

ESCUELA UNIVERSITARIA POLITECNICA DE LAS PALMAS DE GRAN CANARIA

TITULO : REDES DE ORDENADORES.APLICACION A UN MODULO DE
----- ENSEÑANZA CON APPLE II

Autor:

Juan Manuel Silva Silva

Tutor:

Sebastian Suárez Gil

PREFACIO

La gran proliferación de computadores y ordenadores personales está a la orden del día , y la principal razón es sin duda el abaratamiento de los costes. Al mismo tiempo están surgiendo las nuevas redes de ordenadores y por supuesto las REDES LOCALES .

Por ello , en la enseñanza al igual que en otras tantas ramas de la industria se va introduciendo cada vez más la informatización y telecomunicación .

Luego de ahí que la idea de este trabajo estriba en conseguir una comunicación real entre alumnos y profesor a través de un ordenador personal como es el APPLE II .

Entonces en la primera parte del trabajo se plantean conceptos básicos relacionados con las redes de ordenadores y redes locales , dando una idea general y viendo también los distintos protocolos de acceso posibles.

En la segunda parte se plantea el sistema en sí . A nivel software el desarrollo del Apple II como terminal y a nivel hardware el establecimiento del protocolo de acceso que hace posible la comunicación entre los distintos Apples con el Apple principal.

Agradecimiento:

- al laboratorio de ordenadores
- al laboratorio de electrónica
- a los APPLE II

y en general gracias por su colaboración a todos aquellos que de una manera u otra han hecho posible la implementación de este trabajo fin de carrera .

Las Palmas de Gran Canaria , 15 de Octubre de 1986 .

INDICE.

PRIMERA PARTE

Capítulo 1 : LAS COMUNICACIONES DE DATOS3

Introducción a las redes de comunicaciones de datos --
arquitectura de los sistemas distribuidos -- servicios de
transmisión de datos -- estructura física de los circuitos
para transmisión de datos.

Capítulo 2 : REDES LOCALES.....24

Conceptos -- topologías en redes locales -- reparto de
recursos del sistema -- técnicas de compartición de un medio
de comunicación -- técnicas de compartición -- técnicas de
contienda -- técnicas de reserva -- técnicas híbridas
contienda-reserva -- técnicas de selección -- técnicas de
paso de testigo .

SEGUNDA PARTE

Capítulo 3 : ESTRUCTURACION DE UN SISTEMA DE ORDENADORES..43

Estructura software -- estructura hardware.

Capítulo 4 : ESTRUCTURA A NIVEL SOFTWARE.....50

Communications Interface Card -- Serial Interface Card --
introducción a la tarjeta SIC . trabajo realizado con la
tarjeta SIC .Conclusiones -- Super Serial Card --
introducción a la tarjeta SSC -- trabajo realizado con la
tarjeta SSC . Conclusiones .

Capítulo 5 : ESTRUCTURA A NIVEL HARDWARE.....78

Especificación del módulo de enseñanza informatizado --
elección de las distintas partes del sistema -- modo de

funcionamiento .

TERCERA PARTE : APENDICES

Apèndice I : MANUAL DE USO DEL APPLE II TRABAJANDO COMO
TERMINAL.....91

Conceptos -- uso de comandos.

Apèndice II : LENGUAJE ENSAMBLADOR DEL 6502 . GRUPO DE
INSTRUCCIONES.....96

Registros CPU y flags de estado -- modos de direccionamiento
del 6502 -- grupo de instrucciones del 6502 -- dispositivos
de E/S (UART).

Apèndice III : EL MODELO DE CONEXION RS-232-C.....120

Introducción -- bases de la conexión -- acoplamiento --
compatibilidad RS-232-C -- entrada / salida del procesador --
la UART y la conexión RS-232-C -- definiciones lógicas --
margen de ruidos -- orden de transmisión de los bits -- marca
y espacio -- niveles lógicos de control y acoplamiento --
reglas básicas RS-232-C.

Apèndice IV : LISTADOS DE SOFTWARE

Apèndice V : ESQUEMAS HARDWARE

BIBLIOGRAFIA

PRIMERA PARTE

CAPITULO 1

CAPITULO 1: LAS COMUNICACION DE DATOS

Introducción a las Redes de Comunicación de Datos

Hoy día es de vital importancia la posesión y distribución de información , así como la capacidad de comunicación de los hombres .Pero no basta con procesar los datos ,hay que poder acceder a ellos ,transformarlos , divulgarlos , etc, y para ello se requiere la capacidad de comunicación de datos.

En general Sistemas de Comunicación de Datos son todos aquellos sistemas informáticos cuyos procesadores y terminales no están en un mismo lugar y la información va a través de algún sistema de telecomunicación entre todos ellos. Es palpable la tendencia actual de la arquitectura informática hacia el multiproceso en todos los equipos , por lo que , en sentido estricto, se puede hablar en todos ellos de comunicación de datos .

Los objetivos que se persiguen cuando se decide comunicar datos a distancia , se pueden ver como objetivos empresariales convencionales: organizativos (coordinación, unificar métodos y procedimientos , aumentar la capacidad de expansión , centralización de datos y decisiones, etc), económicos (ahorros de equipos ,etc) y de servicio (rapidez , disponibilidad , comodidad , etc).

De estos factores el más importante es sin duda , el tipo de servicio en su concepto caudal versus urgencia . Aparecen así servicios clásicos de :proceso remoto en batch , proceso

remoto interactivo , sistemas transaccionales sobre bases de datos , sistemas de control de procesos , junto a servicios más actuales como el del correo electrónico en sus versiones documentales , con imagen estática y con mensajes asociados , los servicios de archivos documentales , bases de datos de acceso público , servicios de información , venta o banca doméstica , etc.

Las redes de comunicaciones de datos se hacen e instalan para dar uno de estos servicios , y a veces varios .

El soporte básico de la comunicación de datos son las redes públicas de telecomunicación y , en especial la red telefónica . El servicio telefónico , por su ubicuidad y coste , ha sido casi el único medio de comunicación de datos y en varios países lo sigue siendo . La prestación inicial ha sido transmitir datos a través de los enlaces telefónicos conmutados o a través de circuitos telefónicos punto a punto dedicados permanentemente al enlace entre los dos equipos del usuario . Las compañías explotadoras han recibido simultáneamente el impacto de la demanda de los circuitos de transmisión de datos y el impacto de la revolución generada por la tecnología de estado sólido.

Las redes de comunicación de datos son todavía un aspecto secundario de innovación en las compañías de telecomunicación. Hasta la fecha los criterios de renovación venían impuestos independientemente por la compañía explotadora , y el predominio de la prudencia sobre la

innovación era casi absoluto . Una de las excepciones a esto está en el área de los servicios de transmisión de datos.

La comunicación de datos utiliza los servicios de las compañías de telecomunicación .Estos servicios pueden ser: sólo de transmisión , a través de una red especial ;o bien un servicio de comunicaciones hacia una aplicación de gran público .

Actualmente todos los países desarrollados , además del servicio de alquiler de circuitos , tienen una red con tecnología de paquetes o planes para tenerla en breve , con interfaz internacional normalizado.

Algunos aspectos que van a dar que hablar en la próxima década son :

- Por una parte el impacto de los micros . De ello , de la gran venta de ordenadores personales que se están dando se llega a la conclusión de que los servicios públicos de comunicaciones , no tienen una estructura técnica y económica capaz de atender a la demanda de comunicación de datos que va a surgir en la próxima década.

- Otro impacto desconocido va a ser el de las redes locales en liza con las centralitas telefónicas privadas . Redes locales con voz o centralitas con terminales informáticos , híbridos y variantes poblarán la Tierra y unas harán desaparecer a las otras.

Las soluciones vendrán de la mano de las fibras ópticas , de los satélites , de los servicios móviles ,de las novedades telemáticas , de la integración de funciones de

comunicaciones en sistemas operativos y , sobre todo , de la mano de los expertos , de su imaginaci3n y de su capacidad de previsi3n y organizaci3n.

1. ARQUITECTURA DE LOS SISTEMAS DISTRIBUIDOS

La arquitectura de la tecnologia ha acercado a los electr3nicos a la inform3tica . O sea el profesional de la electr3nica se ha acercado a la inform3tica , y de hecho en el camino que ha tenido que recorrer le ha sido necesario adquirir y dominar conceptos de arquitectura de computadores, y de ingenieria de software , conceptos todos ellos reservados antano a lo que se consideraba como una 3lite de profesionales de la inform3tica .

La causa fundamental de este fen3meno ha sido sin duda alguna , la presi3n econ3mica : el microprocesador se utiliza porque es barato .

De todas maneras las bases de la arquitectura de sistemas distribuidos est3 en plena evoluci3n , por lo que todavfa no puede afirmarse que exista un cuerpo de doctrina estable en torno al tema .

1.1 Clasificaci3n.

El t3rmino sistema distribuido se utiliza para denominar indistintamente a diferentes clases de sistemas inform3ticos, en los que la potencia de tratamiento de la informaci3n se encuentra repartida en el espacio y que por diferentes motivos han aparecido como una alternativa a aquellos sistemas , en los que la potencia de tratamiento de la

información se encuentra concentrada en un único computador. En la figura 1.1 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 un mecanismo de Comunicación e Interconexión (MCI) .

En una clasificación a groso modo tenemos :

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

-a) Redes de computadores

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

Los trabajos en el campo de las redes de computadores parten de los recursos ya existentes , luego en un principio lo que se utiliza es la red telefónica .

Durante la última década , los fabricantes de equipos informáticos ha ido desarrollando arquitecturas para la realización de redes de computadores , ya sea proporcionando todos los elementos para construir la red , o 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 .

-b) Redes locales

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

El tema de las redes locales de computadores es hoy día un campo de conocimientos en completa evolución sin tener el recorrido necesario que tienen otros campos de la informática, es decir la disponibilidad de una serie de productos comercializados tal como en las redes de computadores.

-c) Sistemas multicomputadores

El fenómeno no es nuevo y así tenemos que ya en los 60 , las máquinas clásicas tenían unidades especializadas tanto en la manipulación de periféricos como en la gestión de comunicaciones .

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

-d) 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 , ha estado en la mente del arquitecto de computadores desde mucho antes de la

aparición del microprocesador . En muchos casos estos esfuerzos para la realización de "máquinas paralelas" se han visto superados por la propia evolución de la tecnología al permitir la realización de máquinas secuenciales potentes y rápidas .

Por otro lado ya hay información de la realización de componentes integrados complejos basados en la utilización de más de una unidad de proceso dedicada a la realización de funciones especializadas como solución para aumentar la potencia de tratamiento de información de un componente , sistemas que podríamos catalogar dentro de la categoría de los multimicroprocesadores .

1.2 Componentes de un sistema distribuido.

Un sistema informático desde el punto de vista del usuario estará constituido por un conjunto de elementos (hardware o software) , capaces de realizar conjuntamente , una determinada función orientada a la resolución de un determinado problema .

Dichos elementos serán , bien tareas (programas) , bien recursos (periféricos , ficheros) entre los que podrán establecerse determinadas relaciones . Por ejemplo comunicación entre tareas (para intercambiar información al objeto de resolver alguna función determinada) , o comunicaciones entre tareas y recursos (para que las primeras utilicen los recursos de las segundas) .

En un sistema centralizado , es decir basado en un

computador , la responsabilidad del establecimiento de dichas relaciones corresponde generalmente al sistema operativo . Un sistema operativo estará constituido básicamente por un conjunto de programas que se ejecutan en el propio computador y que realizan dicho servicio de comunicación entre los elementos mencionados .

Desde el punto de vista del usuario un sistema informático distribuido se verá igual que cualquier otro sistema informático , es decir , estará formado por un conjunto de elementos (tareas , recursos) entre los cuales podrán establecerse relaciones . La particularidad consistirá en que podrán estar ubicados en máquinas distintas .

El mecanismo que permita materializar dichas relaciones estará formado básicamente por un conjunto de elementos software (programas, a los que se denomina sistema operativo distribuido) , y además por un conjunto de elementos hardware y/o software que constituye el mecanismo de comunicación entre los elementos de tratamiento de información .

El término arquitectura de un sistema distribuido se utiliza habitualmente para designar su estructura y su funcionamiento , que son las herramientas utilizadas para poner en comunicación los sistemas informáticos distribuidos, con el objeto de permitir su cooperación .

La materialización de relaciones entre los elementos de un sistema distribuido implica el establecimiento de enlaces

entre ellos , con objeto de permitir el intercambio de información . Existen dos caminos :

- La realización de un enlace implica la utilización de una vía de comunicación a la que puede darse el nombre de camino lógico (o conexión).

- Por otra parte , en un determinado sistema , un camino lógico deberá materializarse utilizando los medios físicos existentes en el sistema .

1.3 Materialización de un camino lógico

El establecimiento de un camino lógico entre elementos de un sistema distribuido implicará la utilización de los caminos físicos integrantes del mecanismo de comunicación e interconexión del sistema . (figura 1.2)

Si consideramos el problema de las formas de utilizar un único camino físico podremos encontrar a su vez dos soluciones extremas : la primera consistirá en utilizarlo para la realización de un único camino lógico , la segunda supondrá permitir que sea utilizado de forma compartida , esto es multiplexando sobre él diferentes caminos lógicos .

En cuanto a las características estructurales de los caminos físicos que se utilizan para la materialización de los caminos lógicos de conmutación , es decir lo que habitualmente se denomina topología del sistema , damos cuatro tipos de soluciones que comentamos a continuación.

- Interconexión total

Se ve que una posible solución consiste en

interconectar directamente todos los ETI del sistema mediante un conjunto de caminos físicos que los enlacen dos a dos .

En este caso todos los caminos lógicos entre cada par de ETI , compartirían el mismo camino físico , directo entre ellos .

El coste hace que este tipo de interconexión esté limitado a los casos en que , bien existan pocos ETI , o que las distancias de separación sean cortas .

- Interconexión irregular

La realización de redes de computadores situados a distancias considerables plantea la necesidad de minimizar los costes de comunicación entre las máquinas , luego lo que se hace es disponer de un número mínimo de caminos físicos , que a través de nodos de interconexión , permitan establecer de forma indirecta , los diferentes caminos lógicos necesarios en el sistema . Al mecanismo de comunicación e interconexión en este tipo de sistemas se le suele denominar subred .

- Interconexión tipo bus

Como se puede suponer este tipo de topología estriba en el establecimiento de todos los caminos lógicos a través de un único camino físico bidireccional : el bus .

El problema fundamental en este tipo de sistemas reside básicamente en la asignación de este recurso (el bus) a los diferentes usuarios que deseen utilizarlo .

La solución tipo bus compartido , bien paralelo , bien serie , se utiliza en la realización de sistemas multicomputadores .

- Interconexión en anillo

En este caso los diferentes caminos lógicos se realizan sobre un único camino físico que interconecta a los diferentes elementos del sistema, cerrándose sobre sí mismo.

El anillo está constituido, pues, por un conjunto de nodos, enlazados mediante conexiones punto a punto. Las funciones que realizan dichos nodos pueden revestir diferentes grados de complejidad según el tipo de sistema que se trate.

1.4 Transmisión de la información

La comunicación entre sistemas informáticos aparece cuando ya existía una amplia tradición y experiencia en las comunicaciones tanto analógicas como digitales, luego de ello se deduce que el inicio de las comunicaciones entre sistemas informáticos esté influenciado por toda esta experiencia.

Se sabe que en la comunicación entre un terminal y un computador los elementos básicos de información transmitidos son los códigos asociados al juego de caracteres del teclado y de la unidad de presentación de la información en dicho terminal. Habitualmente en dicha comunicación se utiliza el código ASCII. También hay que decir que la transmisión entre dos sistemas informáticos puede realizarse según transmisión asíncrona o síncrona.

- En transmisión asíncrona cada elemento de información se transmite individualmente, acompañado de un conjunto de 2 a 3 bits de sincronismo, lo que constituye la unidad de información transmitida.

- Un equipo terminal de datos puede generar información en forma de carácter y transmitirlos en modo síncrono , para lo cual formará un bloque de N caracteres que acompañará con las informaciones de control apropiadas según esté previsto en el formato de trama del procedimiento de transmisión utilizado .

También existe la transmisión en paquetes . La ISO (Organización Internacional de Normalización) define un paquete como un conjunto de datos y otros elementos binarios de control que están organizados según un determinado formato y que se transmiten como un todo de acuerdo con un determinado procedimiento de transmisión .

1.5 Control de la comunicación

La comunicación entre dos elementos del sistema se realiza mediante la transmisión de la información a través del camino lógico que los une y consistirá en la transferencia de un conjunto finito de bits .

Se pueden producir errores en la transmisión de la información , debido a las características del camino físico utilizado . Al mismo tiempo se ha ido desarrollando técnicas para la detección de estos errores . Así a nivel de carácter se utiliza el método de detección de paridad ; a nivel de bloque de caracteres se complementa la detección a nivel de cada carácter , que suele denominarse detección vertical , con una detección a nivel de bloque denominada detección horizontal (checksum), que consiste en generar un nuevo

elemento de comprobación ; si la transmisión es sincrónica suelen utilizarse los métodos de control de error denominados de detección longitudinal o cíclica (CRC) .

En el caso de que el camino lógico utilizado para la transmisión de la información comparta con otros la totalidad o parte de su recorrido , será necesario que la información transmitida vaya acompañada de una identificación de dicho camino .

La comunicación de dos elementos a través de un camino lógico se realizará mediante el establecimiento de un diálogo entre ellos . Un diálogo consistirá en un intercambio de mensajes entre los elementos de origen y destino de la información .

Cada mensaje transmitido exige una confirmación de su recepción correcta o incorrecta , luego esto crea en principio que el camino lógico esté más saturado al tener que mandar de vuelta un mensaje de confirmación ; pero esto se subsana devolviendo sólo un mensaje específico, y esto lleva consigo el fijar el número de mensajes sin haber recibido confirmación del anterior ("ventana" de mensajes sin confirmar) .

Además de los mensajes conteniendo los datos será necesario intercambiar otro tipo de mensajes para la realización de funciones complementarias , tales como inicio y terminación de una conversación , confirmación de recepción de mensajes entre otras .

Por último en este punto nos queda hablar de los protocolos , que se definen como el conjunto de reglas y algoritmos que regulan el intercambio de información entre usuarios que quieran comunicar entre si . Es un problema de especificación de software restringido a un tipo de algoritmos muy particulares . En un sistema distribuido un protocolo permitirá iniciar , mantener y terminar un diálogo entre los elementos del sistema , entre otras cosas ; asimismo , regulará todo lo relacionado con el control de errores ; igualmente estarán previstas en un protocolo la forma de identificar el camino que se utiliza para el intercambio de información y la identificación del tipo de mensajes . Los elementos del diálogo de un protocolo serán los mensajes que estará dado entre entes que se comunican .

En general los protocolos se están haciendo cada vez más complejos . Por un lado se necesita especificar el servicio que un protocolo debe dar a unos usuarios que quieran comunicar entre si y por otro , especificar el protocolo que deba dar dicho servicio . Después de la especificación el diseñador se enfrenta a dos problemas , por una lado debe verificar que el protocolo verifica la especificación del servicio y por otro lado debe generarse el protocolo en forma ejecutable para la o las máquinas en cuestión .

2. SERVICIOS DE TRANSMISION DE DATOS

Hoy día los términos de Transmisión de Datos (TD), Telettratamiento o Teleproceso se usan para describir el movimiento de la información codificada sobre algún sistema

de comunicación y cuyo destino final , de una forma directa o indirecta , es su tratamiento por parte de un ordenador .

A partir de la década de los ochenta la simbiosis de la informática y las telecomunicaciones ha llegado a un grado tal que podemos hablar de la aparición de una nueva ciencia llamada telemática o teleinformática mediante la cual el hombre podría hacer el mejor uso de la información , concebida ésta como el auténtico motor de toda actividad .

Podemos agrupar los servicios de telecomunicaciones que ha ido surgiendo en :

- Servicios de Transmisión de Datos , que posibilitan la comunicación de datos (terminal/terminal y terminal/ordenador) sin comprender el tratamiento informático de los datos transmitidos .

- Servicio de Mensajes y Textos , basados en la comunicación terminal/terminal de mensajes o textos.

- Servicios de Telefotografía y Fácsmil , comunicación también persona/persona para la reproducción a distancia de documentos de información alfanumérica gráfica o fotográfica.

- Servicios de Telealarma , Telecontrol , Telemando y Telemedida , de transmisión de impulsos generados automáticamente en función de la variación de magnitudes físicas sometidas a control y sin comprender el proceso de las señales transmitidas .

Un sistema telinformático es un conjunto de equipos informáticos (ordenadores y terminales) , para la captura ,

proceso , envío y transmisión de datos y la red de telecomunicaciones constituida por circuitos y equipos de transmisión y conmutación .

Luego al considerar un proyecto de transmisión de datos se han de tener en cuenta unas necesidades de medios , definidas en las tres vertientes siguientes .

- Ordenador ; hardware y software ,ya sea el específico o el que resulta de incrementar el ya utilizado en los procesos batch para las nuevas aplicaciones .

- Terminales ; a utilizar en los puntos fuentes de información .

- Medios de comunicación ; considerados como el vínculo de unión entre el ordenador y el terminal .

Al constituir un sistema informático se puede elegir entre el establecimiento a través de circuitos punto a punto alquilados de una red dedicada o utilizar el servicio de una red pública conmutada .

Los sistemas que utilizan la transmisión de datos por líneas de telecomunicación , con respecto al ordenador , pueden ser " en línea " o " fuera de línea " .

Diremos que un sistema de proceso trabaja en línea (on-line) cuando acepta directamente directamente la entrada de datos desde el área de la fuente de información y envía los resultados del proceso , controlados por el ordenador , al área que los necesita . Los requisitos de este tipo de proceso son pues "acceso remoto" y "mantenimiento y actualización de memoria" .

Un sistema de teleproceso está fuera de línea (off-line) cuando los medios de telecomunicación para transmitir los datos no están directamente conectados al ordenador, sino que son grabados o perforados en un terminal periférico para ser procesados posteriormente. Un sistema fuera de línea no puede ser interactivo o conversacional, ya que el ordenador no está conectado directamente al lugar donde se envían los datos por lo que no se recibe su contestación, aunque puedan recibirse señales de control para vigilar el funcionamiento de los equipos y verificar si no ha habido errores.

Diremos que un sistema trabaja en tiempo real cuando controla un determinado entorno, donde se produce la información, la recibe, la procesa y devuelve los resultados con suficiente rapidez como para satisfacer las necesidades de dicho entorno.

Otra forma de trabajo será el llamado de tiempo compartido o time-sharing, por el que varios usuarios remotos pueden tener acceso simultáneo a los recursos de un ordenador y ejecutar trabajos independientes con programas propios de modo interactivo, de forma que el tiempo de respuesta para cada uno sea tal que el ordenador esté aparentemente dedicado en exclusiva. El ordenador trata independientemente los distintos trabajos mediante técnicas de asignación dinámica de recursos, pero en la escala de tiempos de usuario se produce ilusión de exclusividad al obtener el tiempo de respuesta que precisa.

3. ESTRUCTURA FISICA DE LOS CIRCUITOS PARA TRANSMISION DE DATOS

3.1 Introducci3n

Por transmisi3n de datos se entiende el movimiento de informaci3n que ha sido o va a ser procesada , codificada sobre alg3n sistema de transmisi3n el3ctrica .

Se necesita , entonces una fuente de datos , un destinatario de los mismos y un camino de uni3n entre ambos . Las t3cnicas que se emplean para llevar a cabo esta transmisi3n varian en funci3n de la distancia .

En la figura 1.3 se ven los elementos que constituyen un sistema de transmisi3n entre dos puntos A y B ;tales elementos son :

- ETD: El Equipo Terminal de Datos tiene dos funciones b3sicas : ser fuente o destino final de los datos y controlar la comunicaci3n . Aqu3 se engloban tanto los terminales como el m3s complejo ordenador .

- ETCD: El Equipo de Terminaci3n del Circuito de Datos tiene la misi3n de transformar las seales portadoras de la informaci3n a transmitir para luego enviarlas mediante los medios de telecomunicaci3n existentes .

- LINEA: Es el conjunto de medios de transmisi3n que une los dos ETCD , cuya constituci3n depender3 de la distancia , velocidad , etc. y que debe cumplir unas especificaciones dependiendo de la infraestructura de la comunicaci3n .

- ED: El Enlace de Datos constituye la uni3n entre la fuente y el colector de datos , y est3 formado por los controladores de comunicaciones , ETCD y LINEA .

- CD: El Circuito de Datos está formado por los ETCD y la LINEA cuya misión es entregar en el interface con el ETD colector las señales bajo la misma forma y con la misma información que recibió con el ETD fuente .

3.2 Conceptos relacionados con la transmisión de datos

La comunicación de datos es un concepto que abarca toda la transferencia de información entre dos puntos cualesquiera dentro del campo cubierto por un sistema de comunicación , basado en el tratamiento por ordenador de aquella información.

La transmisión puede darse en serie o en paralelo:

- En la transmisión de datos en serie los datos se transfieren bit a bit utilizando un canal único . Es la forma de transmitir datos a grandes distancias .

- En la transmisión de datos en paralelo se transmiten simultáneamente todos los bits de un carácter o palabra , luego hay tantos canales como bits contenga el elemento base. Esto conlleva mayor complejidad , pero mayor velocidad . Se utiliza sólo en transmisiones dentro del propio centro de cálculo .

Como es necesario que la fuente y el colector tengan una base de tiempos común tiene que existir un sincronismo entre el transmisor y el receptor . Esto puede hacerse al menos en tres niveles .

- Sincronismo de bit, para determinar el instante en que , teóricamente debe comenzar a contarse un "bit".

- Sincronismo de carácter, en que el receptor sabe cuál es el primer bit de un carácter .

- Sincronismo de mensaje o de bloque, que define el conjunto de caracteres que van a constituir la unidad base para el tratamiento de errores , etc. y forman parte del protocolo de comunicaciones .

En cuanto a los modos de explotar el circuito de datos , existen tres modos básicos :

- El modo fundamental es el "simplex" , cuya transmisión se realiza solamente en un sólo sentido . Si se utiliza más de un receptor entonces el modo puede llamarse " bus de distribución simplex " .

- El modo semidúplex (half-duplex) realiza la transmisión alternativamente en los dos sentidos exigiendo cierto tiempo para cada inversión, lo cual reduce la eficacia del sistema.

- En el modo dúplex integral (full-dúplex) la transmisión se realiza simultáneamente en los dos sentidos .

La extensión lógica del funcionamiento half-duplex es "multiplex" . En el funcionamiento multiplex el flujo de datos entre dos o tres equipos terminales no es simultánea . La ventaja primaria del funcionamiento multiplex sobre el simplex está en los costes de líneas . Y las desventajas incluyen tanto circuiteria más compleja como la lógica requerida para el intercambio de datos (handshaking).

3.3 Interface del circuito de datos

En general se define interface como el conjunto de reglas y convenios que gobiernan y controlan la interconexión entre

estaciones , a fin de intercambiar informaci3n y datos .

El establecimiento del circuito de datos servir3 para transportar de extremo a extremo la informaci3n que se le entrega en forma de se3ales digitales . Se destacan dos aspectos importantes en este interface :

- su relativa complejidad debida al hecho de que debe completar y resolver gran variedad de casos en cuanto al modo de explotaci3n del circuito de datos .

- su gran importancia dentro del sistema de transmisi3n de datos , al ser frontera natural dentro entre la transmisi3n y el propio tratamiento de los datos .

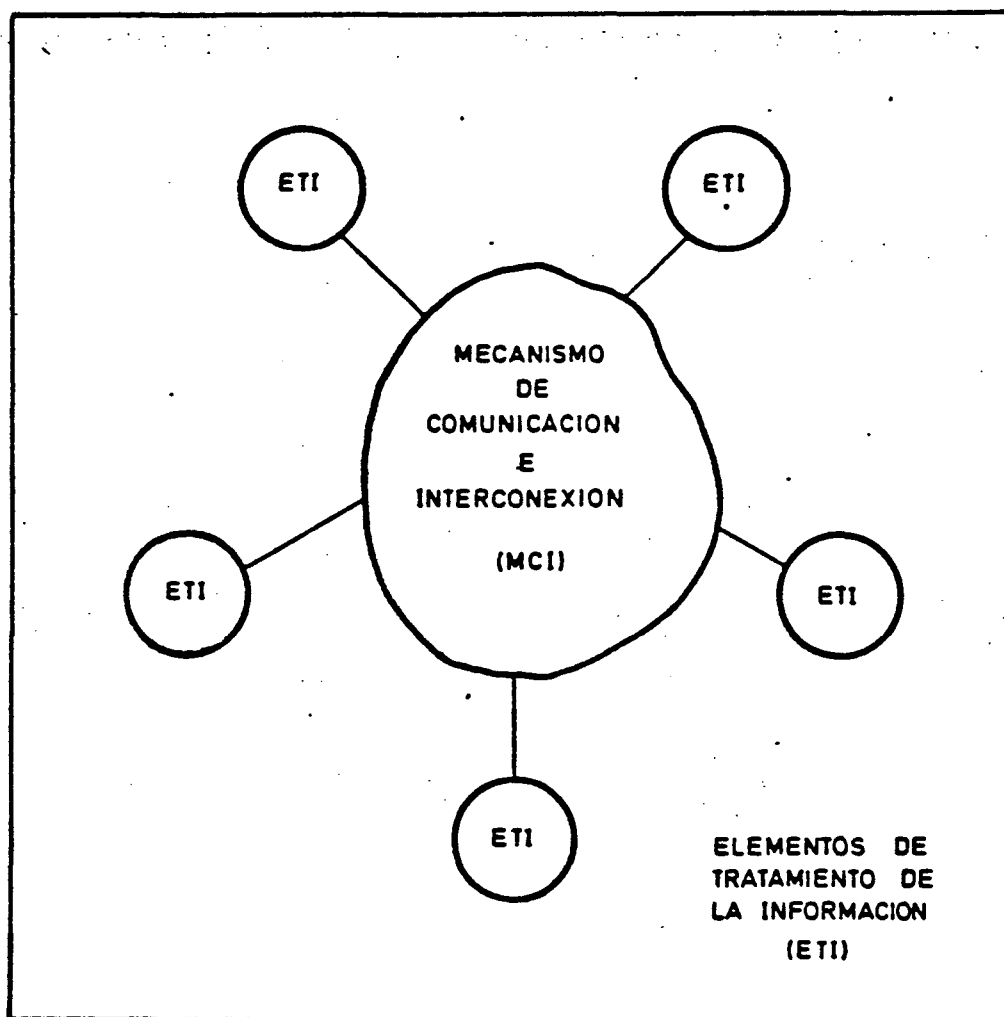


Figura 1.1 Esquema general de un sistema distribuido.

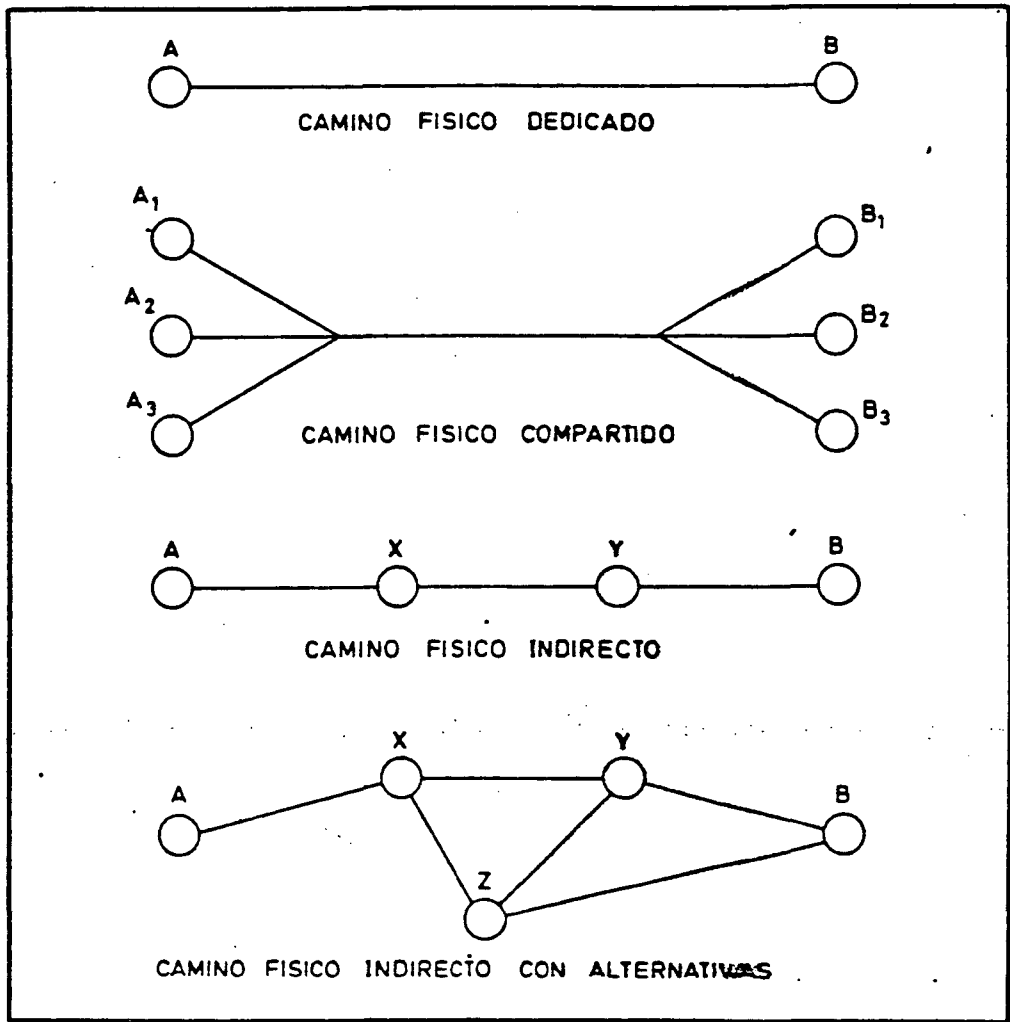


Figura 1.2 Alternativas en la materialización de un camino lógico.

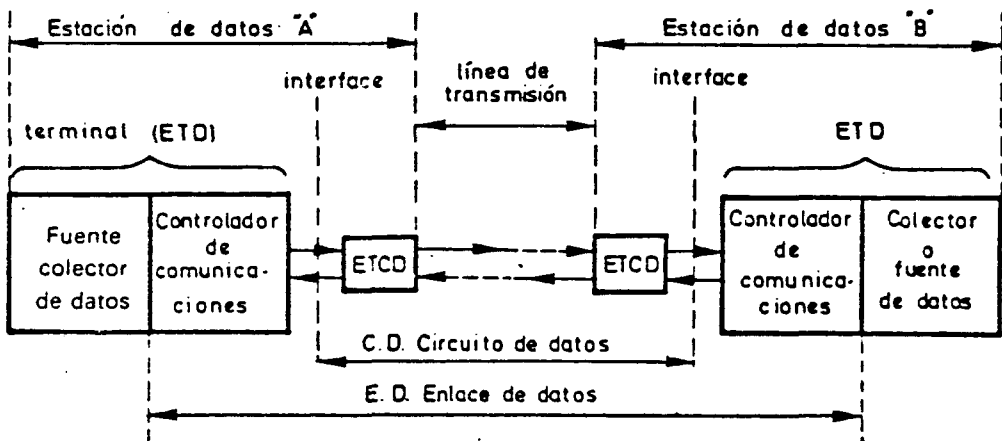


Figura 1.3 Elementos de un sistema de transmisión de datos.

CAPITULO 2

CAPITULO 2: CONCEPTOS

2.1 Introducción

Este tipo de redes se refiere a aquellas en que las distancias son relativamente cortas (desde unos centenares de metros a unas decenas de kilómetros), además su otra característica es la elevada capacidad de comunicación entre los nodos que constituyen la red (desde unas decenas de kbps a decenas de Mbps) .

El éxito de este tipo de sistemas estriba en la relación prestaciones/precio de los sistemas de tratamiento de la información , conjugado con los avances de las tecnologías de comunicación digital .

Una red local es un sistema de comunicación de datos que permite que un número de dispositivos de tratamiento de la información independientes se comuniquen entre ellos con las siguientes características :

- Área moderada ,por ejemplo una oficina ,etc.
- Canal de comunicación de capacidad media-alta
- Probabilidad de error baja en los mensajes internodo .

Las áreas de aplicación caen en una o más de las categorías siguientes : datos , voz y gráficos .

Los objetivos primordiales de la red local son :

- Compatibilidad de productos fabricados y diseñados por empresas distintas .
- Que sea de bajo coste y que se pueda utilizar en nodos de bajo costo .
- Debe tener un nivel de estructuración , de forma que un

cambio en un nivel sólo afecte a ese nivel .

Las prestaciones funcionales de forma general son las siguientes :

- La red local debe dar el servicio de enviar a una o más direcciones de destino unidades de datos a nivel de enlace .

- En una red local las comunicaciones se realizan entre procesos que tienen el mismo nivel .

En cuanto a las características físicas de las redes locales , deberán satisfacer los objetivos funcionales siguientes :

- Posibilidad de comunicación directa entre dos nodos de una red local sin necesidad de almacenado y reenvío a través de un tercer nodo de la red excepto casos en los que se tienen distintas clases de servicios .

- Las redes locales deben permitir la adición y supresión de nodos de la red de una forma fácil .

- Los recursos físicos de la red deben ser compartidos de forma justa por los distintos nodos .

Dadas ya las características básicas de las redes locales éstas se pueden clasificar según el estado actual de la tecnología en :

- Sistemas de bajas prestaciones y bajo costo .
Normalmente utilizan cable trenzado como medio de transmisión .

- Sistemas de prestaciones medias y medio costo .
Normalmente utilizan cable coaxial como medio de transmisión con codificación de señales en banda base .

- Sistemas de altas prestaciones y costo elevado .
Utilizan cable coaxial blindado con codificación de senales
en banda ancha .

2.2 Topologías en redes locales

La estructura topológica de la red es un parámetro
primario que condiciona mucho las prestaciones que se sacan
de la red . El acierto entre una u otra estructura depende en
cada caso al tipo de tráfico que debe cursar y de una
valoración de la importancia relativa de las prestaciones que
de la red se pretende obtener . Criterios básicos a tener en
cuenta son :

- Coste-modularidad en cuanto al coste en medios de
comunicación y a la sencillez de instalación y mantenimiento .

- Flexibilidad-complejidad por la dificultad que supone
incrementar o reducir el número de estaciones .

- Fiabilidad-adaptabilidad por los efectos que un fallo en
una estación o en el medio de comunicación pueden provocar en
la red , así como las facilidades de reconfiguración .

- Dispersión-concentración por su adecuación a
instalaciones con poca o mucha dispersión geográfica .

- Retardo-caudal por el retardo mínimo introducido por la
red , o su facilidad de manejar grandes flujos de
información sin que se produzcan bloqueos o congestiones .

Una fuerte exigencia de alguna de esta características
puede obligar a renunciar a la instalación de una determinada
red local por el tipo de topología que utiliza . Así por

ejemplo si se quiere fiabilidad no se debe utilizar la red en estrella ya que en ésta, una avería en el módulo central bloquea toda la red.

2.1.1 Topología en estrella

Todas las estaciones están unidas mediante medios bidireccionales, a un nodo o módulo central que efectúa funciones de conmutación (fig. 2.1a). Es de aplicación frecuente en redes muy centralizadas o en sistemas de control.

El nodo central asume además las labores de control y dispone de gran parte de los recursos informáticos comunes (memorias masivas, etc.).

El nodo aísla una estación de otra, resultando una red fiable en cuanto a averías en las estaciones; sin embargo una avería en un nodo deja totalmente bloqueada la red y sin posibilidad de reconfiguración. La flexibilidad-complejidad es buena permitiendo cambiar con sencillez el número de estaciones, ya que las modificaciones son sencillas y localizadas todas en el nodo central. Puede resultar costoso por la longitud del medio de comunicación a instalar, además no permite grandes flujos de tráfico; luego no es adecuado para redes con gran dispersión geográfica, pero el retardo es mínimo.

2.1.2 Topología en árbol

Es una extensión de la arquitectura en estrella interconexionando varias (fig. 2.1b). Se puede establecer una gerarquía clasificando a las estaciones en grupos y

niveles según el nodo central a que están conectadas y su distancia jerárquica al nodo central .

Sus características son similares a la red en estrella , reduce la longitud de los medios de comunicación incrementando el número de nodos . Se adapta a redes de grandes distancias y predominancia de tráfico local .

2.1.3 Topología en malla

Cada estación está conectada con otras (red completa) o varias (red incompleta) estaciones formando una estructura regular o irregular (fig. 2.2).

El coste en medios de comunicación depende del número de conexiones y suele ser elevado , ganando sin embargo en fiabilidad frente a fallos y en posibilidades de reconfiguración . El coste de instalación es grande por aumentar el número de estaciones . No se adapta a amplias áreas geográficas , pero permite tráficos elevados con retardos medios bajos . La dificultad de diseño reside en minimizar el número de conexiones y el desarrollo de potentes algoritmos de encaminamiento y distribución de flujos .

Suelen ser de uso frecuente en redes de ordenadores , unidos a estructuras en estrella o en árbol , que en redes locales .

2.1.4 Topologías de anillos simples y múltiples

Los módulos de comunicaciones de las estaciones están interconectados formando un anillo , de forma que todas las informaciones pasan por todos los módulos que únicamente

envían a la estación los paquetes destinados a ella (fig. 2.3).

Se suele recurrir a dos líneas separadas : una de transmisión y otra de conexión , luego ello repercute en una velocidad de transmisión mayor , además de un transceptor más sencillo .

