso de no estar encendida la impresora ó estar en OFF LINE el programa pasa a un estado de espera hasta que reciba la confirmación de "impresora preparada"

Debe tenerse cuidado de no presionar el Control I accidentalmente sin la intención de imprimir y con la impresora apagada, puesto que el programa puede quedarse en un bucle permanente de espera.

- + c) Apretar la tecla RETURN; de éste modo el ordenador recibe la orden de listado y comienza a enviar los datos que seran imprimidos posteriormente.
 - + d) La impresora se desactivará automáticamente al detectar la llegada de los caracteres D1 ó D3 (indicando fin de listado).
- V) Para almacenar en la unidad de Disco del Sistema de Desarrollo, se deben seguir los siguientes pasos:
- + a) Escribir el comando de listado del programa/fichero, a almacenar en el disco, en el ordenador.
 - + b) Antes de pulsar la tecla RETURN, presionar el CONTROL K, y en pantalla aparecerá:

NOMBRE DEL FICHERO A CREAR EN EL S.D. ?
 - + c) Teclear el nombre del fichero en el que queramos almacenar los datos recibidos. Si ya existe un fichero con ese nombre, pierde la información que contenía, y si no, se crea un nuevo fichero en el disco.
 - + d) Pulsar la tecla RETURN. A partir de entonces el ordenador comenzará a mandar los datos a almacenar en el disco.
 - + e) El disco se desactivará automáticamente al recibir uno de los dos caracteres ASCII D1 ó D3 (indicando fin de listado).
- VI) Para enviar un fichero desde el Sistema de Desarrollo al ordenador pueden seguirse varios caminos distintos:
- 1) - Entrar en el EDITOR (del ordenador)
 - Teclear ADD
 - Pulsar el CONTROL A, y aparecerá en pantalla:

NOMBRE DEL FICHERO DEL S.D. A ENVIAR ?

- Teclar el nombre del fichero teniendo en cuenta las normas del ISIS.
- Presionar la tecla RETURN.
- Una vez finalizado el envío del fichero al ordenador, utilizar el comando KEEP de éste para guardarlo en un fichero permanente del HP-3000.

2) - Entrar en el Subsistema de copias de ficheros del ordenador: FCOPY.
(esperar hasta que el ordenador nos devuelva el signo >)

- Teclar:

FROM=;TO= (nombre del fichero);NEW

Donde:

+ (nombre del fichero) es el nombre del fichero a crear en el ordenador.

+ y NEW se utiliza para indicar que el fichero a crear es nuevo, es decir que no existe ya en un fichero permanente del ordenador.

- Pulsar el CONTROL A, y aparecerá en pantalla:

NOMBRE DEL FICHERO DEL S.D. A ENVIAR ?

- Teclar el nombre del fichero.
- Pulsar RETURN.

3) Entrar en el Subsistema BASIC.

- Pulsar el CONTROL A, y aparecerá en pantalla:

NOMBRE DEL FICHERO DEL S.D. A ENVIAR ?

- Teclar el nombre del fichero.
- Pulsar la tecla RETURN.

Una vez finalizado el envío del fichero al ordenador, utilizar el comando SAVE de éste para guardarlo en un fichero permanente del HP-3000.

Aparte de estos tres procedimientos pueden seguirse otros que gestionen el almacenamiento de información en el ordenador. Estos tres métodos son los que hemos comprobado por ahora, quedando abierta la posibilidad de usar otros subsistemas del ordenador HP-3000. Para ello basta tener en cuenta que el envío de datos desde el disco del Sist. de Desarrollo al ordenador se realiza de forma análoga al envío de datos por teclado al ordenador.

-VII) Otras teclas y Controles de interés:

+ HOME: Pulsando ésta tecla ponemos a la Usart en la condición BREAK, consiguiendo el mismo efecto que la tecla BREAK del terminal. (Si teníamos algún fichero abierto dentro del sistema de Desarrollo, se cerrará automáticamente).

+ ESCAPE: Pulsando ésta tecla, se interrumpe la ejecución del programa QUILLY y se devuelve el control del Sistema de Desarrollo al Sistema Operativo ISIS II.

3.3 EXPLICACION DEL PROGRAMA QUILLEY

El núcleo central de éste programa, que se encarga de la comunicación con el ordenador, es igual al del programa MODEM. Esto quiere decir que:

- Las características de la comunicación, tales como velocidad de Transmisión, etc. son iguales.
- Y el protocolo (software) "Enquiry-Acknowledge" es exactamente el mismo.

Por lo tanto, todo lo referente a ésta ya se explicó en el programa MODEM y considero que no es necesario volver a insistir sobre lo mismo.

Hablemos pues, ya, del programa en sí. En éste programa, como ya dijimos anteriormente, se distinguen claramente las secciones (tres) en que está dividido, cada una de las cuales está asociada a una etiqueta en el programa ensamblado.

En la primera sección se programan por un lado los circuitos a utilizar en la comunicación y se inicializan el puntero del buffer y los flags (asociada a la etiqueta PRINCP), y por otro se establece el bucle principal, el cual se identifica con la etiqueta BUCPRI.

La segunda sección se encarga tanto del almacenamiento en disco como del control de la impresora, y está asociada a la etiqueta RDIOGD.

Y por último, la tercera sección, que se encarga del envío de un fichero desde el Sistema de Desarrollo al ordenador se identifica con la rutina ENVFIC.

El funcionamiento básico del programa es bastante simple. El núcleo central ó bucle principal es igual al del programa MODEM. En este núcleo, las teclas presionadas se envían al ordenador (con la excepción de los controles que ya veremos), y los datos recibidos desde éste se sacan por lo pantalla. La única diferencia que hay con respecto a MODEM es que en caso de que se pulse alguno de los controles de impresora o de dis-

co, se activan los flags respectivos, y aparte de sacar los datos por pantalla se van almacenando en un buffer. Una vez lleno éste buffer, ó bien, cuando haya llegado uno de los caracteres ASCII D1 ó D3, indicando fin de listado, se para momentáneamente la comunicación con el ordenador enviando a éste un Control S, y se pasa a imprimir ó a almacenar en disco el contenido del buffer.

Una vez hecho esto, se vuelve a continuar la comunicación, ó bien, si llegó un D1 ó un D3 se desactivan los flags que estaban activados, y se continúa la comunicación. En caso de que el control pulsado sea para enviar un fichero por bloques (cuyo tamaño depende del tamaño del buffer), se van enviando al ordenador, después de leído el fichero a enviar.

3.3.1 ANALISIS DEL PROCESO DE COMUNICACION, Y DEL CONTROL DEL TECLADO DEL PROGRAMA QUILEY

Como ya hemos visto anteriormente, en éste programa se utiliza un buffer de 1 K bytes, cuyo tamaño está definido por la constante TAMBUF. Por lo tanto, simplemente con variar el valor de ésta constante, podemos cambiar el tamaño del buffer. Asimismo, utilizamos 2 posiciones de memoria consecutivas a las que llamamos puntero del buffer, direccionadas por la etiqueta PBUFFER, y en las que almacenamos la dirección del siguiente carácter a almacenar en el buffer.

