

300617

4  
2ej



**UNIVERSIDAD LA SALLE**

**ESCUELA DE INGENIERIA**

**Incorporada a la U.N.A.M.**

Diseño de una Terminal  
de  
Video

**Tesis Profesional**

Que para obtener el Título de  
**INGENIERO MECANICO ELECTRICISTA**

presenta

**JORGE DOMENECH MUELAS**

TESIS CON  
FALLA DE ORIGEN

México, D. F.

enero 1986



Universidad Nacional  
Autónoma de México



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

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

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

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

## INDICE

### DISEÑO DE UNA TERMINAL DE VIDEO.

- Prologo.

I. Definición de las características de la terminal a diseñar.

1.1- Descripción del tipo de aplicación para la cual va a ser utilizada la terminal.

II. Arquitectura de la terminal.

III. Diseño del microcomputador que controla a la terminal.

3.1- Arquitectura del microcomputador.

3.1.1- Descripción del microcomputador.

3.2- Microprocesador.

3.3- Memoria. Mapas de memoria.

3.3.1- Consideraciones de carga.

3.4- Dispositivos I/O. Mapa de I/O.

3.5- Decodificación de memoria e I/O

3.6- Controlador de DMA (DMAC).

#### IV. Diseño de la sección de Video.

4.1- Descripción de las señales requeridas por un monitor de video EIA RS170.

4.1.1- Señales de sincronía y de borrado.

4.1.2- Señales de luminancia y crominancia.

4.2- Formato de pantalla de la terminal. Modos alfabético, gráfico 1 y gráfico 0.

4.2.1- Formato de la información contenida en la pantalla de video.

4.2.1.1- Modo alfabetico y calculo de las  
señales requeridas.

4.2.1.2- Modo grafico 1.

4.2.1.3- Modo grafico 0.

4.3- Diseño del generador de video.

4.3.1- Arquitectura del VG.

4.3.1.1- Grupos del VG.

4.3.1.2- Canal de comunicaciones con el CRTC.

4.3.1.3- Canal de comunicaciones con el CRT.

4.3.1.4- Programación del VG.

4.3.2- Diseño del circuito del VG.

4.3.2.1- Diseño de CGEN, AGEN y DOTGEN.

4.3.2.2- Diseño de VSYN.

4.3.2.3- Diseño de VMUX.

4.3.2.4- Diseño de COLORMAT, VADD y CLBINJ.

4.4- Controlador de CRT (CRTC).

4.4.1- Arquitectura del CRTC.

4.4.1.1- Operación del CRTC.

4.4.1.2- Grupos del CRTC.

4.4.1.3- Registros del CRTC.

4.4.2- Diseño del circuito del CRTC.

V. Diseño de la interfaz de comunicaciones computador/  
terminal.

5.1- Diseño de la interfaz EIA RS232C.

5.1.1- Protocolos de datos.

5.1.2- Protocolos de electrónica.

5.1.3- Diseño del circuito de la interfaz RS232C.

5.2- Diseño de la interfaz de lazo de corriente de 20  
mA.

5.2.1- Diseño del circuito de lazo de corriente de  
20 mA.

VI. Diseño del teclado.

6.1- Arquitectura del teclado.

6.1.1- Configuración externa.

6.1.2- Configuración interna.

6.2- Diseño del controlador y del explorador de teclado.

6.2.1- Diseño del KBC.

6.2.2- Diseño del KBE.

6.3- Programa de control del KBC.

## VII. Diseño de la fuente de alimentación.

7.1- Estructura de la fuente.

7.2- Multivibrador asimétrico y filtros de salida.

7.3- Etapa de entrada.

7.4- Conmutador de potencia.

7.5- Protección contra corto circuito.

7.6- Especificaciones generales.

## VIII. Programa de control de la terminal.

B.1- Diagrama de flujo. Modo remoto.

B.2- Descripción de la programación de la sección de video.

B.2.1- Características de la operación en modo remoto.

B.2.2- Tiempo utilizado para refrescar el CRT.

B.2.3- Prioridad de las interrupciones.

B.3- Programación del DATACOMM.

B.4- Programación del KBC.

- Apendices:

A-1. Diagramas completos.

A-2. Principales abreviaturas y terminos.

A-3. Resumen de las especificaciones de la terminal.

A-4. Código ASCII.