En redes centralizadas el anillo incluye un controlador, lo que no es frecuente en redes locales , donde se prefieren los procedimientos distribuidos por ser más flexibles .

El flujo viene limitado por el ancho de banda del recurso de transmisión . El retardo introducido por cada estación puede hacer que el total sea demasiado grande para determinadas aplicaciones en tiempo real . Suelen utilizarse para conectar sistemas informáticos de capacidad media y alta , especialmente si esán bastante separados geográficamente .

La relación coste-modularidad es buena así como la flexibilidad para incrementar el número de estaciones . La aparición de un fallo en el medio de comunicación bloquea totalmente la red sin posibilidad de reconfiguración .

Al instalar una red en anillo suele dotarse del concentradores de conexiones lo que permite aislar y separar con rapidez las averías , cortocircuitando y separando el bucle averiado del resto .

2.1.5 Topología bus y multipunto

Los módulos de conexiones están conectados de un único medio de transmisión (bus) que recorre todas las estaciones .

Al igual que en la estructura en anillo no es necesario efectuar encaminamientos . Aquí todas las estaciones reciben simultáneamente la información .

En aplicaciones a redes locales el control de acceso al medio suele ser distribuido . Pero en redes más complejas la conexión suele ser a través de un controlador que también gestiona el bus (estructura multipunto) .

Dentro de la topología del bus se distingue entre :

- Bus bidireccional . Se transmite en ambas direcciones por el mismo medio o medios conductores (bus paralelo) .

- Bus unidireccional . Permite alcanzar distancias mayores con unos pequeños amplificadores , ello requiere que se aumente la longitud del cable utilizado . Las tres formas de conexión utilizadas dentro de este tipo son :

- a) Lazo . Es un bus que se inicia y termina en un controlador que centraliza la gestión .

- b) Horquilla . Puede estar formada por una sola o dividirse en ramales para adaptarse por ejemplo a los distintos pisos de un edificio .

- c) Espiral . En este tipo el tiempo que una estación tarda en recibir su propio mensaje es igual para todas las estaciones . Mientras el retardo de propagación es más reducido que en otras topologías como el anillo , presenta mayores dificultades para una utilización eficiente de la capacidad del recurso , dando lugar a complejos algoritmos de control de acceso .

2.1.6 Topología múltiple

Cuando las estaciones pueden agruparse en conjuntos de forma que el tráfico hacia otro conjunto es mucho menor que el interior, puede resultar preferible distribuir las en varias redes, en lugar de una conectadas a través de un puerto sin que sea necesario que todas las redes tengan la misma topología.

2.2 Reparto de recursos del sistema

El que en lugar de uno sean varios los que utilicen un mismo recurso permite rentabilizarlo sin deteriorar significativamente la calidad del servicio, si se utiliza una técnica de asignación eficaz. Para resolver el problema de la asignación puede pensarse en repartirlo entre los usuarios de forma que a cada uno se le asigna una fracción del recurso que puede ser igual para todos, o preferiblemente proporcional a su demanda media si existen apreciables diferencias entre ellos.

Las técnicas basadas en compartición resultan complejas de controlar ya que la asignación de recursos varía con el tiempo, por actuar bajo criterios de demanda.

2.3 Técnicas de repartición de un medio de comunicación

Convendrá distinguir entre si los usuarios están conectados directamente al recurso o no, y si el control de acceso al recurso está centralizado o no.

Si los usuarios están conectados directa y

permanentemente al recurso ; las técnicas se denominan de acceso múltiple .

Cuando la conexión se efectúa de forma indirecta a través de un equipo multiplexor , éste , además de efectuar el reparto y conectar usuario y recurso , suele centralizar el control del acceso y las técnicas suelen denominarse de multiplexación .

Las técnicas de acceso múltiple se utilizan fundamentalmente en los enlaces por satélite , donde la separación geográfica de los usuarios hace inviable la utilización de un multiplexor .

Las técnicas de multiplexación suelen emplearse en redes con bajo número de estaciones por su sencillez de realización , o como una primera etapa en el acceso a la red.

2.4 Técnicas de compartición

En redes locales las técnicas de compartición suelen ser siempre de acceso directo ; cada usuario está conectado directamente al medio de comunicación y no a través de un concentrador .

La capacidad del canal suele ser del orden de la velocidad de transmisión de los usuarios . Las técnicas de concentración suelen utilizarse con más profusión en grandes redes de ordenadores con enlaces de alta capacidad y terminales con velocidades muy dispares .

Existen tres procedimientos básicos para compartir un recurso cuando la conexión entre usuarios es directa : selección , reserva y contienda . Los dos primeros pueden

efectuarse con control de acceso centralizado o distribuido . La técnica de contienda (también llamada de acceso aleatorio), es específica para redes con control de acceso distribuido .

- Selección . Se avisa al usuario al llegar su turno y toma control hasta que termina de transmitir los mensajes que tiene pendientes en la cola de espera . La asignación de turnos no es en el tiempo . En todo caso los usuarios se seleccionan por turno y desconocen cuando van a serlo nuevamente .

- Contienda . Cuando el usuario necesita el canal intenta tomarlo , estableciéndose una contienda con otros usuarios que desean también utilizarlo .

- Reserva . El usuario conoce con adelanto cuando va a poder utilizar el recurso . O bien dispone de una reserva permanente o , en su caso , antes de tomar el recurso solicita que se le haga y confirme una determinada reserva .

2.5 Técnicas de contienda

La resolución del proceso de contienda utilizado como técnica de acceso , requiere la detección y resolución de las colisiones .

Las técnicas de acceso pueden clasificarse en técnicas con o sin escucha según tengan información o no de cuando el canal está libre . Normalmente esta escucha se efectúa por detección de presencia de señal .

En la transmisión sorda cuando un usuario tiene un paquete a transmitir lo envía por el canal ; si no había

nadie transmitiendo o lo hace mientras dure el envío , el paquete llega intacto , en caso contrario habrá colisión que al detectarse provocará una retransmisión .

En cuanto a los métodos de detección de errores se utilizan dos procedimientos básicos :

- En los métodos de detección por interferencia se asocia la aparición de interferencias sobre la señal enviada al canal por el transceptor .

- En el segundo procedimiento , la detección de colisiones se efectúa por no recibir un aviso de conformidad (ACK) una vez terminada la transmisión del paquete . A diferencia del anterior no se detectan realmente colisiones sino errores en la transmisión lo que puede provocar que se inicie un algoritmo de resolución de colisiones sin ser necesario por no haber existido una colisión .

2.6 Técnicas de reserva

Las estaciones solicitan una reserva y no inician la transmisión de la información hasta que se les concede . Son técnicas libres de colisión en lo que a transmisión de información se refiere . Pueden existir durante el proceso de solicitud de la reserva si ésta se resuelve mediante un procedimiento de contienda .

Los sistemas centralizados disponen de un controlador que recibe y concede las demandas de reserva resolviendo los problemas de concurrencia de acuerdo a una relación de prioridades .

En los sistemas descentralizados todas las estaciones deben recibir las solicitudes formuladas por las demás y aplicar un algoritmo común a todas para calcular desde cuándo y por cuánto tiempo tiene derecho a una reserva .

Dentro de las técnicas de reserva hay dos clases de sistemas .

- de Control Distribuido .- Cuando se utiliza una técnica implícita no existe propiamente un procedimiento de solicitud y resolución de reserva , previa a la transmisión , sino que una estación que quiere utilizar el canal intenta tomarlo directamente ; si lo consigue , las demás estaciones entienden que aquella lo tiene reservado hasta que señalice lo contrario .

- de Control Centralizado .- Suelen estar compuestos por dos canales : uno asignado a la transmisión de los mensajes y el otro para resolver las demandas de reservas .

2.7 Técnicas híbridas contienda-reserva

Tratan de unir las buenas prestaciones de las técnicas de contienda cuando el tráfico es bajo , con las mucho mejores de las técnicas de reserva para tráficos altos . La reserva evita las colisiones y retransmisiones que tienden a saturar el sistema .

2.8 Técnicas de selección

Fueron las primeras que se emplearon para compartir un recurso . Al principio se utilizaron en líneas de comunicación con una topología en estrella , y con

posterioridad se aplicaron a una topología en bus .

Pueden ser centralizadas o distribuidas según el control de selección de usuario esté o no centralizado . En ambos casos los usuarios deben almacenar sus mensajes en una cola hasta ser seleccionados y no conocen cuándo esto va a ocurrir .

Las técnicas de selección centralizadas pueden ser de tres tipos básicos : sondeo (polling) , daisy chaining y peticiones independientes .

- En las técnicas de sondeo , se selecciona un usuario enviándole su dirección que se recibe también en todos los demás ; esto hace que cuando ese usuario reconozca su dirección tome control del canal avisando al controlador una vez finalizado el envío de sus mensajes , o devuelve el control inmediatamente si no tiene ningún mensaje en la cola de espera .

- El daisy-chaining es una técnica muy utilizada en buses internos de ordenador . Necesita un hilo extra que recorra todos los usuarios . Cuando un usuario recibe un pulso a través de este hilo , toma el control del canal o pasa el pulso al siguiente sino tienen ningún mensaje que enviar . El hilo extra conexionado en anillo sólo se utiliza para la selección .

- En las técnicas de peticiones independientes , los usuarios envían solicitudes al controlador , de forma que éste pueda identificarlos . Cuando el controlador ha recibido las solicitudes las ordena según prioridades y selecciona

sucesivamente los distintos usuarios solicitantes .

2.9 Técnicas de paso de testigo (token passing)

Son técnicas de selección que hacen uso de una palabra clave o testigo (token) para establecer los turnos en el uso del canal de comunicación . Existen dos tipos :

- Paso de testigo en anillo (token ring) .- Mientras no haya paquetes que enviar estará recirculando un testigo que es recibido y retransmitido por las distintas estaciones . Cuando una estación quiere transmitir , espera a recibir el testigo , y no lo vuelve a retransmitir tomando así el control del canal . Entonces puede enviar sus paquetes de transmisión , y finaliza transmitiendo de nuevo el testigo .

- Paso de testigo en bus (token bus) .- A diferencia del token ring , la topología en bus no impone el orden en que van a seleccionarse las estaciones . La selección se efectúa enviando directamente el testigo a la estación que corresponde tomar el turno . El testigo debe , en este caso incluir un campo de dirección . Se establece así un anillo lógico sobre la topología de bus .

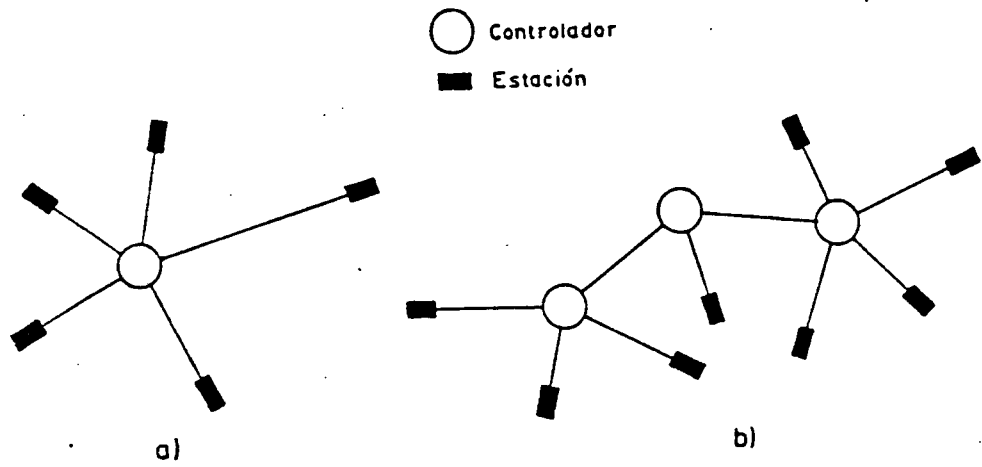


Figura 2 .1 Topologías en: a) estrella; b) árbol.

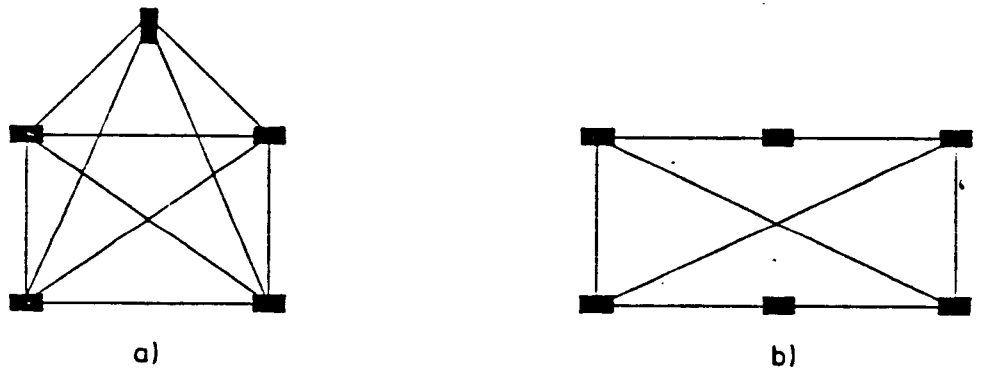


Figura 2 .2 Topología en malla: a) completa; b) incompleta regular.

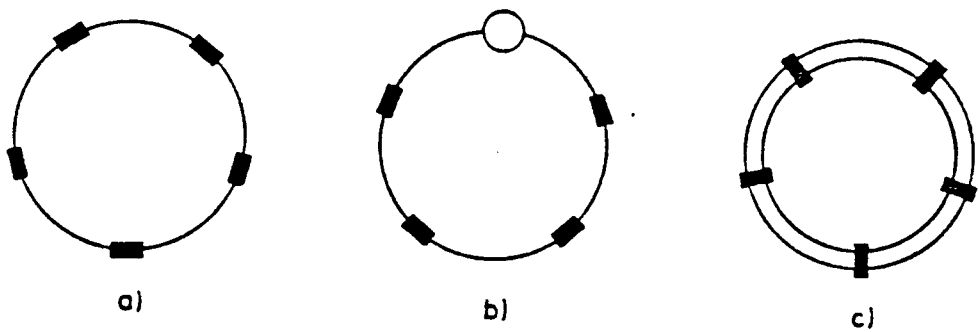


Figura 2 .3 Anillo simple: a) descentralizado; b) centralizado; c) anillo doble descentralizado.

SEGUNDA PARTE

CAPITULO 3

CAPITULO 3 : ESTRUCTURACION DE UN SISTEMA DE ORDENADORES

La idea de realización de este trabajo surge viendo las nuevas redes de ordenadores que se están dando hoy día (por ejemplo IBM , etc) ; y sobre todo de LA RED LOCAL DE COMUNICACIONES CON " MSX " (varias marcas de software compatible que se pueden interconectar fácilmente sin ningún tipo de problema). Y ello lo queríamos aplicado a la enseñanza , o sea la idea de un MASTER (maestro) , y una serie de alumnos con sus respectivos terminales (ver fig 3.1). El porqué de esto anterior no es ni más ni menos porque como se está viendo en la enseñanza al igual que en tantas ramas (en muchas ; por no decir en todas) se está introduciendo la informatización y telecomunicación (que hoy por hoy van estrechamente unidas).

Pero dirándonos al tema , lo que teníamos a mano para la realización de tal tarea eran los ordenadores " APPLE " (o imitación), y con ellos trabajamos.

La idea consiste en tener uno de ellos (APPLE) como maestro y los demás como seguidores o terminales que acceden al principal cuando le necesitan para buscar alguna información . De esta idea es por lo que se ha hablado de "redes de ordenadores" en el primer capítulo.

Por otra parte además del diseño del circuito conformador del protocolo de acoplamiento hardware y de el bus de interconexión RS-232-C , se ha trabajado en el software de dos tarjetas diferentes . La una SERIAL INTERFACE

CARD que trabaja sin ACIA y que por lo tanto lo hace todo como si dijéramos por software, y la otra SUPER SERIAL CARD (SSC) cuya ACIA (6551) controla la comunicación ; existe también otra tarjeta con ACIA (6850) , la COMMUNICATIONS INTERFACE CARD de cuya tarjeta no hemos encontrado la suficiente información como para trabajar con ella . Pero de todo ello hablaremos más ampliamente en el capítulo correspondiente.

Es de señalar que con la que más se ha trabajado es con la SUPER SERIAL CARD , cuya tarjeta venía ya preparada para trabajar en modo TERMINAL o modo MODEM y con la cual se ha trabajado para acceder a ficheros de l disco (guardar y sacar información) . A su vez hemos establecido el protocolo software para que trabaje como terminal conectado al MODEM de que disponemos (de ello hablaremos más ampliamente en el capítulo correspondiente).

Dentro de la estructuración vamos a hacer incapié en dos apartados completamente distintos :

- Estructuración software.-

a) Software de la COMMUNICATIONS INTERFACE CARD.-

Con esta tarjeta se implementa el modo de comunicación half-duplex (modo terminal) . Hay que señalar que no se cree difícil su utilización con ficheros .

b) Software de la SERIAL INTERFACE CARD .-

Como ya tiene establecido el modo terminal , se intenta que por una parte funcione como el terminal del HP-3000 que está conectado al MODEM , y por otra parte se intenta que

trabaje con ficheros , esto es :

- Se puedan mandar ficheros existentes en disco por la tarjeta.

- Se puedan almacenar en disco ficheros llegados por tarjeta .

Desde luego hay otras opciones , pero las anteriores son las más importantes.

Hay que señalar en este apartado de estructuración software que debido a la poca información que se tiene , muchas dificultades se resuelven a base de pruebas . También se ha de decir que cuando se empezó sólo se disponía de un miniensamblador para el desarrollo de los programas ; ahora se dispone de un ensamblador mucho más potente y con más posibilidades , que hubiera resuelto muchas dificultades . Por eso se va a hacer un inciso aquí para explicar brevemente las posibilidades de este ensamblador .

* Sistema profesional de Desarrollo de lenguaje ensamblador en Computadoras Apple (LISA).-

- Introducción

Es un ensamblador interactivo del 6502 para el Apple II. Fue diseñado para solucionar necesidades tanto de principiantes como de programadores avanzados . Debido a sus características el trabajar con ensamblador (y el que dice ensamblador implica lenguaje máquina) es casi tan fácil como BASIC .

Se dispone en él de muchos pseudocódigos , mnemónicos y comandos que permiten la flexibilidad del uso de DOS 3.2 .

- Formato fuente .- Consta de :

ETIQUETA MNEMÓNICO OPERANDO ; COMENTARIO

* El campo de etiqueta contiene de 1-8 caracteres que empiezan en la columna 1 y termina en un blanco o un pto.

* El campo de mnemónico delimitado por un blanco debe contener tres caracteres mnemónicos . Estos mnemónicos pueden ser :

a) Mnemónicos pertenecientes al 6502, tales como ADC, AND, ... etc

b) Otros mnemónicos extendidos , tales como BTR, BFL , ...y otros.

c) Otros mnemónicos fáciles , tales como SET, LDR, etc.

d) Pseudocódigos tales como OBJ, ORG, EQU, y otros.

* El campo de Operando , también delimitado por un blanco , contiene la expresión de dirección y cualquier información de modo de direccionamiento real.

* El campo de comentario es opcional , debe empezar en ";" y debe estar separado del campo de operando al menos por un blanco . Si no hay campo de operando , el de comentario puede seguir al de mnemónico . Como se supone , este campo se ignora a la hora de ensamblar.

También se puede hablar de modos de direccionamiento , comandos , etc , pero se remite al interesado al manual de LISA para más información .

- Estructuración hardware.-

Esta parte es la que define el protocolo de acceso , por ello en su circuitería se trabaja con señales de control

que dejan paso o no a las señales de datos .

El trabajo se ha configurado por el uso de dos tarjetas distintas , cuyo protocolo de acceso (control de acceso) , por tanto es diferente, y origina más circuitería.

En esta estructura se han de distinguir dos apartados :

a) Inicialización .- Donde se procede a comprobar , si al menos hay un master y un esclavo para establecer la comunicación . De otra manera se hace que se desconecte la tarjeta o tarjetas que piden entrada .

b) prioridad de acceso .- Se establece prioridad cuando llegan varios a la vez, pero de una vez un alumno conectado al maestro no puede comunicar otro alumno hasta que el otro termine .

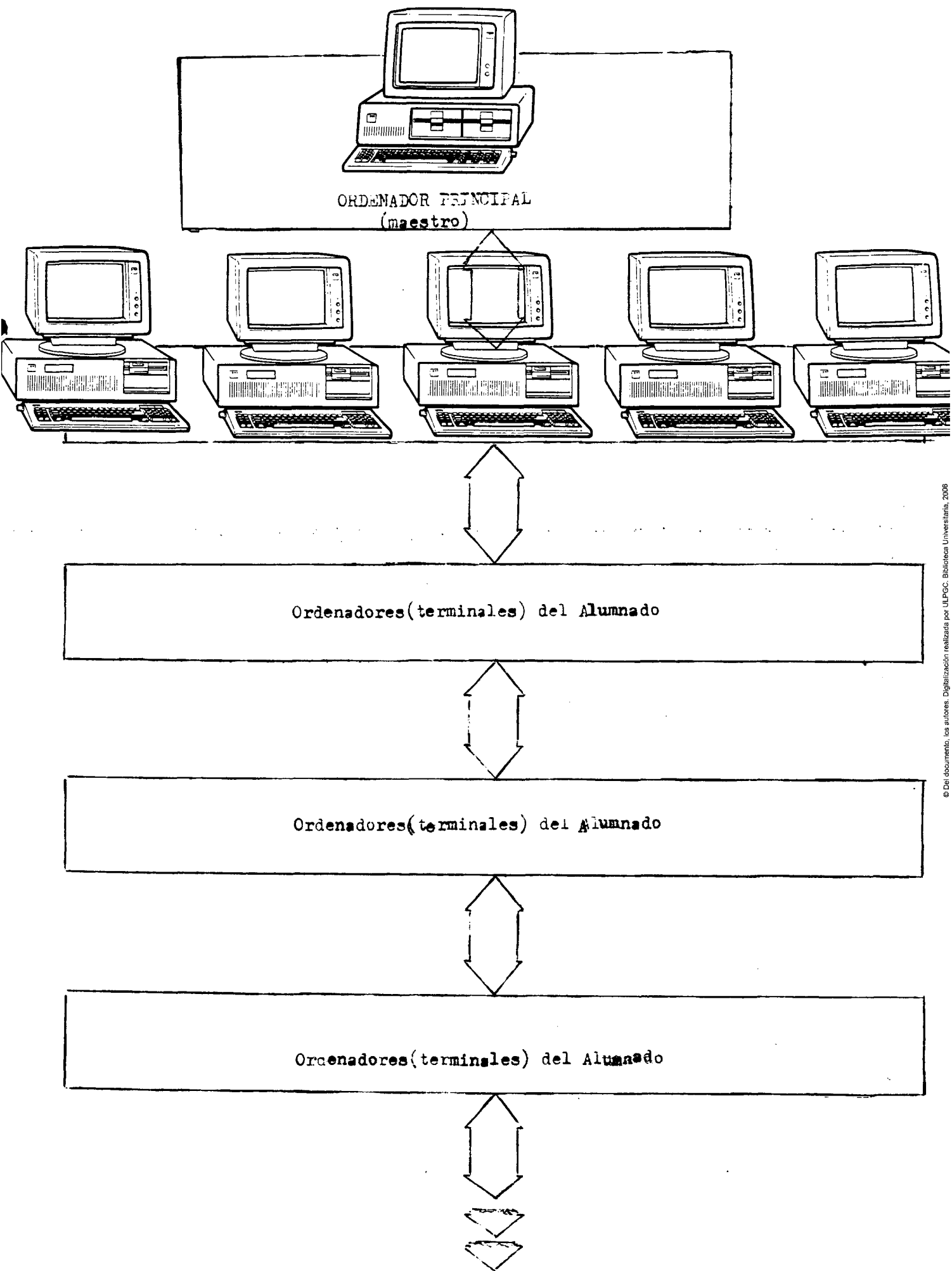


figura 3.1

CAPITULO 4

Capítulo 4 : ESTRUCTURA A NIVEL SOFTWARE

En este capítulo se describen los distintos programas que se utilizan para interconexión de los Apples, así como las distintas tarjetas que se utilizan.

Para dar una idea orientativa de lo que se ha hecho, comenzaremos diciendo que los Apple pueden utilizar varias tarjetas de comunicaciones, tales como la SUPER SERIAL CARD, la SERIAL INTERFACE CARD ó la COMMUNICATIONS INTERFACE CARD.

Estas tarjetas se colocan en las ranuras (slots) que el Apple tiene, si bien la ranura 0 se refiere al propio Apple, por ejemplo si se pone PR#0 es la propia pantalla Apple.

Estas tarjetas de conexiones al igual que otras que posee el Apple tienen la particularidad de trabajar a partir de la posición "CN00" y hasta "CNFF" siendo N el número del slot en que está la tarjeta y después se tiene un espacio de PROM de 2K (\$C800-\$CFFF) el cual está disponible para cualquier tarjeta, y más de una tarjeta puede tener expansión ROM. Sin embargo sólo una expansión puede estar activa en un determinado momento. Para ello cada tarjeta con expansión ROM tiene un flip-flop para habilitarla.

Entonces una única tarjeta a la vez puede acceder a esos 256-bytes PROM refiriéndose a la localización \$CFFF en su subrutina de inicialización. Toda tarjeta periférica reconoce esta localización especial, entonces pone sus flip-flops inactivos e inhabilita su expansión ROM. Por supuesto esto también inhabilita la expansión de ROM de la propia tarjeta que está tratando de utilizar ese espacio, pero la ROM se habilita de nuevo cuando el microprocesador obtiene

otra instrucción de los 256-bytes impulsados a PROM . Ahora la expansión de ROM está habilitada y su espacio ocupado . Las distintas subrutinas pueden entonces saltar directamente dentro de programas en la ROM , donde pueden trabajar con esos 2K limpiamente .

COMMUNICATIONS INTERFACE CARD (CIC)

De esta tarjeta se posee poca información , luego con ella no hemos trabajado , si bien es una tarjeta propicia para establecer comunicaciones por disponer de la ACIA 6850 . Luego esta tarjeta con la información adecuada y/o un estudio riguroso seguro que vale para la conexión que queremos establecer.

SERIAL INTERFACE CARD (SIC)

-Introducción a la tarjeta SIC.-

Esta tarjeta fué la primera que se utilizó en los intentos de comunicar dos Apples . Para ello se idearon programas en BASIC y luego viendo que en BASIC no se podía conseguir se trabajó en ensamblador .

Hay que señalar que aunque se consiguió una comunicación half-duplex (que no podíamos conseguir desde ensamblador), ésta no es una tarjeta propia para comunicación entre Apples trabajando como terminales sino más bien para trasladar datos a impresora y otros . Se dice esto anterior porque aunque esta tarjeta tiene muchas salidas (cables de salida/entrada) no tiene unas señales activas para controlar dichas funciones de comunicación como sería necesario (unicamente se activan las señales 2 , 3 y 23). Luego primero vamos a ver las

características más relevantes de dicha tarjeta y luego se verá lo que se llegó a hacer con ella .

- Características fundamentales con un RS-232 estándar:

* La salida del Apple II puede enviarse a una impresora serie u otro dispositivo externo , a pantalla Apple o ambos.

* La entrada puede tomarse desde otro dispositivo externo o del teclado Apple o ambos simultáneamente .

* El Apple II puede manejar comunicación half-duplex en el rango de 75 a 19.200 baudios , en ambas direcciones con una impresora , otro Apple , un terminal , modem u otro dispositivo externo RS-232 .

* La SIC puede conectarse también funcionando en lazo de corriente con un teletipo .

- Inicialización de la SIC .-

Al empezar a usar la SIC se ha de inicializar poniendo sus parámetros de funcionamiento a sus valores DEFAULT .

- Parámetros de funcionamiento del interface serie .-

Se tienen 10 parámetros de funcionamiento definidos por el usuario , de ellos 5 se determinan por los 7 niveles de los switch que están en la tarjeta :

a) Velocidad en baudios . En nuestro caso hemos de ponerla a 1200 baudios (switch 1 a OFF, 2 a OFF, y 3 a ON).

b) Delay de CR. Para que no haya se pone el switch 4 a OFF.

c) Ancho de línea más video Apple . Para 40 caracteres por línea y video habilitado ponemos los switch 5 y 6 a ON .

d) Line Feed. Si el dispositivo externo alimenta su propio Line Feed después de un CR se pone el switch 7 a ON con lo

que desactivamos el Line Feed .

Luego los otros 5 DEFAULT permanentes son :

- a) No hay bit de paridad .
- b) No hay carácter de cheksum .
- c) Convierte todos los caracteres de llegada de mayúsculas a minúsculas .
- d) El número de bits de datos es 8 más un bit de start.
- e) 2 bits de stop.

Estos valores que se ponen cada vez que se inicializa el sistema pueden cambiarse luego por comandos software a los valores deseados.

Para más información les remitimos al manual de la SIC.

Una vez descritas las características más relevantes de esta tarjeta vamos a describir el programa que se hizo para la comunicación que se pudo realizar gracias al listado de referencia de ensamblador que se tiene .

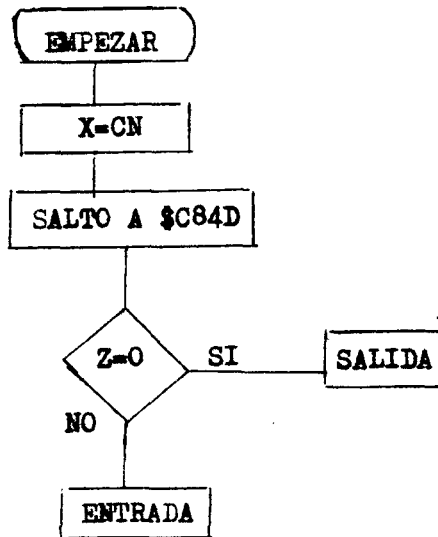


Figura 4.1

- Trabajo realizado con la tarjeta SIC . Conclusiones .

El programa base que se realizó fue el dado en la figura anterior . En síntesis el programa es lo siguiente :

a) Inicialización de la tarjeta , que incluye la puesta de flags de entrada , borrado de pantalla , y la puesta en el registro X del valor de "CN" siendo N el número de la ranura (slot) donde se coloca la tarjeta .

b) Se entra en la subrutina de llegada de carácter (JMP \$C84D) y al volver de esta subrutina tenemos el flag de cero puesto a cero o uno dependiendo de si es entrada o salida respectivamente .

c) Luego de detectar si es entrada o salida se recoge el dato que la subrutina anterior guarda en la posición \$05B8,X que la tarjeta llama BYTE (+ \$CN) y si es entrada se manda a pantalla , pero si es salida se manda al exterior mediante otra subrutina que ya viene implementada en la propia tarjeta que es un JSR \$C9AA , y también se manda o no a pantalla dependiendo de si se quiere o no el modo ECO .

Pero luego queremos que aparezca un cursor en pantalla para saber en cada momento donde estamos , luego por ello se nos complica un poco más el programa , cuyo organigrama es el dado en la pagina siguiente.

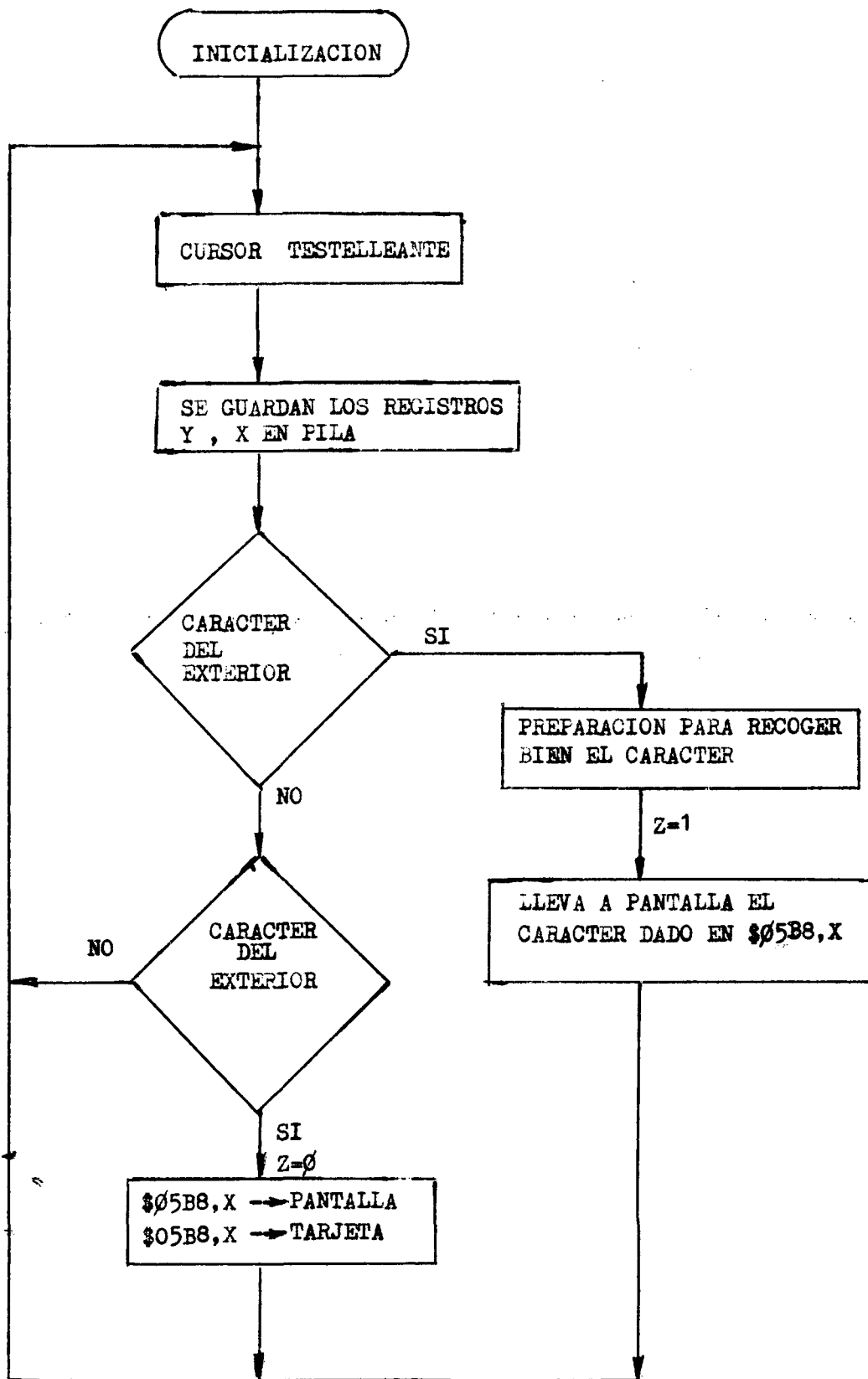


Figura 4.2

Los pasos dados en el organigrama de la página anterior son:

a) Primero se inicializa la tarjeta , poniendo el stack , 1 bit de stop , borrado de pantalla y X=CN .

b) Luego se ponen los flags necesarios para que aparezca un cursor testelleante en pantalla , pero que no lo haga ni muy rápido ni muy lento .

c) Después se pasa a la subrutina de entrada de carácter donde si hay carácter lo mete en la posición \$05BB,X y pone el flag de cero a 1 o 0 dependiendo de si viene desde fuera o viene desde teclado . Y si no hay carácter sigue dentro del bucle , si bien tiene que sacar el cursor testelleando en pantalla.

d) Cuando se vuelve de la subrutina de entrada de carácter tenemos entonces el flag de cero a 1 o 0 dependiendo de si es entrada o salida respectivamente , luego dependiendo de ello nos vamos a entrar mandar el dato a pantalla , y si es salida lo manda a través de la tarjeta mediante la subrutina que está en la posición \$C9AA , y a su vez puede mandarlo a pantalla o no dependiendo de si se quiere eco o no .

SUPER SERIAL CARD (SSC)

En esta tarjeta es donde se trabajado más , motivado quizás por las habilidades que de por sí ya vienen implementadas en la propia tarjeta . Aquí las señales si que son activas .

Vamos a ver primero las características más relevantes de dicha tarjeta y luego se expondrá lo que se pretende hacer

con ella y lo que se consigue .

-Introducción a la SSC.- La SSC tiene dos formas de interconexión serie para una amplia variedad de dispositivos incluyendo impresoras , terminales , plotters y otros computadores. Todos estos dispositivos pueden conectarse a la SSC directamente o mediante MODEM .

La SSC reemplaza las variedades PB y PBA de Serial Interface Card , también a la Communications Card y soporta el modo terminal . Finalmente la SSC soporta los comandos software de la tarjeta de interconexión del Apple II. La SSC puede configurarse para conectarse a otro dispositivo externo en 3 formas :

- a) Por la propia puesta de los switch de la tarjeta .
- b) Por comandos desde teclado , tanto en monitor , Integer Basic , Applesoft o DOS .
- c) Por comandos emitidos desde lenguaje ensamblador o programas en BASIC o PASCAL .

Echando una ojeada a la tarjeta nos encontramos con :

- El jumper block . Ordinariamente apunta hacia el modo terminal , pero si tenemos que conectarlo a un MODEM hemos de ponerlo en modo modem .
- Los switch. Van numerados desde SW1-1 a SW1-7 los de la izquierda y SW2-1 a SW2-7 los de la derecha .
- El borde del conector que se une al slot es importante que no se toque con los dedos .
- El zócalo de cables .

Modo de Impresión .-

Para estar en este modo se ha de poner SW1-5 OFF y SW1-6 ON y el jumper block apuntando hacia terminal .

El modo de impresión permite usar la SSC con una impresora o terminal local (esto es directamente conectado) y también con otros dispositivos series locales . Después de instalar la SSC , se puede controlar su funcionamiento desde BASIC , PASCAL o programa en lenguaje ensamblador e incluso directamente desde el teclado .

La puesta de switch determina entre otras las opciones de rango de velocidad (en baudios) , el formato de los datos , la paridad , delay de carry return , ancho de línea y video ON/OFF (eco o noeco) , y generación o no de Line Feed ; pero estas opciones se pueden cambiar luego mediante comandos (por medio del CTRL-I) , tal es el caso del tiempo de delay de CR , LF y FF .

Modo de Comunicación .-

Para estar dentro de este modo se ha de poner SW1-5 ON y SW1-6 ON , además el jumper block debe apuntar hacia MODEM .

El modo de comunicación permite usar la SSC con un MODEM conectado a un dispositivo remoto (tal como una impresora , terminal u otro computador remoto). Después de instalar la SSC se puede controlar su funcionamiento desde un programa en BASIC , PASCAL o lenguaje ensamblador e incluso desde el teclado.

Al igual que en el modo de impresión las opciones dadas con la puesta de los switch se pueden cambiar mediante

comandos , pero ahora en vez de CTRL-I es CTRL-A.

En nuestro caso particular se han puesto los switch para una transmisión en modo de comunicación , a 1200 baudios , 8 bits de longitud de carácter , sin paridad y 1 bit de stop. Esto es así porque también pretendemos usar el Apple como terminal conectado al MODEM que nos conecta al ordenador Hewlett-Packard de la serie HP-3000 ; de hecho, la comunicación se ha hecho , si bien al principio creó algunos problemas que estaban en los cables de conexión , y luego era que recibíamos mal . Analizando tal situación se vió que había un error de framing (no se recibía bien los bits de stop) , luego como lo teníamos puesto a 2 bits de stop , decidimos ponerle 1 y sorprendentemente funciona bien... Pero hay que decir que cuando la pantalla empieza a subir se pierden 1 o 2 caracteres al principio de la línea , esto se puede deber a un error de overrun (no se recupera el carácter recibido desde el registro receptor de datos antes que llegue el otro carácter), o sea al tener que mover la pantalla se pierde un tiempo que es esencial para recoger el dato . Con esto no podemos hacer nada , si bien en la tabla dedicada a los problemas en el dispositivo nos viene que la mejor solución es poner la tarjeta de 80 columnas , lo cual resultaría mucho mejor como terminal ; pero dicha tarjeta hasta el momento no se ha podido conectar estando la SSC en el mismo Apple, ello se cree debido a problemas de desactivación del flag de expansión de los 2K (\$C800-CFFF) para que puedan utilizar esas posiciones una u otra tarjeta .

Dentro del modo de comunicación , la SSC puede entrar en modo terminal y hacer al Apple II como un terminal inteligente . Esto es útil para conectar al Apple II a un computador en tiempo compartido , o simplemente trabajando con otro Apple II .

En este modo la tecla ESCAPE nos sirve para pasar de mayúsculas a minúsculas y viceversa .

Para entrar en modo terminal hay que poner :

IN#n (siendo n el número del slot) <RETURN>

<CTRL-A> T <RETURN>

Comandos de modo terminal .-

- T entra en modo terminal.
- B transmite una señal de Break.
- E <E/D> pone o quita eco .
- S <E/D> pone o quita los caracteres especiales.
- <n> T determina tratamiento de caracteres de llegada.
- X <E/D> pone o quita reconocimiento XOFF.
- Q sale de modo terminal .

Trabajando con el SSC.-

La SSC puede trabajar con datos en serie o en paralelo , claro está que cuando se transmite a cortas distancias es en paralelo y cuando son largas es en serie .

El formato de los datos viene dado por las normas estándar del RS-232C , esto es : 1 bit de start al comienzo, un bit de paridad opcional al final de los 5 a 8 bits de datos y finalmente 1 o 2 bits de stop .