Por otro lado, como se explica más detalladamente en el proyecto GESTION DE FICHEROS. APLICACION A LA CONEXION DE UN SISTEMA DE DESARROLLO CON UN ORDENADOR de Carlos Ley Bosch, utilizamos una posición de memoria a la que llamamos byte de Flags direccionada por la etiqueta FLAG, y en la que activamos o desactivamos una serie de flags para el control de la impresora y del disco.

El resto de las etiquetas del programa ya están explicadas en el propio listado del mismo, y por tanto, no voy a explicar las de nuevo aquí.

A continuación voy a hablar solamente de lo que concierne

al proceso de comunicación, del control de los flags y del buffer, y del control del teclado.

Como podemos ver en el programa, los primeros pasos a realizar son, aparte de poner a cero todos los flags, la programación de los circuitos a utilizar en la comunicación, tal y como se explicó en el programa MODEM.

Una vez hecho esto, se inicializa el puntero del buffer con la primera dirección del buffer.(INDBUF). Después de esto, se entra en el bucle principal (BUCPRI) para comprobar si se ha presionado alguna tecla y/o ha contestado el ordenador (esto ya se explicó en MODEM).

En caso de que se haya presionado alguna tecla se salta a la rutina KEYB. En esta rutina, primero se lee la tecla presionada y a continuación se comprueba si es alguna de las teclas especiales definidas por nosotros.

Si la tecla presionada es la tecla HOME, se envía a la Usart el comando 2FH (ver User's Manual MCS 85 de Intel) a través del port F7H para poner a ésta en estado de Break. A continuación, se hace una llamada a la rutina DELAY para efectuar un retardo, ya que el estado Break debe durar un tiempo mínimo para que el ordenador lo entienda. Después se vuelve a restituir la Usart enviando el comando 37H a través del port F7H, y por último, se vuelve al bucle principal.

En caso de que la tecla presionada sea la tecla de control de Impresora: CONTROL I (etiqueta TCIMP en el programa) se salta a la rutina CFLAGI en la que se activa el flag de impresora, y se vuelve al bucle principal para comenzar (o continuar) a llenar el buffer.

Si la tecla presionada fué la tecla de control de Escritura en el Disco: CONTROL K (etiqueta TCDISK, en el programa) se salta a la rutina CFLAGD en la que comprueba si está activado el flag de Disco. Si lo está, no hace falta volverlo a activar, y por tanto, se vuelve al bucle principal. Si no lo está, lo activa, y salta a la rutina de abrir fichero (ABFICH) que está detalladamente explicada en el proyecto GESTION DE FICHEROS.

APLICACION A LA CONEXION DE UN SISTEMA DE DESARROLLO CON UN ORDENADOR de Carlos Ley Bosch. Una vez abierto el fichero, vuelve al bucle principal para comenzar a llenar el buffer.

Si se presionó la tecla ESCAPE (etiqueta TRISIS) se salta a la dirección de memoria 08H para devolver el control al Sistema Operativo ISIS.

En caso de que la tecla presionada fuese la tecla de control de enviar fichero al ordenador: CONTROL A (TADISK en el programa) se salta a la Rutina ENVFIC para abrir el fichero a leer y enviarlo dividido en bloques al ordenador. Una vez hecho esto, vuelve a INDBUF para inicializar el puntero del buffer y volver al bucle principal. (Esto se explica más detalladamente en el proyecto de Carlos Ley Bosch -Octubre 83-)

Si la tecla pulsada fué RUBOUT, como el ordenador no entiende el código de ésta tecla, hay que cambiarlo por el código de BACKSPACE, y enviarlo al ordenador mediante la rutina TRES.

Y por último, si la tecla pulsada no era ninguna de estas teclas especiales, se envía al ordenador mediante la rutina TRES (similar a la rutina LAZO2 del programa MODEM), y se vuelve al bucle principal.

En caso de que el ordenador haya contestado, se salta a la rutina SPUSAR. En esta rutina, se lee el dato por el port F6H y se enmascara el octavo bit. A continuación se comprueba si es un Enquiry (05H), en cuyo caso se salta a la etiqueta CUATRO para contestar automáticamente con un Acknowledge (06H) y volver al bucle principal. Si no, se presenta en pantalla el dato y se comprueba si está activado algún flag. Si no lo está se vuelve al bucle principal. Pero si hay algún flag activado, se carga en el par de registros HL la dirección del buffer (contenido de PBUFER) donde se va a almacenar el carácter leído. Después de almacenado el carácter se comprueba si es alguno de los dos caracteres ASCII D1 ó D3. Estos caracteres son enviados desde el ordenador al terminal para indicarle a éste que ya terminó de ejecutar el último comando (independientemente del Subsistema del ordenador en que se esté).

Por lo tanto, si ha llegado alguno de estos caracteres, quiere decir que ya hay que parar de llenar el buffer y terminar de imprimir ó almacenar en disco éste último bloque. Para ello se salta a la rutina RDIOGD que está perfectamente explicada en el Proyecto de Carlos Ley Bosch -Octubre 83-. Solamente quiero destacar que estos caracteres especiales también se almacenan en el buffer, ya que después, a la hora de imprimir o grabar en disco se comprueba leyendo el buffer si llegaron estos caracteres, en cuyo caso se desactivan los respectivos flags.

En caso de que el carácter leído por la Usart no sea un D1 ó un D3 se incrementa el puntero del buffer (contenido de PBUFER) y se comprueba si ya está lleno el buffer. Para ello basta con comparar el contenido de PBUFER con la dirección de comienzo del buffer más el tamaño del buffer (TAMBUF) menos "uno". El restarle "uno" se debe a que cuando se salta a RDIOGD y se para la comunicación, hay que leer el último carácter que ha llegado en el tiempo que se tarda desde que llegó el último carácter hasta que se para la comunicación. Por eso se reserva una posición en el buffer para éste último carácter.

Si el buffer no está lleno, se vuelve al bucle principal para continuar llenándolo, y si lo está se salta a RDIOGD.

Esto ha sido, pues, todo lo referente a la comunicación y al control del teclado en el programa QUILEY. Todo lo referente al control de la impresora y al control de ficheros se estudia en el proyecto GESTION DE FICHEROS. APLICACION A LA CONEXION DE UN SISTEMA DE DESARROLLO XON UN ORDENADOR de Carlos Ley Bosch.

A continuación se presenta el listado del programa en lenguaje ensamblador, en el cual se han insertado también comentarios para identificar sobre el papel cada uno de los bloques del algoritmo.