- Bibliografia.

## PROLOGO

Nuestro tiempo es el tiempo de la electrónica. Quienquiera que pretenda negar la tremenda importancia de esta rama de la ingeniería en la civilización actual debe ser alguien que camina sin ver las maravillas que lo rodean, pues basta con mirar a nuestro alrededor para darnos cuenta de los tremendos cambios que se llevaron, se llovan y se llevarán a cabo en nuestro habitat natural como consecuencia del advenimiento de la era de la electrónica.

En forma aveces sutil, aveces prepotente, la electrónica se ha convertido en un elemento dominante en nuestra sociedad. Es extraño aquel de nosotros que no tiene contacto diario con ella; ya sea al ver la hora en un reloj electrónico, al cerrar un negocio por vía telefónica, al enterarse, vía televisión, de lo que paso en el otro extremo del mundo hace apenas unas horas, al resolver un complejo problema en una calculadora programable, al oír música en el radio, o al editar su tesis en una computadora personal.

Sin embargo la electrónica no parece venir a llenar una necesidad fundamental; no necesitamos electrónica para comer, vestirnos o construir nuestra casa. Entonces ¿Por qué la electrónica parece estar, casi como un dios, en todas partes?. La respuesta es, a mi entender, sencilla: la

electrónica nos ha dado una capacidad de manejo de la información que el hombre jamás había poseído.

Por medios electrónicos el dirigente de una empresa puede enterarse de lo que está ocurriendo en el mercado de sus productos, no importa lo lejano que éste se halle de su centro de producción, a unas pocas horas de que los hechos hayan ocurrido, y tomar las acciones pertinentes. O puede recopilar una gran cantidad de información de su proceso de producción y recibir solo un informe estadístico, que puede analizar con rapidez, habiendo sido hecha la tarea de adquisición y proceso estadístico de la información por equipo electrónico.

Un banco puede manejar una gran cantidad de cuentas y dar una gran cantidad de servicios distribuidos en diferentes sucursales, gracias a que la información de cada una de estas puede ser compartida por cada sucursal gracias a una red electrónica de comunicaciones.

Un proceso industrial que antes requiriera de varios hombres para su control puede ahora ser controlado por un dispositivo electrónico que adquiera la información necesaria del proceso, la valore y tome las acciones adecuadas.

Vemos pues que el principal valor de la electrónica radica en la posibilidad de, por medios electrónicos, adquirir, transmitir y procesar información.

El elemento central que nos permite procesar grandes cantidades de información con gran rapidez es la computadora. En efecto, la definición más exacta de computadora nos dice que esta es "una máquina capaz de adquirir información (mediante los adecuados transductores), procesarla de acuerdo a una lista de instrucciones que le hayan sido dadas con anterioridad, y que pueden ser cambiadas para adecuarse a diferentes procesos, y actuar sobre el exterior de acuerdo a la información adquirida".

En realidad, casi es posible dividir la electrónica en dos partes: las comunicaciones (para transmitir información entre puntos remotos), y el proceso de información (adquisición de datos, computación, acción sobre el proceso).

Uno de los principales elementos con los cuales una computadora debe ser capaz de comunicarse es con el ser humano. Para recibir información procedente de este la computadora debe poseer algún sistema de adquisición de datos adecuado; el sistema que parece ser más útil es un teclado de máquina de escribir. La información es generalmente entregada desde la computadora al ser humano en forma escrita, pero es impráctico (excesivamente costoso) imprimir todos los intercambios de información entre la máquina y el ser humano, por lo que generalmente se utiliza un CRT o una pantalla de cristal líquido como dispositivo de salida.

Existen otros dispositivos de comunicación computador/humano, entre ellos el más prometedor es quizás la comunicación oral, pero este sistema presenta aun gran cantidad de problemas que lo hacen impracticable.

El objetivo de esta tesis es desarrollar una terminal de video para computadoras, con capacidad grafica y manejo de color, e inteligente, es decir, que tiene la capacidad para actuar como microcomputadora. Este tipo de terminal es el que se esta usando con mayor exito como interfaz hombre/computador.

No se trata de desarrollar un sistema nuevo de comunicación hombre/máquina, sino de estudiar el funcionamiento de uno ya existente (la terminal de video) y de probada eficacia, y de mejorarlo, generando una implementación totalmente personal y altamente eficiente.