Las señales RS-232C son compatibles para trabajar con un

MODEM a través de una línea telefónica, por eso tienen esta similitud dada por ejemplo en DTR (Data Set Ready) es descolgar el teléfono, y así seguiríamos sucesivamente con todas las demás señales, esto es con DSR (Data Set Ready), RTS (Request To Send), CTS (Clear To Send), TxD (Transmisión de datos), RI (Ring Indicator) y DCD (Data Carrier Detect) entre otras.

- Trabajo realizado con la tarjeta SSC . Conclusiones

Ya descrita la tarjeta pasamos a mencionar lo que pretendemos hacer con ella, esto es, una especie de terminal inteligente. Para ello nos basamos en dos programas: uno en BASIC y el otro en lenguaje ensamblador con los que pretendemos sobre todo realizar tareas con ficheros, esto es leer datos del fichero o escribir dentro de él. Al leer esos datos podemos tanto sólo enviarlos a pantalla, como también enviarlos por la tarjeta SSC al otro Apple.

Pero dado que para tal tarea hemos de realizar un programa en BASIC y otro en lenguaje ensamblador que estén estrechamente relacionados, no podríamos cargar otro programa en BASIC por ejemplo, cuando trabajamos en modo terminal. Este problema se soluciona cuando se sepa exactamente que posiciones ocupa un programa en BASIC y entonces guardarlo como un fichero como hacemos con los ficheros normales que utilizamos. Esta cuestión no es difícil, pero requiere un estudio más amplio.

Civándonos a lo que pretendemos en principio, lo primero que hacemos es establecer la comunicación con un programa en

ensamblador que tenga el protocolo ENQUIRY/ACKNOWLEDGE , esto es para prepararlo para trabajar como el terminal de ordenador del HP-3000 que se tiene y cuyo protocolo consiste en que cuando se recibe un "05" desde el ordenador se tiene que contestar con un "06" para que no se rompa la transmisión y sepa que sigue conectado (esto lo hace cada cierto tiempo).

Esto se consiguió sin grandes dificultades (el establecimiento del protocolo) . Para ello se coge parte del programa que ya tiene la tarjeta implementada y se va a la entrada de datos desde fuera de tal manera que cuando llegue un dato mira si es de protocolo : si es , devuelve lo ya dicho , pero si no es lo trata normalmente. Hay que decir que lo que se hizo para poder realizar esto y en general todo el programa en sí, es copiar el programa de la tarjeta a partir de una posición dada ; y trabajar con la parte de programa que nos interesa. Al mismo tiempo que se puede realizar las modificaciones oportunas en nuestro programa y acceder al verdadero de tarjeta cuando sea necesario .

Pero lo que nos originó más problemas fuè la intención de trabajar con ficheros. Esto suponía en principio tener dos programas uno en BASIC y otro en lenguaje ensamblador, ya que desde ensamblador no sabemos cómo acceder a la unidad de disco. Luego el primer problema estribaba no en saltar desde BASIC a ensamblador (puesto que para ello sólo hay que saber a la posición a la que se tiene que acceder y hacer un CALL o llamada a esa posición desde BASIC) , sino en retornar de ensamblador a BASIC . Sabemos que hay una posición a donde se

salta y hace este cometido (JMP \$FEB0), pero al retornar a BASIC elimina o borra todo el programa y las variables que hay en memoria. Este salto nos sirve cuando queremos salirnos del modo terminal, pero para volver simplemente no nos sirve.

Luego en principio lo que se intentó fué pasar de BASIC a ensamblador y luego retornar con un simple RTS, y desde luego que lo hacia y ejecutaba la siguiente instrucción en BASIC, pero el RTS no lo podíamos usar en un programa largo donde hubiera que meter y acceder muchas veces a la pila ya que se crearían muchos problemas. Entonces lo que intentamos averiguar era la posición que accedía a BASIC, puesto que había una posición que el RTS tomaba como referencia para volver.

Ahora vamos a explicar cómo conseguimos averiguar esa posición de retorno a BASIC que se pretendía hallar.

Todo se basa en ir a la pila, colocarla a partir de una determinada posición y observar cuál es la posición que se graba en ella cuando se viene desde BASIC; no si grandes dificultades descubrimos la importante posición (\$DB24).

Pero esto plantea otro problema como es el de que cuando se pasa de BASIC a ensamblador se graban en pila 2 posiciones de memoria que luego no se utilizan para retornar. Aquí hay que hacer el inciso de que no se utiliza el STACK para retornar a BASIC porque podrían haber problemas de no saber cómo se encuentra dicho Stack al retornar; pero de hecho podría utilizarse, si bien habría que contemplar todas las

posibilidades de retorno y tener la pila bien posicionada.

Entonces para la resolución de tal problema lo que se hace es simplemente vaciar la pila dos veces cuando se vuelve de BASIC mediante dos PLA seguidos , con lo cual la pila queda intacta , pero para mayor seguridad lo que se hace además es guardar las dos últimas posiciones antes de ir a BASIC y reponerlas después de volver desde allí.

Luego ya se empieza con los ficheros y se empieza con los ficheros de acceso aleatorio ya que los secuenciales , aunque permiten mayores posibilidades como son las de añadir algo a un fichero determinado y otras peculiaridades, son mucho más complejos a la hora de trabajar con ellos por el hecho de que cada vez que se hace un acceso se tiene que abrir/leer/escribir y luego cerrar ese fichero . Por lo demás si los de acceso aleatorio nos sirven para la función que queremos desempeñar , para qué vamos a complicar más el asunto o mejor dicho el software.

Ahora vamos a hacer un inciso para detallar brevemente como se accede a los comandos de ficheros desde un programa en BASIC.

- Uso de comandos DOS en programas.-

Estos comandos DOS dentro de un programa en BASIC se colocan dentro de una sentencia PRINT , prefijada mediante un código ASCII de cuatro caracteres . Normalmente se usa CTRL-D para crear este prefijo , luego por seguir más o menos una norma aquí se utilizará también este prefijo .

El CTRL-D debe ser el primer carácter que se saca por la

sentencia PRINT . El PRINT anterior no debe terminar en ","
ni ";" .

Por ejemplo :

```
10 D$=" "  
20 PRINT D$:"CATALOG"  
30 END
```

Sólo hay que tener en cuenta que dentro de las comillas hay que poner CTRL-D , y ya ejecutando el programa nos daría el CATALOGO del disco .

Para eliminar el CTRL-D sólo hay que usar el comando nulo, esto es poner PRINT D\$ en una línea determinada .

Como de los ficheros aleatorios no necesitamos más por ahora cerramos este inciso y volvemos con la resolución de los problemas planteados .

Hay que decir que el programa en BASIC está preparado para que cuando se corra cargue el programa que hay en ensamblador, o sea sólo hay que preocuparse de correr el programa en BASIC llamado SI90 o SI90A dependiendo de si se quieren ficheros hexadecimales o normales y ya entraríamos en el modo terminal de la tarjeta.

Bien , el programa ahora va a empezar yendo de BASIC a ensamblador , poner el modo terminal y cuando se detecte un CTRL- ó comando de fichero se vuelve a BASIC a realizarlo . Pero eso vuelve a plantear problemas . Un primer problema es que al hacer el acceso a ficheros de disco se rompe parte del programa , luego cada vez que se hace un acceso a ficheros se ha de reinicializar la tarjeta ; y esto se hace sacando el

carácter nulo por la tarjeta cuando se vuelve desde BASIC. Luego vemos que sólo hace el bien el acceso a BASIC cuando en una dirección de trabajo de RAM que utiliza la tarjeta y que es la 07FA ó 073B,X contiene el valor 2E , y por otra parte si tiene 2E ese flag en vez de 6F no responde bien a los propios comandos de la tarjeta (cuando tiene 6F responde bien). Luego lo que se hace es colocar 2E cuando se va a producir un salto a BASIC y cuando se retorna se procede a colocar otra vez 6F con lo cual los comandos propios de la tarjeta también funcionan bien .

También se ha de señalar el problema que se nos plantea cuando cada vez que se corre el programa se desestabiliza la tarjeta . Ello se debe a que hay posiciones de memoria que aunque se creen libres , no lo son tanto . Estas posiciones son , por ejemplo los \$2A00 bytes que transfiere el DOS al final de la carga inicial y que nos recorta el espacio que pretendemos tener para grabar a fichero o sacar de él. Otras posiciones que se cargan en principio son desde la 0C00 a 0C6F y desde la 1B00 a 1FFF .

Una vez descritos los problemas planteados vamos a describir el programa en sí . Podemos establecer primero la descripción del programa generalizado (fig 4.3) y luego ya iremos al programa más detallado (fig 4.4, 4.5, 4.6 y 4.7).

-Programa generalizado.-

Como se ve en el organigrama de flujo el programa empieza en BASIC para luego seguir en ensamblador y más concretamente en modo terminal .

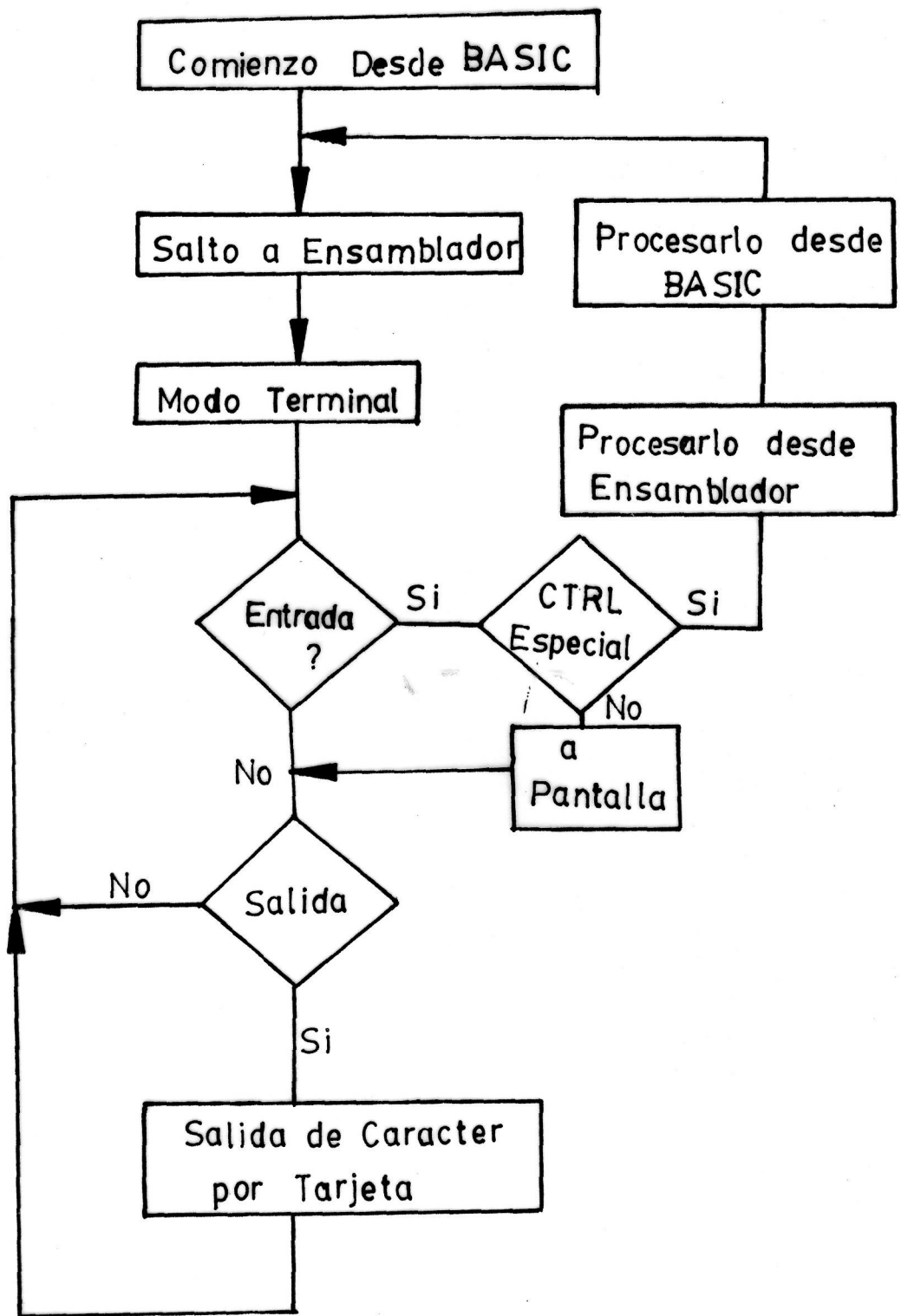


FIG.N°4.3

DISTRIBUCION DE POSICIONES (\$6000-\$BFFF)
PARA EL PROGRAMA SSC

\$ 6000	PROGRAMA DESARROLLADO
\$ 61FF	
\$ 6200	PROGRAMA DE TARJETA
\$ 69FF	
\$ 6A00	PROGRAMA DESARROLLADO
\$ 6C70	
	LIBRE(X)
\$ 7000	BUFFER PARA FICHEROS DE DISCO
\$ 95FF	
\$ 9600	UTILIZADAS POR DOS
\$ BFFF	

(X) = PARA ULTERIORES AMPLIACIONES

Cuando se está en este modo se testea si hay entrada , si la hay se procede a ver si es un control especial . Si se da ese control especial , por ejemplo CTRL-B se procede a hacer el procesamiento adecuado , primero en ensamblador y luego en BASIC , volviendo cuando se haya procesado al modo terminal en ensamblador . Si no hay ningún control especial simplemente se toma la entrada y se lleva a pantalla .

Si no hay entrada , se procede al testeo de si hay salida. Si la hay se procede a sacar el carácter por tarjeta y también a pantalla si está establecido el modo ECO , luego al igual que si no hubo salida se vuelve al bucle de entrada .

-Programa detallado.-

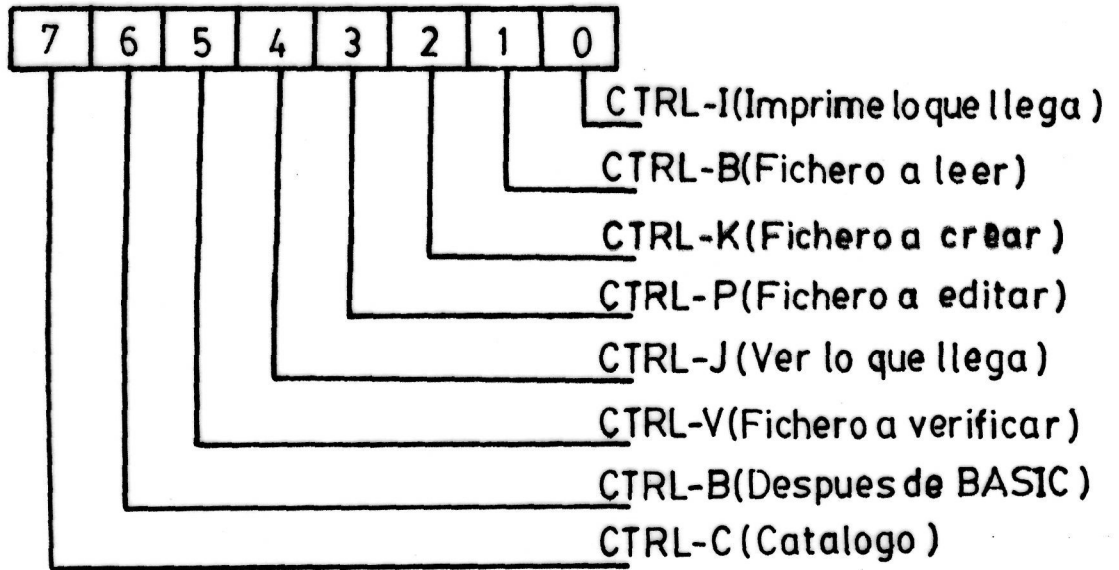
En el programa se han de distinguir dos partes claramente diferenciadas :

- Programa en ensamblador donde se establece el modo terminal y donde se ejecuta parte del procesamiento de los controles o comandos especiales.
- Programa en BASIC , que se encarga de la carga inicial y de otra parte del procesamiento de los controles o comandos especiales.

Lo que vamos a hacer es un seguimiento evolutivo de las distintas partes por donde pasa el programa empezando desde BASIC , pasando por ensamblador y volviendo de nuevo a BASIC cuando sea necesario . Estos pasos los podemos ir viendo en las figuras 4.4, 4.5, 4.6 y 4.7 .

a) El primer paso es la inicialización desde el programa BASIC , el cual realiza la activación de los comandos DOS

FLAG[\$1000]



\$07F8+S (MISFLAG)

0	Genera <LF> después de <CR> cuando 1
1	Modo IMPRESION cuando 0; Modo COMUNICACION si 1
2	1 → Teclado habilitado
3	<CTRL-S> (0XFF) Chequeo de entrada <CTRL-R> y CTRL-T si 1
4	PASCAL — 1; BASIC — 0
5	1 → Descarga <LF>
6	1 → Habilita generación de minúsculas y caracteres especiales (Modo COMUNICACION)
6	1 → Opción de Tabulado (Modo IMPRESION)
7	1 → Modo ECO

desde BASIC y luego realiza la carga del programa ensamblador correspondiente (CALMA90) que se encuentra en el diskette . Una vez hecho esto se pasa a ensamblador.

b) En ensamblador se procede igualmente a la inicialización del sistema que conlleva una serie de pasos:

- * Inicialización de los flags (cero, carry, ...)
- * Inicialización del puntero de la pila
- * Inicialización de los registros X e Y, donde X=CN e Y=N0.

* Puesta de flags propios del modo terminal , y dentro de ellos , la puesta de 6F en 07FA para que reconozca los comandos propios de la tarjeta en modo terminal.

* Reseteo del espacio reservado para llegada de datos de fichero y del espacio reservado al nombre del fichero.

* Reseteo del flag de comando (\$1000) , otros que se utilizan como registros temporales y por último uno que se utiliza para el conocimiento de la longitud del buffer cuando es necesario.

c) Luego ya se establece el modo terminal dentro del cual se tienen unos subpasos a seguir .

Se establecen los distintos testeos que se van a ir detallando y resolviendo uno a uno.

- Testeo de flag activo .- Si no se da se sigue al siguiente testeo , pero si se da salta a la posición \$6AA0 donde se tiene la subrutina de tratamiento de flag que se verá más tarde .

- Testeo de entrada .- Si se da , se comprueba si el

carácter llegado es un ENQUIRY (\$05) , a lo cual se responde con un ACKNOWLEDGE (\$06) . Si la llegada no es un ENQUIRY , simplemente se recoge el carácter y se lleva a pantalla.

- Testeo de salida .- Si no hay salida se vuelve al bucle que denominamos principal , o sea al testeo de si hay flag activo. Pero si hay salida se verifica si es un control o comando especial :

1) Verificación de CTRL-C , que de darse pondría en flag el valor #\$80 y luego pasaría a BASIC a presentar el CATALOGO en pantalla.

2) Verificación de CTRL-Z , ante el cual se realiza un BREAK durante un corto periodo de tiempo .

3) Verificación de ESCAPE , dado el cual se rompería el programa , ya que se vuelve a BASIC , pero elimina el programa y variables que allí residen .

4) Verificación de CTRL-S , ante lo cual se realiza un bucle de espera hasta que llegue el CTRL-Q.

5) Verificación de CTRL-I , ante lo cual se pondría el flag \$1000 a #\$01 y se vuelve al bucle principal.

6) Verificación de CTRL-K , dado el cual se pone el flag a #\$04 y pasa a la subrutina en la \$6A00 donde se pone :

" NOMBRE DE FICHERO A CREAR " y se espera a que se le ponga el nombre , dado el cual se vuelve al bucle principal .

7) Verificación de CTRL-B , dado que es análogo al caso anterior sólo que ahora se pone flag a #\$02 y el comentario a sacar es : " NOMBRE DE FICHERO A LEER " .

8) Verificación de CTRL-P , caso análogo a los dos

anteriores donde ahora se pone flag a `##08` y la subrutina que sigue saca el comentario : " NOMBRE DE FICHERO A EDITAR " , y espera por dicho nombre , dado el cual vuelve al bucle principal .

9) Verificación de CTRL-J , ante lo cual se pone el flag a `##10` y se vuelve al bucle principal.

10) Verificación de CTRL-V , caso análogo al del CTRL-K , sólo que ahora la subrutina siguiente saca el comentario :

" NOMBRE DE FICHERO A VERIFICAR " y se hace la espera por dicho nombre para luego retornar al bucle principal .

11) Verificación de CTRL-A , dado el cual pasaría a poner el comentario : " ENTRE EL COMANDO " y ya pasaría a la subrutina propia de tratamiento de comandos de tarjeta , para luego retornar al bucle principal .

12) Si no se presenta ningún caso de los anteriores se pasa a sacar el carácter por tarjeta y si está establecido el modo ECO también se presenta en pantalla.

d) En este apartado vamos a comentar la subrutina propia de tratamiento de ficheros en ensamblador , que comienza en la dirección `$6AA0` y a la cual ya hemos hecho mención antes.

Se establecen las siguientes verificaciones:

1) Verificación de fichero de entrada , dado el cual pasa a coger el carácter y sigue con las verificaciones siguientes:

- Verificación de ENQUIRY , a lo cual se responde con ACKNOWLEDGE y vuelta al bucle principal .

- Verificación de final de fichero , para lo cual se

comprueba si el carácter llegado es un ##D1 o ##D3 . Esto es por hacerlo similar al programa QUILLY o sea para que trabaje como el terminal .del HP-3000 , pero podría haberse adoptado otra nomenclatura para cuando se trabajase entre APPLES y de hecho se tiene que hacer puesto que esos caracteres se corresponden con los caracteres de teclado "S" y "Q" y esto lleva consigo el que no se puedan teclear dentro de un fichero .

Si se da el final de fichero se pasa a BASIC para el tratamiento del comando de fichero dado.

- Verificación de Buffer lleno , dado el cual se pasa a poner el comentario : " BUFFER DEMASIADO GRANDE " y se retorna a BASIC.

- Si no se da ningún caso de los anteriores , se guarda el carácter en el BUFFER y se retorna al bucle principal.

2) Fichero de salida .- Dado que no era fichero de entrada quiere decir que es fichero de salida . Entonces se pasa a comprobar si es salida por tarjeta (CTRL-B) , si no es pasa ya a BASIC , pero si es se pasa a la subrutina de tratamiento de caracteres de salida que está en la \$6C00 . Los pasos a seguir entonces son :

- Carga posición desde el Buffer , claro está que antes y desde BASIC se ha cargado el Buffer con un fichero de disco.

- Verificación de final de fichero , dado lo cual retorna al bucle principal .

- Verificación de Buffer lleno , a lo cual se responde con el comentario : " BUFFER DEMASIADO GRANDE " y luego retorna

al bucle principal.

- Si no se da ninguno de los casos anteriores se saca el carácter por tarjeta y luego se retorna al bucle principal.

e) En este apartado se comenta el programa dado en BASIC después de volver desde ensamblador. Los pasos a seguir son determinados por el valor del flag que se pasa a la variable " C " .

- 1) Verificación de CTRL-C (C=##80), dado el cual se pasa a volcar el Catálogo del disco en pantalla, y luego se vuelve a ensamblador.

- 2) Si no es un control o comando que necesite "nombre" pasa directamente al paso 4, de otra manera recoge el nombre del fichero ya grabado en unas posiciones dadas y lo mete en la variable E\$.

- 3) Si es Fichero a Crear pasa directamente al paso 4, de otra manera indica que es salida (CTRL-V o CTRL-B). Por lo tanto pasa a cargar el fichero desde el disco al Buffer.

- 4) Verifica si es modo de impresión en cuyo caso se activa.

- 5) Salida de valores del Buffer a pantalla y a impresora si está activada.

- 6) Deshabilita el control de impresión si lo había.

- 7) Reseta Comandos DOS desde BASIC.

- 8) Si no es fichero a crear pasa directamente a ensamblador, de otra manera antes de volver carga la longitud del Buffer y lo graba en disco.

f) Se comenta ahora la reinicialización que se hace al volver a ensamblador. Los pasos a seguir son :

- Restablecer el verdadero puntero del Stack.
- Reinicialización de la tarjeta haciendo salir por ella el carácter espacio.
- Si es comando de " fichero a leer " pasa a la subrutina de sacar caracteres por tarjeta . De otra manera reinicializa flags y buffer , y de nuevo se pasa al modo terminal.

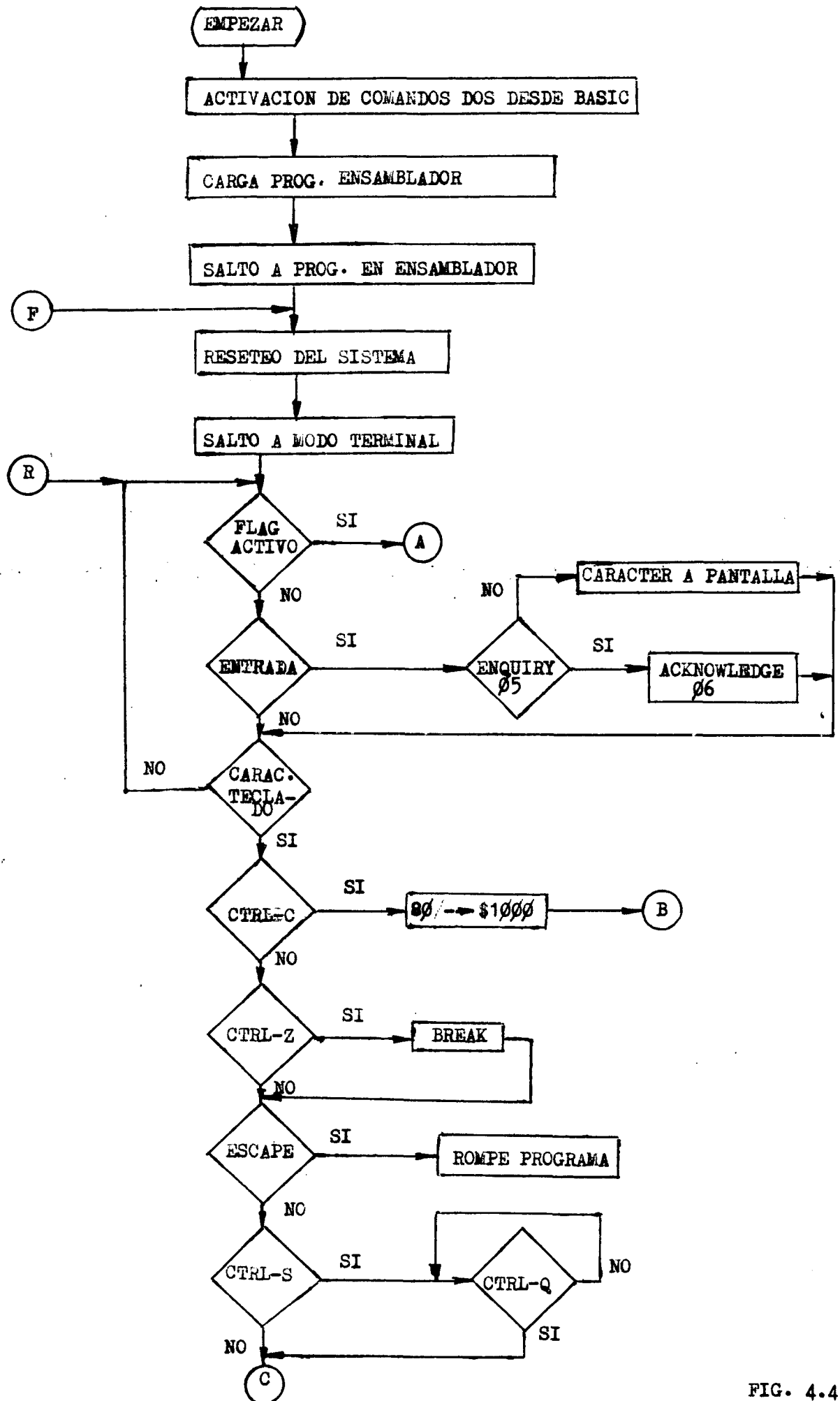


FIG. 4.4

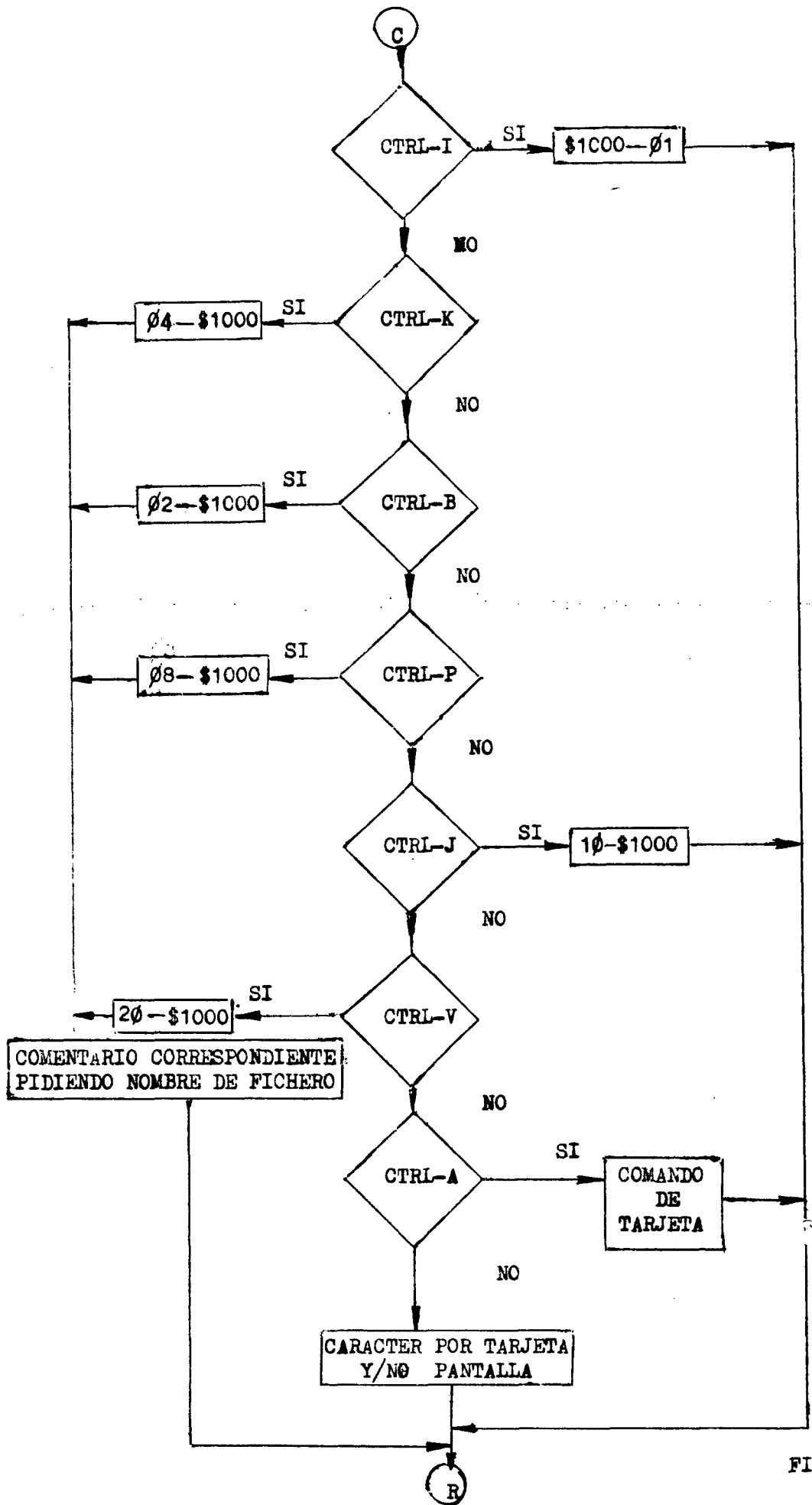


FIG. 4.5

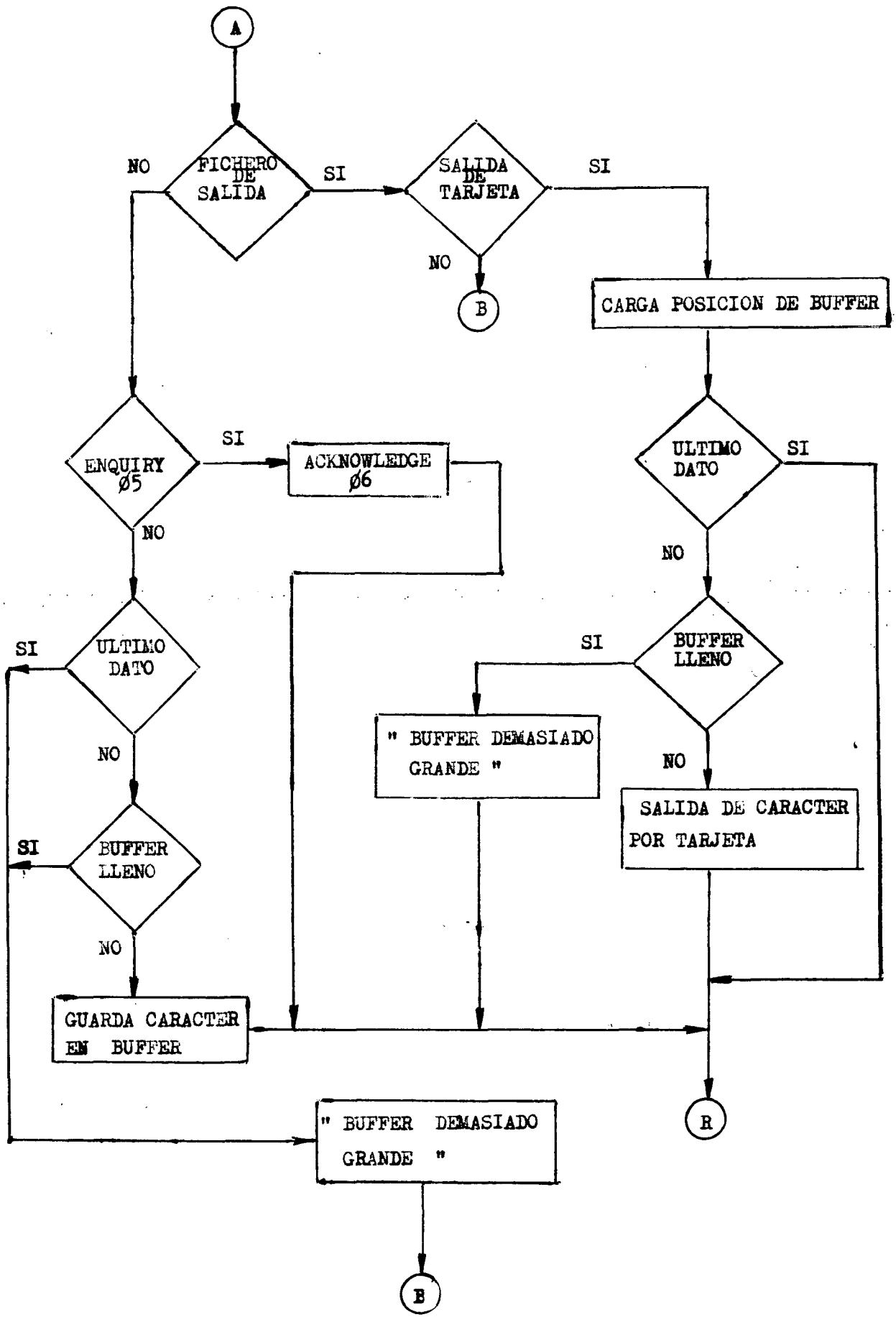


FIG. 4.6

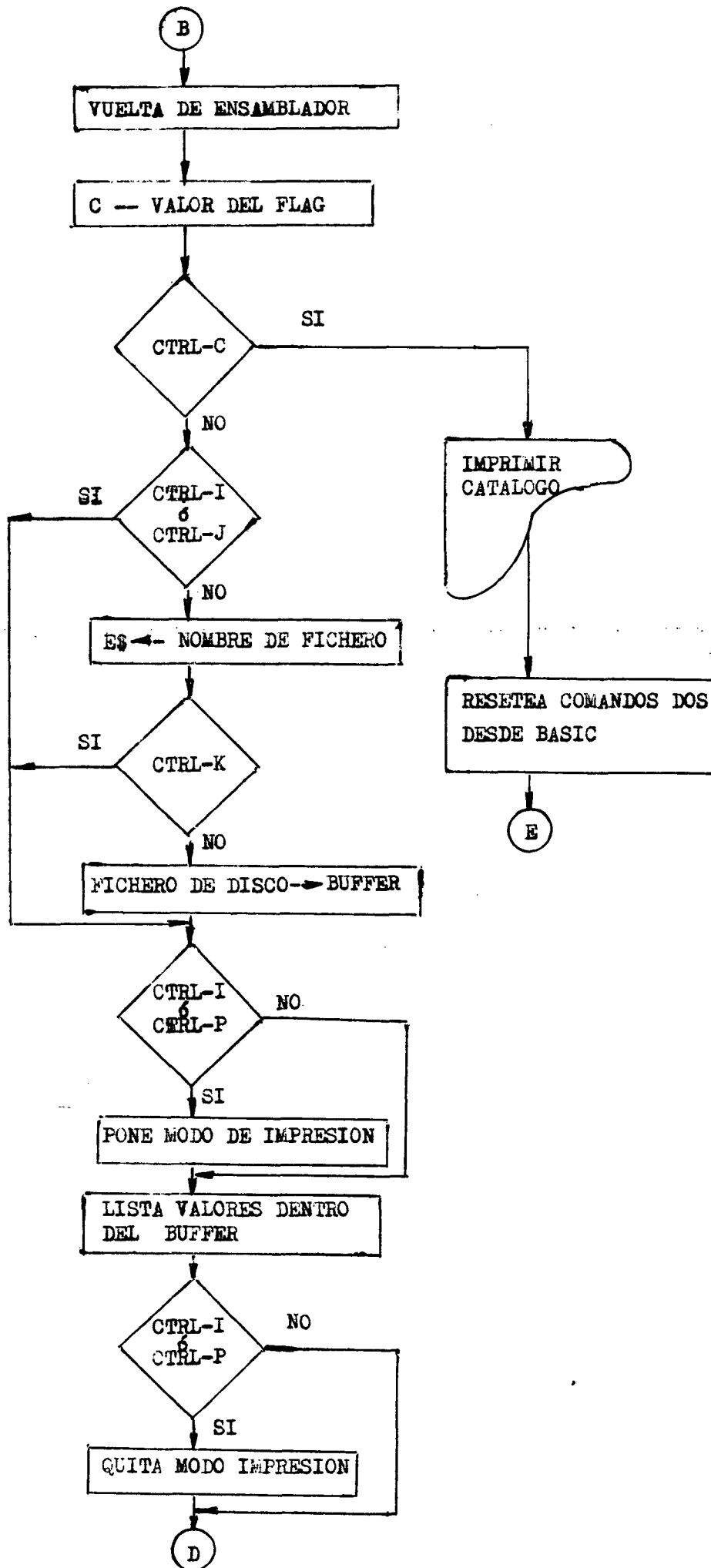


FIG. 4.7

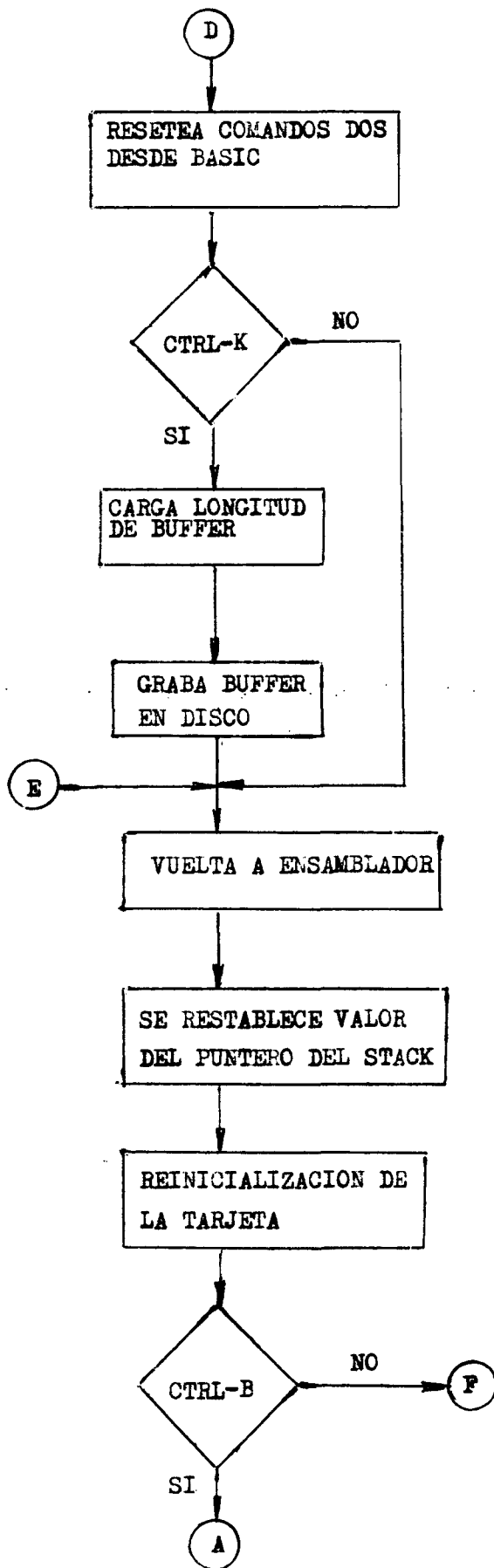


FIG. 4.8

CAPITULO 5

CAPITULO 5 : ESTRUCTURA A NIVEL HARDWARE

En este capítulo se hace mención de la estructura de la tarjeta SSC (de una manera muy concisa) y luego ya hablaremos del hardware propio del circuito de protocolo .