LOC	OBJ	LINE	SOURCE STATEMENT
		1	*****
		2	Este programa mantiene la comunicacion entre un Sistema de Desarrollo In-
		3	tellec MDS 221 y un Ordenador de la serie HP-3000. La comunicacion es:
		4	Serie, Asincrona *64, velocidad de transmision 1200 baudios, 8 bits de lon-
		5	gitud del caracter, sin paridad, y con 2 bits de stop.
		6	
		7	Asimismo, realiza la gestion y control de la impresora, y de la Unidad de
		8	diskette del Sistema de Desarrollo. Para ello, se utilizan los siguientes
		9	controles:
		10	
		11	- CONTROL I: Esta tecla se utiliza para el manejo de la impre-
		12	sora. Para imprimir, basta con pulsar "CONTROL I" a partir
		13	del momento en que queramos comenzar a imprimir. La parada
		14	es automatica cuando se termina el listado.
		15	
		16	- CONTROL K: Esta tecla se utiliza para almacenar en el floppy
		17	del Sistema de Desarrollo cualquier programa o fichero del
		18	del ordenador. Para que realice esta funcion basta con pul-
		19	sar CONTROL I a partir del momento en que queramos almace-
		20	nar, si bien es recomendable que se haga despues de escri-
		21	bir el comando de listado del ordenador y antes de pulsar la
		22	tecla RETURN.
		23	
		24	- CONTROL A: Esta tecla se utiliza para enviar un fichero desde
		25	la unidad de floppy del Sistema de Desarrollo al ordenador.
		26	Para ello basta con entrar en el Subsistema del Ordenador
		27	en que queramos volcar el fichero, y cuando el ordenador
		28	este preparado esperando los datos, pulsar CONTROL A.

```

LOC  OBJ          LINE          SOURCE STATEMENT
                                30 ;          - HOME: Esta tecla se utiliza para enviar un condicion BREAK
                                31 ;          al ordenador.
                                32 ;
                                33 ;          - ESCAPE: Esta tecla se utiliza para abortar la ejecucion
                                34 ;          de este programa y devolver el control al Sistema Opera-
                                35 ;          tivo ISIS II del Sistema de Desarrollo.
                                36 ;
                                37 ;*****
                                38 ;
                                39          EXTRN  CI,CO,CSTS,ISIS
                                40          CSEG
                                41 ;
0000  42 DISABL  EQU      0DH
0005  43 ENABL  EQU      05H
00FF  44 CPUC   EQU      0FFH
00F8  45 PIOC   EQU      0F8H
00F8  46 PIOC   EQU      0F8H
00F9  47 PIOC   EQU      0F9H
00F9  48 PIOC   EQU      0F9H
0004  49 FO     EQU      04H
0002  50 IBF    EQU      02H
0001  51 OBF    EQU      01H
0001  52 DEVRDY EQU      01H
0014  53 LPTC   EQU      014H
0015  54 LSTC   EQU      015H
0011  55 D1     EQU      11H      ; Caracteres que controlan la parada automatica
0013  56 D3     EQU      13H
0009  57 TCIMP   EQU      09H      ; Ctl I: Tecla para activar la impresora
0008  58 TCDISK EQU      08H      ; Ctl K: Tecla para crear un fichero en el floppy
001B  59 TRISIS  EQU      1BH      ; ESCAPE: Tecla de Retorno a Isis

```

LOC	OBJ	LINE	SOURCE STATEMENT
007F		60	RBOUT EQU 7FH
0008		61	BSPACE EQU 08H
0013		62	CPCOMU EQU 13H ; Ctl S: control para parar la comunicacion
0011		63	CCCOMU EQU 11H ; Ctl Q: control para continuar la comunicacion
0000		64	OPEN EQU 0 ; Parametros para control de la Rutina Isis
0003		65	READ EQU 3
0004		66	WRITE EQU 4
0001		67	CLOSE EQU 1
000C		68	ERROR EQU 12
0010		69	BAUD EQU 010H ; Velocidad de Transmision
0400		70	TAMBUF EQU 1024 ; Longitud ,en bytes, del buffer
0001		71	TADISK EQU 01H ; Ctl A: Tecla para enviar un fichero desde el floppy
0019		72	MFF EQU 019H ; Ctl Y: Marca de Fin de fichero
001D		73	HOME EQU 1DH ; HOME: Tecla para enviar la condicion BREAK.
		74	;
		75	;
		76	PRINCP:
0000	218904	C 77	LXI H,FLAG ; Reseteo de los flags
0003	3600	78	MVI M,0H
0005	3E40	79	MVI A,40H ; Se resetea la Usart
0007	D3F7	80	OUT 0F7H
0009	3ECF	81	MVI A,0CFH ; Programacion de la Usart del Canal 2
000B	D3F7	82	OUT 0F7H
		83	;
000D	3E76	84	MVI A,076H ; Se programa el Timer para trabajar
000F	D3F3	85	OUT 0F3H ; en modo 3 con una frecuencia de
0011	211000	86	LXI H,BAUD ; 75.8 Khz.
0014	7D	87	MOV A,L
0015	D3F1	88	OUT 0F1H
0017	7C	89	MOV A,H

LOC	OBJ	LINE	SOURCE STATEMENT
001B	D3F1	90	OUT 0F1H
		91 ;	
001A	3E07	92	MVI A,07H ; Se activan las lineas DTR y los flags
001C	D3F7	93	OUT 0F7H ; R*E y T*E de la Usart.
		94 ;	
001E	21C203	C 95	LXI H,INICIA
0021	CD7303	C 96	CALL SDDPP
		97 ;	
0024	DBF7	98	IN 0F7H ; Se comprueba si esta activada la linea
0026	E680	99	ANI 80H ; DSR del modem.
0028	C23800	C 100	JNZ UNO
002B	218C03	C 101	LXI H,CEM
002E	CD7303	C 102	CALL SDDPP
0031	DBF7	103	DOS: IN 0F7H ; Si no esta activada, se espera a que lo este
0033	E680	104	ANI 80H ; despues de sacar el comentario por pantalla.
0035	CA3100	C 105	JZ DOS
		106 ;	
0038	3E37	107	UNO: MVI A,37H ; Se esta activada, se activan entonces: DTR,
003A	D3F7	108	OUT 0F7H ; RTS, R*E, y T*E.
		109 ;	
003C	218C04	C 110	INDBUF: LXI H,BUFFER; Inicializacion del puntero del buffer
003F	228A04	C 111	SHLD PBUFER
		112 ;	
0042	CD0000	E 113	BUCPRI: CALL CSTS ; Bucle principal:
0045	0F	114	RRC
0046	DC5300	C 115	CC KEYB ; - Si se pulsado una tecla salta a KEYB
0049	DBF7	116	RESPUE: IN 0F7H
004B	E602	117	ANI 02H
004D	C22701	C 118	JNZ SPUSAR ; - Si el ordenador ha contestado salta a SPUSAR
0050	C34200	C 119	JMP BUCPRI

```

LOC  OBJ          LINE      SOURCE STATEMENT
                                120 ;
                                121 ;
0053 CD0000   E   122 KEYB:  CALL    CI      ; Se lee la tecla pulsada.
0056 4F              123      MOV    C,A
0057 FE1D              124      CPI    HOME   ; Si no es la tecla HOME salta a KEYB1
0059 C26B00   C   125      JNZ    KEYB1
005C 3E2F              126      MVI    A,2FH  ; Si es la tecla HOME se pone a la Usart
005E D3F7              127      OUT   0F7H  ; en estado BREAK momentaneamente
0060 CD2403   C   128      CALL  DELAY
0063 3E37              129      MVI    A,37H
0065 D3F7              130      OUT   0F7H
0067 CD5802   C   131      CALL  CEFICH
006A C9              132      RET
006B FE09              133 KEYB1:  CPI    TCIMP  ; Si la tecla pulsada es Ctl I salta a
006D CA9100   C   134      JZ     CFLAGI ; CFLAGI
0070 FE0B              135      CPI    TCDISK ; Si es Ctl K salta a CFLAGD
0072 CA9C00   C   136      JZ     CFLAGD
0075 FE1B              137      CPI    TRISIS ; Si es ESCAPE retorna al ISIS.
0077 CA0800              138      JZ     08H
007A FE01              139      CPI    TADISK ; Si es Ctl A salta a ENVFIC.
007C CA7902   C   140      JZ     ENVFIC
007F FE7F              141      CPI    RBOUT  ; Si es Rubout, sustituye su codigo por.
0081 C28600   C   142      JNZ    TRES
0084 0E08              143      MVI    C,SPACE; el de Backspace
                                144 ;
                                145 ;*****
                                146 ;
                                147 ; Esta Rutina envia el caracter que hay en el registro C
                                148 ; hacia la Usart.
                                149 ;