En el capítulo I se discuten las características de mi terminal, haciendo énfasis en su versatilidad, diferencias con terminales ya existentes y aplicaciones.

Jorge Domenech Muelas.

México D.F., a 3 de Noviembre de 1985.

## I. DEFINICIÓN DE LAS CARACTERÍSTICAS DE LA TERMINAL A DISEÑAR

Solo existe una justificación para diseñar un nuevo tipo de equipo, y es que venga a solucionar una necesidad de alguno de nuestros clientes, que constituya una mejora real sobre los equipos ya existentes, por lo que justo despues de concebir una nueva idea debemos preguntarnos: ¿ es esto necesario?, ¿ llena este dispositivo una necesidad, presente o futura de alguno de nuestros clientes?. Si la respuesta a estas preguntas es afirmativa debemos proceder con el diseño; sino debemos archivar nuestra idea, hasta un momento en que se vuelva la solución a un problema existente.

Un ingeniero no es solo un observador; es un creador, y es, ante todo, un solucionador de problemas. Para solucionar un problema debemos, sobre todo, reconocer su existencia, ¿ cuál es, pues, el problema que justifica el que yo pretenda diseñar una terminal de video, existiendo ya otras muchas en el mercado? La respuesta es muy sencilla: el diseño que yo propongo reúne las condiciones de funcionabilidad, versatilidad simplicidad de manejo y costo necesarias para convenir al cliente que necesite una terminal/microcomputadora con capacidad de graficación y manejo de color.

## 1.1- DESCRIPCIÓN DEL TIPO DE APLICACIÓN PARA LA CUAL VA A SER UTILIZADA LA TERMINAL.

Esta terminal ha sido concebida para utilizarse en un medio en el cual se necesite una terminal de video altamente eficiente, con capacidades de graficación a color y de bajo costo, que pueda ser utilizada, en un momento dado, como una microcomputadora de baja capacidad.

Un ejemplo de este tipo de aplicación puede encontrarse en un despacho de diseño de ingeniería, en donde cada uno de los ingenieros puede requerir de una microcomputadora en donde ejecutar programas sencillos, obtener graficas y editar textos, pero que en un momento dado puede tener la necesidad de acceder a un banco central de datos en donde se reuna el trabajo de cada diseñador, para conocer cual es el status general del trabajo, etc.

Esta terminal utiliza un monitor de video de acuerdo al estandar RS170, con un ancho de banda de 4.0 MHz, por lo cual la resolución que puede alcanzarse con ella es bastante limitada, y no pueden usarse las 80 columnas que es comun encontrar en una terminal estandar, sin embargo esto presenta dos ventajas, en primer lugar, el costo de este tipo de monitor es generalmente bajo; en segundo lugar, esta terminal puede utilizarse como un generador de caracteres y de graficas para televisión, e incluso, en modo de

microcomputadora, como un generador de juegos de video.

La decisión de utilizar este tipo de monitor le da a esta terminal una pequeña desventaja de resolución, frente a una terminal normal, y una ventaja al aumentar drásticamente el tipo de aplicaciones en donde se puede utilizar el diseño básico aquí utilizado.

Debe entenderse que este es un proyecto académico, por lo cual es razonable que se haya dado prioridad a la versatilidad sobre la excelencia en una sola aplicación, ya que así es posible usar el mismo diseño, con pocas modificaciones, en otras áreas; por ejemplo, si quisieramos una terminal de alta resolución bastaría con modificar la programación del CRTIC, y alterar ligeramente la configuración de los generadores de sincronía (la terminal del registro de desplazamiento de donde se obtendría VSYNC y HSYNC debería cambiarse).

Por lo demás las capacidades gráficas y de manejo de color de esta terminal son realmente buenas, y pueden encontrar gran aplicación en el área de los juegos de video y de CAD/CAM, así como (volvemos a mencionarlo) en la generación de caracteres y de gráficas para televisión.

## II. ARQUITECTURA DE LA TERMINAL

La terminal debe poseer tres funciones basicas, que son:

1. Comunicación con el computador maestro (bidireccional).
2. Recepción de datos dados por el usuario a través del teclado.
3. Visualizar la información recibida desde el computador maestro; a través del CRT, o en forma audible, a través del generador de audio.