Estructura física de la SSC

Los principales componentes de la SSC (fig 5.1) son :

- 50 pines de conexión al slot de conexiones.
- 1 bus de direcciones de 12 líneas.
- direccionamiento y control lógico
- una ROM de 2K * 8 bits.
- 2 bloques de 7 switch cada uno
- 2 registro para la puesta de switch
- la ACIA (6551)
- un oscilador para la ACIA de 1.8432 Mhz
- un interface transmisor y otro receptor
- un bus de datos de 8 líneas
- un buffer para el bus de datos
- un jumper block que puede funcionar como eliminador de modem.
- una cabecera de 10 pines para conectar el SSC al DB-25 (del RS-232-C)

Las 12 líneas de datos proveen el direccionamiento necesario en la SSC. El control lógico y las señales de RESET, SELECCION DE DISPOSITIVO , SELECCION E/S y STROBE E/S asegura la ruta a seguir a las direcciones apropiadas.

La SSC sigue el protocolo de acceso al uso del espacio direccionado a partir de la \$CB00 , para ello dispone de un

latch.

La ACIA está ampliamente descrita ya en el Apéndice 2 , luego sólo decir de ella que ejecuta conversiones de datos paralelo / serie y serie / paralelo y transferencias de datos en simples buffers.

Para los datos de entrada y salida se utilizan los MC-1489 (cambio de voltajes RS-232-C a TTL) y los MC-1488 (cambio de voltajes de niveles TTL a niveles RS-232-C).

El jumper block tiene dos posiciones : cuando apunta hacia MODEM trabaja como un DTE (Data Terminal Equipment) que está preparado para hablar con el DCE (Data Communication Equipment) , tal como un Modem . Si apunta hacia terminal es como un DCE , luego puede comunicarse directamente con una impresora o cualquier DTE.

ESPECIFICACION DEL MODULO DE ENSEANZA INFORMATIZADO

Pretendemos realizar un módulo o caja negra que establezca el protocolo necesario para conectar 1 terminal de un alumno de entre 5 terminales , al del maestro (MASTER) y a su vez este módulo pretendemos que se pueda conectar a otro similar de manera que se pueda establecer una cadena (fig 5.2)

En principio esta caja estaría conformada de tal manera que se establezca una prioridad de entre los terminales de acceso ,pero si ya uno está conectado no puede entrar otro aunque tenga mayor prioridad , hasta que el primero no se desconecte.

También se pretende con este módulo que esté conformado de manera tal que para establecer la comunicación han de estar

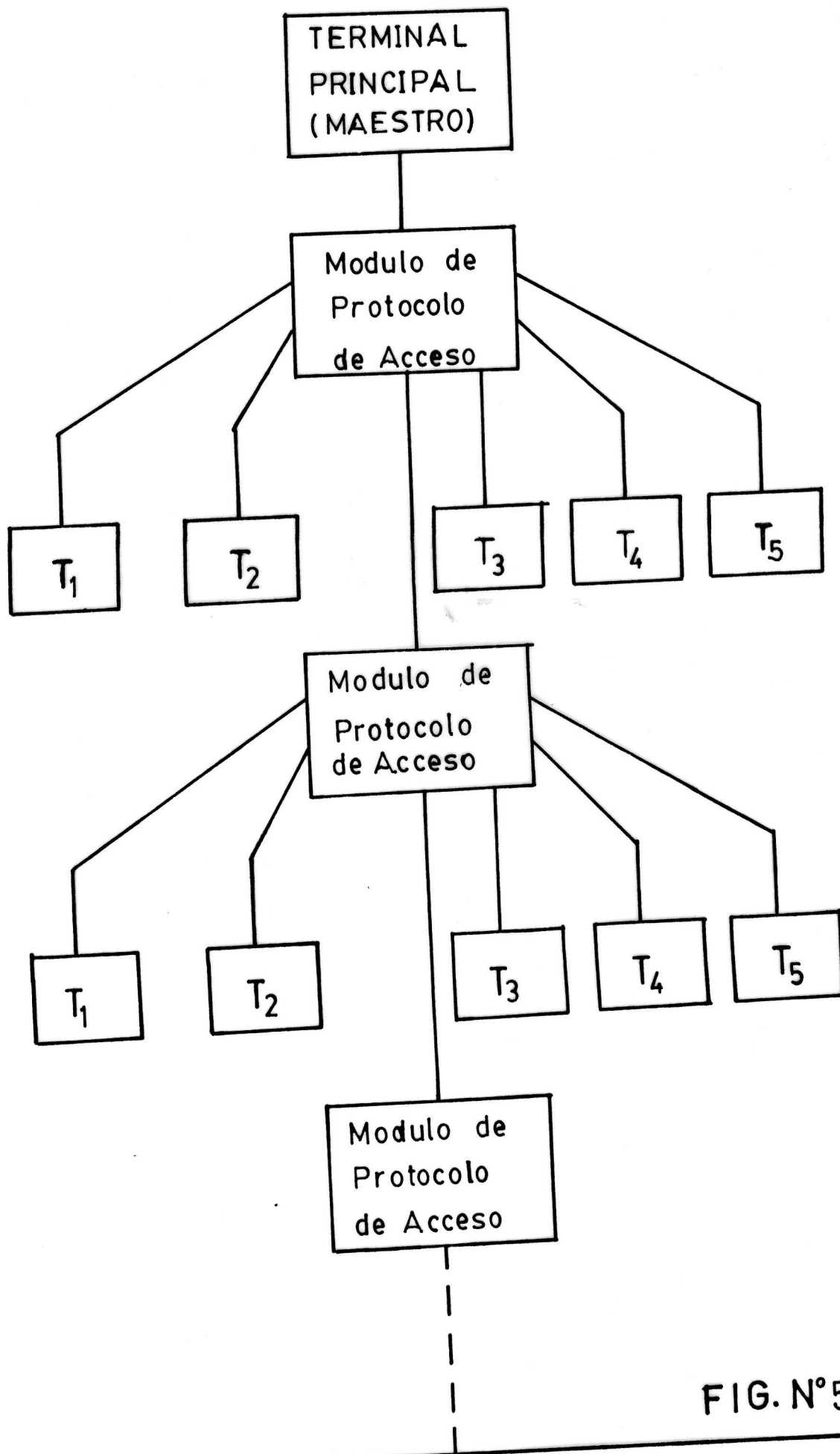


FIG. N°5.2

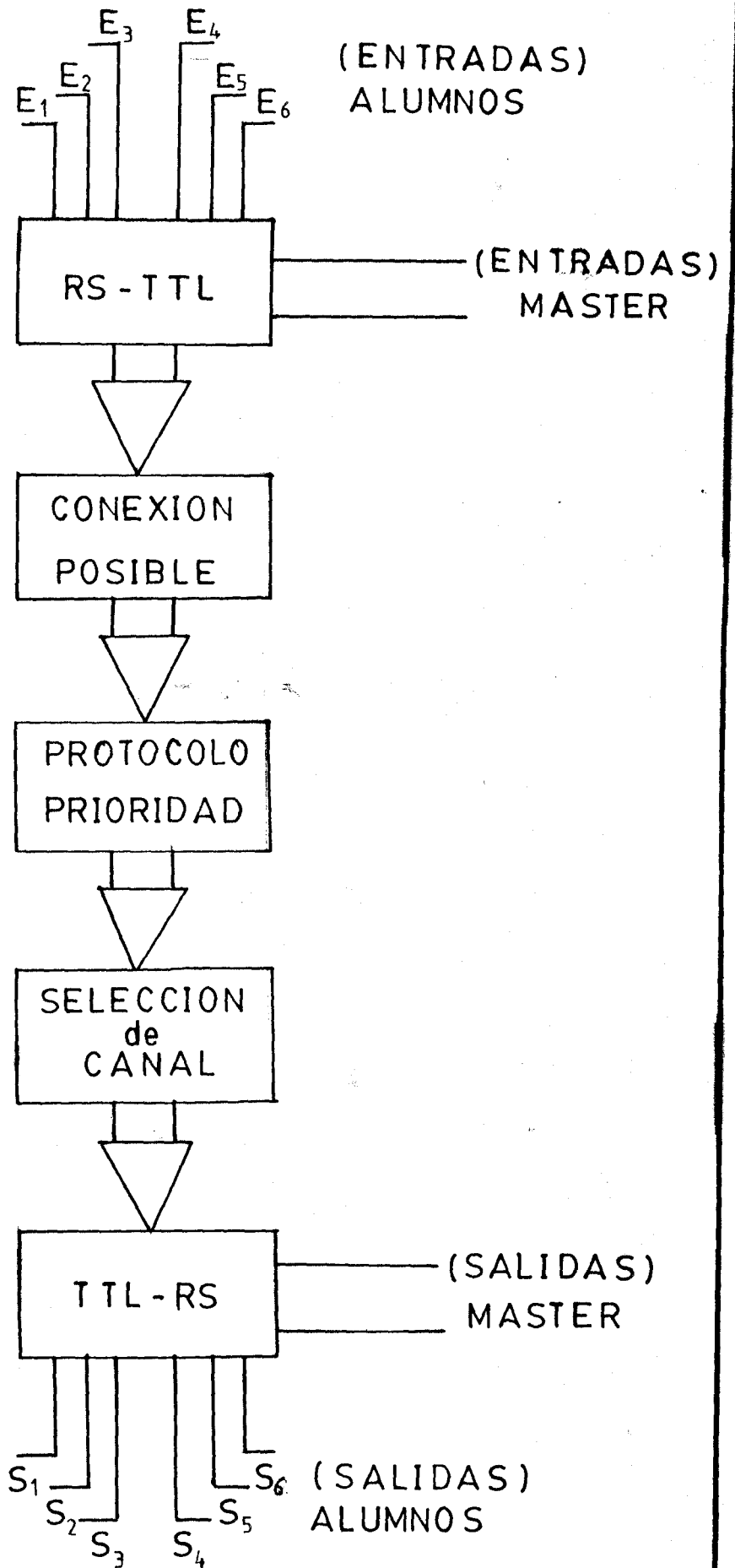


FIG.5.3

preparados para comunicar , al menos un alumno y por supuesto el maestro . Si no hay al menos un alumno desconectaría al maestro ; de igual forma si hay uno o varios alumnos intentando conectar , pero el maestro está desconectado , desconecta a dichos alumnos .

Esto de "desconecta a" tiene el sentido de decir :

- al terminal del profesor , que no hay alumnos pidiendo comunicación .

- al terminal del alumno , que el profesor no está conectado , o que hay otro alumno en comunicación .

ELECCION DE LAS DISTINTAS PARTES DEL SISTEMA

El sistema ha de constar de varias partes bien definidas :

- Conversor de señales de control/datos de RS-232-C a TTL para su posterior procesado.

- Módulo capacitado para detectar el pedido de comunicación y establecerla cuando sea posible .

- Módulo de establecimiento del protocolo de entrada a comunicar, que deja pasar al de mayor prioridad cuando llegan varios a la vez y no deja pasar a nadie cuando hay alguien comunicando (esto es, referido a los alumnos).

- Módulo de salida de señales de control/datos que además de seleccionar el canal adecuado hace la conversión de niveles TTL a niveles RS-232-C.

Todo esto puede verse mejor en la fig 5.3 .

MODO DE FUNCIONAMIENTO

Para el establecimiento de tal modo se han de tener en cuenta los voltajes reflejados en la tabla de la fig 5.4,

además de los datos significativos siguientes:

a) La tarjeta SIC se puede conectar en modo terminal con la Super Serial en modo modem, luego quiere decir que la SIC se comporta como una SSC en modo terminal.

b) Trabajamos en lógica negativa, esto es, un nivel alto es -12 voltios y un nivel bajo es +12 voltios.

c)	DTE	TXD 2 SALIDA	DCE	TXD 2 ENTRADA
		RXD 3 ENTRADA		RXD 3 SALIDA
		RTS 4 SALIDA		RTS 4 ENTRADA
		CTS 5 ENTRADA		CTS 5 SALIDA
		DSR 6 ENTRADA		DSR 6 SALIDA
		DTR 20 SALIDA		DTR 20 ENTRADA
		DCD 8 ENTRADA		DCD 8 SALIDA

Del apartado (a) se deduce que si establecemos el convenio de que la SIC sea establecida como maestro, otro maestro también podría ser la tarjeta SSC trabajando en modo terminal; y a su vez los alumnos tendrían que sus tarjetas SSC trabajan en modo modem y si utilizan tarjetas SIC éstas, han de tener los cables de TXD y RXD cambiados.

tarjeta SIC	tarjeta SIC (con TxD y RxD cambiados)
ò	ò
tarjeta SSC (modo TERMINAL)	tarjeta SSC (modo MODEM)

Del apartado (c) y de la tabla de la fig 5.8 se deduce que	
Apple SSC (TERMINAL)	Apple SSC (MODEM)
3 NEG--DCE	2 NEG--DTE
6 SAL ! 20 ENTR	4 SAL ! 6 ENTR
8 SAL ! 4 ENTR	20 SAL ! 8 ENTR

Apple SIC

3 NEG ---DCE

23 Pasa a positiva cuando se entra en modo de comunicacón.

Luego de la Super Serial (modo terminal) , cogéramos la 4 como salida y la 6 como entrada , y nos referimos a entrada y salida de control para el protocolo de acceso , del mismo modo para la SSC (modo modem) se escoge la 6 como salida y la 4 como entrada . Se ve que también se podría haber trabajado con la 8 y la 20 .

Pero , llegamos a la SIC y nos preguntamos cómo realizaremos el control para el protocolo de acceso a comunicar.

Resulta que la 23 que nosotros creíamos activa porque se paraba la comunicacón cuando la conectábamos a la patilla 3, no lo era tal , ya que si la conectamos exteriormente con un voltaje negativo no se para la comunicacón . Pero si conectamos un voltaje positivo en la patilla 3 si se para , la que realmente era activa era la patilla 3 . Pero esta patilla 3 es salida , luego puede provocar un caos en la entrada de la otra tarjeta , luego tenemos que buscar otra patilla activa. De todas formas la patilla 23 , no nos servía para lo que pretendíamos ,pero ya veremos cómo le encontramos utilidad más adelante.

Seguimos haciendo pruebas para encontrar una patilla activa que parase la comunicacón en la tarjeta SIC ya que la activación de salida podría salir de la propia patilla 3 que cambia de nivel de voltaje cuando se conecta. Pero no

damos con ninguna , y e ahí la sorpresa si quitamos la patilla 2 o le aplicamos un voltaje positivo se desconecta la tarjeta , luego es la patilla 2 , la que se ha de utilizar para el control de activación o desactivación.

Luego de lo acontecido anteriormente se deduce que hemos de tener varios módulos :

- a) desconexión del sistema
- b) protocolo de entrada
- c) protocolo de activaciones
- d) protocolo de salida

a) Para el desconexión de tarjetas se ha de tener en cuenta que si no hay Super Serial metidas en los conectores , después de la entrada de la señal de control y después del MC-1489A (convertor RS-232-C a TTL) un cero o al aire equivale a la salida de un 1 (que tomamos como desconectado), luego en esta tarjeta no hay problemas de que no esté metida en la clavija correspondiente.

Pero con la tarjeta SIC es un caso aparte. Si estuviese siempre conectada sacando la entrada de la patilla 3 (ò de la 2 si estuviese trabajando en modo modem), implicaría +10 voltios cuando estuviese desconectada y -10 voltios si está conectada y luego después del MC-1489 y de un inversor nos podría servir , pero si el conector está separado sería lo mismo que si estuviese conectado ya que a la salida del MC-1489 nos saldría un 1 que al pasar por el inversor correspondiente nos daría un 0. Luego esta patilla no nos sirve ,pero he ahí la solución : la patilla 23 había de

SEG. 23	MASTER 6	SEG. 4	MAST. 23	SAL.
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

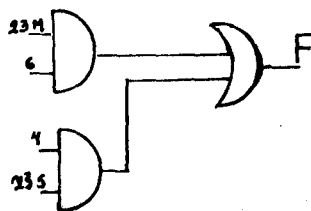
L(0) ACTIVA

23 (0 vol. → 1)
12 vol. → 0

4 (-12 → 1)
6 (+12 → 0)

23S 6 \ 4 23M		4 23M			
		00	01	11	10
23S 6	0 0	0	0	0	0
	0 1	0	1	1	0
	1 1	0	1	1	1
	1 0	0	0	1	1

$$F = (23M \times 6) + (4 \times 23S)$$



$$\bar{F} = (23S + 6) \cdot (6 + 23M)$$

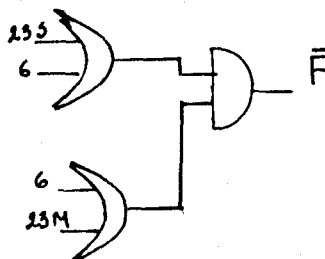


FIG. 5.5

servir para algo . Si sale un cero indica que está desconectada la tarjeta del slot o que simplemente no pide comunicación ; y si tiene +10 voltios indica que quiere entrar , luego nos sirve para el propósito esperado.

Para encontrar el circuito necesario tomamos 2 master y 2 seguidores como principio y el resultado se presenta en la fig. 5.5 , si bien después hemos de generalizarlo a todas las entradas del sistema.

Luego se ve como resulta más cómoda la opción de la función por unos que por ceros.

$$f = (23M*6) + (4*23SEG)$$

Luego ahora hemos de generalizarlo para 6 seguidores y ya está.

- Respecto a la tarjeta SIC , como lo que nos interesa es saber si hay alguna de ellas activa , simplemente podemos unir todas las patillas 23.

- Respecto a la tarjeta SSC , para saber si hay alguna de ellas activa se unen mediante puertas AND (ver fig 5.6).

b) Protocolo de entrada .-

En este apartado como se ha mencionado anteriormente se pretende que se conecte el Apple alumno de mayor prioridad y que llegue antes con el principal.

Para ello , en principio sólo se tomaban las patillas de entradas de control y la que tuviera mayor prioridad era la que conectaba y desactivaba a las otras ; pero cuando ya hay una conectada y llega otra de mayor prioridad surge el problema de que entra la de mayor prioridad en comunicación y

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

C	D	E
0	0	0
0	1	0
1	0	0
1	1	1

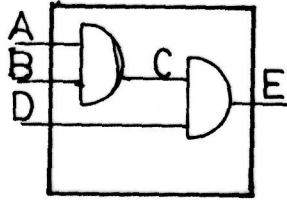
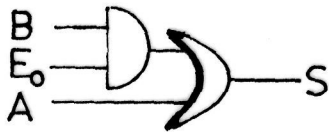


FIG. 5.6

	A	B	E_0	S
0	0	0	0	X
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	X
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

		E_0	
		0	1
A	B	X	0
		0	1
1		1	1
		X	1

$$F = B \times E_0 + A$$



$$\bar{F} = (A+B) \times (A+E_0)$$

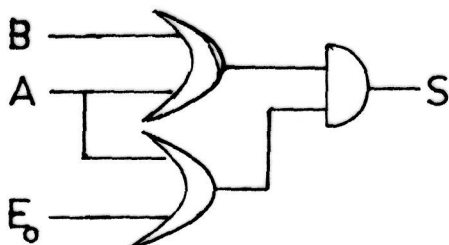


FIG. 5.7

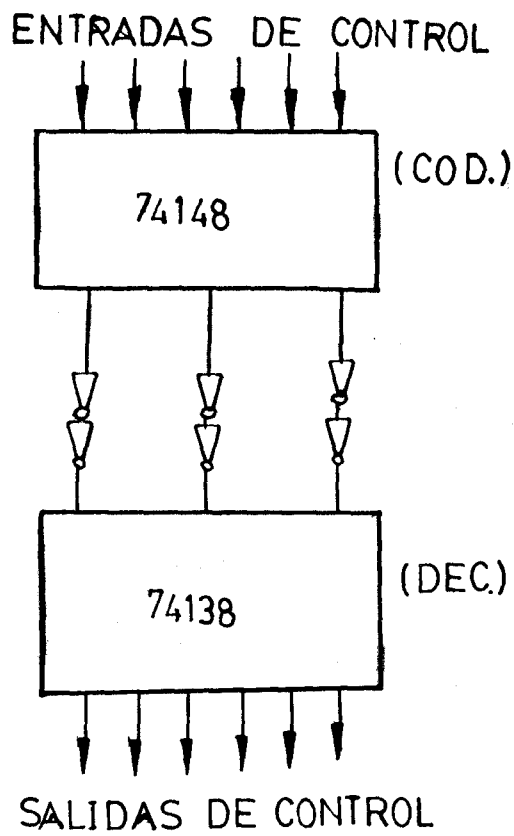


FIG. 58

SSC	SIC	S
0	0	X
0	1	0
1	0	0
1	1	1

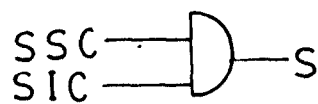


FIG. 59

la otra la hace que se desconecte . Luego ha de realizarse un circuito que contemple el caso de una conexión anterior ya establecida para que no la corte hasta que se termine. Para ello se tienen en cuenta la entrada actual , si el circuito está activado y la entrada anterior. Ver fig 5.7

Del esquema de la figura 5.7 se deduce que la función por unos resulta más sencilla .

$$f = B * E_0 + A$$

También se ve del esquema que los casos 0 y 4 no pueden darse ya que si el circuito está desactivado la respuesta ha de ser entonces 1.

c) Protocolo de activaciones .-

Como sabemos , cada tarjeta presenta una señal de control de salida y otra de entrada .Estas señales de salida se procesan , de manera que sólo una tarjeta puede entrar en comunicación a la vez . Luego de las señales de control de salida de la tarjeta se elige la de mayor prioridad , a cuya tarjeta se le dice por medio de su entrada de control que puede enviar caracteres ; y a las demás se les envia otro voltaje que las hace inhibirse y no poder conectarse. Este protocolo se realiza mediante el Codificador de prioridad 74148 y el Decodificador 74138 .Ver fig 5.8 .

d) Protocolo de salida .-

Se establece de una manera muy sencilla . O sea sólo tenemos que hacer que de las 6 señales de Transmisión sólo una llegue al receptor , para que no haya confusiones , ni cargas de señales no debidas . Para ello se toman las

distintas señales de Transmisión y se llevan al multiplexor 74151 , el cual sólo deja pasar la señal (cuyo código le viene del codificador) a la entrada del otro Apple .

Y para la recepción se hace de forma similar , luego se toma la señal de transmisión del principal (que se supone uno sólo conectado , ya sea SIC o SSC) y se llevaría a un 74138 por una entrada activa de nivel bajo (porque sino sale la señal invertida) que mediante la codificación adecuada deja pasar a la salida correspondiente la transmisión del principal.

Pero como podemos conectar tanto una tarjeta SIC como una SSC en modo terminal (MASTER) se han de tener en cuenta los casos que se pueden dar .

Si la patilla 3 es negativa quiere decir que al pasar por el 1489 dará un 1 e implicará que hemos de meterla por la entrada de habilitación positiva del 74138.

Y la otra podríamos negarla con una puerta y meterla por una entrada de habilitación negativa . Pero resulta que para que la SSC funciona bien , ya que cuando está activa y la otra inactiva o ninguna (fuera del conector) da:

0
-12 -----1

Pero , para la SIC , que de la misma forma tendría que resultar ya que

0
-12 -----1 y negado sería un 0

la señal al pasar por el inversor sale invertida y el receptor no la entiende .

Entonces se ha de implementar un circuito que observe tales características , aunque lo ideal hubiese sido que el 74138 tuviera dos entradas de activación activas a nivel alto. Ver fig 5.9

Vemos como el caso 0 no nos importa

Y vemos como con una puerta AND se soluciona el problema si bien la entrada se ha de meter por una entrada de activación negativa del 74138.

tarjeta SUPER SERIAL	tarjeta SUPER SERIAL	tarjeta SIC
modo TERMINAL	modo MODEM	normal
SIN TRANSMITIR	SIN TRANSMITIR	SIN TRANSM
(al aire)	(al aire)	(al aire)
1 -----	1 -----	1 -----
2 -----	2 -10 vol	2 0 vol
3 -10 vol	3 -----	3 +10 vol
4 +2.2 vol	4 -10 vol	4 -----
5 +2.2 vol	5 +2.2 vol	5 -----
6 -10 vol	6 +2.2 vol	6 -----
7 tierra	7 tierra	7 tierra
8 -9 vol	8 +2.2 vol	8 -----
19 -----	19 -----	12 -----
		13 -----
20 +2.2 vol	20 -10 vol	20 -----
		23 -----

tarjeta SUPER SERIAL	tarjeta SUPER SERIAL	tarjeta SIC
modo TERMINAL	modo MODEM	normal
en transmisiòn	en transm.	en trans.
(al aire)	(al aire)	(al aire)
1 -----	1 -----	1 -----
2 -----	2 -10 vol	2 0 vol
3 -10 vol	3 -----	3 -10 vol
4 +2.2 vol	4 +10 vol	4 -----
5 +2.2 vol	5 +2.2 vol	5 -----
6 +10 vol	6 +2.2 vol	6 -----
7 tierra	7 tierra	7 tierra
8 +9 vol	8 +2.2 vol	8 -----
19 -----	19 -----	12 -----
		13 -----
20 +2.2 vol	20 +10 vol	20 -----
		23 -----

fig 5.4

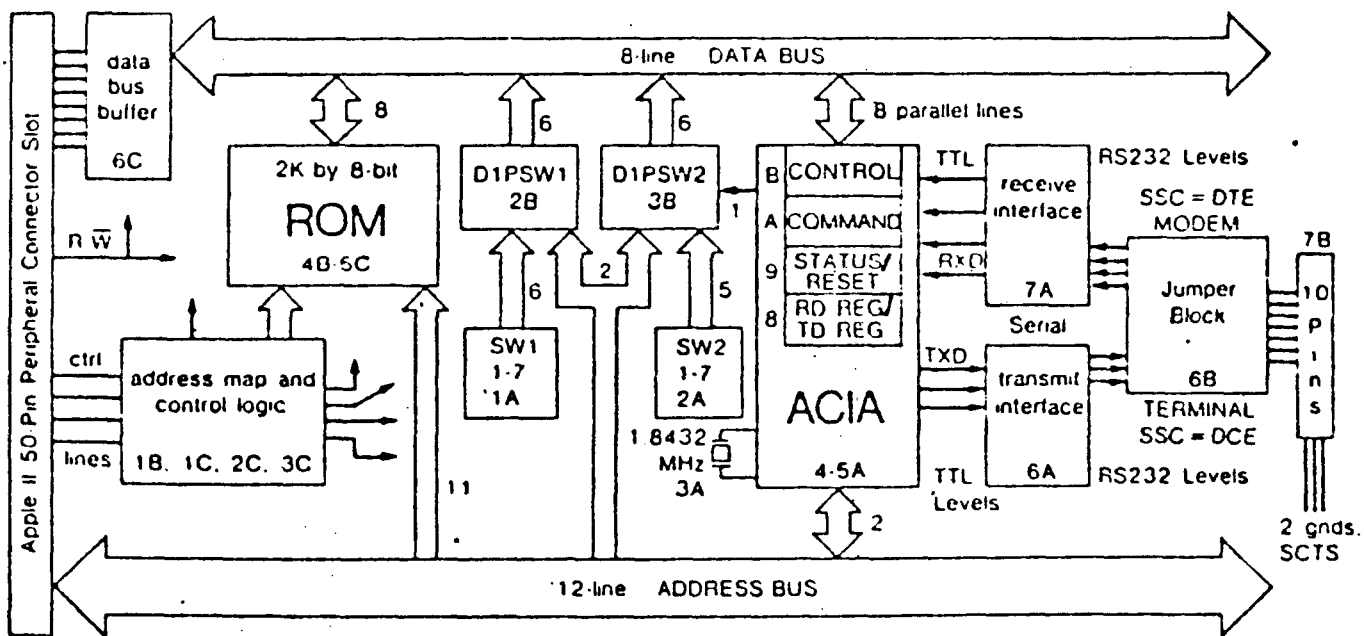


Figure 5.1 Overall Block Diagram of the SSC

TERCERA PARTE

APÉNDICE I

APENDICE 1: MANUAL DE USO DEL APPLE II TRABAJANDO COMO TERMINAL

Conceptos

Este manual de uso está conformado a partir de una estructura software, más que hardware, pero ésta última también se ha de tener en cuenta.

Lo primero que se ha de verificar es si los Switches están debidamente colocados, esto es para tener todos los APPLES conectados las mismas características de trabajo:

- velocidad de 1200 baudios
- 8 bits de longitud de carácter
- sin paridad
- 1 bit de stop
- Line Feed de salida deshabilitado
- 40 caracteres por línea y video habilitado

Para ello en la tarjeta SUPER SERIAL CARD, los switches han de estar de la siguiente manera:

```
SW1- ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 !  
-----  
! OFF! ON! ON! ON! ON! ON! ON!  
-----  
  
SW2- ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 !  
-----  
! ON! ON! ---! ON! OFF! OFF! OFF!  
-----
```

Y en la tarjeta SERIAL INTERFACE CARD, los switches estarán así:

```
! 1 ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 !  
-----  
! OFF! OFF! ON! OFF! ON! ON! ON!  
-----
```

Después de verificado lo anterior, se ha de comprobar que

sólo una tarjeta MASTER esté colocada y que lo esté en modo TERMINAL y que la del alumnado lo estén en modo MODEM .

Una vez verificado lo anterior se procede al encendido o puesta en funcionamiento del circuito protocolario y luego a conectar los APPLEs .

El paso siguiente estriba en acceder al modo terminal que trabaja con ficheros . Para ello se ha de contar en el disco con el programa BASIC SI90 o SI90A (si se quieren ficheros en hexadecimal o no) y el correspondiente programa en ensamblador llamado " CALMA90 " .

Ahora sólo nos queda correr el programa BASIC correspondiente , por ejemplo : " RUN SI90 " , y ya él se encarga de cargar el CALMA90 y pasar a trabajar en modo terminal.

Antes que nada se ha de señalar que el protocolo hardware está preparado para que si un MASTER y un alumno estén conectados a él , proceda a comunicarlos . Pero si no existe alguno de los dos , desconecte al que lo intenta acceder . Este protocolo se vió ya en el capítulo dedicado a hardware.

Uso de comandos

Supongamos que ya estamos trabajando en modo terminal y comunicándonos con otro APPLE II . Luego esto lleva consigo que podamos usar los distintos comandos disponibles :

- a) comandos propios de la tarjeta
 - b) comandos desarrollados por nuestro software particular
- a) Comandos propios de la tarjeta.-

Para el acceso a tales comandos se ha de poner:

CTRL-A <RETURN>

y en la pantalla saldrá :

ENTRE EL COMANDO :

Entonces podemos poner uno de los siguientes comandos seguidos por <RETURN>.

- T entra en modo terminal
- B transmite una señal de BREAK durante 233 milisegundos .
- E <E/D> habilita/deshabilita ECO. El default está puesto a E D (full-duplex) ; se pone E E para half-duplex .
- S <E/D> habilitación/deshabilitación de caracteres especiales . El default es "S E" ; permite/dificulta la generación de letras minúsculas y caracteres especiales (ver tabla A-1).
- <n> T Determina el tratamiento de los caracteres de llegada , esto es :
 0. Todos los caracteres de llegada pasan de minúsculas a mayúsculas antes de pasar a un programa BASIC o a pantalla.
 1. Los caracteres minúsculas quedan sin cambiar.
 2. Displaya los caracteres minúsculas como mayúsculas inversos
 3. Pasa los caracteres minúsculas a programas sin cambiarlos , pero se muestran en pantalla

- los caracteres minúsculas como mayúsculas y los mayúsculas como mayúsculas inversos.
- X <E/D> habilitación/deshabilitación de reconocimiento XOFF. El default es X E ; en modo terminal X E, la SSC detecta <CTRL-R> y <CTRL-T> (control remoto OFF y ON respectivamente), pero no <CTRL-S>.

b) Comandos desarrollados .-

- CTRL-C Imprime el CATALOGO de ficheros que hay en disco.
- CTRL-B " FICHERO A LEER " ; coge un fichero del disco y lo envia a través de la tarjeta.
- CTRL-K " FICHERO A CREAR " ; coge los datos venidos desde el otro terminal por la tarjeta y los almacena en el BUFFER hasta que llegue el final del fichero , y luego lo guarda en disco.
- CTRL-V " FICHERO A VERIFICAR " ; coge un fichero de disco y lo imprime .
- CTRL-I Simplemente imprime lo que le está llegando.
- CTRL-J Sólo saca por pantalla lo que le está llegando, luego es útil cuando se utilizan ficheros hexadecimales.
- CTRL-Z BREAK durante un cierto periodo de tiempo .
- ESCAPE Retorna a BASIC , eliminando el programa y las variables que allí residen .
- CTRL-S Bucle de espera hasta que llegue el CTRL-Q.
- CTRL-Q Ya definido en el párrafo anterior.

APENDICE II

APENDICE 2 : LENGUAJE ENSAMBLADOR DEL 6502 . GRUPO DE INSTRUCCIONES.

Registros CPU y flags de estado.-

El microprocesador 6502 tiene un Acumulador, un registro de estado (P), dos registros indices ("X" e "Y"), un stack pointer y un contador de programa . Todos los registros son de 8 bits (7 - 0), excepto el contador de programa que es de 16 bits (15 - 0).

El registro de estado del 6502 contiene seis flags de estado y un bit de control de interrupción . Los seis flags de estado son :

- flag de carry (C)
- flag de cero (Z)
- flag de overflow (V)
- flag de signo (S)
- flag de modo decimal (D)
- flag de break (B)

Las posiciones asignadas dentro del registro de estado son :

7	6	5	4	3	2	1	0							
!-----!-----!-----!-----!-----!-----!-----!-----!														
!	S	!	V	!	B	!	D	!	I	!	Z	!	C	!

Las localizaciones de memoria comprendidas entre 0100 hasta la 01FF están permanentemente asignadas al Stack.

El flag de carry sostiene el acarreo del bit más significativo en cualquier operación aritmética . La única característica no usual de este flag es que tiene significado invertido en operaciones de sustracción .

El flag de cero y el de signo son flag estándar.

Cuando se pone el estado de modo Decimal causa que las instrucciones de Suma con Carry y Sustracción con carry ejecuten operaciones BCD .

El estado de Break pertenece a interrupciones software . Cuando se ejecuta una interrupción software (instrucción Break) , la lógica de la CPU del 6502 pone el flag de estado de Break.

El estado de overflow es un típico overflow , excepto que puede usarse como una entrada de control en el microprocesador 6502 .

Modos de direccionamiento de memoria en el 6502.-

- Direccionamiento inmediato de memoria.-

En esta forma de direccionamiento , uno de los operandos tiene presente en el byte inmediatamente siguiente el primer byte del código objeto . Un operando inmediato se especifica poniendo delante del operando el símbolo #. Por ejemplo ,

AND #\$08

requiere al ensamblador que genere la instrucción que haga la AND lógica de \$08 con el contenido del acumulador.

- Direccionamiento directo de memoria.-

Esta forma de direccionamiento usa el segundo , o segundo y tercer bytes de la instrucción para identificar la dirección del operando en memoria . La versión de página cero se especifica cuando la expresión usada como operando en la instrucción se reduce a un valor comprendido entre \$00 y \$FF . Por ejemplo ,

AND \$30

requiere que el ensamblador genere una instrucción AND la cual haga el AND lógico de la localización \$0030 con el contenido del acumulador .

La versión absoluta (non-zero-page) es similar excepto que la dirección del operando ocupa dos bytes . Por ejemplo:

AND \$31F6

requiere que el ensamblador genere una instrucción AND que haga el AND lógico del valor en la posición de memoria \$31F6 con el contenido del Acumulador.

- Direccionamiento implícito o inherente.-

Este modo significa que direcciones para ejecutar la instrucción . Ejemplos típicos son CLC (borrado de carry) y TAX (transferir el registro A al registro X).

- Direccionamiento Acumulador.-

Este modo significa que la instrucción opera con los datos que están en el acumulador . En el microprocesador 6502, las instrucciones con acumulador son sólo las notaciones: ASL (cambio lógico a la izquierda), ROL (cambio lógico a la izquierda a través del carry), LSR (cambio lógico a la derecha), y ROR (cambio lógico a la derecha a través del carry).

- Direccionamiento indirecto pre-indexado.-

En este modo el segundo byte de la instrucción se suma al contenido del registro índice X para acceder al lugar de memoria en los 256 primeros bytes , donde se encontrará la dirección indirecta. Se usa suma de reiniciación cíclica , que significa que cualquier carry formado en una dirección

suma será descargado. Por ejemplo

AND (\$20,X)

requiere que el ensamblador genere la instrucción que hace el AND lógico del contenido del Acumulador con el contenido del byte direccionado por el lugar de memoria de la página cero (zero-page) dado por la suma de \$20 y el contenido de registro índice X . El uso de paréntesis en el campo de dirección indica que es indirecto o "contenido de".

El direccionamiento indirecto se almacena con su bit más significativo primero (en las direcciones más bajas) . También hay que decir que las direcciones ocupan dos bytes de memoria.

Una nota importante es que para este tipo de direccionamiento sólo se puede usar el registro X .

- Direccionamiento indirecto post-indexado.-

En este modo de direccionamiento el segundo byte de la instrucción contiene una dirección en los primeros 256 bytes de memoria . Esa dirección y la próxima localización contienen una dirección que se suma al contenido del registro Y para obtener la dirección efectiva .

Las diferencias con el direccionamiento pre-indexado son:

- En el direccionamiento indirecto pre-indexado la indexación se ejecuta antes de la indirección , mientras que en que en el post-indexado se hace primero la indexación y luego la indirección .

- El anterior usa el registro X , mientras que éste usa

el registro Y.

- El direccionamiento indirecto pre-indexado es útil para escoger una de un grupo de direcciones a usar , mientras que el post-indexado es útil para acceder a elementos en un array o tabla para la cual la dirección base ha sido obtenida indirectamente .

Un ejemplo de direccionamiento indirecto post-indexado es,

AND (#20),Y

donde se requiere que el ensamblador genere la instrucción que el AND lógico del Acumulador con el contenido del byte direccionado por la suma del registro índice Y a la dirección de memoria #0020 . Téngase en cuenta que sólo está entre paréntesis el #20 ya que sólo esa parte de la dirección se usa indirectamente .

- Direccionamiento indexado.-

Esta forma de direccionamiento usa el segundo o segundo y tercer bytes de la instrucción para especificar la dirección base . Esta dirección base se suma entonces al contenido del registro índice X o Y para obtener la dirección efectiva . X e Y no son intercambiables ya que ninguna instrucción tienen ambas formas de simple indexado con ambos X e Y . De hecho las únicas instrucciones que permiten indexado de página cero con Y son LDX (cargar registro X) y STX (almacenar registro X) . Un ejemplo típico de direccionamiento indexado de página cero es,

AND #20,X

donde se requiere que el ensamblador genere la instrucción

que haga el AND lógico del contenido del Acumulador con el contenido del byte de la dirección dada por la suma de \$20 y el contenido del registro índice X . Esta es una instrucción de dos bytes porque la dirección cae dentro de los primeros 256 bytes de memoria .

Un ejemplo típico de direccionamiento indexado absoluto es,

AND \$31FE,Y

donde se requiere al ensamblador que genere la instrucción que haga el AND lógico del Acumulador con el contenido del byte de la dirección dada por la suma de \$31FE y el contenido del registro índice Y . Esta es una instrucción de tres bytes ya que la dirección base no está dentro de los primeros 256 bytes de memoria .

Aunque en este modo pueden usarse los registros índices X e Y hay instrucciones que sólo permiten el registro índice X en su modo , tal es el caso de las instrucciones ASL, DEC, INC, LSR, ROL, y ROR .

- Direccionamiento indirecto.-

Este modo de direccionamiento sólo se aplica al JMP (salto a una nueva localización) . En este modo , el segundo y tercer bytes de la instrucción contienen la dirección en la cual se localiza la dirección efectiva . Como se ve el direccionamiento indirecto puede tener cualquier valor y estar localizado en cualquier lugar de memoria . Obviamente este modo puede ser considerado como un caso particular dentro del direccionamiento indirecto tanto post-indexado

como pre-indexado donde el registro índice es cero. Un ejemplo típico es ,

JMP (\$31FE)

donde se requiere que el ensamblador genere una instrucción JMP que cargará el contador de programa del lugar de memoria direccionado por los contenidos de los lugares de memoria \$31FE y \$31FF . Hay que tener en cuenta que las direcciones absolutas tienen 16 bits de longitud y ocupan dos bytes , sin embargo los datos localizados en una dirección son de 8 bits.

- Direccionamiento relativo.-

Las instrucciones de salto con condiciones usan direccionamiento relativo de programa ; un simple byte de desplazamiento se trata como un número binario con signo que se suma al contador de programa , entonces el contador de programa se incrementa para direccionar la próxima dirección secuencial . El rango de desplazamiento va desde +129 a -126 bytes. Un ejemplo típico es,

BCC *+5

donde se requiere que el ensamblador genere un salto si el carry está borrado a un valor dado por el contador de programa más cinco ; si el carry no está borrado entonces sigue con la próxima instrucción que le sigue.

Grupo de instrucciones en el 6502.-

- ADC .- Suma memoria con carry al acumulador .

Esta instrucción tiene ocho métodos de direccionamiento de datos de memoria y permite que el contenido de los datos de memoria y el estado de carry sean sumados al Acumulador .