```



```

LOC  OBJ          LINE      SOURCE STATEMENT
0086 DBF7          150 TRES:   IN      0F7H    ; - Se espera a que la Usart este preparada
0088 E601          151          ANI      01H    ; para transmitir.
008A CAB600       C   152          JZ       TRES
008D 79           153          MOV     A,C    ; - Salida del caracter que hay en el registro
008E D3F6         154          OUT     0F6H  ; C por la Usart
0090 C9           155          RET
156 ;
157 ;*****
158 ;
159 ;
160 ;*****
161 ;
162 ; Rutina de activacion del flag de impresora
163 ;
0091 3A8904       C   164 CFLAGI: LDA     FLAG    ; - Si se ha pulsado Ctl I, se activa
0094 F601         165          ORI     01H
0096 328904       C   166          STA     FLAG
0099 C34900       C   167          JMP     RESPUE
168 ;
169 ;*****
170 ;
171 ;
172 ;*****
173 ;
174 ; Esta rutina activa el flag de disco y abre un nuevo fichero
175 ;
009C 3A8904       C   176 CFLAGD: LDA     FLAG    ; - Si se ha pulsado Ctl K se comprueba si es-
009F E604         177          ANI     04H    ; ta activado el flag de disco.
00A1 C24900       C   178          JNZ     RESPUE ; - Si lo esta, se retorna
00A4 F604         179          ORI     04H    ; - Y si no lo esta, se activa, y se abre el
    
```

LOC	OBJ		LINE	SOURCE STATEMENT
00A6	328904	C	180	STA FLAG ;nuevo fichero
00A9	210200		181	LXI H,02H ; - Parametro para que el nuevo fichero sea
00AC	220F01	C	182	SHLD ACCES ; abierto para escritura
00AF	21A203	C	183	LXI H,NDF
			184	*****
			185	;
			186	;
			187	Rutina para abrir un fichero, estando el comentario
			188	a sacar en pantalla preguntando por el nombre del
			189	programa en los registros HL cuando se hace la llamada
			190	*****
			191	;
00B2	CD7303	C	192	ABFICH: CALL SDDPP
00B5	0610		193	MVI B,10H
00B7	211701	C	194	LXI H,NFILE
00BA	CD0000	E	195	CINCO: CALL CSTS ; - Se lee el nombre del fichero a abrir, y
00BD	0F		196	RRC ; se almacena en las posiciones de memoria
00BE	D2BA00	C	197	JNC CINCO ; a partir de NFILE
00C1	CD0000	E	198	CALL CI
00C4	4F		199	MOV C,A
00C5	FE7F		200	CPI RBOU
00C7	C2D400	C	201	JNZ CIN
00CA	2B		202	DCX H
00CB	04		203	INR B
00CC	0E08		204	MVI C,08H
00CE	CD0000	E	205	CALL CO
00D1	C3BA00	C	206	JMP CINCO
00D4	CD0000	E	207	CIN: CALL CO
00D7	79		208	MOV A,C
00D8	FE0D		209	CPI 0DH

LOC	OBJ		LINE	SOURCE STATEMENT
00DA	CAE300	C	210	JZ SEIS
00DD	05		211	DCR B
00DE	77		212	MOV M,A
00DF	23		213	INX H
00E0	C3BA00	C	214	JMP CINCO
00E3	3620		215	SEIS: MVI M,20H
00E5	23		216	INX H
00E6	05		217	DCR B
00E7	C2E300	C	218	JNZ SEIS
			219 ;	
			220 ;	
00EA	0E0A		221	MVI C,0AH ; Se saca un <LF> automatico por pantalla
00EC	CD0000	E	222	CALL CO
			223 ;	
00EF	0E00		224	MVI C,OPEN
00F1	110B01	C	225	LXI D,OBLK
00F4	CD0000	E	226	CALL ISIS
00F7	3A2501	C	227	LDA STATUS
00FA	B7		228	ORA A
00FB	C27F03	C	229	JNZ ERR
00FE	2A1501	C	230	LHLD OAFT
0101	229F02	C	231	SHLD RAFT
0104	225002	C	232	SHLD WAFT
0107	226A02	C	233	SHLD CAFT
010A	C9		234	RET
			235 ;	
010B	1501	C	236	OBLK: DW OAFT
010D	1701	C	237	DW NFILE
010F			238	ACCES: DS 2H
0111	0000		239	ECO: DW 0H