Para llevar a cabo cada una de estas funciones se definira un grupo independiente, que poseera la inteligencia necesaria para poder realizar su tarea, y además para poder comunicarse con un grupo de control, que le suministrara la información necesaria para poder efectuar su trabajo. No existira una comunicación directa entre los diversos grupos, sino que todas las comunicaciones se llevaran a cabo a través del grupo de control, el cual interpretara las informaciones procedentes del teclado y del computador maestro, procediendo de acuerdo con ellas. Los grupos que nos permitiran realizar estos trabajos son:

### 1. DATACOMM:

Grupo de comunicación de datos. Permite una

comunicación bidireccional entre el computador maestro y la terminal. Es un grupo ejecutor.

2. KB:

Grupo de teclado (keyboard). Permite la recepción de información desde el usuario, mediante un teclado tipo QWERTY. Es un grupo ejecutor.

3. VIDEO:

Grupo de video. Genera todas las señales necesarias para visualizar la información recibida en un monitor de video RS170. Es un grupo ejecutor.

4. AUDIO:

Grupo de audio. Genera los diversos efectos de sonido que puede producir la terminal. Es un grupo ejecutor.

6. VTKBP:

Grupo microcomputador. Es el controlador de la terminal. Interpreta toda la información recibida por esta ya sea desde el computador anfitrión o desde el usuario, determinando las acciones que deben tomarse e instruyendo de estas a los diversos grupos que deben tomarlas. Es un grupo de control.

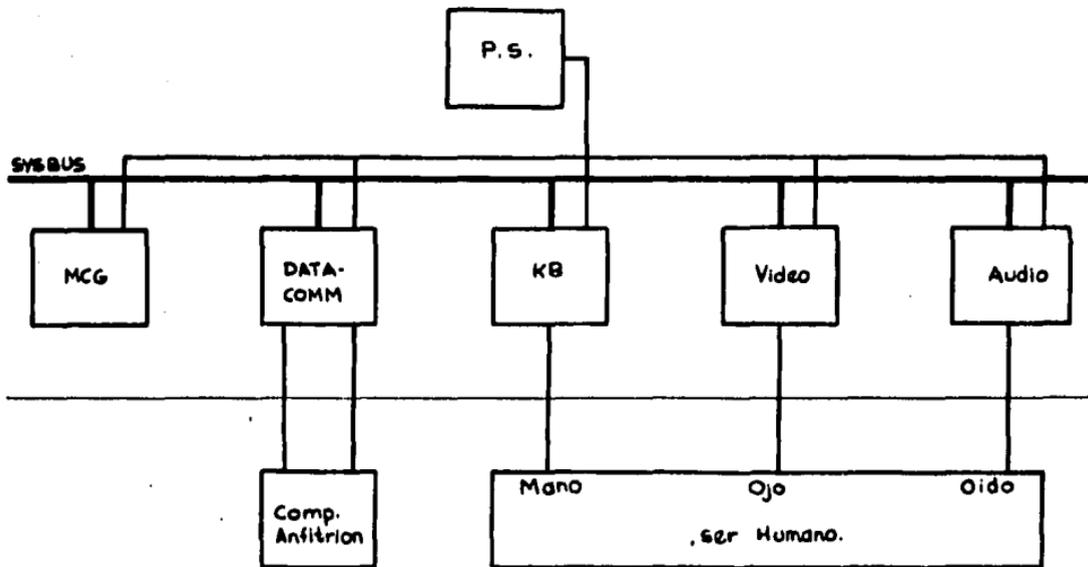
7. PS:

Alimentación de potencia. Este grupo se encarga de generar las diversas señales de alimentación

necesitadas por los otros bloques de la terminal. Es un grupo de soporte.

Como puede observarse existen tres tipos de grupos: de control, de ejecución y de soporte. El grupo de control no esta capacitado para llevar a cabo ninguna acción de interfaz con el exterior, tarea que le es asignada a los grupos de ejecución o perifericos; el grupo de soporte tiene como unica función el permitir que los otros grupos tengan potencia.