Los ocho métodos de direccionamiento son :

- Inmediato -ADC dato
- Absoluto (directo) -ADC dirección 16 bits
- De página cero -ADC dirección
- Pre-indexado con registro índice X -ADC (direcc,X)
- Post-indexado con registro índice Y -ADC (direcc),Y
- Indexado de página cero con registro índice X -direcc,X
- Indexado absoluto con registro índice X -ADC direcc16,X
- Indexado absoluto con registro índice Y -ADC direcc16,Y

Por ejemplo si hacemos un ADC con direccionamiento inmediato se hace la suma de los contenidos del próximo byte de memoria y del estado de Carry al Acumulador.

- AND.- Suma memoria con Acumulador.

Esta instrucción hace el AND lógico del contenido de una posición de memoria con el contenido del Acumulador . Esta instrucción ofrece las mismas opciones de direccionamiento que la instrucción ADC .

- ASL.- Rotación a la izquierda en Acumulador o memoria.

Ejecuta un cambio a la izquierda de un bit del contenido del Acumulador o del contenido de seleccionado byte de memoria . El ASL tiene cuatro opciones de direccionamiento de memoria :

- De página cero (directo) -ASL dirección
- Absoluto (directo) -ASL dirección de 16 bits
- Indexado de página cero con registro índice X

ASL -dirección,X

-Indexado de página cero con registro índice Y
ASL -dirección16,Y

- BCC .- Salto si el carry está borrado (C=0)

Esta instrucción salta con direccionamiento relativo , pero sólo salta cuando el estado del Carry es cero ; si el estado del Carry es uno ejecuta la siguiente instrucción. Esta instrucción no afecta a ningún registro ni status excepto el contador de programa .

- BCS .- Salto si el carry está puesto (C=1)

Funciona como la instrucción anterior , sólo que ahora el salto se ejecuta cuando el estado del carry es igual a uno; si el estado es cero ejecuta la siguiente instrucción .

- BEQ .- Salto si es igual a cero (Z=1)

En esta instrucción el salto se produce cuando el estado de cero es igual a uno ; si es cero ejecuta la próxima instrucción .

- BIT .- Test de bit

Esta instrucción ejecuta un AND lógico entre el contenido del Acumulador con el contenido de una posición de memoria seleccionada , pone los flags de acuerdo con ello , pero no altera los contenidos del Acumulador o del byte de memoria . Los únicos modos de direccionamiento permitidos son absoluto (directo) y de página cero (directo).El primer byte del código objeto determina el modo de direccionamiento.

- BMI .- Salto si menos (S=1).

Esta instrucción trabaja como la instrucción BCC excepto

que se ejecuta sólo cuando el estado del signo es uno ; de otra manera se ejecuta la instrucción siguiente.

- BNE .- Salto si no es igual a cero ($Z=0$).

Volvemos a estar en un caso idéntico al de BCC , sólo que ahora el salto se efectúa cuando el estado de cero es cero ; de otra manera se ejecuta la próxima instrucción .

- BPL .- Salto si más ($S=0$)

Se repite un caso idéntico al anterior , sólo que ahora se salta cuando el estado del signo es cero ; de otra manera se ejecuta la siguiente instrucción .

- BRK .-Se fuerza un break (interrupción software).

El contador de programa se incrementa dos posiciones y el estado de Break se pone a uno , entonces el contador de programa y el registro de estado (P) se guardan en el Stack.

La máscara de interrupción se pone a uno . Esto quita la habilidad del servicio de interrupción del 6502, por ejemplo el procesador no responde a una interrupción de un dispositivo externo. Los contenidos del puntero de interrupción se cargan en el contador de programa .

La instrucción Break puede usarse para muchas funciones. Puede proveernos de un punto de rotura para el depurado de programas o puede transferir el control a un importante sistema software tal como el sistema operativo de disco o un monitor .

- BVC .- Salto si overflow está borrado ($V=0$).

Funciona igual que la instrucción BBC , sólo que ahora

salta cuando el estado de Overflow es cero ; si no es así sigue con la próxima instrucción .

- BVS .- Salto si overflow está puesto (V=1).

Estamos en el mismo caso anterior , sólo que ahora se salta cuando el estado de Overflow es uno .

- CLC .- Borrado de Carry.

Simplemente significa borrar el estado de carry . Ningún otro estado o registro se afectado. Se usa en operaciones de suma por sólo tener el 6502 la instrucción ADC , que suma con carry.

- CLD .- Borrado de modo decimal .

Borra el estado de modo decimal . No afecta a ningún otro estado o registro . Se usa para retornar al 6502 al modo binario en el cual las instrucciones ADC y SBC producen resultados binarios más que BCD .

- CLI .- Borrado de la máscara de interrupción (habilita interrupciones).

Borra el bit de máscara de interrupciones en el registro de estado (P). Esta instrucción habilita el servicio de interrupciones del 6502 , por ejemplo el 6502 responde a la línea de control de requisito de interrupción .

- CLV .- Borrado de overflow.

Borra el bit de overflow en el registro de estado. Ningún otro registro o estado se ve afectado. Hay que tener en cuenta que el 6502 no dispone de instrucción para poner overflow.

- CMP .- Compara memoria con acumulador.

Esta instrucción sustrae el contenido de un byte de memoria seleccionado del contenido del acumulador , pone los flags de acuerdo con ello ,pero no altera los contenidos del acumulador o del byte de memoria. Esta instrucción ofrece las mismas opciones de direccionamiento de memoria que la instrucción ADC .

- CPX .- Compara el registro indice X con memoria.

Hace lo mismo que la instrucción anterior , sólo que ahora el byte de memoria se sustrae del registro indice X en vez del acumulador . Sólo se permiten los modos de direccionamiento inmediato, de página cero (directo), y absoluto (directo).

- CPY .- Compara el registro indice Y con memoria.

Igual que la anterior sólo que ahora se hace con el registro indice Y.

- DEC .- Decrementa memoria (por uno).

Esta instrucción decrementa por uno el contenido de una localización de memoria determinada . Aquí se usan cuatro modos de direccionamiento:

-De página cero (directo) -DEC dirección

-Absoluto (directo) -DEC dirección de 16 bits

-Indexado de página cero con registro indice X

- DEC dirección,X

-Absoluto indexado con registro indice X -DEC direcc16,X

- DEX .-Decrementar el registro indice X (por uno)

Esta instrucción decrementa por uno el contenido del

registro índice X . Los estados de cero y signo se ven afectados .

- DEY .- Decrementar el registro índice Y (por uno)

Esta instrucción decrementa por uno el contenido del registro índice Y . Al igual que las dos instrucciones anteriores afecta a los estados de cero y signo .

- EOR .- OR EXCLUSIVO de Ac con memoria.

Hace la OR-EXCLUSIVA del Ac con el contenido de un byte de memoria seleccionado . Ofrece las mismas opciones de direccionamiento que la instrucción ADC . El primer byte del código objeto selecciona el modo de direccionamiento .

EOR se usa para en testeos para cambiar el bit de estado. También si se hace EOR #\$FF se complementa el contenido del acumulador .

- INC .- Incrementa memoria (por uno).

Esta instrucción incrementa por uno el contenido de una posición de memoria seleccionada . La instrucción INC ofrece cuatro opciones de direccionamiento de datos de memoria que son idénticos a los dados por la instrucción DEC .

- INX .- Incrementa el registro índice X (por uno).

Esta instrucción incrementa por uno el contenido del registro índice X . Los estados de cero y signo se ven afectados al igual que en la instrucción ADC .

- INY .- Incrementa el registro índice Y (por uno).

Lo mismo que la anterior sólo que ahora el que se incrementa es el registro índice Y.

- JMP .- Salto en modo absoluto o direccionamiento indirecto.

Esta instrucción es la única que tiene modo de direccionamiento verdaderamente indirecto .

La instrucción JMP puede usar el modo de direccionamiento absoluto (directo). En este caso , el segundo byte de la instrucción se carga en el byte más bajo del contador de programa y el tercer byte de la instrucción se carga en el byte alto del contador de programa.

- JSR .- Salto a subrutina.

Esta instrucción guarda el contador de programa en el stack y transfiere el control a una instrucción especificada. Sólo se permite modo de direccionamiento absoluto (directo).

-LDA .- Carga Acumulador de memoria .

Carga el contenido del byte seleccionado en el Acumulador .Esta instrucción ofrece las mismas opciones de direccionamiento que la instrucción ADC.

-LDX .- Carga registro índice X de memoria.

Carga el contenido del byte de memoria seleccionado en el registro índice X . Los modos de direccionamiento permitidos son :

-Inmediato -LDX dato

-Absoluto (directo) -LDX dirección de 16 bits

-De página cero (directo) -LDX dirección

-Absoluto indexado con Y -LDX direcc16,Y

-Indexado de página cero con Y -LDX direcc,Y

-LDY .-Carga registro índice Y de memoria.

Es el mismo caso anterior sólo que ahora es con el registro índice Y.

- LSR .- Cambio lógico a la derecha de Ac. o memoria.

Realiza una rotación lógica de 1 bit a la derecha del Acumulador o memoria. Los cuatro modos de direccionamiento disponibles son :

-De página cero -LSR dirección

-Absoluto (directo) -LSR dirección de 16 bits

-Indexado página cero con registro índice X -LSR dir,X

-Indexado absoluto con registro índice X -LSR dirección,X

-NOP .- No opera.

Esta es una instrucción de un byte , y no hace nada , excepto incrementar el contador de programa. Puede utilizarse para reemplazar instrucciones , así como para efectuar delays.

- ORA .- OR-LOGICO de memoria con Acumulador.

Realiza el OR-LOGICO del contenido de un lugar de memoria con el contenido del Acumulador. Ofrece los mismos modos de direccionamiento que la instrucción ADC.

- PHA .- Guarda Acumulador en el stack.

Almacena el contenido del Acumulador en el tope de la pila y luego el stack se decrementa en 1 .No afecta a ningún registro o estado.

- PHP .-Guarda el Registro de Estado en el stack.

Lo mismo que la instrucción anterior sólo que ahora es con el registro de estado .

- PLA .- Recupera el contenido del Ac del Stack.

Primero incrementa el Stack Pointer en 1 y luego carga el Acumulador del tope del Stack.

- PLP .- Recupera el contenido del Registro de Estado del Stack.

Incrementa en 1 el Stack Pointer y carga el Registro de Estado del tope de la pila .No afecta a ningún registro ,pero todos los estados pueden cambiar.

- ROL .- Rota el Acumulador o memoria a la izquierda a través del carry.

Ofrece 4 modos de direccionamiento de datos de memoria:

-De página cero (directo) -ROL dirección

-Absoluto (directo) -ROL dirección de 16 bits

-Indexado página cero con registro índice X -ROL dir,X

-Absoluto indexado con registro índice X ROL direcc16,X

- ROR .- Rota Acumulador o memoria a la derecha a través del carry. Ofrece los mismos de direccionamiento que el caso anterior.

- RTI .- Retorno de interrupción .

Recupera el registro de estado (P) y el contador de programa del stack .Se pierden entonces los viejos valores del registro de estado y el contador de programa , y el puntero del stack sube 3 posiciones.

- RTS .- Retorno de subrutina .

Busca un nuevo valor de contador de programa del stack y lo incrementa antes de usarlo para buscar una instrucción. Luego se usa normalmente al final de una instrucción JSR .

- SBC .- Sustraer memoria de Ac con borrow .

Sustraer el contenido de un byte de memoria seleccionado y el complemento del estado de carry del Acumulador . Ofrece los mismos modos de direccionamiento que la instrucción ADC.

- SEC .- Puesta de Carry.

Pone el estado de carry a 1 . Ningún otro registro se ve afectado . Se requiere como parte de una resta ya que la única sustracción posible en el 6502 es SBC que sustraer también el estado de carry complementado

- SED .- Puesta de modo decimal .

Pone el estado de modo decimal a 1 . No afecta a ningún otro registro . Se usa para colocar el procesador 6502 en modo decimal en el cual las instrucciones SBC y ADC producen más bien resultados BCD que binarios.

- SEI .- Pone máscara de interrupción (desabilita interrupciones). Al ponerla el 6502 no responde a la línea de control de interrupciones.

- STA .- Almacena Acumulador en memoria .

Ofrece los mismos de direccionamiento que la instrucción ADC , si bien no se dispone de modo de direccionamiento inmediato.

- STX .- Almacena registro índice X en memoria .

Los modos de direccionamiento son :

-De página cero (directo) -STX dirección

-Absoluto (directo) -STX dirección de 16 bits

-Indexado página cero con registro índice Y -STX dir,Y

- STY .- Almacena registro índice X en memoria .

Estamos en el mismo caso anterior sólo que ahora se trabaja con el registro índice Y.

- TAX .- Mueve de Acumulador a registro índice X.

Pone los estados de signo y cero de acuerdo con ello .

- TAY .- Mueve de Acumulador a registro índice Y.

Análogo al caso anterior sólo que ahora es con Y.

- TSX .- Mueve de Stack Pointer a registro índice X.

Es la única instrucción que te permite acceder al valor dentro del Stack Pointer.

- TXA .- Mueve de registro índice X a Acumulador .

- TXS .- Mueve de registro índice X a Stack Pointer.

Es la única instrucción que permite determinar el valor dentro del Stack.

- TYA .- Mueve de registro índice Y a Acumulador .

DISPOSITIVOS DE E/S (UART)

En este apartado hablaremos de la ACIA 6850 y de la ACIA 6551 que son las que utilizan 2 de las tarjetas de comunicaciones de que disponemos .

- ACIA 6850 (Interface adaptador de comunicaciones asíncronas).

Esta ACIA es una UART específicamente diseñada para trabajar con microcomputadores basados en el 6800 y 6502 . Ocupa dos direcciones de memoria y contiene dos registros de sólo lectura (datos recibidos y estado) y dos registros de sólo escritura (datos transmitidos y control). Las tablas

adjuntas (A2.1 y A2.2) describen los contenidos de estos registros .

* Características especiales de la ACIA 6850 :

a) Los ciclos de lectura y escritura direccionan físicamente distintos registros .No se pueden usar los registros ACIA como direcciones para instrucciones como incremento , decremento , o cambios que tengan ambos ciclos de lectura y escritura .

b) El registro de control ACIA no puede leerse por la CPU . Si el programa necesita su valor se ha de sacar de salvar una copia del registro de control de memoria .

c) La ACIA no tiene reset de entrada . Puede resetearse colocando unos en los bits 0 y 1 del registro de control . Este procedimiento llamado MASTER RESET es necesario antes de usar ACIA .

d) Las señales RS-232 se activan a nivel bajo , luego si por ejemplo RTS (Request to Send) no se usa se pone a nivel alto.

c) La ACIA requiere un reloj externo . Típicamente se alimenta a 1760 hz y se usa en el modo %16 (bits 0 y 1 de registro de control a 1 y 0 respectivamente).Se usa el clock para centrar la recepción en orden a no permitir falsos bits de start por ruidos en las líneas .

d) El flag de Datos Preparados es el bit 0 del registro de estado . El flag de periférico preparado es el bit 1 del registro de estado .

- ACIA 6551 (Interface adaptador de comunicaciones asincrónicas) .

Es una variación de la 6850 que también puede usarse en sistemas basados en el 6800 y 6502 .Tiene la mayoría de las características de la ACIA 6850 y también tiene un chip para generar el rango de velocidad que puede programarse en 15 formas distintas derivadas de un cristal estándar de 1.8432 Mhz. Así la ACIA 6551 puede proveer virtualmente cualquier común rango de velocidad sin tener un "timer" externo o un generador de reloj.

El dispositivo tiene 4 registros internos direccionables. Su funcionamiento se controla por dos registros :

a) El registro de control (fig A2.4) controla el generador de reloj , la longitud de la palabra , el número de bits de stop y la fuente receptora de reloj .

b) El registro de comando (fig A2.5) controla el chequeo y generación de paridad , habilitado de interrupciones y las señales de comienzo de comunicación RS-232 . Puede resetarse la ACIA 6551 con sólo escribir cualquier dato dentro de las direcciones de registro de estado .

Table 3-5. 6502 Instruction Object Codes in Numerical Order

Object Code	Instruction		Object Code	Instruction	
00	BRK		68	PLA	
01 pp	ORA	(addr,X)	69 pp	ADC	data
05 pp	ORA	addr	6A	ROR	A
06 pp	ASL	addr	6C ppqq	JMP	(label)
08	PHP		6D ppqq	ADC	addr16
09 pp	ORA	data	6E ppqq	ROR	addr16
0A	ASL	A	70 pp	BVS	disp
0D ppqq	ORA	addr16	71 pp	ADC	(addr),Y
0E ppqq	ASL	addr16	75 pp	ADC	addr,X
10 pp	BPL	disp	76 pp	ROR	addr,X
11 pp	ORA	(addr),Y	78	SEI	
15 pp	ORA	addr,X	79 ppqq	ADC	addr16,Y
16 pp	ASL	addr,X	7D ppqq	ADC	addr16,X
18	CLC		7E ppqq	ROR	addr16,X
19 ppqq	ORA	addr16,Y	81 pp	STA	(addr,X)
1D ppqq	ORA	addr16,X	84 pp	STY	addr
1E ppqq	ASL	addr16,X	85 pp	STA	addr
20 ppqq	JSR	label	86 pp	STX	addr
21 pp	AND	(addr,X)	88	DEY	
24 pp	BIT	addr	8A	TXA	
25 pp	AND	addr	8C ppqq	STY	addr16
26 pp	ROL	addr	8D ppqq	STA	addr16
28	PLP		8E ppqq	STX	addr16
29 pp	AND	data	90 pp	BCC	disp
2A	ROL	A	91 pp	STA	(addr),Y
2C ppqq	BIT	addr16	94 pp	STY	addr,X
2D ppqq	AND	addr16	95 pp	STA	addr,X
2E ppqq	ROL	addr16	96 pp	STX	addr,Y
30 pp	BMI	disp	98	TYA	
31 pp	AND	(addr),Y	99 ppqq	STA	addr16,Y
35 pp	AND	addr,X	9A	TXS	
36 pp	ROL	addr,X	9D ppqq	STA	addr16,X
38	SEC		A0 pp	LDY	data
39 ppqq	AND	addr16,Y	A1 pp	LDA	(addr,X)
3D ppqq	AND	addr16,X	A2 pp	LDX	data
3E ppqq	ROL	addr16,X	A4 pp	LDY	addr
40	RTI		A5 pp	LDA	addr
41 pp	EOR	(addr,X)	A6 pp	LDX	addr
45 pp	EOR	addr	A8	TAY	
46 pp	LSR	addr	A9 pp	LDA	data
48	PHA		AA	TAX	
49 pp	EOR	data	AC ppqq	LDY	addr16
4A	LSR	A	AD ppqq	LDA	addr16
4C ppqq	JMP	label	AE ppqq	LDX	addr16
4D ppqq	EOR	addr16	80 pp	BCS	disp
4E ppqq	LSR	addr16	81 pp	LDA	(addr),Y
50 pp	BVC	disp	84 pp	LDY	addr,X
51 pp	EOR	(addr),Y	85 pp	LDA	addr,X
55 pp	EOR	addr,X	86 pp	LDX	addr,Y
56 pp	LSR	addr,X	88	CLV	
58	CLI		B9 ppqq	LDA	addr16,Y
59 ppqq	EOR	addr16,Y	8A	TSX	
5D ppqq	EOR	addr16,X	8C ppqq	LDY	addr16,X
5E ppqq	LSR	addr16,X	8D ppqq	LDA	addr16,X
60	RTS		8E ppqq	LDX	addr16,Y
61 pp	ADC	(addr,X)	C0 pp	CPY	data
65 pp	ADC	addr	C1 pp	CMP	(addr,X)
66 pp	ROR	addr	C4 pp	CPY	addr

Table 3-5. 6502 Instruction Object Codes in Numerical Order (Continued)

Object Code	Instruction		Object Code	Instruction	
C5 pp	CMP	addr	E4 pp	CPX	addr
C6 pp	DEC	addr	E5 pp	SBC	addr
C8	INY		E6 pp	INC	addr
C9 pp	CMP	data	E8	INX	
CA	DEX		E9 pp	SBC	data
CC ppqq	CPY	addr16	EA	NOP	
CD ppqq	CMP	addr16	EC ppqq	CPX	addr16
CE ppqq	DEC	addr16	ED ppqq	SBC	addr16
D0 pp	BNE	disp	EE ppqq	INC	addr16
D1 pp	CMP	(addr),Y	F0 pp	BEQ	disp
D5 pp	CMP	addr,X	F1 pp	SBC	(addr),Y
D6 pp	DEC	addr,X	F5 pp	SBC	addr,X
D8	CLD		F6 pp	INC	addr,X
D9 ppqq	CMP	addr16,Y	F8	SED	
DD ppqq	CMP	addr16,X	F9 ppqq	SBC	addr16,Y
DE ppqq	DEC	addr16,X	FD ppqq	SBC	addr16,X
E0 pp	CPX	data	FE ppqq	INC	addr16,X
E1 pp	SBC	(addr,X)			

Figure 11-36. Block Diagram of the 6850 ACIA

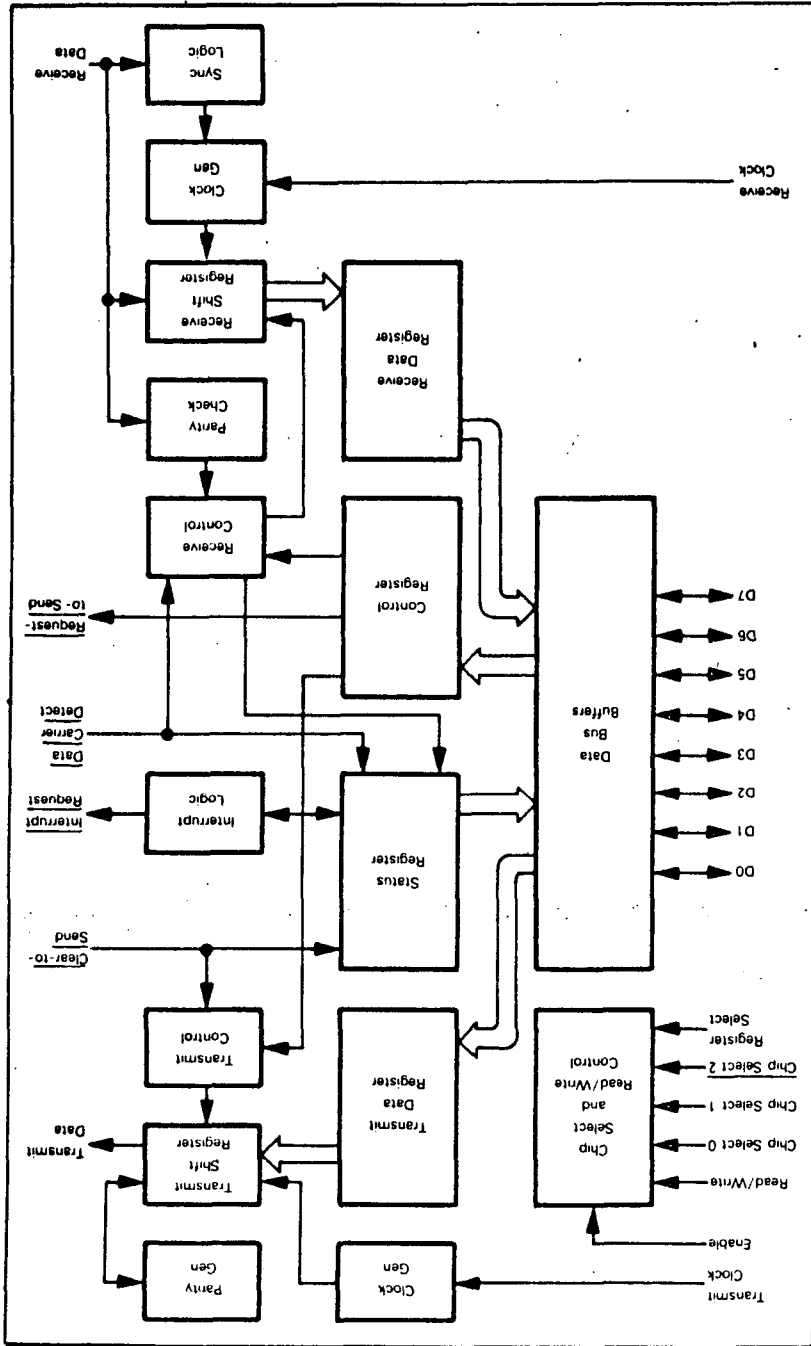


Table A2.1 Definition of 6850 ACIA Register Contents

Data Bus Line Number	Buffer Address			
	RS- \bar{R} /W Transmit Data Register	RS-R/W Receive Data Register	\bar{R} S- \bar{R} /W Control Register	\bar{R} S-R/W Status Register
	(Write Only)	(Read Only)	(Write Only)	(Read Only)
0	Data Bit 0*	Data Bit 0	Counter Divide Select 1 (CR0)	Receive Data Register Full (RDRF)
1	Data Bit 1	Data Bit 1	Counter Divide Select 2 (CR1)	Transmit Data Register Empty (TDRE)
2	Data Bit 2	Data Bit 2	Word Select 1 (CR2)	Data Carrier Detect (DCD)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	Clear-to-Send (CTS)
4	Data Bit 4	Data Bit 4	Word Select 3 (CR4)	Framing Error (FE)
5	Data Bit 5	Data Bit 5	Transmit Control 1 (CR5)	Receiver Overrun (OVRN)
6	Data Bit 6	Data Bit 6	Transmit Control 2 (CR6)	Parity Error (PE)
7	Data Bit 7***	Data Bit 7**	Receive Interrupt Enable (CR7)	Interrupt Request (IRO)

* Leading bit = LSB = Bit 0
 ** Data bit will be zero in 7-bit plus parity modes
 *** Data bit is "don't care" in 7-bit plus parity modes

Table A2.2 Meaning of the 6850 ACIA Control Register Bits

CR6	CR5	Function	
0	0	$\bar{R}TS$ = low. Transmitting Interrupt Disabled	
0	1	$\bar{R}TS$ = low. Transmitting Interrupt Enabled	
1	0	$\bar{R}TS$ = high. Transmitting Interrupt Disabled	
1	1	$\bar{R}TS$ = low. Transmits a Break level on the Transmit Data Output. Transmitting Interrupt Disabled	
CR4	CR3	CR2	Function
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bit
0	1	1	7 Bits + Odd Parity + 1 Stop Bit
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bit
1	1	0	8 Bits + Even Parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit
CR1	CR0	Function	
0	0	+ 1	
0	1	+ 16	
1	0	+ 64	
1	1	Master Reset	

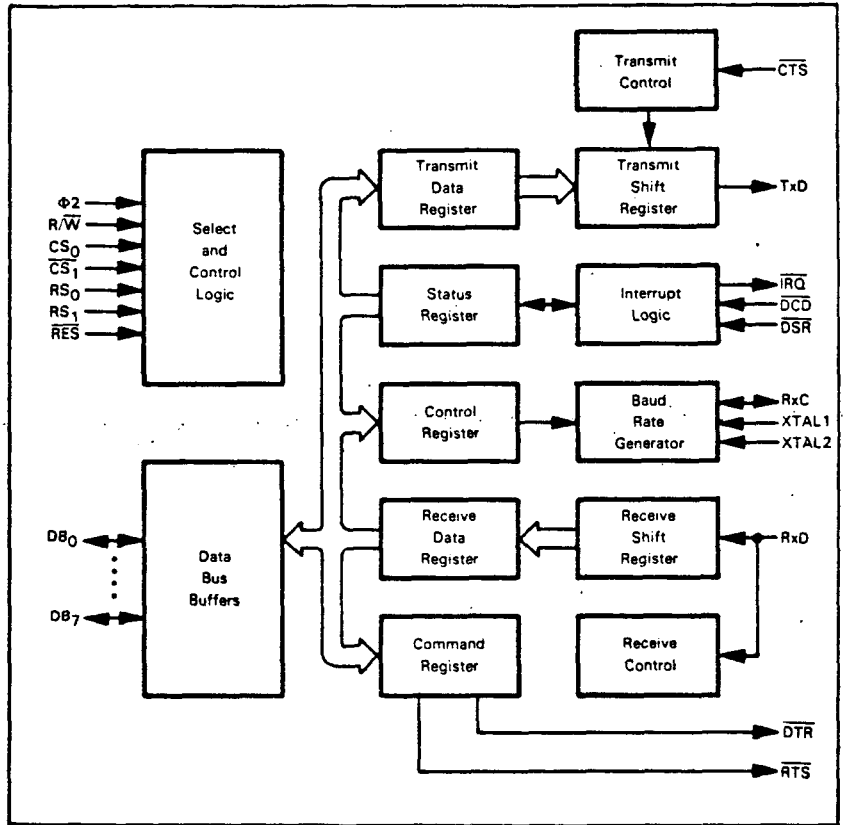


Figure 11-37. Block Diagram of the 6551 ACIA

Table 11-18. Addressing 6551 ACIA Internal Registers

RS ₁	RS ₀	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	0	Control Register	

The table shows that only the Command and Control registers are read/writes. The Programmed Reset operation does not cause any data transfer, but is used to clear the SY6551 registers. The Programmed Reset is slightly different from the Hardware Reset (RES) and these differences are described in the individual register definitions.

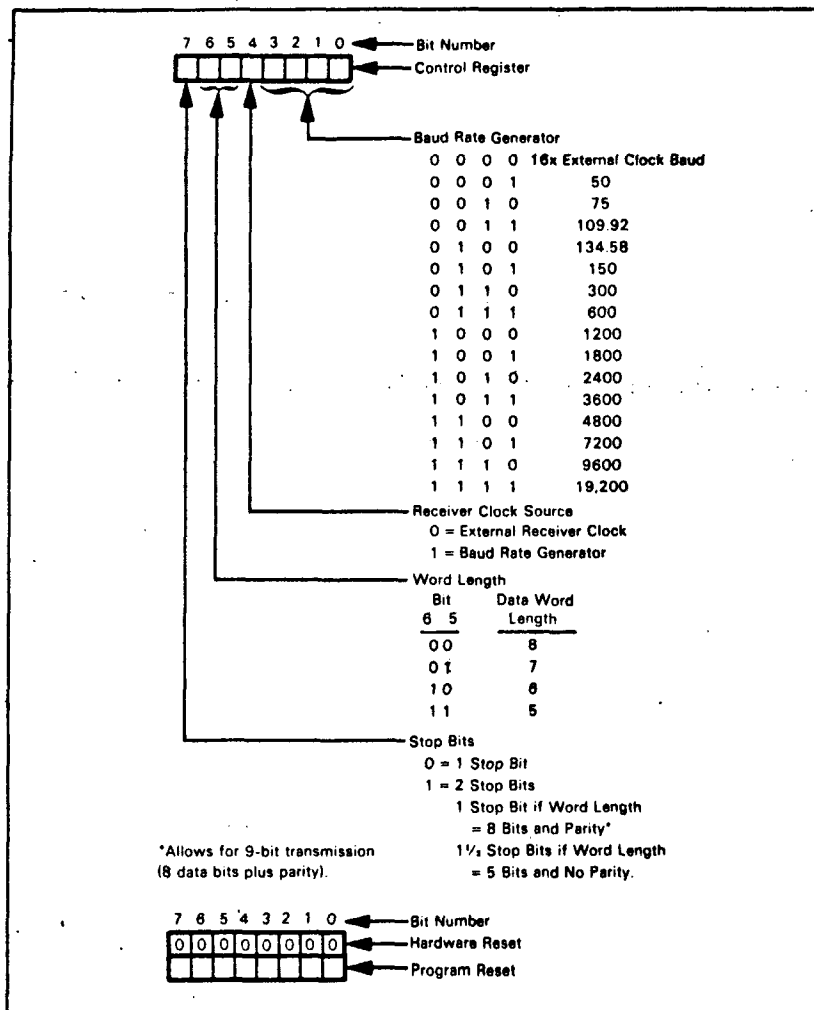


Figure A2.4 Definition of 6551 ACIA Control Register Contents

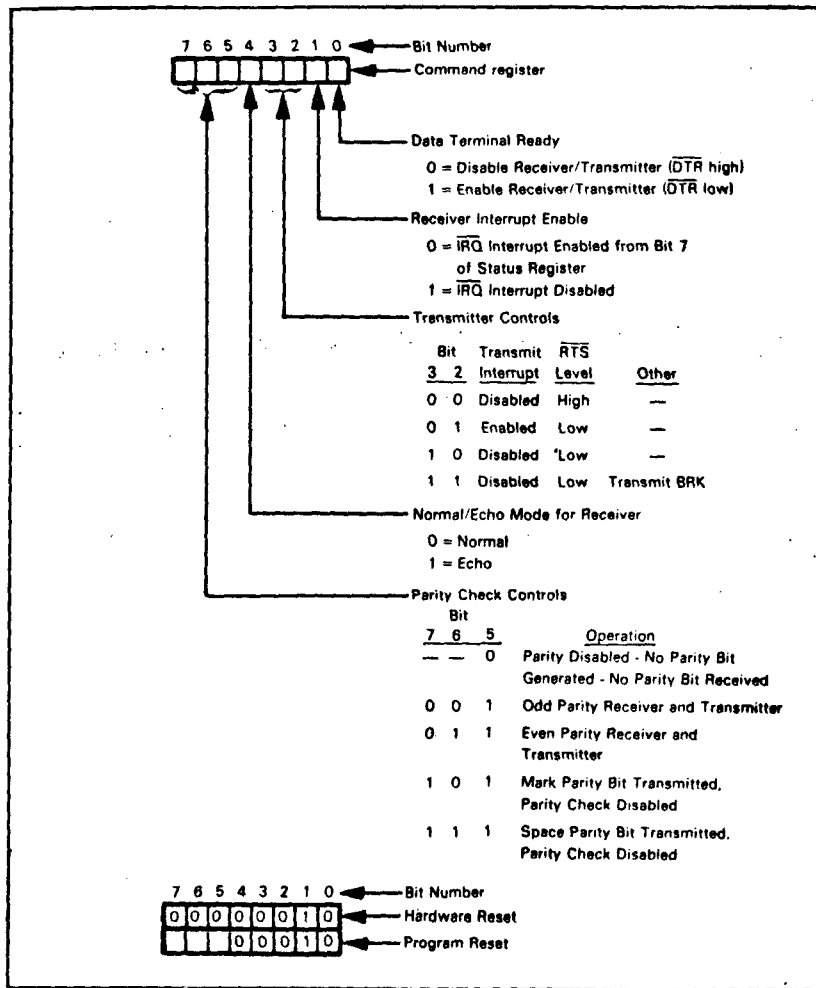


Figure 2.5 Definition of 6551 ACIA Command Register Contents

APENDICE III

APENDICE 3 : EL MODELO DE CONEXION RS-232-C

En este apéndice se pretende dar una idea de lo que es el RS-232-C , pero a nivel más práctico que teórico .

INTRODUCCION

En 1969 , la EIA (Asociación de Industrias Electrónicas), los laboratorios Bell y los fabricantes de equipos de comunicaciones formularon cooperativamente , y emitieron , el EIA RS-232, que casi inmediatamente experimentó revisiones menores , convirtiéndose en la RS-232-C . Un modelo similar fué emitido por la Organización Internacional de Modelos , Comité Consultivo Internacional sobre Telefonía y Telegrafía (CCITT). Se debe recalcar que la RS-232-C fué desarrollada para un único propósito , establecida formalmente por su título :

Conexión entre un Equipo Terminal de Datos y un Equipo de Comunicación de Datos empleando un Intercambio de Datos binarios en Serie.

Cada palabra del título es significativa: describe la conexión entre un terminal (Equipo Terminal de Datos, o DTE) a un modem (Equipo de Comunicación de Datos, o DCE) para la transmisión de datos en serie . El documento consta de cuatro partes .

* **Características de la señal eléctrica .-** Esto describe el "lado" eléctrico que presenta la conexión y que requiere del mundo externo . Se definen aquí los niveles de voltaje que representarán al 0 y 1 lógicos .

*** Características mecánicas de la conexión (conectores).-**

Esta sección establece que la conexión debe consistir en una clavija y un receptáculo , y que el receptáculo estará en el DCE . Se especifica la asignación de números a las patillas, pero debería notarse que el propio conector no ha sido especificado . El conector familiar en forma de "D", el DB-25, que ahora es casi sinónimo de conexiones en serie , se deriva de otro organismo de modelos , la Organización Internacional de Modelos , o ISO . Los detalles de la conexión se muestran en la fig A3.1 .

*** Descripción funcional de circuitos de intercambio.-**

Esta sección define y da nombres a las funciones de las señales eléctricas a usar . Así por ejemplo, se asignan los DATOS TRANSMITIDOS a la patilla 2 . Hay 21 definiciones similares , pero sólo unas pocas son relevantes a los microordenadores.

*** Conexiones modelo para configuraciones seleccionadas de sistemas de comunicación .-** Son ejemplo de tipos comunes de conexión entre modem y terminal .

BASES DE LA CONEXION

En su forma más simple , la conexión RS-232-C consta solamente de dos cables (uno para llevar datos y , además , un "circuito común". A menudo este circuito común se denomina erróneamente " circuito de tierra " , pero no tiene nada que ver con la tierra . Es solamente la referencia de voltaje absoluta para todos los circuitos de la conexión , el punto

desde el cual se miden todos los voltajes del circuito . La única cosa que hay que recordar es que esta conexión entre las patillas 7 debe hacerse en cada conector RS-232-C, no importa cuán simple o complejo sea .

Como no es usual que los terminales y modem no sean dispositivos de un sólo sentido sino que cualquiera puede hacer la función opuesta es por lo que ambos dispositivos han de ser capaces de transmitir y recibir datos , de modo que ahora tenemos una línea 3 para recibir datos , claro está tomando como referencia el DTE . Entonces es tiempo de establecer la diferencia entre los DTE y los DCE:

- Los DTE transmiten desde la patilla 2 y reciben por la patilla 3.

- Los DCE transmiten desde la patilla 3 y reciben por la patilla 2 .

Hay que tener en cuenta que aunque los nombres de las patillas para los dispositivos DTE y DCE puedan ser idénticos, tienen funciones complementarias.

ACOPLAMIENTO

El acoplamiento o handshaking es el modo en que se regula y controla el flujo de datos a través de la conexión . Se distinguen dos modos de acoplamiento :

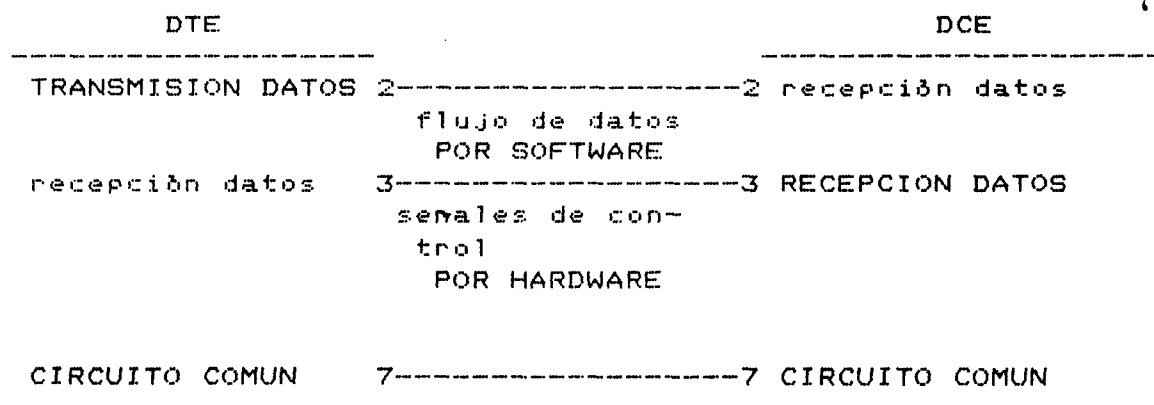
- por software
- por hardware

El acoplamiento por software existe cuando un dispositivo controla a otro por medio del contenido de los datos . Por ejemplo una manera de controlar a una impresora es hacer que

el ordenador le envíe sus caracteres línea a línea . Al final de cada línea el ordenador dice a la impresora : "este es el final de línea" y espera que la impresora le envíe un carácter cómo que está lista para aceptar otra línea.

En contraste , con el acoplamiento por hardware trabajamos a un nivel más fundamental , donde la impresora puede obligar al ordenador a pararse en el envío de los caracteres cambiando simplemente el voltaje de un cable . Resuelve los problemas de la comunicación al nivel más básico mecánico (con cables y voltajes). La desventaja de este tipo de acoplamiento es que sólo puede usarse cuando los dispositivos pueden conectarse físicamente a través de un cable .

El diagrama siguiente muestra una distinción importante entre los tipos de señales de conexión : señales de datos y señales de control .



COMPATIBILIDAD RS-232-C

Mientras algunas de las señales de la conexión se usan casi universalmente en los micros , otras se aplican libremente sin tener en cuenta ninguna práctica establecida .

Lo que vamos a plantear a continuación es un sumario exacto de lo que se puede esperar de una conexión RS-232-C.

Áreas de Compatibilidad RS-232-C .-

" Las necesidades eléctricas de la conexión (voltajes, etc), por necesidad, se observan estrictamente ". Si un dispositivo dice ser compatible RS-232-C significa que puedes conectarlo a otro dispositivo igualmente compatible, sin dañar o fundir ninguno. Esto significa que se compenetrarán eléctricamente lo suficientemente bien para el intercambio de datos.

" Los niveles de voltaje asignados para el cero y el uno corresponderán a aquellos descritos en el modelo ". Las definiciones de lógica y de voltaje son diferentes para las señales de datos y las señales de control, por ello el uso de 0 (cero) y 1 (uno) puede ser causa de confusiones o malentendidos.

" Las clavijas y enchufes (conectores macho y hembra) han de encajar perfectamente " .

Incluso cuando las tareas de cada patilla están dictaminadas para el conector normalizado, los propios conectores no. El conector en forma de "D", DB-25, de cualquier modo, se ha convertido en una norma multiuso.