LOC	OBJ		LINE	SOURCE STATEMENT
0113	2501	C	240	DW STATUS
0115			241	OAFI: DS 2H
0117			242	NFILE: DS 14
0125			243	STATUS: DS 2H
			244	;
			245	;
			246	;
			247	*****
			248	;
			249	; Rutina para leer los datos que llegan desde el ordenador
			250	; y sacarlos por pantalla. En caso de que haya activado al-
			251	; gun flag, se almacena el dato en el BUFFER.
			252	;
			253	*****
0127	DBF6		254	SPUSAR: IN 0F6H ; - Se lee el dato
0129	E67F		255	ANI 7FH ; - Se resetea el octavo bit
012B	4F		256	MOV C,A
012C	FE05		257	CPI 05H ; - Si ha llegado un Enquiry (protocolo) se
012E	CA6201	C	258	JZ CUATRO ; salta a cuatro
0131	CD0000	E	259	CALL CO
0134	3A8904	C	260	LDA FLAG ; - Se saca el dato por pantalla, y si no hay
0137	E605		261	ANI 05H ; ningun flag activado se vuelve al bucle
0139	CA4200	C	262	JZ BUCPRI ; principal.
013C	2A8A04	C	263	LHLD PBUFER ; - Si hay activado algun flag se mete el dato
013F	71		264	MOV M,C ; en el buffer
0140	79		265	MOV A,C
0141	FE11		266	CPI D1 ; - Si el dato es un D1 o un D3 (parada automa-
0143	CA6A01	C	267	JZ RDIOGD ; tica), se salta a RDIOGD.
0146	FE13		268	CPI D3
0148	CA6A01	C	269	JZ RDIOGD

LOC	OBJ	LINE	SOURCE STATEMENT
014B	23	270	INX H ; - Se incrementa el puntero del buffer
014C	228A04	C 271	SHLD PBUFER
014F	118B08	C 272	LXI D,BUFFER+TAMBUF-1
0152	2A8A04	C 273	LHLD PBUFER
0155	7B	274	MOV A,E
0156	BD	275	CMP L ; - Si el buffer esta lleno se salta a
0157	C24200	C 276	JNZ BUCPRI ; RDIOGD, si no se vuelve al bucle
015A	7A	277	MOV A,D ; principal.
015B	BC	278	CMP H
015C	C24200	C 279	JNZ BUCPRI
015F	C36A01	C 280	JMP RDIOGD
0162	0E06	281 CUATRO:	MVI C,06H ; - Si llego un Enquiry se contesta automati-
0164	CD8600	C 282	CALL TRES ; camente con un Acknowledge
0167	C34200	C 283	JMP BUCPRI
		284 ;	
		285 ;	
016A	0E13	286 RDIOGD:	MVI C,CPCOMU; Se para la comunicacion
016C	CD8600	C 287	CALL TRES
016F	DBF7	288 HOLA:	IN 0F7H ; - Se lee el ultimo caracter que llego
0171	E602	289	ANI 02H ; a la Usart cuando se paro la comunica-
0173	CA6F01	C 290	JZ HOLA ; cacion, y se almacena en el buffer.
0176	DBF6	291	IN 0F6H
0178	E67F	292	ANI 7FH
017A	4F	293	MOV C,A
017B	CD0000	E 294	CALL CO
017E	2A8A04	C 295	LHLD PBUFER
0181	71	296	MOV M,C
0182	3A8904	C 297	LDA FLAG ; - Se leen los flags.
0185	E601	298	ANI 01H ; - Si esta activado el flag de impresora
0187	C49A01	C 299	CNZ IMPRIM ; se salta a IMPRIM

LOC	OBJ		LINE	SOURCE STATEMENT
018A	3A8904	C	300	LDA FLAG
018D	E604		301	ANI 04H ; - Si esta activado el flag de disco
018F	C40402	C	302	CNZ ALDISC ; se salta a ALDISC
0192	0E11		303	MVI C,CCCOMU; - Se continua la comunicacion y vuelve
0194	CD8600	C	304	CALL TRES ; a inicializar el puntero del buffer
0197	C33C00	C	305	JMP INDBUF ; y al bucle principal
			306 ;	
			307 ;	
			308 ;*****	
			309 ;	
			310 ; Rutina para imprimir el contenido del buffer	
			311 ;	
			312 ;*****	
			313 ;	
019A	118C04	C	314	IMPRIM: LXI D,BUFFER;
019D	EB		315	SIETE: XCHG ; - Se carga en HL la direccion de comienzo del buffer
019E	4E		316	MOV C,M ; - Se lee el caracter y se almacena en el registro C
019F	1615		317	MVI D,LSTC ; - Se cargan los comandos de control de la impresora
01A1	1E14		318	MVI E,LPTC
01A3	CD8D01	C	319	CALL OUTDVR
01A6	79		320	MOV A,C
01A7	FE11		321	CPI D1
01A9	CAC201	C	322	JZ SALIDA
01AC	FE13		323	CPI D3
01AE	CAC201	C	324	JZ SALIDA
01B1	23		325	INX H
01B2	EB		326	XCHG
01B3	2A8A04	C	327	LHLD PBUFER
01B6	23		328	INX H
01B7	7B		329	MOV A,E

LOC	OBJ		LINE	SOURCE STATEMENT
01B8	BD		330	CMP L
01B9	C29D01	C	331	JNZ SIETE
01BC	7A		332	MOV A,D
01BD	BC		333	CMP H
01BE	C29D01	C	334	JNZ SIETE
01C1	C9		335	RET
			336 ;	
01C2	3A8904	C	337	SALIDA: LDA FLAG
01C5	E6FC		338	ANI 0FCH
01C7	E6FC		339	ANI 0FCH
01C9	32B904	C	340	STA FLAG
01CC	C9		341	RET
			342 ;	
01CD	3E0D		343	OUTDVR: MVI A,DISABL
01CF	D3FF		344	OUT CPUC
01D1	DBF9		345	LAZ01: IN PIOS
01D3	E607		346	ANI FO OR IBF OR OBF
01D5	C2D101	C	347	JNZ LAZ01
01D8	7A		348	MOV A,D
01D9	D3F9		349	OUT PIOC
01DB	DBF9		350	LAZ02: IN PIOS
01DD	E607		351	ANI FO OR IBF OR OBF
01DF	FE01		352	CPI OBF
01E1	C2DB01	C	353	JNZ LAZ02
01E4	DBF8		354	IN PIOI
01E6	E601		355	ANI DEVRDY
01E8	CAD101	C	356	JZ LAZ01
01EB	DBF9		357	LAZ03: IN PIOS
01ED	E607		358	ANI FO OR IBF OR OBF
01EF	C2EB01	C	359	JNZ LAZ03

LOC	OBJ	LINE	SOURCE STATEMENT
01F2	7B	360	MOV A,E
01F3	D3F9	361	OUT PIOC
01F5	DBF9	362 LAZ04:	IN PIOS
01F7	E607	363	ANI FO OR IBF OR OBF
01F9	C2F501	C 364	JNZ LAZ04
01FC	79	365	MOV A,C
01FD	D3F8	366	OUT PIOC
01FF	3E05	367	MVI A,ENABL
0201	D3FF	368	OUT CPUC
0203	C9	369	RET
		370 ;	
		371 ;*****	
		372 ;	
		373 ;	
		374 ;	
		375 ;*****	
		376 ;	
		377 ;	Rutina para contar el numero de bytes a almacenar en el disco
		378 ;	
		379 ;*****	
0204	110000	380 ALDISC:	LXI D,0H ; - Se inicializa en DE el contador de bytes
0207	218C04	C 381	LXI H,BUFFER; - Se carga en HL la direccion de comienzo del buffer
020A	7E	382 OCHO:	MOV A,M ; - Se lee el caracter
020B	FE11	383	CPI D1
020D	CA1502	C 384	JZ VEINTE ; - Si es el caracter de parada automatica salta a
0210	FE13	385	CPI D3 ; VEINTE
0212	C22002	C 386	JNZ TREINT
0215	3A8904	C 387 VEINTE:	LDA FLAG ; - Se activa el flag de "Ultimo bloque de disco"
0218	F608	388	ORI 08H
021A	328904	C 389	STA FLAG