Los grupos que llevan a cabo estas tres funciones son inherentemente complejos y a pesar de que poseen un cierto grado de inteligencia no pueden funcionar en forma independiente, sino que requieren de un dispositivo que los controle, interpretando la información generada por ellos y proporcionandoles los datos necesarios para su correcto funcionamiento. Hasta cierto punto puede considerarse a este dispositivo como a un cerebro que recibe, procesa y genera información desde sus diferentes sentidos. El dispositivo que mejor puede satisfacer estos requerimientos es una microcomputadora (podria utilizarse logica discreta para lograr este proposito, pero el costo en componentes, tamaño del circuito, tiempo de diseño, tiempo de reparación y componentes en stock para servicio seria más alto que el de utilizar un micro) con los diversos grupos de ejecución conectados a ella como dispositivos I/O, mediante un bus de datos, direcciones y control. En nuestra terminal este bus recibe el nombre de SYSBUS. Los diversos grupos y su relación



pueden verse en la figura 2.0.0/1.

Cada uno de los grupos mencionados con anterioridad (a excepción del grupo de alimentación) esta dividido a su vez en tres grupos, que son:

1. Grupo de interfaz con SYSBUS.
2. Grupo de ejecución.
3. Grupo de control.

Como sus nombres lo indican estos subgrupos se encargan de interconectar al grupo con SYSBUS. De ejecutar las tareas que el grupo deba realizar. Y de controlar las diferentes funciones del grupo.

PS esta conectado a los diversos grupos mediante una red de potencia separada de SYSBUS.

Los capitulos que siguen contienen una descripción mas detallada de cada uno de los grupos y de los subgrupos que lo forman, tanto desde el punto de vista estructural (arquitectura), como desde el punto de vista de su implementación física (desarrollo del circuito electrónico). Una descripción detallada de SYSBUS se encuentra en el capítulo III (Diseño del VTKBP).

### III. DISEÑO DEL MICROCOMPUTADOR QUE CONTROLA A LA TERMINAL

Como se describió en el capítulo anterior la terminal tiene varios dos funciones principales, que son:

1. Convertir información procedente del computador anfitrión a un formato inteligible para el ser humano.
2. Recibir información procedente del ser humano y comunicarla al computador anfitrión.

Para lograr estas funciones la terminal posee:

1. Un grupo de comunicaciones con el computador anfitrión.
2. Un grupo de generación de video.
3. Un grupo de generación de audio.
4. Un grupo de colección de información procedente del ser humano.

Es evidente que estos grupos no pueden tener un funcionamiento independiente, sino que interactúan el uno con el otro, para que esta interacción pueda llevarse a cabo existen dos caminos, uno es proporcionar la inteligencia necesaria a cada grupo para que pueda comunicarse con los otros, esto implica que cada grupo pueda comprender a los otros tres. Otro procedimiento implica construir un quinto

grupo con la inteligencia suficiente para controlar a los otros cuatro, y dar a cada uno de estos la inteligencia necesaria para realizar su función propia y para poder comunicarse con el grupo de control. Este es el camino escogido por mí, debido a que es más simple y barato que el procedimiento anterior.

El grupo de control es un microcomputador, y recibe el nombre de Video Terminal and Key Board Processor (VTKBP). El escoger un microcomputador como bloque de control no fue una decisión que siguiera a la decisión de organizar a la terminal en grupos, sino que fue una decisión primaria, de la cual partió la idea de la organización por grupos, que son considerados por el micro como dispositivos I/O. Este tipo de organización tiene la ventaja fundamental de ser totalmente clara y fácil de implementar, dando como resultado un circuito totalmente lógico y simple. Es de hecho esta organización la que se usa en la mayor parte de las terminales de video actuales, siendo solo en terminales antiguas en donde se pueden encontrar otros tipos de organización, semejantes a este pero no tan claramente definidos, sistemas no muy eficientes que fueron adoptados debido a la falta de elementos tales como el microprocesador.

### 3.1- ARQUITECTURA DEL MICROCOMPUTADOR:

El microcomputador esta basado en un microprocesador Intel 8085AH y es una estructura tipica tipo Von Neumann.

El 8085AH no fue escogido en forma secundaria para adaptarse al diseño sino que se le selecciono a priori y el diseño fue llevado a cabo para adaptarse a este microprocesador. Esto se hizo asi debido a que el equipo de desarrollo de Hardware y de Software existente en la Universidad La Salle esta enfocado a los diversos microprocesadores de Intel (8080, 8085, 8086, etc.), por lo que era una decisión lógica el utilizar uno de estos. El 8085 de 8 bits tiene una capacidad suficiente para el trabajo y es más eficiente y requiere menos componentes externos que el 8080, en tanto que los procesadores de 16 bits como el 8086, 8088 estaban más alla de lo requerido.