ENTRADA / SALIDA DEL PROCESADOR

Normalmente hoy se lleva a cabo el proceso completo de E/S en serie mediante un único circuito integrado, conocido genéricamente como UART. Antes de que apareciera la UART, el proceso de entrada / salida se lleva a cabo a su nivel más

fundamental ; esto es , se controla directamente por el mismo procesador del ordenador .

La UART se conoce como un dispositivo de servicio puesto que releva al procesador del tedioso trabajo de las E/S serie. En contraste con una E/S controlada por procesador , un programa con UART no se ocupa de detalles . El programador trata a la UART como un buzón dentro del cual se echan los caracteres de salida para ser despachados , o del que se sacan los caracteres de entrada . Todo lo que sabe el programador es que la conversión serie/paralelo , la sincronización y la lógica asociada las hace la UART.

LA UART Y LA CONEXION RS-232-C

El proceso de lectura de los estados lógicos de las entradas y cambio de los estados lógicos de las salidas es el mecanismo mediante el cual la UART interactúa con la conexión RS-232-C. Se ha de señalar que no todos los dispositivos periféricos en serie contienen realmente UART. La fig A3.2 nos vale por mil palabras , demostrando como la UART se une a la conexión RS-232-C.

FUNCIONES DE LAS PATILLAS

Se hace aquí un sumario de las patillas más importantes de la conexión RS-232-C . Estas definiciones se hacen desde la perspectiva del DTE.

- Patilla 1 . **MASA DE PROTECCION.** También llamada "masa de la carcasa" de forma no oficial . Si un equipo no tiene conector de masa en su enchufe a la corriente , debería conectarse a través de la patilla 1 a uno que la tuviese . Se

usa para prevenir descargas eléctricas en el caso de que falle la corriente . Esto se puede lograr usando un cable que conecte las patillas 1 de ambos dispositivos.

La función de esta patilla se confunde frecuentemente con la de la patilla 7, conocida como "retorno común" . Se supone que la conexión de masa del enchufe a la red la llevará de vuelta a tierra , de ahí el término tierra . Sin embargo, en la práctica , cuando se conectan dos equipos a diferentes líneas de eléctricas (en un gran edificio por ejemplo), sus caminos a tierra pueden ser eléctricamente diferentes . Esto ocasiona que sus carcadas no sean eléctricamente iguales , y puede impedir las comunicaciones , pero se puede eliminar de forma efectiva uniendo ambas carcadas a través de las patillas 1.

Pero de hecho bajo circunstancias ordinarias , es más probable que se causen problemas al conectar las patillas 1 . Los bucles de tierra hacen que el equipo se comporte erráticamente , por ello si los periféricos se comportan como si estuviesen poseídos , y se comprueba que tienen conectadas las patillas 1 , he ahí el problema . La solución , pues, es cortar la conexión entre las patillas 1.

En cualquier caso en las normas RS-232-C , la patilla 1 es opcional.

Patilla 2. **TRANSMISION DE DATOS (TXD)**. Transmite datos desde el DTE al DCE .

Patilla 3. **RECEPCION DE DATOS (RXD)**. Transmite datos del DCE al DTE.

Patilla 4. **PETICION DE ENVIO** . Salida de propósito general . Sus usos varían ampliamente.

Patilla 5. **BORRADO PARA ENVIAR** . Entrada de propósito general . Sus usos varían ampliamente.

Patilla 6. **CONJUNTO DE DATOS LISTO (DSR)** . Entrada de propósito general para comunicar al DTE que el DCE está encendido y listo para funcionar.

Patilla 7. **CIRCUITO COMUN** . Punto de referencia para todos los voltajes de la conexión . **OBLIGATORIO.**

Patilla 8. **DETECTOR DE PORTADORA DE DATOS (DCD)**. Sus usos varían , pero en un DTE se usa frecuentemente para deshabilitar la recepción de datos.

Patilla 20. **TERMINAL DE DATOS LISTO (DTR)**: Salida de propósito general . Generalmente se usa para comunicar al DCE que el DTE está encendido y listo para funcionar.

Se usan muchas otras patillas en las conexiones entre microordenadores , pero la mayoría de la actividad importante se concentra en estas nueve patillas , aunque la 1 no se suele utilizar mucho por lo descrito antes.

DEFINICIONES LOGICAS

En comparación con las convenciones lógicas de uso corriente, la relación voltaje/lógica en la conexión está invertida : un voltaje positivo en la conexión representa un 0 , mientras que un voltaje negativo representa un 1. La fig A3.3 muestra las definiciones lógicas para el RS-232-C.

Hemos de fijarnos en que la lógica está invertida : el 1 está asignado a niveles negativos de voltaje , y el cero a

los positivos . Para garantizar un cero , una patilla de salida debe mantener un voltaje entre +5 y +15 voltios . Similarmente , un 1 garantizado debe estar entre -5 y -15 voltios . La "banda muerta" entre +5 y -5 se conoce como "región de transición" , donde los niveles lógicos no están definidos . Esto significa que cualquier salida entre + 5 y - 5 voltios puede interpretarse ambiguamente como 0 ó 1 . La figura A3.4 muestra las definiciones lógicas para las entradas .

La única diferencia entre esta definición y la de las salidas estriba en ancho de la zona de transición . La zona lógica indefinida de la entrada es únicamente de 6 voltios (de +3 a -3 voltios) , mientras la de la salida es de 10 voltios . Esta diferencia , aparentemente incidental , es de gran importancia .

MARGEN DE RUIDOS .

La diferencia entre las definiciones para los voltajes mínimos permisibles se conoce como margen de ruidos del circuito . Esto significa que se permite algún ruido eléctrico junto con el voltaje de salida , sin afectar adversamente el nivel lógico de la entrada .

La disparidad entre las regiones de transición para entradas y salidas también funciona como un margen general de seguridad compensando algunas pérdidas de voltaje en el cable . Pueden consumirse hasta 2 voltios en la transmisión a través del cable sin que el voltaje caiga en la zona de transición indefinida de la entrada .

Puesto que las señales de control y acoplamiento son de corriente continua , no están (en comparación con las señales de datos) relativamente afectadas por la longitud del cable . Por esta razón , las normas RS-232-C imponen requerimientos menos rigurosos en las señales de control que en las señales de datos.

ORDEN DE TRANSMISION DE LOS BITS

Los datos se transmiten " al revés " : el bit menos significativo se transmite antes , seguido de los otros en orden inverso a su significancia. Puesto que las tablas lógicas (tablas de verdad , mapas de bits , etc) se suelen dibujar con el bit más significativo a la izquierda , se tiende a pensar la transmisión de bits del mismo modo ; pero el hecho cierto es que a nivel físico la transmisión ocurre como se acaba de señalar.

MARCA Y ESPACIO

Los primeros diseñadores de dispositivos electromecánicos (como el teleimpresor) descubrieron que la fiabilidad podía aumentarse grandemente manteniendo una corriente fija en la línea durante los tiempos muertos (esto es, cuando no se transmite dato) ; la transmisión de datos se realizaba interrumpiendo la corriente de reposo . Este estado de reposo (flujo de corriente) fué asignado arbitrariamente al valor 1. En las comunicaciones en serie , este estado también se llama condición de MARCA . De modo complementario , la ausencia de flujo de corriente (esto es, la transmisión real de datos) se define como un 0 o como condición de ESPACIO .

NIVELES LOGICOS DE CONTROL Y ACOPLAMIENTO

En la RS-232-C un voltaje positivo representa un 0 lógico y un voltaje negativo un 1. Luego una salida es positiva (0 lógico), cuando está habilitada. Esto implica, por ejemplo que cuando TERMINAL DE DATOS LISTO (DTR) está en un 0 lógico (FALSO) el terminal de datos está listo.

Para eliminar estas incongruencias, es probablemente más fácil analizar las líneas de datos (TRANSMISION DE DATOS Y RECEPCION DE DATOS) en términos de lógica invertida, y aplicar la lógica convencional a las señales de control y acoplamiento.

REGLAS BASICAS DE RS-232-C

Existen dos especificaciones eléctricas básicas:

- La especificación más importante para los conectoros es casi demasiado buena para ser cierta. Las normas EIA dicen:

El conector de un circuito de intercambio ha de diseñarse para soportar un circuito abierto, un cortocircuito en dicho hilo de intercambio en el cable de conexión y cualquier otro conductor en dicho cable... incluyendo la Masa de Señal, sin causar daños a sí mismo o equipos asociados.

Cualquier patilla se puede conectar a cualquier otra en cualquier momento sin causar daños. Mientras se limite a conectar dispositivos RS-232-C no se puede dañar nada.

Luego ello significa que podemos indiscriminadamente puentear patillas, compartir voltajes, o cortocircuitar

voltajes no deseados sin necesidad de preocuparnos.

Pero he aquí un aviso : no quiere decir que la conexión RS-232-C no sea vulnerable a los daños . Puede dañarse al conectar a cualquier circuito que provoque voltajes fuera del rango entre +25 y -25 voltios . Nunca se ha de suponer que la presencia de un conector DB-25 garantiza la conexión RS-232-C o incluso una conexión "en serie".

- La segunda especificación clave requiere que el transmisor siempre esté a un voltaje negativo (1 lógico o MARCA) cuando se están transmitiendo los datos . Ya sabemos que los datos deben transmitirse por la patilla 2 o la 3 , dependiendo del sexo del equipo (DTE o DCE) . Luego simplemente probando estas patillas podemos saber si un dispositivo es un DTE o un DCE .

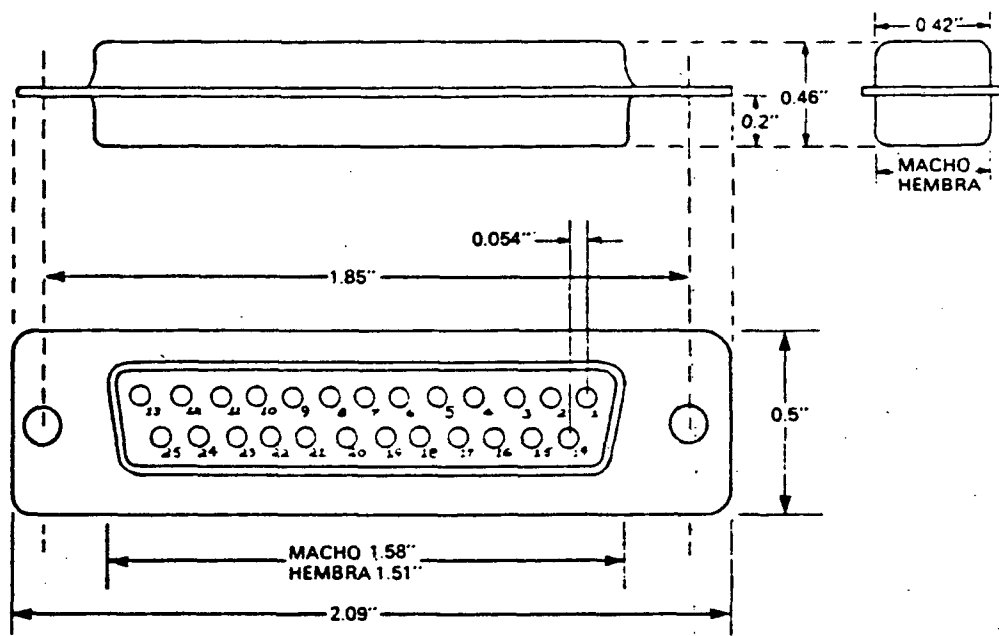
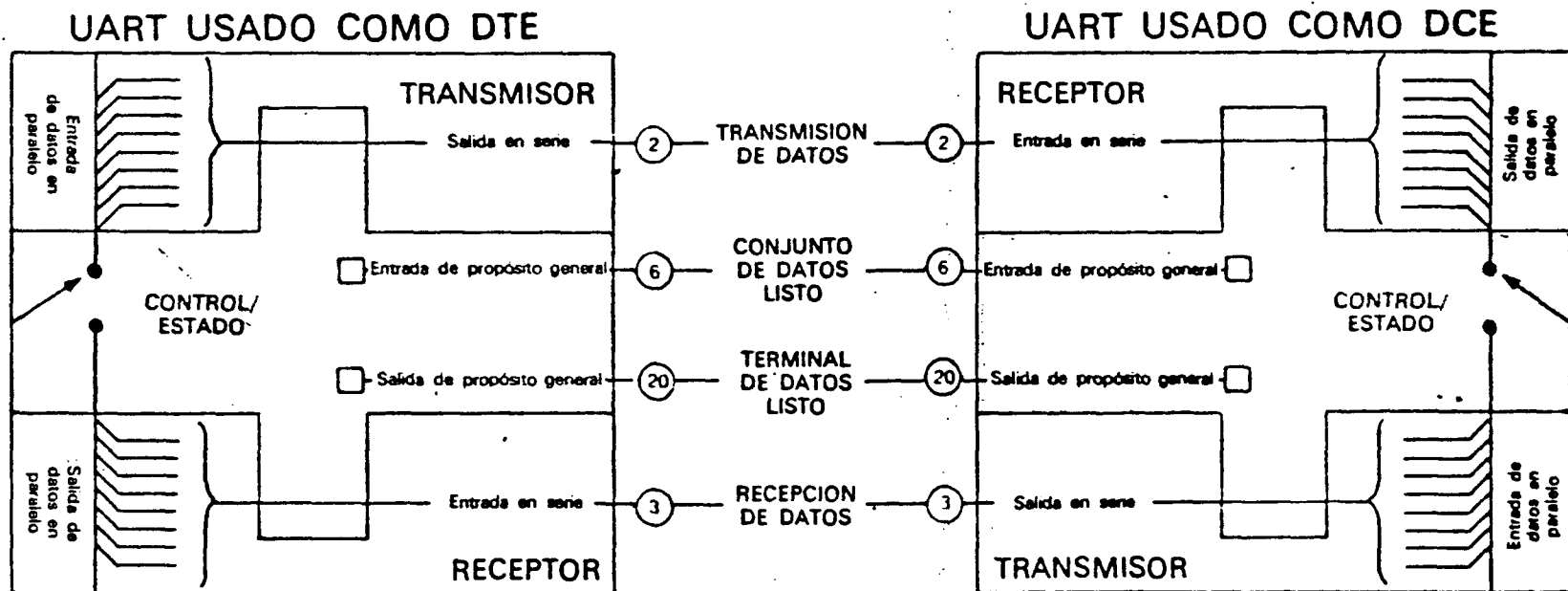


Figura A3.1 Esquema de un conector DB-25.

Figura A3.2



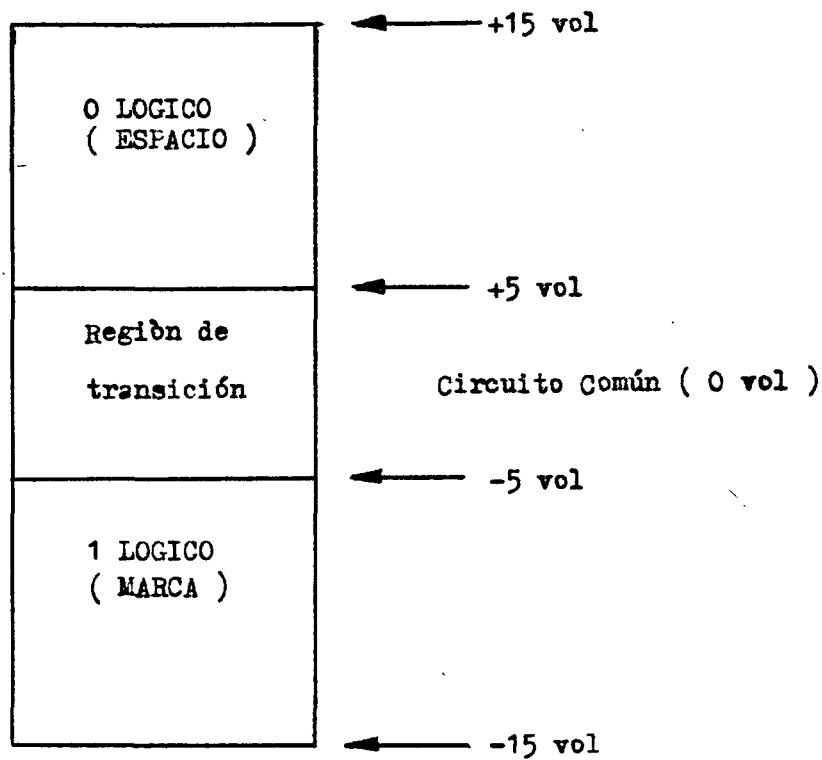


Figura A3.3 Definiciones lógicas para las salidas RS-232-C

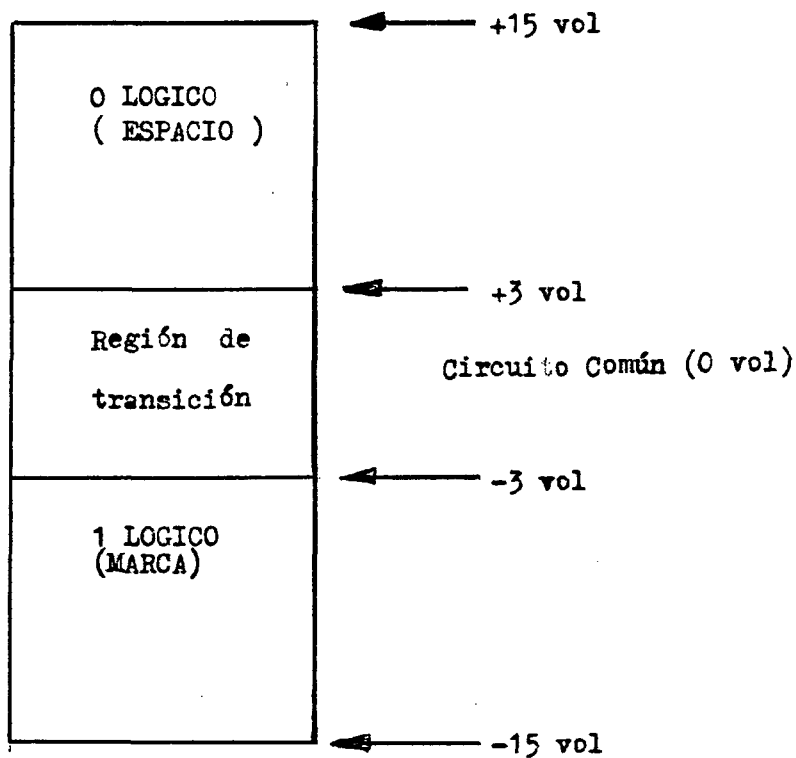


Figura A3.3 Definiciones lógicas para las entradas RS-232-C

APENDICE IV

PROGRAMA ENSAMBLADOR DESARROLLADO PARA LA TARJETA SIC
(ONCEB5).

PROGRAMA DESARROLLADO PARA QUE EL APPLE FUNCIONE COMO TERMI-
NAL CON LA TARJETA DE INTERCONEXION "SERIAL INTERFACE CARD"
QUE HEMOS PUESTO EN EL SLOT CINCO. EN ESTE CASO LE HEMOS
PUESTO EL MODO ECO

```

6000- 2C 58 FF BIT $FF58 ; INICIALIZACION DE FLAGS
6003- A2 FF LDX #$FF ; ESTABLECER PUNTERO PILA
6005- 9A TXS
6006- A9 A0 LDA #$A0 ; INICIALIZACION TARJETA
6008- 20 00 C5 JSR $C500 ; MANDANDO CARACT. NULO
600B- A9 01 LDA #$01 ; ESTABLECE SOLO UN BIT DE
600D- 8D FD 04 STA $04FD ; STOP
6010- 20 58 FC JSR $FC58 ; SE BORRA PANTALLA
6013- A2 C5 LDX #$C5 ; X=CN=C5
6015- A4 24 LDY $24
6017- B1 28 LDA ($28),Y
6019- B5 27 STA $27 ; SALVA CARACTER DE PANTALLA
601B- A9 07 LDA #$07 ; IMPLEMENTA "+" TESTELLEANTE
601D- 25 4F AND $4F ; FORMA UN CURSOR
601F- D0 10 BNE $6031
6021- A4 24 LDY $24
6023- A9 DF LDA #$DF
6025- D1 28 CMP ($28),Y ; CURSOR EN PANTALLA?
6027- D0 02 BNE $602B ; SI NO, PONLO
6029- A5 27 LDA $27 ; DE OTRA MANERA PONE CARACTER
602B- 91 28 STA ($28),Y ; VERDADERO EN PANTALLA
602D- E6 4F INC $4F ; HAZLO TESTELLEAR, PERO NI
602F- E6 4F INC $4F ; LENTO, NI RAPIDO
6031- 98 TYA ; SE GUARDAN Y, X EN
6032- 48 PHA ; STACK
6033- A2 C5 LDX #$C5
6035- 8A TXA
6036- 48 PHA
6037- 20 74 60 JSR $6074 ; HAY DATOS ?
603A- F0 1D BEQ $6059 ; SALTO SI ES ENTRADA

```

SUBROUTINA DE SALIDA DE DATOS A PANTALLA Y HACIA EL EXTERIOR

```

603C- 68 PLA ; RESTAURA LOS REGISTROS
603D- AA TAX ; X E Y
603E- 68 PLA
603F- A8 TAY
6040- BD B8 05 LDA $05B8,X ; COGE EL DATO
6043- C9 8D CMP #$8D ; ES CR?
6045- D0 06 BNE $604D ; SI NO , SALTA
6047- A4 24 LDY $24 ; SI, LUEGO BORRA CURSOR
6049- A9 A0 LDA #$A0 ; DE PANTALLA RESTAURANDOLO
604B- 91 28 STA ($28),Y ; CON CARACTER NULO
604D- BD B8 05 LDA $05B8,X ; VUELVE A CARGAR EL CARACTER
6050- 20 ED FD JSR $FDED ; SALIDA DEL DATO POR PANTALLA
6053- 20 AA C9 JSR $C9AA ; SALIDA HACIA EL OTRO APPLE
6056- 4C 22 60 JMP $6021 ; SALTO A EMPEZAR

```

SUBROUTINA DE SALIDA DE DATOS A PANTALLA
PROCEDENTES DEL EXTERIOR

```
6059- 68          PLA          ; RESTAURA REGISTROS
605A- AA          TAX          ; X,Y
605B- 68          PLA
605C- AB          TAY
605D- BD B8 05    LDA          $05B8,X; COGE DATO
6060- 09 B0       ORA          #$80  ; SUMA $80 YA QUE VIENE
6062- 48          PHA          ; DE MONITOR DEL EXTERIOR
6063- A4 24       LDY          $24
6065- A5 27       LDA          $27  ; SALVA CARACT DE PANTALLA
6067- 91 28       STA          ($28),Y
6069- 68          PLA
606A- C9 BA       CMP          #$8A  ; SI ES LF VUELVE A EMPEZAR
606C- F0 B3       BEQ          $6021
606E- 20 ED FD    JSR          $FDED ; SACA POR PANTALLA
6071- 4C 21 60    JMP          $6021
```

SUBROUTINA DE ENTRADA DE DATOS TANTO DESDE FUERA COMO DESDE
EL PROPIO TECLADO

```
6074- 78          SEI          ; NO INTERRUPT.HARDWARE
6075- BD B8 04    LDA          $04B8,X; PREPARA PARA OPCIONES
6078- 29 03       AND          #$03  ; DE BIT DE PARIDAD/STOP
607A- 4A          LSR
607B- 6E 78 07    ROR          $0778 ; INICIALIZA PARIDAD
607E- 48          PHA          ; SI A=0 SE ESPERA PARIDAD
607F- BD B8 06    LDA          $06B8,X
6082- 85 35       STA          $35  ; INICIALIZA BIT DE CUENTA
6084- AC F8 06    LDY          $06F8
6087- 38          SEC          ; CARRY PARA BIT DE START
6088- B0 26       BCS          $60B0 ; SALTA SIEMPRE
608A- AC F8 06    LDY          $06F8 ; TODOS LOS BIT PUESTOS?
608D- C6 35       DEC          $35  ; SI, OBTIENE BIT PARID/STOP
608F- F0 5E       BEQ          $60EF ; NO, LLEGAN MAS
6091- 18          CLC
6092- 4C B0 60    JMP          $60B0 ; SALTO A VER SI HAY DATOS
```

VUELTA A EMPEZAR SI NO HAY DATOS , PARA QUE EL CURSOR SIGUA
CON SU TESTELLEO

```
6095- 68          PLA          ; DESCARGA LA PILA 3 VECES
6096- 68          PLA          ; Y RESTABLECE REGISTROS
6097- 68          PLA          ; PARA VOLVER BIEN
6098- 68          PLA
6099- AA          TAX
609A- 68          PLA
609B- AB          TAY
609C- 4C 21 60    JMP          $6021
609F- EA          NOP
```

```

60A0- EA NOP
60A1- EA NOP
60A2- EA NOP
60A3- EA NOP
60A4- EA NOP
60A5- EA NOP
60A6- EA NOP ; ESPACIO RESERVADO PARA
60A7- EA NOP ; POSIBLES MODIFICACIONES
60A8- EA NOP
60A9- EA NOP
60AA- EA NOP
60AB- EA NOP
60AC- EA NOP
60AD- EA NOP
60AE- EA NOP
60AF- EA NOP

```

```

*****
AQUI ES DONDE VERDADERAMENTE SE VE SI HAY LLEGADA DE DATOS
*****

```

```

60B0- B9 80 C0 LDA $C080,Y
60B3- 90 1F BCC $60D4 ; SALTO SI DATOS
60B5- 10 0D BPL $60C4 ; SALTO SI BIT DE START
60B7- AD 00 C0 LDA $C000 ; MIRAR TECLADO
60BA- 10 D9 BPL $6095 ; SALTA SI NO DATO DE TECLADO
60BC- 9D B8 05 STA $05B8,X ; ALMACENAMIENTO DE DATO
60BF- 2C 10 C0 BIT $C010 ; BORRA TECLADO
60C2- 68 PLA ; FIJA STACK Y HACE
60C3- 60 RTS ; RETORNO
60C4- BD B8 03 LDA $03B8,X
60C7- A0 09 LDY #$09 ; DELAY PARA TOMAR BIEN
60C9- 88 DEY ; LOS DATOS
60CA- D0 FD BNE $60C9
60CC- E9 01 SBC #$01 ; ESPERAMOS OTRA VEZ?
60CE- F0 BA BEQ $608A ; NO
60D0- A0 0E LDY #$0E ; SI , OTROS 78 CICLOS
60D2- D0 F5 BNE $60C9 ; SALTA SIEMPRE
60D4- A8 TAY
60D5- 2A ROL ; ROTA DATO A TRAVES
60D6- 7E B8 05 ROR $05B8,X ; DEL CARRY
60D9- 98 TYA
60DA- 4D 78 07 EOR $0778 ; ACTUALIZA PARIDAD
60DD- 8D 78 07 STA $0778
60E0- BD B8 03 LDA $03B8,X ; OBTIENE NUM. DE DELAYS
60E3- 38 SEC
60E4- E9 01 SBC #$01
60E6- F0 A2 BEQ $608A
60E8- A0 09 LDY #$09
60EA- 88 DEY ; DELAY=53*(BRATE-1)CICLOS
60EB- D0 FD BNE $60EA
60ED- F0 F5 BEQ $60E4
60EF- 68 PLA ; PARIDAD?
60F0- D0 08 BNE $60FA ; SI NO, SALTA
60F2- B9 80 C0 LDA $C080,Y ; OBTIENE BIT DE PARIDAD
60F5- 4D 78 07 EOR $0778
60F8- 30 16 BMI $6110
60FA- B9 80 C0 LDA $C080,Y

```

60FD-	10 FB	BPL	\$60FA	
60FF-	A9 0A	LDA	##0A	
6101-	FD B8 06	SEC	\$06B8,X	
6104-	A8	TAY		
6105-	88	DEY		
6106-	F0 09	BEQ	\$6111	; ENVIA EL DATO AL LLAMADOR
6108-	7E B8 05	ROR	\$05B8,X	; PONE BIT ALTO A UNO
610B-	38	SEC		
610C-	B0 F7	BCS	\$6105	; SALTA SIEMPRE
610E-	00	BRK		
610F-	00	BRK		
6110-	38	SEC		
6111-	60	RTS		; SALTO SACAR AL EXTERIOR
6112-	00	BRK		; Y POR PANTALLA
6113-	00	BRK		

PROGRAMA DESARROLLADO PARA LA TARJETA SSC.

PROGRAMAS BASIC PARA EL TRATO DE FICHEROS (SI90 y SI90A)

LIST

```
5  REM "PROGRAMA BASIC PARA EL DE
    SARROLLO DEL APPLE COMO TERM
    INAL "
10 D$ = " "
20 PRINT D$;"BLOAD CALMA90"
30 CALL 24576
35 REM "AHORA EMPIEZA EL PROGRA
    MA EN SI"
40 C = PEEK (4096)
45 IF C = 128 THEN 400
50 IF C = 01 OR C = 16 THEN 130
60 E$ = " "
70 FOR I = 1 TO 32
80 IF PEEK (3327 + I) = 00 THEN
    115
90 E$(I) = CHR$ ( PEEK (3327 + I
    ))
100 E$ = E$ + E$(I)
110 NEXT I
115 IF C = 04 THEN 150
120 PRINT D$;"BLOAD";E$;"",A$7000
    "
150 IF C < > 01 AND C < > 08 THEN
    160
155 PRINT D$;"PR#1"
160 FOR I = 28672 TO 38399
170 V = PEEK (I)
175 A$ = CHR$ (V)
180 IF V = 209 OR V = 211 THEN 3
    20
290 PRINT A$;
300 A$ = " "
310 NEXT I
320 IF C < > 01 OR C < > 08 THEN
    325
322 PRINT D$;"PR#0"
325 PRINT D$
330 IF C < > 04 THEN 370
340 Z = (16 * 16 * ( PEEK (4207))
    + PEEK (4208))
350 PRINT D$;"BSAVE";E$;"",A$7000
    ,L";Z
370 CALL 24761
380 GOTO 35
400 PRINT D$;"CATALOG"
410 PRINT D$
420 GOTO 370
```

J

LIST

```

5  REM "PROGRAMA BASIC PARA EL D
    ESARROLLO DEL APPLE COMO TER
    MINAL"
10  D$ = " "
20  PRINT D$;"BLOAD CALMA90"
30  CALL 24576
35  REM "AHORA EMPIEZA EL PROGRA
    MA EN SI"
40  C = PEEK (4096)
45  IF C = 128 THEN 400
50  IF C = 01 OR C = 16 THEN 130
60  E$ = " "
70  FOR I = 1 TO 32
80  IF PEEK (3327 + I) = 00 THEN
    115
90  E$(I) = CHR$ ( PEEK (3327 + I
    ))
100 E$ = E$ + E$(I)
110 NEXT I
115 IF C = 04 THEN 130
120 PRINT D$;"BLOAD";E$;"",A$7000
    "
130 A$ = " "
140 B$ = "ABCDEF"
150 IF C < > 01 AND C < > 08 THEN
    160
155 PRINT D$;"PR#1"
160 FOR I = 28672 TO 38399
170 V = PEEK (I)
180 IF V = 209 OR V = 211 THEN 3
    20
190 R = INT (V / 16)
200 FOR J = 1 TO 2
210 IF R < 10 THEN 240
220 W$ = MID$ (B$,R - 9,1)
230 GOTO 250
240 W$ = STR$ (R)
250 A$ = A$ + W$
260 X = ((V / 16) - R) * 16
270 R = X
280 NEXT J
290 PRINT A$;" ";
300 A$ = " "
310 NEXT I
320 IF C < > 01 OR C < > 08 THEN
    325
322 PRINT D$;"PR#0"
325 PRINT D$
330 IF C < > 04 THEN 370
340 Z = (16 * 16 * ( PEEK (4207))
    + PEEK (4208))
350 PRINT D$;"BSAVE";E$;"",A$7000
    ,L";Z
370 CALL 24761
380 GOTO 35
400 PRINT D$;"CATALOG"
410 PRINT D$
420 GOTO 370

```

PROGRAMA ENSAMBLADOR PARA LA TARJETA SSC (CALMA90)

```

6075-   A9 96           LDA    ##96
6077-   CD 6C 60       CMP    $606C
607A-   D0 EC         BNE    $606F
607C-   A9 70         LDA    ##70
607E-   8D 6C 60     STA    $606C
6081-   EA            NOP
6082-   EA            NOP
6083-   A2 00         LDX    ##00
6085-   A9 00         LDA    $00      ; SE BORRAN LAS POSICIONES DE
6087-   9D 00 0D     STA    $0D00,X ; MEMORIA DONDE SE ALMACENAN
608A-   E8            INX      ; LUEGO LOS NOMBRES DE FICHEROS
608C-   E0 20         CPX    ##20  ; A SACAR O METER EN EL DISCO
608D-   D0 F6         BNE    $608C
608F-   AE 50 10     LDX    $1050
6092-   A9 00         LDA    ##00      ; RESETEO DE LOS FLAGS
6094-   8D 6F 10     STA    $106F
6097-   8D 00 10     STA    $1000
609A-   20 B5 63     JSR    $63B5      ; SALTO A LA SUBROUTINA DE DESARRO-
609D-   00           BRK      ; LLO DEL PROGRAMA EN SI
609E-   00           BRK
609F-   00           BRK

```

```

*****
SUBROUTINA PARA VOLVER AL PROGRAMA DE GOBIERNO
DESDE BASIC
*****

```

```

60A0-   A9 2D           LDA    ##2D      ; PREPARA AL SISTEMA PARA QUE
60A2-   9D 38 07     STA    $0738,X ; VUELVA BIEN DESDE BASIC
60A5-   8E 40 10     STX    $1040      ; GUARDA EL VALOR DE X, Y ADEMAS
60A8-   BD 00 01     LDA    $0100,X ; DOS POSICIONES DEL PUNTERO DE
60AB-   9D 00 0E     STA    $0E00,X ; LA PILA QUE SE RESTABLECERAN
60AE-   CA            DEX      ; DESPUES DE VOLVER DESDE BASIC
60AF-   BD 00 01     LDA    $0100,X
60B2-   9D 00 0E     STA    $0E00,X
60B5-   4C 23 D8     JMP    $D823      ; SALTO A BASIC PROPIAMENTE DICHO
60B8-   00           BRK
60B9-   68           PLA      ; VUELTA DESDE BASIC, LUEGO SE
60BA-   68           PLA      ; SACAN DOS POSICIONES DE PILA
60BB-   AE 40 10     LDX    $1040      ; Y SE RESTABLECE EL VALOR DEL
60BE-   9A            TXS      ; REGISTRO X , Y LOS VALORES
60BF-   BD 00 0E     LDA    $0E00,X ; DE PUNTERO DE LA PILA
60C2-   9D 00 01     STA    $0100,X
60C5-   CA            DEX
60C6-   BD 00 0E     LDA    $0E00,X
60C9-   9D 00 01     STA    $0100,X
60CC-   AE 50 10     LDX    $1050
60CF-   A9 A0         LDA    ##A0      ; VUELTA A REINICIALIZAR EL SIS-
60D1-   20 00 C2     JSR    $C200      ; TEMA
60D4-   AD 00 10     LDA    $1000      ; MIRAR SI EL FLAG QUE HIZO IR A
60D7-   29 3D         AND    ##3D      ; BASIC ES EL DE ...
60D9-   D0 12         BNE    $60F9
60DB-   09 42         ORA    ##42
60DD-   8D 00 10     STA    $1000

```

```

60E0- 20 00 6C JSR $6C00
60E3- AD 00 10 LDA $1000
60E6- C9 42 CMP ##42
60E8- D0 03 BNE $60F9
60EA- 4C 66 65 JMP $6566
60ED- 4C 12 60 JMP $6012
60F0- 00 BRK
60F1- 00 BRK
60F2- 00 BRK
60F3- 00 BRK

```

```

*****
RUTINA DE ENTRADA DE CARACTERES DESDE FUERA
*****

```

```

60F4- AD 00 10 LDA $1000 ; PRIMERO MIRA SI HAY ALGUN
61F7- 29 7F AND #$7F ; FLAG ACTIVADO
61F9- F0 03 BEQ $610A
61FB- 4C A0 6A JMP $6AA0
61FE- 20 D2 65 JSR $65D2
6101- 90 1E BCC $612D
6103- B9 88 C0 LDA $C088, Y
6106- 09 80 ORA #$80
6108- 29 7F AND #$7F ; SI EL CARACTER DE LLEGADA ES
610A- C9 7F CMP #$7F ; 7F ENTONCES LO DESECHA
610C- F0 13 BEQ $612D
610E- C9 05 CMP #$05 ; SI ES 05 TIENE QUE DEVOLVER
6110- F0 11 BEQ $612F ; UN 06 (PROTOCOLO ENQ/ACK)
6112- C9 8A CMP #$8A
6114- D0 09 BNE $6123
6116- A8 TAY
6117- BD 38 07 LDA $0738, X
611A- 29 20 AND #$20
611C- D0 03 BNE $612D
611E- 98 TYA
611F- 38 SEC
6120- 60 RTS
6121- 18 CLC
6122- 60 RTS
6123- 4C 20 6B JMP $6B20 ; SALTO SUBROUTINA DE
6126- 00 BRK ; SALIDA DEL CARACTER
6127- 00 BRK ; DE DEVUELTA $06
6128- 00 BRK
6129- 00 BRK
612A- 00 BRK
612B- 00 BRK
612C- 00 BRK
612D- 00 BRK

```

```

*****
RUTINA PARA PROCESAR CARACTERES ESPECIALES DE SALIDA COMO SON
LOS DIVERSOS CONTROLES PARA SACAR POR IMPRESORA, SACAR/GUARDAR
EN DISCO Y OTROS
*****

```

612E-	8D 10 C0	STA	\$C010	; GUARDO EL CARACTER DE SALI
6131-	8D 10 10	STA	\$1010	; DA EN DOS POSIC.DISINTAS
6134-	C9 83	CMP	##83	; SI ES CTRL C SALTA A BASIC
6136-	D0 08	BNE	\$6140	; Y SACA EL CATALOGO
6138-	A9 80	LDA	##80	
613A-	8D 00 10	STA	\$1000	
613D-	4C A0 60	JMP	\$60A0	
6140-	C9 9A	CMP	##9A	; MIRAR SI ES CTRL Z CON LO
6142-	D0 28	BNE	\$616C	; CUAL SE PRODUCE UN BUCLE
6144-	A9 0F	LDA	##0F	; DURANTE EL CUAL SE CORTA LA
6146-	99 8A C0	STA	\$C08A,Y	; TRANSMISION , O SEA UN BREAK
6149-	8A	TXA		
614A-	48	PHA		
614B-	98	TYA		
614C-	48	PHA		
614D-	A9 04	LDA	##04	
614F-	8D 80 60	STA	\$6080	
6152-	A2 FF	LDX	##FF	
6154-	A0 FF	LDY	##FF	
6156-	88	DEY		
6157-	D0 FD	BNE	\$6156	
6159-	CA	DEX		
615A	D0 F8	BNE	\$6154	
615C-	CE 80 10	DEC	\$1080	
615F-	D0 F1	BNE	\$6152	
6161-	EA	NOP		
6162-	EA	NOP		
6163-	68	PLA		
6164-	A8	TAY		
6165-	68	PLA		
6166-	AA	TAX		
6167-	A9 0B	LDA	##0B	; SE RESTABLECE LA TRANSMISION
6169-	99 8A C0	STA	\$C08A,Y	
616C-	C9 9B	CMP	##9B	; SI ES ESCAPE SALTA A TERMINAR
616E-	D0 03	BNE	\$6173	
6170-	4C B0 FE	JMP	\$FEB0	
6173-	C9 93	CMP	##93	; SI ES CTRL S ESPERA A QUE LE
6175-	D0 1E	BNE	\$6195	; LLEGUE UN CTRL Q
6177-	A9 82	LDA	##82	
6179-	8D 3A 67	STA	\$673A	
617C-	20 3E 67	JSR	\$673E	
617F-	90 FB	BCC	\$617C	
6181-	60	RTS		
6182-	8D 10 C0	STA	\$C010	
6185-	C9 91	CMP	##91	; ES CRTL-Q?
6187-	D0 09	BNE	\$6192	
6189-	A9 2E	LDA	##2E	; SI, QUITA EL MODO DE ESPERA
618B-	8D 3A 67	STA	\$673A	
618E-	A9 A0	LDA	##A0	
6190-	38	SEC		
6191-	60	RTS		
6192-	4C 3E 67	JMP	\$673E	
6195-	AD 00 10	LDA	\$1000	; VAMOS A VER SI SE HA ACTIVADO
6198-	29 3F	AND	##3F	; ALGUN FLAG ESPECIAL
619A-	F0 03	BEQ	\$619F	; SI ES ASI SALTA A PROCESARLO
619C-	4C A0 6A	JMP	\$6AA0	

```

619F-- AD 10 10 LDA $1010 ; RECUPERA CARACTER DE TECLADO
61A2-- C9 89 CMP #$89 ; MIRA SI ES CTRL I
61A4-- D0 0B BNE $61B1
61A6-- AD 00 10 LDA $1000 ; SI, PONE FLAG=01
61A9-- 09 01 ORA #$01
61AB-- 8D 00 10 STA $1000
61AE-- 1B CLC
61AF-- 60 RTS
61B0-- 00 BRK
61B1-- C9 8B CMP #$8B ; MIRA SI ES CTRL K
61B3-- D0 0B BNE $61C0
61B5-- AD 00 10 LDA $1000 ; SI, PONE FLAG=04
61B8-- 09 04 ORA #$04
61BA-- 8D 00 10 STA $1000
61BD-- 4C 00 6A JMP $6A00
61C0-- C9 82 CMP #$82 ; MIRA SI ES CTRL B
61C2-- D0 0B BNE $61CF
61C4-- AD 00 10 LDA $1000 ; SI, PONE FLAG=02
61C7-- 09 02 ORA #$02
61C9-- 8D 00 10 STA $1000
61CC-- 4C 00 6A JMP $6A00
61CF-- C9 90 CMP #$90 ; MIRA SI ES CTRL P
61D1-- D0 0B BNE $61DE
61D3-- AD 00 10 LDA $1000 ; SI, PONE FLAG=08
61D6-- 09 08 ORA #$08
61D8-- 8D 00 10 STA $1000
61DB-- 4C 00 6A JMP $6A00
61DE-- C9 8A CMP #$8A ; MIRA SI ES CTRL J
61E0-- D0 0B BNE $61ED
61E2-- AD 00 10 LDA $1000 ; SI, PONE FLAG=10
61E5-- 09 10 ORA #$10
61E7-- 8D 00 10 STA $1000
61EA-- 4C 3D 67 JMP $673D
61ED-- C9 96 CMP #$96 ; MIRA SI ES CTRL V
61EF-- D0 0B BNE $61FC
61F1-- AD 00 10 LDA $1000 ; SI, PONE FLAG=20
61F4-- 09 20 ORA #$20
61F6-- 8D 00 10 STA $1000
61F9-- 4C 00 6A JMP $6A00
61FC-- 4C 3C 67 JMP $673C ; SI NO ES NINGUN CARACTER ES-
61FF-- 00 BRK ; PECIAL SIMPLEMENTE LO TRANS-
; MITE