LOC	OBJ		LINE	SOURCE STATEMENT
021D	C33202	C	390	JMP NUEVE
0220	13		391	TREINT: INX D ; - Se incrementa el contador de bytes y se compara
0221	010004		392	LXI B,TAMBUF; con la longitud del buffer. Si son iguales
0224	79		393	MOV A,C ; salta a nueve
0225	BB		394	CMP E
0226	C22E02	C	395	JNZ CINQUE
0229	78		396	MOV A,B
022A	BA		397	CMP D
022B	CA3202	C	398	JZ NUEVE
022E	23		399	CINQUE: INX H ; - Se incrementa HL para leer el siguiente
022F	C30A02	C	400	JMP OCHO ; caracter
0232	EB		401	NUEVE: XCHG ; - Se almacena en CONESC el numero de bytes
0233	225402	C	402	SHLD CONESC ; a guardar en el disco
			403 ;	
			404 ;	
			405 ;*****	
			406 ;	
			407 ;	Rutina para escribir en el fichero abierto definido por su WAFT
			408 ;	el numero de bytes especificado por CONESC
			409 ;	
			410 ;*****	
			411 ;	
0236	0E04		412	ESCRIT: MVI C,WRITE
0238	115002	C	413	LXI D,WBLK
023B	CD0000	E	414	CALL ISIS
023E	3A2501	C	415	LDA STATUS
0241	B7		416	ORA A
0242	C27F03	C	417	JNZ ERR
0245	3A8904	C	418	LDA FLAG
0248	E608		419	ANI 08H

LOC	OBJ	LINE	SOURCE STATEMENT
024A	E608	420	ANI 08H
024C	C25802	C 421	JNZ CEFICH
024F	C9	422	RET
		423	;
		424	WBLK:
0250		425	WAFT: DS 2H
0252	8C04	C 426	DW BUFFER
0254		427	CONESC: DS 2H
0256	2501	C 428	DW STATUS
		429	;
		430	;
		431	*****
		432	;
		433	; Rutina para cerrar el fichero definido por su CAFT.
		434	;
		435	*****
0258	0E01	436	CEFICH: MVI C,CLOSE
025A	116A02	C 437	LXI D,CBLK
025D	CD0000	E 438	CALL ISIS
0260	3A2501	C 439	LDA STATUS
0263	B7	440	ORA A
0264	C27F03	C 441	JNZ ERR
0267	C36E02	C 442	JMP DIEZ
		443	;
		444	CBLK:
026A		445	CAFT: DS 2H
026C	2501	C 446	DW STATUS
026E	3A8904	C 447	DIEZ: LDA FLAG
0271	E6F3	448	ANI 0F3H
0273	E6F3	449	ANI 0F3H

```

LOC  OBJ          LINE      SOURCE STATEMENT
0275 328904      C   450          STA      FLAG
0278 C9          451          RET
452 ;
453 ;
454 ;*****
455 ;
456 ;          Rutina para enviar un fichero desde el Sistema de Desarrollo
457 ;          al Ordenador
458 ;
459 ;*****
460 ENVFIC:
0279 210300      461          LXI      H,03H      ; - Parametro para abrir el fichero para solo
027C 220F01      C   462          SHLD   ACCES      ; lectura
027F 216A04      C   463          LXI      H,NDFAE
0282 CDB200      C   464          CALL   ABFICH      ; - Se abre el fichero
0285 0E0D          465          MVI      C,0DH      ; - Se manda <CR> automatico a la Usart
0287 CD8600      C   466          CALL   TRES
028A CD2403      C   467          CALL   DELAY
468 ;
469 ;*****
470 ;
471 ;          Rutina para leer y volcar en el buffer el fichero
472 ;          definido por su RAFT, almacenandose el numero de bytes
473 ;          leidos en CONLEC
474 ;
475 ;*****
028D 0E03          476 LECFIC: MVI      C,READ
028F 119F02      C   477          LXI      D,RBLK
0292 CD0000      E   478          CALL   ISIS
0295 3A2501      C   479          LDA      STATUS
    
```

LOC	OBJ	LINE	SOURCE STATEMENT
0298	B7	480	ORA A
0299	C27F03	C 481	JNZ ERR
029C	C3AB02	C 482	JMP CUAREN
		483 ;	
		484 RBLK:	
029F		485 RAFT: DS	2
02A1	8C04	C 486	DW BUFFER
02A3	0004	487 RCNT: DW	TAMBUF ; Se especifica el numero de bytes a leer
02A5	A902	C 488	DW CONLEC
02A7	2501	C 489	DW STATUS
02A9		490 CONLEC: DS	2 ; La rutina Isis almacena aqui el numero de bytes
		491 ;	leidos realmente
		492 ;	
		493 ;*****	
		494 ;	
		495 ;	Rutina para enviar todo el contenido del buffer al ordenador
		496 ;	
		497 ;*****	
02AB	210000	498 CUAREN: LXI	H,0H ; Se inicializa el puntero del buffer
02AE	228A04	C 499	SHLD PBUFER
02B1	CD0000	E 500 LAZPRI: CALL	CSTS ; Lazo principal:
02B4	0F	501	RRC ; - Si se ha pulsado una tecla salta a SDT
02B5	DA4F03	C 502	JC SDT
02B8	DBF7	503	IN 0F7H ; - Si el ordenador ha contestado salta a
02BA	E602	504	ANI 02H ; LECDAT
02BC	C20103	C 505	JNZ LECDAT
02BF	DBF7	506	IN 0F7H ; - Y si la Usart no esta preparada para
02C1	E601	507	ANI 01H ; transmitir se vuelve al lazo principal
02C3	CAB102	C 508	JZ LAZPRI
02C6	2AA902	C 509	LHLD CONLEC ; Si el contador del numero real de bytes leidos