```

```

*****
RUTINA PARA SACAR EN PANTALLA LOS DISTINTOS COMENTARIOS DEBI-
DOS A LOS DIVERSOS CONTROLES Y RECOGER NOMBRES DE FICHEROS
QUE SE UTILIZARAN EN BUSQUEDA O CREACION EN DISCO
*****

```

```

6A00-- AD 00 10 LDA $1000 ; CARGA EL FLAG
6A03-- C9 04 CMP #$04 ; ES 04?
6A05-- D0 0D BNE $6A14
6A07-- A9 1C LDA #$1C ; CARGA LONGITUD CARACTERES
6A09-- 8D 41 6A STA $6A41

```



```

6A0C-   A9 CC       LDA   #$CC      ; CARGA POSICION CARACTERES
6A0E-   8D 43 6A   STA   $6A43
6A11-   4C 40 6A   JMP   $6A40
6A14-   C9 02       CMP   #$02      ; ES 02?
6A16-   D0 0D       BNE   $6A25
6A18-   A9 1B       LDA   #$1B      ; CARGA LONGITUD CARACTERES
6A1A-   8D 41 6A   STA   $6A41
6A1D-   A9 AD       LDA   #$AD      ; CARGA POSICION CARACTERES
6A1F-   8D 43 6A   STA   $6A43
6A22-   4C 40 6A   JMP   $6A40
6A25-   C9 08       CMP   #$08      ; ES 08?
6A27-   D0 0D       BNE   $6A36
6A29-   A9 1D       LDA   #$1D      ; CARGA LONGITUD CARACTERES
6A2B-   8D 41 6A   STA   $6A41
6A2E-   A9 8D       LDA   #$8D      ; CARGA POSICION CARACTERES
6A30-   8D 43 6A   STA   $6A43
6A33-   4C 40 6A   JMP   $6A40
6A36-   A9 20       LDA   #$20      ; COMO NO ES NINGUNO DE LOS
6A38-   8D 41 6A   STA   $6A41      ; ANTERIORES ES FLAG=20
6A3B-   A9 6A       LDA   #$6A      ; LUEGO CARGA LONGITUD Y
6A3D-   8D 43 6A   STA   $6A43      ; POSICION DE CARACTERES

```

```

*****
SUBROUTINA DE CARGA DE LAS POSICIONES NECESARIAS PARA SACAR
COMENTARIO CON RESPECTO A LOS DISTINTOS CONTROLES
*****

```

```

6A40-   A0 20       LDY   #$20      ; SE CARGA EN Y LONGITUD
6A42-   B9 6A 6B   LDA   $6B6A,Y ; CARGA POSICION + LONG
6A45-   85 27       STA   $27
6A47-   98         TYA
6A48-   48         PHA
6A49-   20 A3 67   JSR   $67A3      ; SIEMPRE ENVIA A PANTALLA
6A4C-   68         PLA
6A4D-   A8         TAY
6A4E-   88         DEY
6A4F-   10 F1       BPL   $6A42
6A51-   8E 50 10   STX   $1050      ; GUARDA VALOR DE X
6A54-   A2 00       LDX   #$00
6A56-   8E 60 10   STX   $1060      ; INICIALIZA REG BASE
6A59-   AE 50 10   LDX   $1050
6A5C-   20 34 CC   JSR   $CC34      ; ESPERA CARACTER DE TECLADO
6A5F-   10 FB       BPL   $6A5C
6A61-   C9 88       CMP   #$88      ; ES BAKSPACE ?
6A63-   F0 DB       BEQ   $6A40      ; SI EMPIEZA DE NUEVO
6A65-   C9 8D       CMP   #$8D      ; SI ES CR TERMINA COGIDA CARACT
6A67-   F0 13       BEQ   $6A7C
6A69-   85 27       STA   $27      ; SACA CARACT POR PANTALLA
6A6B-   20 A3 CC   JSR   $CCA3
6A6E-   AE 60 10   LDX   $1060
6A71-   9D 00 0D   STA   $0D00,X ; LO ALMACENA
6A74-   EB         INX
6A75-   8E 60 10   STX   $1060      ; INCREMENTA REG BASE
6A78-   E0 20       CPX   #$20

```

6A7A-	D0 DD	BNE	\$6A59	; SI NO HAN LLEGADO 32 CARAC
6A7C-	A2 00	LDX	##00	; SIGUE , SI NO RETORNA
6A7E-	8E 60 10	STX	\$1060	
6A81-	AE 50 10	LDX	\$1050	; VUELVE A RESTABLECER X
6A84-	A9 8D	LDA	##8D	; SACA UN CR
6A86-	85 27	STA	\$27	
6A88-	20 A3 CC	JSR	\$CCA3	
6A8B-	A9 A0	LDA	##A0	; SACA CARACTER NULO
6A8D-	85 27	STA	\$27	
6A8F-	20 A3 CC	JSR	\$CCA3	
6A92-	18	CLC		
6A93-	4C 3D 67	JMP	\$673D	; VUELVE A SUBROUTINA DE
6A96-	00	BRK		; BUSQUEDA DE CARACTERES
6A97-	00	BRK		
6A98-	00	BRK		
6A99-	00	BRK		
6A9A-	00	BRK		
6A9B-	00	BRK		
6A9C-	00	BRK		
6A9D-	00	BRK		
6A9E-	00	BRK		
6A9F-	00	BRK		

SUBROUTINA DE PROCESADO DE COMANDOS ESPECIALES

6AA0-	29 6A	AND	##6A	
6AA2-	D0 66	BNE	\$6B0A	
6AA4-	20 D2 65	JSR	\$65D2	
6AA7-	90 16	BCC	\$6ABF	
6AA9-	B9 88 C0	LDA	\$C088,Y	
6AAC-	09 80	ORA	##80	
6AAE-	EA	NOP		
6AAF-	EA	NOP		
6AB0-	C9 7F	CMP	##7F	
6AB2-	F0 0B	BEQ	\$6ABF	
6AB4-	C9 05	CMP	##05	
6AB6-	F0 0B	BEQ	\$6AC3	
6AB8-	EA	NOP		
6AB9-	EA	NOP		
6ABA-	EA	NOP		
6ABB-	4C CD 6A	JMP	\$6ACD	
6ABE-	EA	NOP		
6ABF-	18	CLC		
6AC0-	60	RTS		
6AC1-	EA	NOP		
6AC2-	EA	NOP		
6AC3-	4C 20 6B	JMP	\$6B20	; SALTO A SACAR UN \$06
6AC6-	EA	NOP		
6AC7-	EA	NOP		
6AC8-	EA	NOP		
6AC9-	EA	NOP		
6ACA-	EA	NOP		

6ACB-	EA		NOP		
6ACC-	EA		NOP		
6ACD-	8E 50 10		STX	\$1050	
6AD0-	AE 60 10		LDX	\$1060	
6AD3-	C9 D1		CMP	#\$D1	; SI ES D1 O D3 SALTA
6AD5-	F0 20		BEQ	\$6AF7	; Y NO RECOGE MAS DATOS
6AD7-	C9 D3		CMP	#\$D3	; SINO LOS ALMACENA EN
6AD9-	F0 1C		BEQ	\$6AF7	; POSICIONES A PARTIR DE
6ADB-	9D 00 70		STA	\$7000,X	; LA \$7000
6ADE-	E8		INX		
6ADF-	E0 00		CPX	##00	
6AE1-	D0 2E		BNE	\$6B11	
6AE3-	EE DD 6A		INC	\$6ADD	
6AE6-	EE 6F 10		INC	\$106F	
6AE9-	A9 96		LDA	##96	; SI SE LLEGA A LA \$68FF
6AEB-	CD DD 6A		CMP	\$6ADD	; SE ACABA EL ESPACIO PARA
6AEE-	D0 21		BNE	\$6B11	; GRABACION DE POSICIONES
6AF0-	4C 50 6C		JMP	\$6C50	; SALTO SI EL BUFFER ES
6AF3-	A9 A0		LDA	##A0	; DEMASIADO GRANDE
6AF5-	85 27		STA	\$27	
6AF7-	A9 70		LDA	##70	; ENTONCES SE RESTAURAN
6AF9-	8D DD 6A		STA	\$6ADD	; REGISTROS Y SE SALTA A
6AFC-	8E 70 10		STX	\$1070	; BASIC
6AFF-	A2 00		LDX	##00	
6B01-	8E 60 10		STX	\$1060	
6B04-	AE 50 10		LDX	\$1050	
6B07-	4C A0 60		JMP	\$60A0	
6B0A-	29 40		AND	##40	; ES CTRL-B
6B0C-	F0 F9		BEQ	\$6B07	; SI, VUELVE A BASIC
6B0E-	4C 00 6C		JMP	\$6C00	; NO, SACA CARACT POR TARJETA
6B11-	8E 60 10		STX	\$1060	
6B14-	AE 50 10		LDX	\$1050	
6B17-	18		CLC		
6B18-	60		RTS		
6B12-	00		BRK		
6B13-	00		BRK		
6B14-	00		BRK		
6B15-	00		BRK		
6B16-	00		BRK		
6B17-	00		BRK		
6B18-	00		BRK		
6B19-	00		BRK		
6B1A-	00		BRK		
6B1B-	00		BRK		
6B1C-	00		BRK		
6B1D-	00		BRK		
6B1E-	00		BRK		
6B1F-	00		BRK		
6B20-	20 F5 65		JSR	\$65F5	; RUTINA PARA DEVOLVER EL 06 CUANDO
6B23-	D0 FB		BNE	\$6B20	; LLEGA UN 05
6B25-	98		TYA		
6B26-	09 89		ORA	##89	
6B28-	A8		TAY		
6B29-	A9 06		LDA	##06	

```

6B2B- 99 FF BF STA $BFFF,Y
6B2E- 18 CLC
6B2F- 60 RTS
6B30- 00 BRK
6B31- 00 BRK
6B32- 00 BRK
6B33- 00 BRK
6B34- 00 BRK
6B35- 00 BRK
6B36- 00 BRK
6B37- 00 BRK
6B38- 00 BRK
6B39- 00 BRK
6B3A- 00 BRK
6B3B- 00 BRK
6B3C- 00 BRK
6B3D- 00 BRK
6B3E- 00 BRK
6B3F- 00 BRK
6B40- 00 BRK
6B41- 00 BRK
6B42- 00 BRK
6B43- 00 BRK
6B44- 00 BRK
6B45- 00 BRK
6B46- 00 BRK
6B47- 00 BRK
6B48- 00 BRK
6B49- 00 BRK
6B4A- 00 BRK
6B4B- 00 BRK
6B4C- 00 BRK

```

```

*****
ZONA DE DATOS RESERVADA PARA LOS COMENTARIOS A SALIR POR PANTA-
LLA DEPENDIENDO DEL CONTROL UTILIZADO
*****

```

```

6B4D- 8D A0 C5 STA $C5A0 ; DESDE LA 6B4D HASTA LA 6B66
6B50- C4 CE CPY $CE ; ESTA INSCRITA LA FRASE:
6B52- C1 D2 CMP ($D2,X) ; "BUFFER DEMASIADO GRANDE"
6B54- C7 ???
6B55- A0 CF LDY #$CF
6B57- C4 C1 CPY $C1
6B59- C9 D3 CMP #$D3
6B5B- C1 CD CMP ($CD,X)
6B5D- C5 C4 CMP $C4
6B5F- A0 D2 LDY #$D2
6B61- C5 C6 CMP $C6
6B63- C6 D5 DEC $D5
6B65- C2 ???
6B66- 8D 00 00 STA $0000
6B69- 00 BRK
6B6A- 8D BA D2 STA $D2BA ; DESDE LA 6B6A HASTA LA 6B89
6B6D- C1 C3 CMP ($C3,X) ; ESTA INSCRITA LA FRASE :
6B6F- C9 C6 CMP #$C6 ; "FICHERO A VERIFICAR"

```

6B71-	C9 D2	CMP	##D2	
6B73-	C5 D6	CMP	\$D6	
6B75-	A0 C1	LDY	##C1	
6B77-	A0 CF	LDY	##CF	
6B79-	D2	???		
6B7A-	C5 C8	CMP	\$C8	
6B7C-	C3	???		
6B7D-	C9 C6	CMP	##C6	
6B7F-	A0 C5	LDY	##C5	
6B81-	C4 A0	CPY	\$A0	
6B83-	C5 D2	CMP	\$D2	
6B85-	C2	???		
6B86-	CD CF CE	CMP	\$CECF	
6B89-	8D 00 00	STA	\$0000	
6B8C-	00	BRK		
6B8D-	8D BA D2	STA	\$D2BA	; DESDE LA 6B8D HASTA LA 6BA9
6B90-	C1 D4	CMP	(\$D4,X)	; ESTA INSCRITA LA FRASE :
6B92-	C9 C4	CMP	##C4	; "FICHERO A EDITAR"
6B94-	C5 A0	CMP	\$A0	
6B96-	C1 A0	CMP	(\$A0,X)	
6B98-	CF	???		
6B99-	D2	???		
6B9A-	C5 C8	CMP	\$C8	
6B9C-	C3	???		
6B9D-	C9 C6	CMP	##C6	
6B9F-	A0 C5	LDY	##C5	
6BA1-	C4 A0	CPY	\$A0	
6BA3-	C5 D2	CMP	\$D2	
6BA5-	C2	???		
6BA6-	CD CF CE	CMP	\$CECF	
6BA9-	8D 00 00	STA	\$0000	
6BAC-	00	BRK		
6BAD-	8D BA D2	STA	\$D2BA	; DESDE LA 6BAD HASTA LA 6BC7
6BB0-	C5 C5	CMP	\$C5	; ESTA INSCRITA LA FRASE :
6BB2-	CC A0 C1	CPY	\$C1A0	; "FICHERO A LEER"
6BB5-	A0 CF	LDY	##CF	
6BB7-	D2	???		
6BB8-	C5 C8	CMP	\$C8	
6BBA-	C3	???		
6BBB-	C9 C6	CMP	##C6	
6BBD-	A0 C5	LDY	##C5	
6BBF-	C4 A0	CPY	\$A0	
6BC1-	C5 D2	CMP	\$D2	
6BC3-	C2	???		
6BC4-	CD CF CE	CMP	\$CECF	
6BC7-	8D 00 00	STA	\$0000	
6BCA-	00	BRK		
6BCB-	00	BRK		
6BCC-	8D BA D2	STA	\$D2BA	; DESDE LA 6BCC HASTA LA 6BE7
6BCF-	C1 C5	CMP	(\$C5,X)	; ESTA INSCRITA LA FRASE :
6BD1-	D2	???		; "FICHERO A CREAR"
6BD2-	C3	???		
6BD3-	A0 C1	LDY	##C1	
6BD5-	A0 CF	LDY	##CF	
6BD7-	D2	???		

6BD8-	C5 C8	CMP	\$C8	
6BDA-	C3	???		
6BDB-	C9 C6	CMP	##C6	
6BDD-	A0 C5	LDY	##C5	
6BDF-	C4 A0	CPY	\$A0	
6BE1-	C5 D2	CMP	\$D2	
6BE3-	C2	???		
6BE4-	CD CF CE	CMP	\$CECF	
6BE7-	8D 00 00	STA	\$0000	
6BEA-	BA	TSX		; DESDE LA 6BEA HASTA LA 6BFB
6BEB-	CF	???		; ESTA INSCRITA LA FRASE :
6BEC-	C4 CE	CPY	\$CE	; "ENTRE EL COMANDO:"
6BEE-	C1 CD	CMP	(#CD,X)	
6BF0-	CF	???		
6BF1-	C3	???		
6BF2-	A0 CC	LDY	##CC	
6BF4-	C5 A0	CMP	\$A0	
6BF6-	C5 D2	CMP	\$D2	
6BF8-	D4	???		
6BF9-	CE C5 BD	DEC	\$8DC5	
6BFC-	00	BRK		
6BFD-	00	BRK		
6BFE-	00	BRK		
6BFF-	00	BRK		

ESTA SUBROUTINA VA SACANDO LOS CARACTERES QUE HAN SIDO CARGADOS
DESDE EL DISCO Y QUE ESTAN A PARTIR DE LA POSICION \$7000

6C00-	8E 50 10	STX	\$1050	; SE GUARDA EL VALOR DE "X"
6C03-	AE 60 10	LDX	\$1060	
6C06-	BD 00 70	LDA	\$7000,X	; SE VAN SACANDO LO QUE HAY A
6C09-	C9 D1	CMP	##D1	; PARTIR DE LA \$7000 HASTA QUE
6C0B-	F0 12	BEQ	\$6C1F	; ENCRONTREMOS UN "D1" O MARCA
6C0D-	EB	INX		; DE FINAL DE FICHERO
6C0E-	E0 00	CPX	##00	
6C10-	D0 28	BNE	\$6C3A	
6C12-	EE 08 6C	INC	\$6C0B	
6C15-	A9 96	LDA	##96	; SOLAMENTE SE PUEDEN GUARDAR
6C17-	CD 08 6C	CMP	\$6C0B	; POSICIONES DESDE LA \$7000
6C1A-	D0 1E	BNE	\$6C3A	; HASTA LA \$95FF
6C1C-	4C 50 6C	JMP	\$6C50	
6C1F-	A9 70	LDA	##70	
6C21-	8D 08 6C	STA	\$6C0B	; SE RESTABLECEN LOS REGISTROS
6C24-	A2 00	LDX	##00	; Y SE SALTA A LA \$6C50
6C26-	8E 60 10	STX	\$1060	
6C29-	AE 50 10	LDX	\$1050	
6C2C-	EA	NOP		
6C2D-	EA	NOP		
6C2E-	EA	NOP		
6C2F-	A9 00	LDA	##00	; SE RESETEA EL VALOR DEL
6C31-	8D 00 10	STA	\$1000	; FLAG Y DEL PUNTERO DE PAN-
6C34-	A9 A0	LDA	##A0	; TALLA
6C36-	85 27	STA	\$27	
6C38-	18	CLC		
6C39-	60	RTS		

```

6C3A-- 8E 60 10    STX    $1060    ; VA SACANDO LO QUE SE HA CAR-
6C3D-- AE 50 10    LDX    $1050    ; GADO EN EL BUFFER DESDE EL
6C40-- 85 27      STA    $27      ; DISCO Y LO MANDA A PANTALLA
6C42-- 20 02 67   JSR    $6702    ; Y HACIA FUERA
6C45-- 18         CLC
6C46-- 60         RTS
6C47-- 00         BRK
6C48-- 00         BRK
6C49-- 00         BRK
6C4E-- 00         BRK
6C4F-- 00         BRK
6C50-- EA         NOP          ; AQUI SE SALTA SOLO CUANDO
6C51-- EA         NOP          ; SE HA REBASADO LA CANTIDAD
6C52-- EA         NOP          ; DE MEMORIA ASIGNADA PARA
6C53-- EA         NOP          ; SACAR FICHEROS DESDE DISCO
6C54-- EA         NOP          ; Y SACARIA EN PANTALLA EL
6C55-- F0 11     BEQ    $6C68
6C57-- A0 1A     LDY    #1A
6C59-- B9 4D 6B   LDA    $6B4D,Y
6C5C-- 85 27     STA    $27
6C5E-- 98         TYA
6C5F-- 48         PHA          ; COMENTARIO:
6C60-- 20 A3 67   JSR    $67A3
6C63-- 68         PLA
6C64-- A8         TAY
6C65-- 88         DEY
6C66-- 10 F1     BPL    $6C59
6C68-- 60         RTS

```

PROGRAMA ENSAMBLADOR PROPIO DE LA TARJETA SSC.

 DE ESTE PROGRAMA SOLO SE UTILIZAN UNAS SUBROUTINAS MUY UTILES
 PARA EL DESARROLLO DEL APPLE COMO TERMINAL , POR ELLO PARA
 MAS INFORMACION REMITIRSE AL MANUAL DE USO DE LA SSC

```

6200- 2C 58 FF BIT $FF58 ; PONE LOS 5 FLAGS
6203- 70 0C BVS $6211
6205- 38 SEC ; ENTRADA BASIC
6206- 90 18 BCC $6220 ; SALIDA BASIC
6208- B8 CLV
6209- 50 06 BVC $6211
620B- 01 31 ORA ($31,X)
620D- 8E 94 97 STX $9794
6210- 9A TXS
6211- 85 27 STA $27
6213- 86 35 STX $35 ; BUFFER INDICE DE ENTRADA
6215- 8A TXA ; SALVA "X" E "Y" EN STACK
6216- 48 PHA
6217- 98 TYA
6218- 48 PHA
6219- 08 PHP ; SALVA FLAGS DE ENTRADA
621A- 78 SEI ; NO INTERRUPCION
621B- 8D FF CF STA $CFFF ; DESHABILITADO DE LA $C800
621E- 20 58 FF JSR $FF58
6221- BA TSX
6222- BD 00 01 LDA $0100,X ; RECOBRA $CN
6225- BD F8 07 STA $07F8
6228- AA TAX ; GENERALMENTE X=CN
6229- 0A ASL
622A- 0A ASL ; DETERMINA $N0
622B- 0A ASL
622C- 0A ASL
622D- 85 26 STA $26
622F- A8 TAY ; GENERALMENTE Y=N0
6230- 28 PLP
6231- 50 29 BVC $625C
  
```

*****INICIALIZACION BASIC*****

```

6233- 1E 38 05 ASL $0538,X ; SE HABILITA COMANDOS
6236- 5E 38 05 LSR $0538,X
6239- B9 8A C0 LDA $C08A,Y
623C- 29 1F AND #$1F
623E- D0 05 BNE $6245
6240- A9 EF LDA #$EF
6242- 20 05 63 JSR $6305
6245- E4 37 CPX $37
6247- D0 0B BNE $6254
6249- A9 07 LDA #$07
624B- C5 36 CMP $36
624D- F0 05 BEQ $6254
624F- 85 36 STA $36
6251- 18 CLC
6252- 90 08 BCC $625C
6254- E4 39 CPX $39
6256- D0 F9 BNE $6251
6258- A9 05 LDA #$05
625A- 85 38 STA $38
  
```

SALTO A LA APROPIADA RUTINA DE ENTRADA/SALIDA BASIC**

```

625C-  BD 38 07   LDA   $0738,X   ; SEPARA MODO CIC DE OTROS
625F-  29 02     AND   #$02     ; NO CERO PARA MODO CIC
6261-  08        PHP           ; SALVA MODO CIC
6262-  90 03     BCC   $6267
6264-  4C BF 63   JMP   $63BF
6267-  BD B8 04   LDA   $04B8,X   ; CHEQUEO ENTRADA CARACT.MINUS
626A-  48        PHA
626B-  0A        ASL
626C-  10 0E     BPL   $627C   ; SALTO SI NO
626E-  A6 35     LDX   $35
6270-  A5 27     LDA   $27
6272-  09 20     ORA   #$20
6274-  9D 00 02   STA   $0200,X   ; RESTAURA CARACT. MINUSC EN
6277-  B5 27     STA   $27     ; BUFFER Y SALIDA DE ECO
6279-  AE F8 07   LDX   $07F8
627C-  68        PLA
627D-  29 BF     AND   #$BF
627F-  9D B8 04   STA   $04B8,X
6282-  28        PLP
6283-  F0 06     BEQ   $628B   ; SALTO SI MODOS PPC O SIC
6285-  20 63 66   JSR   $6663   ; SALIDA DE MODO CIC
6288-  4C B5 63   JMP   $63B5   ; FINALIZA POR CHEQUEAR MODO
628B-  4C FC 63   JMP   $63FC   ; TERMINAL

```

ENTRADA DE INTERFACE EN PASCAL

```

628E-  20 00 63   JSR   $6300
6291-  A2 00     LDX   #$00     ; NINGUN POSIBLE ERROR
6293-  60        RTS
6294-  4C 9B 63   JMP   $639B
6297-  4C AA 64   JMP   $64AA
629A-  4A        LSR           ; SALVA TIPO REQUEST EN CARRY
629B-  20 9B 65   JSR   $659B
629E-  B0 08     BCS   $62AB
62A0-  20 F5 65   JSR   $65F5   ; PREPARADO PARA SALIDA?
62A3-  F0 06     BEQ   $62AB
62A5-  18        CLC
62A6-  90 03     BCC   $62AB   ; BORRA CARRY POR NO PREPARADO
62A8-  20 D2 65   JSR   $65D2   ; CORRIGE EL CARRY
62AB-  BD B8 05   LDA   $05B8,X   ; COGE LOS FLAGS DE ERROR
62AE-  AA        TAX
62AF-  60        RTS

```

RUTINA PARA ENVIAR UN CARACTER A UNA TARJETA

```

62B0-  A2 03     LDX   #$03
62B2-  B5 36     LDA   $36,X
62B4-  48        PHA
62B5-  CA        DEX
62B6-  10 FA     BPL   $62B2

```

*****PONE LA DIRECCION DE LA TARJETA EN ENGANCHE*****

62B8-	AE F8 07	LDX	\$07F8	
62BB-	BD 38 06	LDA	\$0638, X	
62BE-	85 36	STA	\$36	
62C0-	BD B8 04	LDA	\$04B8, X	; OBTIENE NUMERO DE SLOT
62C3-	29 38	AND	#\$38	
62C5-	4A	LSR		
62C6-	4A	LSR		
62C7-	4A	LSR		
62C8-	09 C0	ORA	#\$C0	; FORMA \$CN
62CA-	85 37	STA	\$37	

*****SALIDA AL PERIFERICO*****

62CC-	8A	TXA		; SALVA \$CN
62CD-	48	PHA		
62CE-	A5 27	LDA	\$27	
62D0-	48	PHA		
62D1-	09 80	ORA	#\$80	
62D3-	20 ED FD	JSR	\$FDED	

*****RESTAURACION DE LAS OTRAS TARJETAS*****

62D6-	68	PLA		
62D7-	85 27	STA	\$27	
62D9-	68	PLA		
62DA-	8D F8 07	STA	\$07F8	
62DD-	AA	TAX		
62DE-	0A	ASL		
62DF-	0A	ASL		
62E0-	0A	ASL		
62E1-	0A	ASL		
62E2-	85 26	STA	\$26	
62E4-	8D FF CF	STA	\$CFFF	
62E7-	A5 36	LDA	\$36	
62E9-	9D 38 06	STA	\$0638, X	
62EC-	A2 00	LDX	#\$00	
62EE-	68	PLA		
62EF-	95 36	STA	\$36, X	
62F1-	E8	INX		
62F2-	E0 04	CPX	#\$04	
62F4-	90 F8	BCC	\$62EE	
62F6-	AE F8 07	LDX	\$07F8	
62F9-	60	RTS		
62FA-	C1 D0	CMP	(D0, X)	; ASC "APPLE"
62FC-	D0 CC	BNE	\$62CA	
62FE-	C5 08	CMP	\$08	

 INICIALIZACION DEL PASCAL 1.0

6300-	20 9B 64	JSR	\$649B	; INICIALIZACION PASCAL 1.0
6303-	A9 16	LDA	#\$16	
6305-	48	PHA		
6306-	A9 00	LDA	#\$00	

```

630B- 9D B8 04 STA $04B8,X
630B- 9D B8 03 STA $03B8,X
630E- 9D 38 04 STA $0438,X
6311- 9D B8 05 STA $05B8,X
6314- 9D 38 06 STA $0638,X
6317- 9D B8 06 STA $06B8,X
631A- B9 B2 C0 LDA $C0B2,Y
631D- 85 2B STA $2B
631F- 4A LSR
6320- 4A LSR
6321- 90 04 BCC $6327
6323- 68 PLA
6324- 29 FE AND $$FE
6326- 48 PHA
6327- B8 CLV ; SE BORRA OVERFLOW PARA MO-
6328- B9 B1 C0 LDA $C0B1,Y ; DO CIC
632B- 4A LSR
632C- B0 07 BCS $6335 ; SALTO SI MODO SIC
632E- 4A LSR
632F- B0 0E BCS $633F
6331- A9 01 LDA $$01 ; CTRL-A
6333- D0 3D BNE $6372
6335- 4A LSR
6336- A9 03 LDA $$03
6338- B0 02 BCS $633C ; SALTO PARA PBA
633A- A9 B0 LDA $$B0
633C- 9D B8 04 STA $04B8,X
633F- 2C 58 FF BIT $FF58
6342- A5 2B LDA $2B
6344- 29 20 AND $$20
6346- 49 20 EOR $$20
6348- 9D B8 03 STA $03B8,X
634B- 70 0A BVS $6357

```

 ENTRADA DE LECTURA DEL PASCAL 1.0

```

634D- 20 9B 63 JSR $639B
6350- AE F8 07 LDX $07F8
6353- 9D B8 05 STA $05B8,X
6356- 60 RTS

```

 DONDE ESTAMOS AHORA?

```

6357- A5 2B LDA $2B ; USO DE LOS SWITCHES PARA
6359- 4A LSR ; PONER ANCHO Y DELAY DE CR
635A- 4A LSR
635B- 29 03 AND $$03
635D- A8 TAY
635E- F0 04 BEQ $6364
6360- 68 PLA ; RESET VIDEO PARA 40
6361- 29 7F AND $$7F ; COLUMNAS
6363- 48 PHA
6364- B9 A6 64 LDA $64A6,Y
6367- 9D 38 06 STA $0638,X
636A- A4 26 LDY $26
636C- 68 PLA
636D- 29 95 AND $$95

```

63BE-- 60 RTS

RUTINA DE ENTRADA BASIC

```
63BF-- F0 29      BEQ  $63EA    ; SALTO SI NO MODO CIC
63C1-- BD B8 06   LDA  $06B8,X ; ENTRADA DE BUFFER LLENA?
63C4-- 10 05      BPL  $63CB
63C6-- 5E B8 06   LSR  $06B8,X ; RESETEA DE BUFFER LLENO
63C9-- D0 24      BNE  $63EF
63CB-- 20 3E 67   JSR  $673E    ; DATOS EN TECLADO?
63CE-- 90 1A      BCC  $63EA
63D0-- BD B8 03   LDA  $03B8,X
63D3-- 29 C0      AND  #$C0     ; TRASLADA MINUS. A MAYUSC.
63D5-- F0 0E      BEQ  $63E5
63D7-- A5 27      LDA  $27
63D9-- C9 E0      CMP  #$E0
63DB-- 90 08      BCC  $63E5
63DD-- BD B8 04   LDA  $04B8,X
63E0-- 09 40      ORA  #$40
63E2-- 9D B8 04   STA  $04B8,X
63E5-- 28         PLP
63E6-- F0 D0      BEQ  $63B8    ; SALTO SI NO MODO CIC
63E8-- D0 CB      BNE  $63B5    ; CHEQUEO ENTRADA A MODO TERMINAL
63EA-- 20 FF 65   JSR  $65FF    ; DATOS EN ACIA?
63ED-- 90 DC      BCC  $63CB
63EF-- 20 11 67   JSR  $6711
63F2-- 28         PLP
63F3-- 08         PHP
63F4-- F0 DA      BEQ  $63D0
63F6-- 20 D1 64   JSR  $64D1    ; CARACTERES ESPECIALES?
63F9-- 4C D0 63   JMP  $63D0
```

RUTINA DE SALIDA BASIC

```
63FC-- 20 1A 66   JSR  $661A    ; CHEQUEO MODO DE COMANDO
63FF-- B0 B7      BCS  $63B8    ; SALTO SI MODO COMANDO
6401-- A5 27      LDA  $27     ; SALVA CHARAC. DE STACK
6403-- 48         PHA
6404-- BD 38 07   LDA  $0738,X
6407-- 29 C0      AND  #$C0
6409-- D0 16      BNE  $6421
640B-- A5 24      LDA  $24     ; CHEQUEO TABULADO DE COMA
640D-- F0 42      BEQ  $6451
640F-- C9 08      CMP  #$08
6411-- F0 04      BEQ  $6417
6413-- C9 10      CMP  #$10
6415-- D0 0A      BNE  $6421
6417-- 09 F0      ORA  #$F0
6419-- 3D B8 06   AND  $06B8,X
641C-- 18         CLC
641D-- 65 24      ADC  $24
641F-- 85 24      STA  $24
6421-- BD B8 06   LDA  $06B8,X
6424-- C5 24      CMP  $24
6426-- F0 29      BEQ  $6451
6428-- A9 A0      LDA  #$A0
```

```

642A- 90 08      BCC  $6434
642C- BD 38 07   LDA  $0738,X
642F- 0A         ASL
6430- 10 1F      BPL  $6451
6432- A9 88      LDA  ##88
6434- 85 27      STA  $27
6436- 2C 58 FF   BIT  $FF58      ; PONE V=1 INDICANDO TABULAC
6439- 08         PHP
643A- 70 0C      BVS  $6448
643C- EA         NOP

```

CORTO MOVIMIENTO DE BATCH

```

643D- 2C 58 FF   BIT  $FF58
6440- 50 B8      BVC  $63FA
6442- AE F8 07   LDX  $07F8
6445- 4C EF 64   JMP  $64EF
6448- 20 B5 64   JSR  $64B5      ; AJUSTE CUENTA DE COLUMNAS
644B- 20 6B 66   JSR  $666B
644E- 4C 6B 64   JMP  $646B
6451- 68         PLA
6452- B8         CLV
6453- 08         PHP      ; SALVA INDICACION "NO TAB"
6454- 85 27      STA  $27
6456- 48         PHA
6457- 20 6B 66   JSR  $666B      ; ENTRA DESPUES DE CHEQUEO
645A- 20 B5 64   JSR  $64B5      ; DE MODO COMANDO
645D- 68         PLA
645E- 49 8D      EOR  ##8D      ; FUE CR?
6460- 0A         ASL
6461- D0 05      BNE  $646B
6463- 9D B8 06   STA  $06B8,X   ; SI, PONE COLUMNAS A CERO
6466- 85 24      STA  $24
6468- BD B8 04   LDA  $04B8,X   ; FUERZA CR DESHABILITADO
646B- 10 0D      BPL  $647A
646D- BD 38 06   LDA  $0638,X
6470- F0 08      BEQ  $647A
6472- 18         CLC
6473- FD B8 06   SBC  $06B8,X
6476- A9 8D      LDA  ##8D
6478- 90 DA      BCC  $6454      ; SALTO PARA FORZAR CR
647A- 28         PLP
647B- 70 A4      BVS  $6421      ; SALTO SI TABULADO
647D- BD 38 07   LDA  $0738,X
6480- 30 16      BMI  $649B
6482- BC B8 06   LDY  $06B8,X
6485- 0A         ASL
6486- 30 0E      BMI  $6496      ; PONE CH AL VALOR DE LA
6488- 98         TYA      ; COLUMNA PARA TABULADO
6489- A0 00      LDY  ##00
648B- 38         SEC
648C- FD 38 06   SBC  $0638,X
648F- C9 F8      CMP  ##F8
6491- 90 03      BCC  $6496
6493- 69 27      ADC  ##27
6495- AB         TAY

```

```

6496- 84 24 STY $24
6498- 4C B8 63 JMP $63B8 ; SALIDA DE BASIC
*****
RUTINA DE ENTRADA PASCAL
*****
649B- 8E F8 07 STX $07F8
649E- 84 26 STY $26
64A0- A9 00 LDA #$00
64A2- 9D B8 05 STA $05B8,X
64A5- 60 RTS
*****
TABULA EL ANCHO PARA MODO DE IMPRESION SIC
*****
64A6- 29 48 AND #$48 ; 40,72 COLUMNAS
64A8- 50 84 BVC $642E ; B0,132 COLUMNAS
*****
RUTINA DE ESCRITURA EN PASCAL
*****
64AA- 85 27 STA $27
64AC- 20 9B 64 JSR $649B
64AF- 20 63 66 JSR $6663
64B2- 4C A3 63 JMP $63A3
*****
RUTINA DE AJUSTE DE COLUMNAS (SOLO PARA MODOS SIC Y PPC)
*****
64B5- A5 27 LDA $27
64B7- 49 08 EOR #$08 ; BACKSPACE?
64B9- 0A ASL
64BA- F0 04 BEQ $64C0 ; SI, DECREMENTA COLUMNAS
64BC- 49 EE EOR #$EE ; DELETE?
64BE- D0 09 BNE $64C9
64C0- DE B8 06 DEC $06B8,X ; DECREMENTA CUENTA DE COLUM.
64C3- 10 03 BPL $64C8
64C5- 9D B8 06 STA $06B8,X ; NO PERMITE BAJAR DE CERO
64C8- 60 RTS
64C9- C9 C0 CMP #$C0 ; NO INCREMENTA CUENTA DE CO-
64CB- B0 FB BCS $64CB ; LUMNAS PARA CARAC. CONTROL
64CD- FE B8 06 INC $06B8,X
64D0- 60 RTS
*****
RUTINA DE PROCESO DE CARACTERES ESPECIALES DE ENTRADA
*****
64D1- BD 38 07 LDA $0738,X ; HABILITADA ENTRADA DE
64D4- 29 08 AND #$08 ; CARACTERES DE CONTROL?
64D6- F0 16 BEQ $64EE
64D8- BD B8 04 LDA $04B8,X
64DB- A4 27 LDY $27
64DD- C0 94 CPY #$94 ; CTRL-T?
64DF- D0 04 BNE $64E5
64E1- 09 80 ORA #$80 ; PONE MODO TERMINAL
64E3- D0 06 BNE $64EB
64E5- C0 92 CPY #$92 ; CTRL-R?
64E7- D0 05 BNE $64EE
64E9- 29 7F AND #$7F ; RESETEA MODO TERMINAL
64EB- 9D B8 04 STA $04B8,X
64EE- 60 RTS
*

```

CORTO MOVIMIENTO DE BLOQUE

```
64EF- 8A TXA
64F0- 0A ASL
64F1- 0A ASL
64F2- 0A ASL
64F3- 0A ASL
64F4- 85 26 STA $26
64F6- A9 00 LDA #00
64F8- 9D B8 05 STA $05B8,X ; INDICACION DE ERROR CERO
64FB- 70 0F BVS $650C
64FD- A0 00 LDY #00
64FF- B1 3C LDA ($3C),Y ; OBTIENE BUFFER DE DATOS
6501- 85 27 STA $27
6503- 20 02 67 JSR $6702 ; LOS ENVIA HACIA LA ACIA
6506- 20 BA FC JSR $FCBA
6509- 90 F2 BCC $64FD
650B- 60 RTS
650C- 20 D2 65 JSR $65D2
650F- 90 FB BCC $650C
6511- B9 88 C0 LDA $C088,Y
6514- A0 00 LDY #00
6516- 91 3C STA ($3C),Y ; PONE DATOS ACIA EN BUFFER
6518- 20 BA FC JSR $FCBA
651B- 90 EF BCC $650C
651D- 60 RTS
```


RUTINAS DE MODO TERMINAL


```
651E- BD B8 04 LDA $04B8,X ; ESTAMOS EN MODO TERMINAL?
6521- 10 31 BPL $6554 ; SI NO, VOLVER
```

*****ESTAMOS DENTRO DEL MODO TERMINAL*****

```
6523- A9 02 LDA #02 ; COMIENZO EN MODO MAYUSC.
6525- 48 PHA
6526- A9 7F LDA #7F
6528- 20 E2 68 JSR $68E2 ; RESETEO DE ECO
652B- A4 24 LDY $24
652D- B1 28 LDA ($28),Y
652F- 85 27 STA $27 ; SALVA CARAC.DE PANTALLA
6531- A9 07 LDA #07 ; FORMA UN CURSOR TESTE-
6533- 25 4F AND $4F ; LLEANTE
6535- D0 10 BNE $6547
6537- A4 24 LDY $24
6539- A9 DF LDA #DF
653B- D1 28 CMP ($28),Y ; ES UNDERLINE DE PANTALLA?
653D- D0 02 BNE $6541 ; SI NO, PONLO
653F- A5 27 LDA $27 ; DE OTRA MANERA USA CARAC.
6541- 91 28 STA ($28),Y ; DE PANTALLA VERDADERO
6543- E6 4F INC $4F ; TESTELLEA NI RAPIDO, NI
6545- E6 4F INC $4F ; LENTO
6547- BD B8 04 LDA $04B8,X ; TODAVIA EN MODO TERMINAL?
```


654A-	30 09	BMI	\$6555	; SI, LUEGO CHEQUEAR ACIA
654C-	20 11 67	JSR	\$6711	; REEMPLAZA NUESTRO CURSOR
654F-	68	PLA		; LIMPIA EL STACK
6550-	A9 8D	LDA	##8D	; RETORNA CON CR
6552-	85 27	STA	\$27	
6554-	60	RTS		
6555-	20 F4 60	JSR	\$60F4	; ENTRADA ACIA ?(ESTA ENTRA-
				; DA YA TIENE EL PROTOCOLO
				; ENQUIRY/ACKNOWLEDGE
6558-	90 0C	BCC	\$6566	; SI NO , CHEQUEA TECLADO
655A-	20 11 67	JSR	\$6711	; RESTAURA CURSOR, CARAC. ENTRADA
655D-	20 D1 64	JSR	\$64D1	; CHEQUEO DE CTRL-T CTRL-R
6560-	20 A3 67	JSR	\$67A3	; ENTRADA A PANTALLA
6563-	4C 2B 65	JMP	\$652B	
6566-	20 3E 67	JSR	\$673E	; TECLA PRESIONADA?
6569-	90 C6	BCC	\$6531	; SALTO SI NO
656B-	70 BE	EVS	\$652B	; SALTO SI SECUENCIA DE ESCAPE
656D-	BD 38 07	LDA	\$073B, X	; SHIFT?
6570-	0A	ASL		
6571-	10 22	BPL	\$6595	
6573-	68	PLA		
6574-	A8	TAY		
6575-	A5 27	LDA	\$27	
6577-	C0 01	CPY	##01	; 1=LETRAS SHIFTEADAS
6579-	F0 20	BEQ	\$659B	
657B-	B0 34	BCS	\$65B1	; 2=MODO MAYUSCULAS
657D-	C9 9B	CMP	##9B	; ESC?
657F-	D0 06	BNE	\$6587	
6581-	C8	INY		; INCREMENTA ESTADO
6582-	98	TYA		
6583-	48	PHA		; VUELVE ATRAS EN STACK
6584-	4C 2B 65	JMP	\$652B	
6587-	C9 C1	CMP	##C1	; <A?
6589-	90 08	BCC	\$6593	
658B-	C9 DB	CMP	##DB	; >Z?
658D-	B0 04	BCS	\$6593	
658F-	09 20	ORA	##20	; ES LETRA A TRASLADAR A LC
6591-	85 27	STA	\$27	
6593	98	TYA		
6594-	48	PHA		; PONE ESTADO ATRAS DEL STACK
6595-	20 68 66	JSR	\$6668	; VA FUERA
6598-	4C 2B 65	JMP	\$652B	
659B-	C9 9B	CMP	##9B	; DOS ESCAPES?
659D-	F0 E2	BEQ	\$65B1	
659F-	C9 B0	CMP	##B0	; <0?
65A1-	90 0A	BCC	\$65AD	
65A3-	C9 BB	CMP	##BB	; >PUNTO?
65A5-	B0 06	BCS	\$65AD	

*****ESCAPE(NUMERO) PARA TRASLADAR A CARACTER ASCII*****

65A7-	A8	TAY		
65A8-	B9 09 65	LDA	\$6509, Y	
65AB-	85 27	STA	\$27	
65AD-	A0 00	LDY	##00	; VUELVE A ESTADO CERO
65AF-	F0 E2	BEQ	\$6593	

```

65B1-   C9 9B      CMP   ##9B      ; ESC?
65B3-   D0 DE      BNE   $6593
65B5-   A0 00      LDY   ##00
65B7-   F0 C9      BEQ   $6582

```

TABLA DE TRASLACION

```

65B9-   9B      ???      ; ESC
65BA-   9C      ???      ; FS
65BB-   9F      ???      ; US
65BC-   DB      ???      ;
65BD-   DC      ???      ; SLASH IZQUIERDO
65BE-   DF      ???      ;
65BF-   FB      ???      ;
65C0-   FC      ???      ; BARRA VERTICAL
65C1-   FD FE FF SBC   $FFFE,X ; TILDE, RUBOUT

```

SUBROUTINAS DE CUIDADO

RUTINA DE ESPERA DE PROPOSITO GENERAL

```

65C4-   A2 CA      LDX   ##CA
65C6-   CA      DEX
65C7-   D0 FD      BNE   $65C6      ; LAZO DE 5 MICROSEGUNDOS
65C9-   3B      SEC
65CA-   E9 01      SBC   ##01
65CC-   D0      BNE   $65C4
65CE-   AE FB 07   LDX   $07F8
65D1-   60      RTS

```

RUTINAS DE LECTURA DEL STATUS REGISTER DE LA ACIA

*****CHEQUEO DE STATUS DE ENTRADA DE LA ACIA*****

```

65D2-   A4 26      LDY   $26      ; Y=$N0
65D4-   B9 89 C0   LDA   $C0B9,Y
65D7-   4B      PHA
65D8-   29 20      AND   ##20      ; DCD?
65DA-   4A      LSR      ; ERROR SI NO
65DB-   4A      LSR
65DC-   85 35      STA   $35
65DE-   68      PLA
65DF-   29 0F      AND   ##0F
65E1-   C9 08      CMP   ##08      ; PONE CARRY SI RDR ESTA LLENO
65E3-   90 04      BCC   $65E9
65E5-   29 07      AND   ##07
65E7-   B0 02      BCS   $65EB
65E9-   A5 35      LDA   $35
65EB-   05 35      ORA   $35      ; OBTIENE BIT DE ERROR DCD
65ED-   F0 05      BEQ   $65F4      ; SALTO SI NO ENCUENTRA ERROR
65EF-   09 20      ORA   ##20
65F1-   9D B8 05   STA   $05B8,X

```

65F4- 60 RTS ; CY=1 IMPLICA DATO DISPONIBLE

***** CHEQUEO PARA SALIDA POR LA ACIA *****

65F5- A4 26 LDY \$26
65F7- B9 B9 C0 LDA \$C0B9,Y
65FA- 29 70 AND #\$70
65FC- C9 10 CMP #\$10
65FE- 60 RTS

RUTINA DE ENTRADA GENERAL(TIENE ARREGLOS EN LA 60F4)

65FF- 20 D2 65 JSR \$65D2
6602- 90 15 BCC \$6619
6604- B9 B8 C0 LDA \$C0B8,Y ; OBTIENE ENTRADA ACIA
6607- 09 80 ORA #\$80 ; PONE ALTO BIT PARA BASIC
6609- C9 8A CMP #\$8A ; ES LINEFEED?
660B- D0 09 BNE \$16
660D- A8 TAY
660E- BD 38 07 LDA \$0738,X ; PODEMOS ENTRARLO?
6611- 29 20 AND #\$20
6613- D0 03 BNE \$6618
6615- 98 TYA
6616- 38 SEC ; INDICA DATO
6617- 60 RTS
6618- 18 CLC ; BORRA CARRY PARA NO ENTRADA
6619- 60 RTS

RUTINA DE SALIDA GENERAL

*****COMIENZO DE RUTINA DE CHEQUEO DE COMANDO*****

661A- A4 26 LDY \$26
661C- B9 B1 C0 LDA \$C0B1,Y
661F- 4A LSR
6620- B0 36 BCS \$6658
6622- BD B8 04 LDA \$04B8,X
6625- 29 07 AND #\$07 ; ESTAMOS EN MODO COMANDO?
6627- F0 05 BEQ \$662E
6629- 20 FC 68 JSR \$68FC ; SI , IR A COMANDO CENTRAL
662C- 38 SEC ; INDICA COMANDO
662D- 60 RTS
662E- A5 27 LDA \$27
6630- 29 7F AND #\$7F ; IGNORA EL BIT ALTO
6632- DD 38 05 CMP \$0538,X ; COMIENZO DE MODO COMANDO?
6635- D0 05 BNE \$663C
6637- FE B8 04 INC \$04B8,X ; SALIDAS DE MODO COMANDO
663A- 38 SEC ; INDICA COMANDO
663B- 60 RTS
663C- BD 38 07 LDA \$0738,X ; ESTA XON HABILITADO
663F- 29 08 AND #\$08
6641- F0 15 BEQ \$6658 ; SALTO SI NO
6643- 20 FF 65 JSR \$65FF ; CUALQUIER ENTRADA?
6646- 90 10 BCC \$6658 ; SI NO VA A SALIDA
6648- C9 93 CMP #\$93 ; ES OFF?

```

664A- F0 0E BEQ $665A ; SI ES, ESPERA OTRA ENTRADA
664C- 48 PHA
664D- BD 38 07 LDA $0738,X ; MODO CIC
6650- 4A LSR
6651- 4A LSR
6652- 68 PLA
6653- 90 04 BCC $6659
6655- 9D B8 06 STA $06B8,X ; SI ES ASI, TENEMOS UN BUFFER
6658- 18 CLC ; INDICA NO MODO COMANDO
6659- 60 RTS
665A- 20 AA C8 JSR $CBAA ; OBTIENE DATO ACIA/TECLADO
665D- C9 91 CMP ##91 ; ES XON
665F- D0 F9 BNE $665A ; SI NO, ESPERA
6661- 18 CLC ; DE OTRA MANERA INDICA NO
6662- 60 RTS ; MODO COMANDO Y RETORNA

```

RUTINA ESPERADA

```

6663- 20 1A CB JSR $CB1A
6666- B0 F1 BCS $6659 ; NO SALIDA DE MODO COMANDO
6668- 20 9E CC JSR $CC9E
666B- A4 26 LDY $26
666D- B9 B1 C0 LDA $C0B1,Y
6670- 4A LSR
6671- 90 4E BCC $66C1 ; SALTO ETX/ACK
6673- 4A LSR
6674- 90 4B BCC $66C1 ; SALTO SI NO EMULACION PBA

```

MATERIA ETX/ACK PBA

```

6676- A5 27 LDA $27 ; SALVA CHAR. EN STACK
6678- 48 PHA
6679- BD 38 04 LDA $0438,X ; CARACT. DE CUENTA PARA
667C- C9 67 CMP ##67 ; BUFFER LLENO
667E- 90 10 BCC $6690
6680- C9 6C CMP ##6C ; SI <103 ENTONCES 153 CA-
6682- B0 22 BCS $66A6 ; RACTERES EN BUFFER
6684- C9 6B CMP ##6B ; SI >=108 ENTONCES MENOS
6686- 68 PLA ; DE 149 CARACTERES.
6687- 48 PHA
6688- 49 9B EOR ##9B ; ESC?
668A- 29 7F AND ##7F
668C- D0 18 BNE $66A6
668E- B0 19 BCS $66A9
6690- BD B8 04 LDA $04B8,X
6693- 29 1F AND ##1F
6695- 09 80 ORA ##80
6697- 85 27 STA $27
6699- 20 02 67 JSR $6702
669C- 20 AA C8 JSR $CBAA ; OBTIENE DATO ACIA/TECLADO
669F- 49 86 EOR ##86 ; ACK?
66A1- D0 ED BNE $6690 ; SI NO REPITE HANDSHAKE
66A3- 9D 38 04 STA $0438,X ; INICIA CUENTA A 255
66A6- DE 38 04 DEC $0438,X

```

```

66A9- 68          PLA          ; CARACT. REALES A SALIDA
66AA- 85 27      STA          $27
66AC- 49 8D      EOR          ##8D
66AE- 0A          ASL
66AF- D0 0A      BNE          $66BB
66B1- BD B8 03   LDA          $03B8,X
66B4- 29 30      AND          ##30
66B6- F0 03      BEQ          $66BB
66B8- 9D 38 04   STA          $0438,X
66BB- 20 02 67   JSR          $6702
66BE- 20 EA 66   JSR          $66EA

```

Y ATRAS A SALIDA NORMAL

```

66C1- 20 02 67   JSR          $6702          ; SALIDA DEL CARACTER

```

***** CHEQUEO DE DELAYS DE CR, LF, Y FF *****

```

66C4- 0A          ASL
66C5- AB          TAY
66C6- BD B8 03   LDA          $03B8,X          ; OBTIENE FLAGS DE DELAY
66C9- C0 18      CPY          ##18          ; FORM FEED?
66CB- F0 0C      BEQ          $66D9
66CD- 4A          LSR
66CE- 4A          LSR
66CF- C0 14      CPY          ##14          ; LINE FEED?
66D1- F0 06      BEQ          $66D9
66D3- 4A          LSR
66D4- 4A          LSR
66D5- C0 1A      CPY          ##1A          ; RETORNO DE CARRO?
66D7- D0 25      BNE          $66FE
66D9- 29 03      AND          ##03
66DB- F0 0D      BEQ          $66EA          ; NO SE INDICA DELAY
66DD- AB          TAY
66DE- B9 FE 66   LDA          $66FE,Y
66E1- AB          TAY          ; DELAY INCREM. DE 32 MSEG
66E2- A9 20      LDA          ##20
66E4- 20 C4 65   JSR          $65C4
66E7- 8B          DEY
66E8- D0 F8      BNE          $66E2

```

*****CHEQUEO DE GENERACION DE LINE FEED *****

```

66EA- A5 27      LDA          $27
66EC- 0A          ASL
66ED- C9 1A      CMP          ##1A          ; RETORNO DE CARRO?
66EF- D0 0D      BNE          $66FE
66F1- BD 38 07   LDA          $0738,X          ; HABILITADO GENERADOR LF?
66F4- 6A          ROR
66F5- 90 07      BCC          $66FE
66F7- A9 8A      LDA          ##8A
66F9- 85 27      STA          $27          ; LINE FEED?
66FB- 4C 6B 66   JMP          $666B

```

```

66FE- 60 RTS
66FF- 01 08 ORA ($08,X) ; 32 MSEG,1/4 SEG
6701- 40 RTI ; 2 SEG

```

RUTINA DE SALIDA ACIA

```

6702- 20 F5 65 JSR $65F5 ; PREPARADO PARA SALIR?
6705- D0 FB BNE $6702
6707- 98 TYA
6708- 09 89 ORA ##89 ; PREPARA PARA DIRECCIONAR
670A- A8 TAY ; ACIA
670B- A5 27 LDA $27
670D- 99 FF BF STA $BFFF,Y ; AQUI ESTA TU ACIA
6710- 60 RTS

```

RESTAURACION DE CURSOR(REGISTRO A CONTIENE EL NUEVO CARACTER)

```

6711- 48 PHA ; SALVA NUEVO CARACTER
6712- A4 24 LDY $24
6714- A5 27 LDA $27 ; VIEJO CARACTER
6716- 91 28 STA ($28),Y
6718- 68 PLA
6719- C9 95 CMP ##95 ; PANTALLA LLENA?
671B- D0 0C BNE $6729
671D- A5 27 LDA $27
671F- C9 20 CMP ##20 ; INVERSO?
6721- B0 06 BCS $6729
6723- 20 DF 67 JSR $67DF
6726- 59 DB 67 EOR $67DB,Y
6729- 85 27 STA $27
672B- 60 RTS

```

BUSQUEDA DE TECLADO PASCAL-BASIC

```

672C- 18 CLC ; BORRA CARRY PARA NO DATO
672D- BD 38 07 LDA $0738,X
6730- 29 04 AND ##04 ; NINGUNA RESPUESTA SI TECLA-
6732- F0 09 BEQ $673D ; DO ESTA DESHABILITADO
6734- AD 00 C0 LDA $C000
6737- 10 04 BPL $673D
6739- 4C 2E 61 JMP $613A
673C- 38 SEC ; INDICA DATO
673D- 60 RTS

```

OBTIENE CARACTER DE TECLADO SOLO PARA BASIC

```

673E- E6 4E INC $4E
6740- D0 02 BNE $6744
6742- E6 4F INC $4F
6744- 20 2C 67 JSR $672C ; RUTINA DE BUSQUEDA DE TECLADO
6747- B8 CLV
6748- 90 F3 BCC $673D ; EXIT SI NO SE PRESIONA TECLA
674A- 20 11 67 JSR $6711

```

```

674D- 29 7F AND ##7F
674F- DD 38 05 CMP $0538,X ; ES COMIENZO DE COMANDO?
6752- D0 3D BNE $6791 ; SI NO SALIDA INDICANDO DATO
6754- A4 26 LDY $26
6756- B9 81 C0 LDA $C081,Y
6759- 4A LSR
675A- B0 35 BCS $6791

```

MANEJADOR DE ESCAPE DE TECLADO

```

675C- A0 11 LDY ##11
675E- B9 EA 6B LDA $6BEA,Y
6761- 85 27 STA $27
6763- 98 TYA
6764- 48 PHA
6765- 20 A3 67 JSR $67A3 ; SIEMPRE ENVIA A PANTALLA
6768- 68 PLA
6769- A8 TAY
676A- 88 DEY
676B- 10 F1 BPL $675E
676D- A9 01 LDA ##01 ; COMIENZA SALIDA EN ESTADO 1
676F- 20 7B 69 JSR $697B
6772- 20 34 C0 JSR $C034 ; ESPERA CARACTER DE TECLADO
6775- 10 FB BPL $6772
6777- C9 88 CMP ##88 ; BACKSPACE?
6779- F0 E1 BEQ $675C ; SI, LUEGO EMPEZAR DE NUEVO
677B- 85 27 STA $27
677D- 20 A3 67 JSR $67A3
6780- 20 1A 66 JSR $661A ; AL INTERPRETADOR DE COMANDO
6783- BD B8 04 LDA $04B8,X ; LO HEMOS HECHO?
6786- 29 07 AND ##07
6788- D0 EB BNE $6772 ; SI NO VOLVER
678A- A9 8D LDA ##8D ; FUERZA UN CR
678C- 85 27 STA $27
678E- 2C 58 FF BIT $FF58 ; INDICA QUE HA OCURRIDO
6791- 38 SEC ; SECUENCIA DE COMANDO
6792- 60 RTS
6793- BA TSX ; ASC" : CSS ERTNE"
6794- C3 ???
6795- D3 ???
6796- D3 ???
6797- A0 C5 LDY ##C5
6799- D2 ???
679A- D4 ???
679B- CE C5 8D DEC $8DC5

```

RUTINA PARA IMPRIMIR UN CARACTER EN EL DISPLAY SELECCIONADO

```

679E- BD 38 07 LDA $0738,X
67A1- 10 13 BPL $67B6 ; SI PANTALLA DESHABILITADA
67A3- BD 38 07 LDA $0738,X ; ENTRA DESPUES CHEQUEO ECO
67A6- 29 02 AND ##02
67A8- F0 0D BEQ $67B7 ; SIEMPRE USA LA PANTALLA
67AA- BD B8 04 LDA $04B8,X ; PANTALLA ES DE APPLE?
67AD- 29 38 AND ##38
67AF- F0 06 BEQ $67B7 ; SLOT 0=PANTALLA APPLE

```

```

67B1- 8A TXA
67B2- 48 PHA
67B3- A9 AF LDA #$AF ; PARA VECTORIZAR AL PERI-
67B5- 48 PHA ; FERICO EN SLOT ESCOGIDO
67B6- 60 RTS

```

***** DRIVER APPLE DE 40 COLUMNAS *****

```

67B7- 20 DF 67 JSR $67DF
67BA- 09 B0 ORA #$B0
67BC- C9 E0 CMP #$E0 ; MINUSCULAS?
67BE- 90 06 BCC $67C6
67C0- 59 D3 67 EOR $67D3,Y ; HACER SOLTAR MINUSCULAS
67C3- 4C F6 FD JMP $FDF6 ; SE GUARDAN LOS REGISTROS

```

***** SI SON MAYUSCULAS SOLO SE MAPEAN LETRAS*****

```

67C6- C9 C1 CMP #$C1 ; <A?
67C8- 90 F9 BCC $67C3
67CA- C9 DB CMP #$DB ; >Z?
67CC- B0 F5 BCS $67C3
67CE- 59 D7 67 EOR $67D7,Y
67D1- 90 F0 BCC $67C3

```

***** MASCARAS PARA CASO DE TRASLACION *****

```

67D3- 20 00 E0 JSR $E000
67D6- 20 00 00 JSR $0000
67D9- 00 BRK
67DA- C0 00 CPY #$00
67DC- 00 BRK
67DD- E0 C0 CPX #$C0
67DF- BD B8 03 LDA $03B8,X
67E2- 2A ROL
67E3- 2A ROL
67E4- 2A ROL
67E5- 29 03 AND #$03
67E7- A8 TAY
67E8- A5 27 LDA $27
67EA- 60 RTS

```

TABLA DE COMANDOS (USADOS POR LA RUTINA PROCESADORA DE
COMANDOS)

```

*****
67EB- 42 ??? ; B(REAK)
67EC- 67 ??? ; CIC
67ED- C0 54 CPY #$54 ; T(ERMINAL)
67EF- 47 ??? ; CIC
67F0- A6 43 LDX $43 ; C(R)
67F2- 87 ??? ; PPC
67F3- A6 51 LDX $51 ; Q(UIT)
67F5- 47 ??? ; CIC
67F6- B8 CLV
67F7- 52 ??? ; R(ESET)
67F8- C7 ??? ; CIC PPC
67F9- AC 5A E7 LDY $E75A ; COMANDO Z ,SIC PPC PAS
67FC- F3 ???

```



```

67FD- 49 90      EOR  #90      ; CPMANDO I, PPC
67FF- D3         ???
6800- 4B         ???      ; COMANDO K
6801- 90 DF      BCC  $67E2   ; PPC
6803- 45 43      EOR  $43     ; E(CO), CIC
6805- 80         ???
6806- 46 E3      LSR  $E3     ; F(ROM TECLADO), CIC PPC PAS
6808- 04         ???
6809- 4C E3 01   JMP  $01E3   ; L(F), CIC PPC PAS
680C- 58         CLI         ; X(OFF)
680D- E3         ???      ; CIC PPC PAS
680E- 08         PHP
680F- 54         ???      ; T(ABULADO)
6810- 83         ???      ; PPC
6811- 40         RTI
6812- 53         ???      ; S(HIFTEADO)
6813- 43         ???      ; CIC
6814- 40         RTI
6815- 4D E3 20   EOR  $20E3
6818- 00         BRK

```

***** FINAL PRIMERA PARTE *****

```

6819- 42         ???      ; B(AUDIOS)
681A- F6 7C      INC  $7C,X   ; CIC PPC PAS
681C- 50 F6      BVC  $6814   ; P(ARIDAD), CIC PPC PAS
681E- 9A         TXS
681F- 44         ???      ; D(ATOS)
6820- F6 9B      INC  $9B,X   ; CIC PPC PAS
6822- 46 F6      LSR  $F6     ; F(F DELAY), CIC PPC PAS
6824- 46 4C      LSR  $4C     ; L(F DELAY)
6826- F6 40      INC  $40,X   ; CIC PPC PAS
6828- 43         ???      ; C(R DELAY)
6829- F6 3A      INC  $3A,X   ; CIC PPC PAS
682B- 54         ???
682C- D6 34      DEC  $34,X
682E- 4E 90 EB   LSR  $E890   ; COMANDO N, PPC
6831- 53         ???      ; S(LOT DE PANTALLA)
6832- 56 60      LSR  $60,X   ; CIC
6834- 00         BRK      ; FIN DE TABLA

```

RUTINAS DE COMANDO

```

6835- A9 3F      LDA  #$3F     ; PONE LA OPCION DE TRAS-
6837- A0 07      LDY  #$07     ; LACION
6839- D0 10      BNE  $684B
683B- A9 CF      LDA  #$CF     ; PONE DELAY DE CR
683D- A0 05      LDY  #$05
683F- D0 0A      BNE  $684B
6841- A9 F3      LDA  #$F3     ; PONE DELAY DE LF
6843- A0 03      LDY  #$03
6845- D0 04      BNE  $684B
6847- A9 FC      LDA  #$FC     ; PONE DELAY DE FF
6849- A0 01      LDY  #$01
684B- 3D B8 03   AND  $03B8,X ; NO MOLESTA OTROS FLAGS
684E- 85 2A      STA  $2A
6850- BD 38 04   LDA  $0438,X
6853- 29 03      AND  #$03
6855- 18         CLC
6856- 6A         ROR

```

6857-	2A	ROL		; CAMBIA DIRECCIONES
6858-	88	DEY		
6859-	D0 FC	BNE	\$6857	; PREPARA PARA OREAR FLAGS
685B-	05 2A	ORA	\$2A	
685D-	9D B8 03	STA	\$03B8,X	
6860-	60	RTS		
6861-	29 07	AND	##07	; PONE EL COMANDO SLOT
6863-	0A	ASL		
6864-	0A	ASL		
6865-	0A	ASL		
6866-	85 2A	STA	\$2A	
6868-	0A	ASL		
6869-	C5 26	CMP	\$26	; ASEGURARSE DE NO PONER
686B-	F0 0F	BEQ	\$687C	; TU PROPIO SLOT
686D-	BD B8 04	LDA	\$04B8,X	
6870-	29 C7	AND	##C7	
6872-	05 2A	ORA	\$2A	
6874-	9D B8 04	STA	\$04B8,X	
6877-	A9 00	LDA	##00	; ALMACENA CERO
6879-	9D 38 06	STA	\$0638,X	; RESETEO DE SLOT
687C-	60	RTS		
687D-	29 0F	AND	##0F	; NUEVA VELOCIDAD (BAUD)
687F-	D0 07	BNE	\$6888	
6881-	B9 81 C0	LDA	\$C081,Y	
6884-	4A	LSR		
6885-	4A	LSR		
6886-	4A	LSR		
6887-	4A	LSR		
6888-	09 10	ORA	##10	; GENERADOR DE VELOC (BAUD)
688A-	85 2A	STA	\$2A	
688C-	A9 E0	LDA	##E0	
688E-	85 2B	STA	\$2B	
6890-	B9 8B C0	LDA	\$C08B,Y	
6893-	25 2B	AND	\$2B	
6895-	05 2A	ORA	\$2A	
6897-	99 8B C0	STA	\$C08B,Y	
689A-	60	RTS		
689B-	88	DEY		
689C-	0A	ASL		; SALVA NUEVO # BIT DATOS
689D-	0A	ASL		
689E-	0A	ASL		
689F-	0A	ASL		
68A0-	0A	ASL		
68A1-	85 2A	STA	\$2A	
68A3-	A9 1F	LDA	##1F	
68A5-	D0 E7	BNE	\$688E	
68A7-	1E B8 04	ASL	\$04B8,X	; PONE MODO TERMINAL
68AA-	38	SEC		
68AB-	B0 10	BCS	\$68BD	
68AD-	99 89 C0	STA	\$C089,Y	; CAE RTS, DTR
68B0-	20 93 FE	JSR	\$FE93	; PR#0
68B3-	20 89 FE	JSR	\$FE89	; IN#0
68B6-	AE F8 07	LDX	\$07F8	
68B9-	1E B8 04	ASL	\$04B8,X	; BORRA MODO TERMINAL

```

68BC-- 18          CLC
68BD-- 7E B8 04   ROR    $04B8,X
68C0-- 60          RTS
68C1-- B9 8A C0   LDA    $C08A,Y ; ENVIO DE BREAK DURANTE
68C4-- 48          PHA                    ; 233 MILISEG.
68C5-- 09 0C          ORA    #$0C
68C7-- 99 8A C0   STA    $C08A,Y
68CA-- A9 E9          LDA    #$E9 ; DELAY DE 233 MILISEG.
68CC-- 20 C4 65   JSR    $65C4
68CF-- 68          PLA
68D0-- 99 8A C0   STA    $C08A,Y ; RESTAURACION COMANDOS
68D3-- 60          RTS
68D4-- A9 28          LDA    #$28
68D6-- 9D 38 06   STA    $0638,X ; ANCHO IMPRESION A 40
68D9-- A9 80          LDA    #$80
68DB-- 1D 38 07   ORA    $0738,X ; PONE ECO DE PANTALLA
68DE-- D0 05          BNE    $68E5
68E0-- A9 FE          LDA    #$FE ; RESETEO FLAG GENER. LF
68E2-- 3D 38 07   AND    $0738,X
68E5-- 9D 38 07   STA    $0738,X
68E8-- 60          RTS
68E9-- C9 28          CMP    #$28 ; >=40?
68EB-- 90 0E          BCC    $68FB ; SI NO, SALE
68ED-- 9D 38 06   STA    $0638,X ; PONE NUEVO ANCHO IMPRESION
68F0-- A9 3F          LDA    #$3F ; DESHABILITA PANTALLA
68F2-- D0 EE          BNE    $68E2
68F4-- 1E 38 05   ASL    $0538,X ; DESHAB.RECONOCER COMANDO
68F7-- 38          SEC
68F8-- 7E 38 05   ROR    $0538,X
68FB-- 60          RTS

```

VECTOR DE ACUERDO A ESTADO DE COMANDO

```

68FC-- A8          TAY                    ; REG-A=COMAN. DE ESTADO
68FD-- A5 27          LDA    $27
68FF-- 29 7F          AND    #$7F
6901-- C9 20          CMP    #$20 ; SALTA ESPACIOS EN TODOS MODOS
6903-- D0 09          BNE    $690E
6905-- C0 03          CPY    #$03 ; EXCEPTO MODO 3
6907-- F0 01          BEQ    $690A
6909-- 60          RTS
690A-- A9 04          LDA    #$04
690C-- D0 6D          BNE    $697B
690E-- C9 0D          CMP    #$0D ; RETORNO DE CARRO?
6910-- D0 12          BNE    $6924
6912-- 20 79 69   JSR    $6979
6915-- C0 07          CPY    #$07
6917-- F0 01          BEQ    $691A
6919-- 60          RTS
691A-- A9 CD          LDA    #$CD ; TODO PROCESO EMPIEZA EN
691C-- 48          PHA                    ; LA PAGINA $CD
691D-- BD 38 04   LDA    $0438,X
6920-- 48          PHA
6921-- A4 26          LDY    $26
6923-- 60          RTS
6924-- B5 35          STA    $35
6926-- A9 CE          LDA    #$CE

```

; TODAS LAS RUTINAS DEBEN
; EMPEZAR EN LA PAGINA \$CE

6928- 48 PHA
6929- B9 30 CE LDA \$CE30,Y
692C- 48 PHA
692D- A5 35 LDA \$35
692F- 60 RTS

; RTS A PROCED. COMANDO

TABLA DE SALTO DE ESTADO

6930- A7 ??? ; MAL ESTADO
6931- 37 ???
6932- 61 B9 ADC (\$B9,X) ; PARAMETRO ACUMULADO
6934- 8A TXA
6935- A7 ??? ; ESTADO ILEGAL
6936- B9 ??? ; SALTO HASTA CR
6937- B9 ??? ; SALTO HASTA CR, LUEGO CMD

COMANDO DE ESTADO 1

6938- DD 38 05 CMP \$0538,X ; ES <CMD>?
693B- D0 06 BNE \$6943
693D- DE B8 04 DEC \$04B8,X ; PONE ESTADO ANTERIOR A CERO
6940- 4C 02 67 JMP \$6702
6943- C9 30 CMP #\$30 ; >=0?
6945- 90 0D BCC \$6954
6947- C9 3A CMP #\$3A ; <=9?
6949- B0 09 BCS \$6954
694B- 29 0F AND #\$0F ; ES UN NUMERO
694D- 9D 38 04 STA \$0438,X
6950- A9 02 LDA #\$02
6952- D0 27 BNE \$697B ; PONE MODO 2 Y RETORNA
6954- C9 20 CMP #\$20 ; ES CARACT. DE CONTROL?
6956- B0 06 BCS \$695E
6958- 9D 38 05 STA \$0538,X ; PONE NUEVO CARACT.COMANDO
695B- 4C 79 69 JMP \$6979 ; RESETEA ESTADO A CERO
695E- A0 00 LDY #\$00 ; USA TABLA DE COMANDO
6960- F0 4D BEQ \$69AF

COMANDO DE ESTADO 2 : PARAMETRO ACUMULADOR

6962- 49 30 EOR #\$30 ; CONVIERTE \$30-\$39 A 0-9
6964- C9 0A CMP #\$0A ; 0-9?
6966- B0 0D BCS \$6975
6968- A0 0A LDY #\$0A ; ES UN NUMERO, SUMA
696A- 7D 38 04 ADC \$0438,X
696D- 88 DEY
696E- D0 FA BNE \$696A
6970- 9D 38 04 STA \$0438
6973- F0 15 BEQ \$698A
6975- A0 2E LDY #\$2E ; USA TABLA DE COMANDO
6977- D0 36 BNE \$69AF

PONE COMANDO DE ESTADO

6979- A9 00 LDA #\$00
697B- 85 2A STA \$2A
697D- AE F8 07 LDX \$07F8

```

6980-   BD B8 04   LDA   $04B8,X
6983-   29 F8     AND   #$F8
6985-   05 2A     ORA   $2A
6987-   9D B8 04   STA   $04B8,X
698A-   60       RTS

```

COMANDO DE ESTADO 4 (E/D)

```

698B-   AB       TAY
698C-   BD 38 04   LDA   $0438,X
698F-   C0 44     CPY   #$44      ; D(ESHABILITA)?
6991-   F0 09     BEQ   $699C
6993-   C0 45     CPY   #$45      ; H(ABLITA)?
6995-   D0 11     BNE   $69A8      ; SI NO , IGNORA ESTE COMANDO
6997-   1D 38 07   ORA   $0738,X   ; PONE FLAG
699A-   D0 05     BNE   $69A1
699C-   49 FF     EOR   #$FF      ; INVIERTE PARA DESHABILITAR
699E-   3D 38 07   AND   $0738,X   ; RESETEA FLAG
69A1-   9D 38 07   STA   $0738,X

```

ESCAPE PARA ESTADO 6

```

69A4-   A9 06     LDA   #$06
69A6-   D0 D3     BNE   $697B
69A8-   A9 20     LDA   #$20      ; CODIGO PARA MAL COMANDO
69AA-   9D B8 05   STA   $05B8,X
69AD-   D0 F5     BNE   $69A4

```

TABLA CONDUCTORA DE COMANDO PROCESADOR

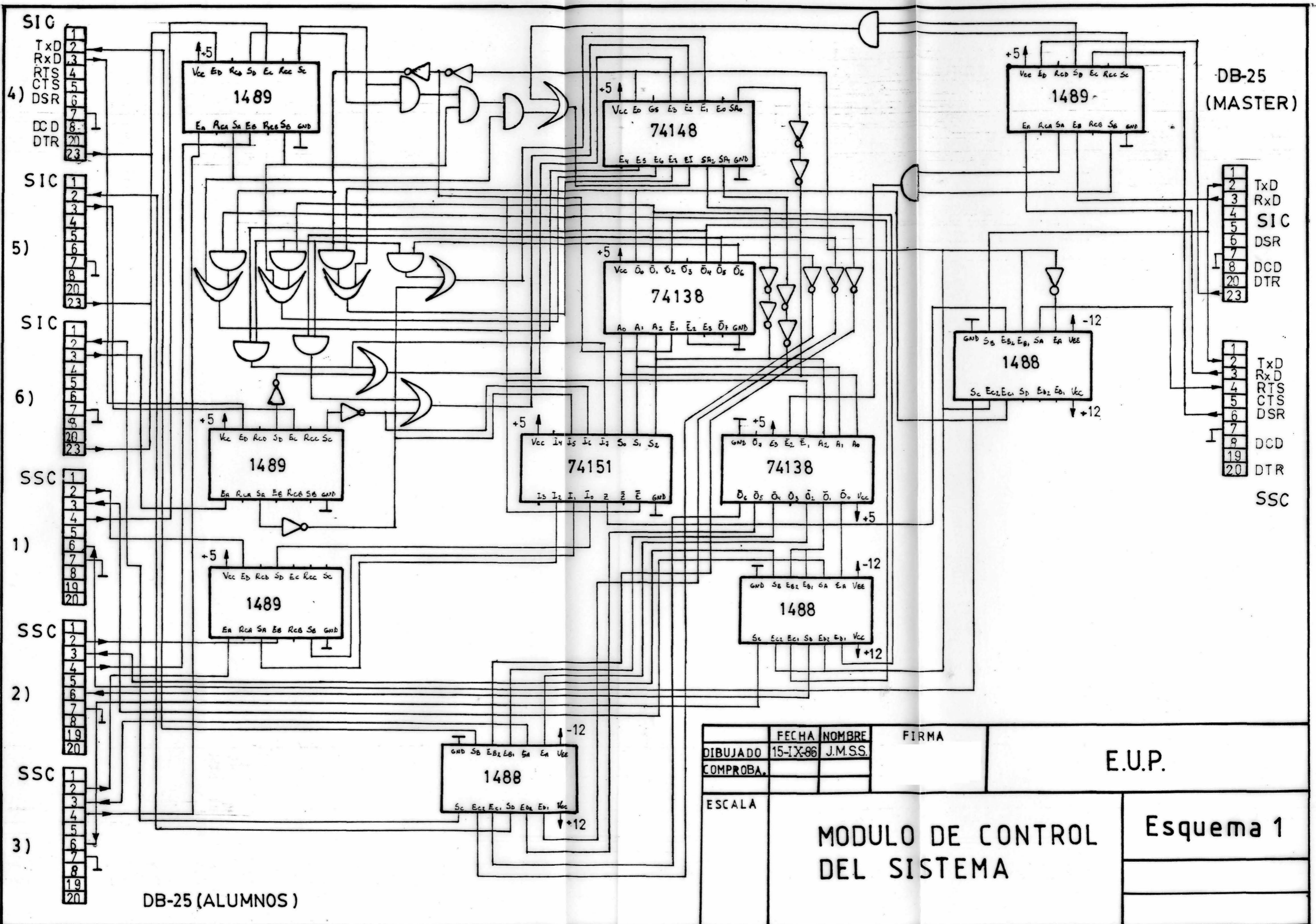
```

69AF-   B9 EB 67   LDA   $67EB,Y   ;OBTIENE CARACTER CANDIDATO
69B2-   F0 F4     BEQ   $69A8      ; CERO MARCA FINAL DE TABLA
69B4-   C5 35     CMP   $35
69B6-   F0 05     BEQ   $69BD
69B8-   CB       INY           ; REENTRA MODOS EQUIVOCADOS
69B9-   CB       INY           ; LONGITUD DE ENTRADA=3
69BA-   CB       INY
69BB-   D0 F2     BNE   $69AF
69BD-   CB       INY
69BE-   B9 EB 67   LDA   $67EB,Y
69C1-   85 2A     STA   $2A
69C3-   29 20     AND   #$20      ; CHEQUEO HABILIT.PASCAL
69C5-   D0 07     BNE   $69CE
69C7-   BD 38 07   LDA   $0738,X
69CA-   29 10     AND   #$10
69CC-   D0 EB     BNE   $69B9
69CE-   BD 38 07   LDA   $0738,X   ; OBTIENE BIT CIC/PPC
69D1-   4A       LSR
69D2-   4A       LSR
69D3-   24 2A     BIT   $2A
69D5-   B0 04     BCS   $69DB      ; SALTO SI MODO CIC
69D7-   10 E0     BPL   $69B9      ; NO OK PARA PPC
69D9-   30 02     BMI   $69DD
69DB-   50 DC     BVC   $69B9      ; NO OK PARA CIC
69DD-   A5 2A     LDA   $2A
69DF-   48       PHA
69E0-   29 07     AND   #$07

```

69E2-	20 7B 69	JSR	\$697B	:	PONE PROXIMO ESTADO
69E5-	C8	INY			
69E6-	68	PLA			
69E7-	29 10	AND	##10		
69E9-	D0 07	BNE	\$69F2	:	SI SE PONE BIT 4 , SALTA
69EB-	B9 EB 67	LDA	\$67EB,Y	:	A RUTINA
69EE-	9D 38 04	STA	\$0438,X		
69F1-	60	RTS			
69F2-	A9 CD	LDA	##CD	:	RUTINAS DEBEN ESTAR EN \$CD
69F4-	48	PHA			
69F5-	B9 EB 67	LDA	\$67EB,Y		
69F8-	48	PHA			
69F9	A4 26	LDY	\$26		
69FB-	BD 38 04	LDA	\$0438,X	:	MUCHAS RUTINAS NECESITAN ESTO
69FE	60	RTS			
69FF-	00	BRK			

A P E N D I C E V



	FECHA	NOMBRE	FIRMA	
DIBUJADO	15-IX-86	J.M.S.S.		E.U.P.
COMPROBA.				
ESCALA	MODULO DE CONTROL DEL SISTEMA			Esquema 1

BIBLIOGRAFIA

- * A PROFESIONAL ASSEMBLY LANGUAGE DEVELOPMENT SYSTEM FOR APPLE COMPUTERS (LISA).
Julio - 1980
- * A REFERENCE MANUAL FOR THE APPLE II AND APPLE II PLUS PERSONAL COMPUTERS (MANUAL DE REFERENCIA DEL APPLE II).
- * 6502 ASSEMBLY LANGUAGE PROGRAMMING
Lance A. Lenthal
Osborne / McGraw-Hill - 1979
- * MANUAL DE REFERENCIA DE LA SUPER SERIAL CARD (SSC)
- * SERIAL INTERFACE MANUAL (SIC)
INSTALLATION AND OPERATING MANUAL
- * MANUAL DE REFERENCIAL DE LA COMMUNICATIONS INTERFACE CARD (CIC)
- * EL LIBRO DEL RS-232-C
Joe Campbell
Ediciones Anaya Multimedia , S.A.-1985
- * TTL DATA BOOK
Texas Instruments
- * TELEINFORMATICA Y REDES DE COMPUTADORES
Serie: Mundo Electrónico .
Marcombo - Boixareu Editores -1984
- * LINEAR INTERFACE DATA BOOK
Fairchild - 1978
- * APPLE II GUIA DE USUARIO
Lon Poole / Martin McNiff / Steven Cook
Osborne / Mc Graw-Hill - 1985