LOC	OBJ	LINE	SOURCE STATEMENT
02C9	7C	510	MOV A,H ; es cero se salta a EXIT
02CA	B5	511	ORA L
02CB	CA1903	C 512	JZ EXIT
02CE	2ABA04	C 513	LHLD PBUFER ; El contenido del puntero del buffer (PBUFER) se
02D1	EB	514	XCHG ; suma a la direccion de comienzo del buffer y se
02D2	218C04	C 515	LXI H,BUFFER; almacena en los registros HL
02D5	7D	516	MOV A,L
02D6	83	517	ADD E
02D7	6F	518	MOV L,A
02D8	7C	519	MOV A,H
02D9	8A	520	ADC D
02DA	67	521	MOV H,A
02DB	4E	522	MOV C,M ; - Se lee un caracter del buffer y se saca
02DC	CD8600	C 523	CALL TRES ; por la Usart
02DF	79	524	MOV A,C
02E0	FE0A	525	CPI 0AH ; - Si el caracter enviado es un LF se salta
02E2	CC2403	C 526	CZ DELAY ; a DELAY
02E5	2AA902	C 527	LHLD CONLEC ; - Si el puntero del buffer (PBUFER) no es
02E8	2B	528	DCX H ; igual al numero real de bytes leidos (CONLEC)
02E9	7B	529	MOV A,E ; se salta a SIGBLO para continuar con el siguien-
02EA	BD	530	CMP L ; te caracter.
02EB	C2F902	C 531	JNZ SIGCAR
02EE	7A	532	MOV A,D
02EF	BC	533	CMP H
02F0	C2F902	C 534	JNZ SIGCAR
02F3	CD2403	C 535	CALL DELAY ; - Si PBUFER=CONLEC se efectua un retraso para
02F6	C38D02	C 536	JMP LECFIC ; para volver a leer el siguiente bloque.
02F9	EB	537	SIGCAR: XCHG
02FA	23	538	INX H ; - Se incrementa el puntero del buffer.
02FB	228A04	C 539	SHLD PBUFER

LOC	OBJ		LINE	SOURCE STATEMENT
02FE	C3B102	C	540	JMP LAZPRI
			541 ;	
0301	DBF6		542	LECDAT: IN 0F6H ; - Se lee el dato enviado por el ordenador
0303	E67F		543	ANI 7FH
0305	4F		544	MOV C,A
0306	FE05		545	CPI 05H ; - Si es un Enquiry se salta a PROTOC
0308	CA1103	C	546	JZ PROTOC
0308	CD0000	E	547	CALL CO ; - Si no, se saca el dato por pantalla y se
030E	C3B102	C	548	JMP LAZPRI ; vuelve al lazo principal
0311	0E06		549	PROTOC: MVI C,06H ; - Si llego un Enquiry se contesta con un Ack-
0313	CD8600	C	550	CALL TRES ; nowledge y se vuelve al lazo principal
0316	C3B102	C	551	JMP LAZPRI
			552 ;	
0319	CD5802	C	553	EXIT: CALL CEFICH ; - Se cierra el fichero leído, y se envia al
031C	0E19		554	MVI C,MFF ; ordenador una marca de "Fin de Fichero" (Ctl Y).
031E	CD8600	C	555	CALL TRES ; Se vuelve a inicializar el puntero del buffer y al
0321	C33C00	C	556	JMP INDBUF ; bucle principal.
			557 ;	
			558 ;	
			559	*****
			560 ;	
			561 ;	Rutina para efectuar un retraso especificado por el valor del
			562 ;	par de registros DE, y que a su vez lee todos los datos que
			563 ;	puedan llegar desde el ordenador durante este tiempo
			564 ;	
			565	*****
			566 ;	
0324	D5		567	DELAY: PUSH D
0325	11FF2F		568	LXI D,2FFFH
0328	DBF7		569	DEL1: IN 0F7H

LOC	OBJ	LINE	SOURCE STATEMENT
032A	E602	570	ANI 02H
032C	CA3F03	C 571	JZ DEL2
032F	DBF6	572	IN 0F6H
0331	E67F	573	ANI 7FH
0333	FE05	574	CPI 05H
0335	CA4703	C 575	JZ DEL3
0338	4F	576	MOV C,A
0339	CD0000	E 577	CALL CO
033C	C32803	C 578	JMP DEL1
033F	1B	579 DEL2:	DCX D
0340	7B	580	MOV A,E
0341	B2	581	ORA D
0342	C22803	C 582	JNZ DEL1
0345	D1	583	POP D
0346	C9	584	RET
0347	0E06	585 DEL3:	MVI C,06H
0349	CD8600	C 586	CALL TRES
034C	C32803	C 587	JMP DEL1
		588 ;	
		589 ;	
		590 ;	
034F	CD0000	E 591 SDT:	CALL CI ; Esta rutina lee la tecla pulsada cuando se estaban
0352	FE1B	592	CPI TRISIS ; enviando datos desde un fichero al ordenador (solo
0354	C25D03	C 593	JNZ SDT1 ; lee las teclas ESCAPE y HOME).
0357	CD5802	C 594	CALL CEFICH
035A	C30800	595	JMP 08H ; - Si la tecla pulsada es ESCAPE, cierra el fichero
035D	FE1D	596 SDT1:	CPI HOME ; abierto y vuelve al ISIS.
035F	C2B102	C 597	JNZ LAZPRI
0362	3E2F	598	MVI A,2FH ; - Si es HOME, pone a la Usart en estado de BREAK,
0364	D3F7	599	OUT 0F7H ; cierra el fichero abierto, y vuelve a iniciali-

LOC	OBJ	LINE	SOURCE STATEMENT
0366	CD2403	C 600	CALL DELAY ; zar el puntero del buffer y al bucle principal.
0369	3E37	601	MVI A,37H
036B	D3F7	602	OUT 0F7H ; - Si es cualquier otra tecla, la ignora y continua
036D	CD5802	C 603	CALL CEFICH ; la ejecucion normal del programa.
0370	C33C00	C 604	JMP INDBUF
		605 ;	
		606 ;	
		607 ;	*****
		608 ;	
		609 ;	Rutina para presentar en pantalla cualquier cadena de caracteres,
		610 ;	cuya direccion de comienzo este especificada en el par de registros
		611 ;	HL, y cuyo ultimo caracter sea: "&"
		612 ;	
		613 ;	*****
		614 ;	
0373	7E	615 SDDPP:	MOV A,M
0374	FE26	616	CPI '&'
0376	08	617	RZ
0377	4F	618	MOV C,A
0378	CD0000	E 619	CALL CO
037B	23	620	INX H
037C	C37303	C 621	JMP SDDPP
		622 ;	
		623 ;	
		624 ;	*****
		625 ;	
		626 ;	Rutina para presentar en pantalla los errores de Isis cometidos en
		627 ;	la gestion de un fichero del disco.
		628 ;	
		629 ;	*****

LOC	OBJ	LINE	SOURCE STATEMENT
037F	0E0C	630	ERR: MVI C,ERROR
0381	118803	C 631	LXI D,EBLK
0384	CD0000	E 632	CALL ISIS
0387	C9	633	RET
		634	;
0388	2501	C 635	EBLK: DW STATUS
038A		636	DS 2H
		637	;
		638	;
		639	***** COMENTARIOS A PRESENTAR EN PANTALLA *****
		640	;
		641	;
038C	0D	642	CEM: DB 0DH,0AH,'CONECTAR EL MODEM',0DH,0AH,'&'
038D	0A		
038E	434F4E45		
0392	43544152		
0396	20454C20		
039A	4D4F4445		
039E	4D		
039F	0D		
03A0	0A		
03A1	26		
03A2	0D	643	NDF: DB 0DH,0AH,'NOMBRE DEL FICHERO A CREAR ? &'
03A3	0A		
03A4	4E4F4D42		
03A8	52452044		
03AC	454C2046		
03B0	49434845		
03B4	524F2041		
03B8	20435245		

LOC	OBJ	LINE	SOURCE STATEMENT
03BC	4152203F		
03C0	2026		
03C2	1B	644	INICIA: DB 1BH,4BH,1BH,4AH,'***** DESARROLLADO POR ANTONIO '
03C3	48		
03C4	1B		
03C5	4A		
03C6	2A2A2A2A		
03CA	2A2A2A2A		
03CE	2A2A2A2A		
03D2	2A2A2A2A		
03D6	20444553		
03DA	4152524F		
03DE	4C4C4144		
03E2	4F20504F		
03E6	5220414E		
03EA	544F4E49		
03EE	4F20		
03F0	5155494E	645	DB 'QUINTANA Y CARLOS LEY *****',0DH
03F4	54414E41		
03F8	20592043		
03FC	41524C4F		
0400	53204C45		
0404	59202A2A		
0408	2A2A2A2A		
040C	2A2A2A2A		
0410	2A2A2A2A		
0414	2A2A		
0416	0D		
0417	2A2A2A2A	646	DB '***** SISTEMA DE'
041B	2A2A2A2A		

LOC	OBJ	LINE	SOURCE STATEMENT
041F	2A2A2A2A		
0423	2A2A2A2A		
0427	2A2A2A2A		
042B	2A2A2A20		
042F	53495354		
0433	454D4120		
0437	4445		
0439	20444553	647	DB ' DESARROLLO PREPARADO *****',0DH,0AH,'&'
043D	4152524F		
0441	4C4C4F20		
0445	50524550		
0449	41524144		
044D	4F202A2A		
0451	2A2A2A2A		
0455	2A2A2A2A		
0459	2A2A2A2A		
045D	2A2A2A2A		
0461	2A2A2A2A		
0465	2A2A		
0467	0D		
0468	0A		
0469	26		
046A	0D	648	NDFAE: DB 0DH,0AH,'NOMBRE DEL FICHERO A LEER ? &'
046B	0A		
046C	4E4F4D42		
0470	52452044		
0474	454C2046		
0478	49434845		
047C	524F2041		
0480	204C4545		

LOC OBJ LINE SOURCE STATEMENT

0484 52203F20
0488 26

649 ;

650 ;

651 ;***** ZONA DE MEMORIA *****

652 ;

0489 653 FLAG: DS 1
048A 654 PBUFER: DS 02H
048C 655 BUFFER: DS TAMBUF

656 ;

0000 C 657 END PRINCP

BUFFER C 048C
CFLAGI C 0091
CONESC C 0254
D1 A 0011
DIEZ C 026E
ERR C 037F
HOME A 001D
KEYB1 C 006B
LECFIC C 028D
NUEVE C 0232

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

CI E 0000 CO E 0000 CSTS E 0000 ISIS E 0000

USER SYMBOLS

ABFICH C 00B2	ACCES C 010F	ALDISC C 0204	BAUD A 0010	BSPACE A 0008	BUCPRI C 0042
CAFT C 026A	CBLK C 026A	CCCOMU A 0011	CEFICH C 0258	CEM C 038C	CFLAGD C 009C
CI E 0000	CIN C 00D4	CINCO C 008A	CINCUE C 022E	CLOSE A 0001	CO E 0000
CONLEC C 02A9	CPCOMU A 0013	CPUC A 00FF	CSTS E 0000	CUAREN C 02AB	CUATRO C 0162
D3 A 0013	DEL1 C 0328	DEL2 C 033F	DEL3 C 0347	DELAY C 0324	DEVRDY A 0001
DISABL A 000D	DOS C 0031	EBLK C 0388	ECO C 0111	ENABL A 0005	ENVFIC C 0279
ERROR A 000C	ESCRIT C 0236	EXIT C 0319	FLAG C 0489	FO A 0004	HOLA C 016F
IBF A 0002	IMPRIM C 019A	INDBUF C 003C	INICIA C 0302	ISIS E 0000	KEYB C 0053
LAZO1 C 01D1	LAZO2 C 01DB	LAZO3 C 01EB	LAZO4 C 01F5	LAZPRI C 02B1	LECDAT C 0301
LPTC A 0014	LSTC A 0015	MFF A 0019	NDF C 03A2	NDFAE C 046A	NFILE C 0117

OAFI	C 0115	OBF	A 0001	OBLK	C 010B	OCHO	C 020A	OPEN	A 0000	OUTDVR	C 01CD
PIOC	A 00F9	PIOI	A 00F8	PIOO	A 00F8	PIOS	A 00F9	PRINCP	C 0000	PROTCC	C 0311
RBLK	C 029F	RBOUT	A 007F	RCNT	C 02A3	RDIOGD	C 016A	READ	A 0003	RESPUE	C 0049
SDDPP	C 0373	SDT	C 034F	SDT1	C 035D	SEIS	C 00E3	SIETE	C 019D	SIGCAR	C 02F9
STATUS	C 0125	TADISK	A 0001	TAMBUF	A 0400	TCDISK	A 000B	TCIMP	A 0009	TREINT	C 0220
TRISIS	A 001B	UNO	C 003B	VEINTE	C 0215	WAFT	C 0250	WBLK	C 0250	WRITE	A 0004

ASSEMBLY COMPLETE, NO ERRORS

PBUFER C 048A
RAFT C 029F
SALIDA C 01C2
SPUSAR C 0127
TRES C 0086

BIBLIOGRAFIA

- "HARDWARE REFERENCE MANUAL". Intellec Series II
Microcomputer Development System.
- "USER'S MANUAL MCS 85". Intel
- "USING THE 8251 UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS
RECEIVER TRNSMITTER". Nota de Aplicación AP-16
de Intel.
- "INTRODUCTION TO LOCAL AREA NETWORKS". Digital Equip-
ment Corpotation. 1982.
- "TELEINFORMSTICA Y REDES DE COMPUTADORES". Editado
por MARCOMBO Boixareu Editores. Coordinado por
A. Alabau. 1982.
- "DESIGN AND ANALYSIS OF COMPUTER COMMUNICATINS NET-
WORKS". Vijay Ahuja. Editado por McGraw-Hill.1982.
- "DATA COMMUNICATIONS FOR MICROCOMPUTER". E. Nichols,
J. Nichols, y K. Musson. Editado por McGraw-Hill.
1982.