Utilizar el 8085AH determino desde un principio la necesidad de usar un controlador de acceso directo a memoria (DMAC), la posibilidad de utilizar una estructura de direccionamiento con espacios separados para memoria e I/O y la posibilidad de utilizar 4 diferentes interrupciones sin requerir un controlador de interrupciones (de haberse usado un procesador de la familia 6800 de Motorola o 6500 de Synertek/Mostek quizas no hubiera sido necesario utilizar el DMAC, pero si hubiera sido preciso usar un controlador de

interrupciones, en cuanto a la estructura de direccionamiento, habria sido comun el espacio para I/O y memoria), sin embargo practicamente cualquier microprocesador de 8 bits podria haberse utilizado exitosamente para el diseño, siendo los precios y el rendimiento comparable en casi todos ellos.

### 3.1.1- Descripción del microcomputador.

El microcomputador tiene 32K bytes de memoria RAM estática que da cabida a las dos imágenes de video (a ser discutidas en el capítulo IV) más un área de memoria utilizada por VTKBOS para mantener un stack y cierta cantidad de variables de sistema, y que puede ser usada también por el usuario para ejecutar programas propios mediante un programa monitor perteneciente a VTKBOS, y 8K bytes de memoria EPROM que contienen el sistema operativo de la terminal llamado VTKBOS (Video Terminal and Key Board Operating System), más un espacio libre que está reservado para nuevas versiones del sistema operativo, y para software de aplicación para manejo de gráficas, etc. que no es comprendido dentro de VTKBOS, esta estructura se explica en el capítulo VII.

Se mantienen, así mismo, 23 registros de I/O de 1 byte cada uno, en el área de I/O. Estos registros corresponden a 7 dispositivos I/O.

### 3.2- Microprocesador:

A continuación aparece una lista de las principales características del 8085AH, esta lista no es exhaustiva, información completa sobre el 8085AH puede encontrarse en el MCS-80/85 TM FAMILY USER'S MANUAL de Intel (ver bibliografía).

#### 8085AH:

Base de tiempo: Cristal de cuarzo, frecuencia resonante en paralelo: 6.144 MHz  $\pm$  0.005%, Capacitancia de carga  $\leq$  30 pF, Capacitancia en paralelo  $\leq$  7 pF, Resistencia equivalente en paralelo  $\leq$  75 E, Nivel de conducción  $\leq$  10 mW.

#### Frecuencia de operación:

3.072 MHz  $\pm$  0.005%

#### Consumo de potencia:

$I_{cc} = 120 \text{ mA @ } V_{cc} = 5.0 \text{ V} \pm 10\%$

$P_d (\text{max}) = 1.5 \text{ W}$

#### Niveles logicos de entrada y salida:

Compatibles con TTL

#### Niveles de corriente de salida:

$I_{OL} = 2 \text{ mA}$

IOH = -400  $\mu$ A

La frecuencia maxima a la que puede trabajar el 8085AH es de 3.125 MHz, yo escogi una frecuencia de 3.072 (6.144/2) para que la señal de reloj del puerto RS232C pudiera derivarse de la señal de reloj del 8085 (SYSCLK) sin que diera lugar a velocidades de transmisión no estandar.

Como puede observarse el fan out del microprocesador es de apenas cinco cargas LS TTL, por lo cual no podemos utilizarlo en una configuración como la actual sin darle una mayor capacidad de conducción, para lograr esto se añaden una serie de amplificadores de corriente (buffers) a las salidas del 8085, de ahora en adelante nos referiremos a esta reunion 8085/buffers con la palabra procesador (ver plano 1). Como podemos ver el procesador se comunica con la memoria y dispositivos I/O por 8 líneas de datos (D0-7), 16 líneas de direcciones (A0-15) y 14 líneas de control, el nombre que se le da a estas líneas de comunicación es de SYSBUS (System Bus).

Las líneas de datos son bidireccionales y tienen un buffer transceiver octal de tres estados 74LS245, que les da un IOL = 24 mA y un IOH = -15 mA, con un tiempo de propagación de 8 nS. Estas líneas de datos estan multiplexadas con las líneas de direcciones A0-7, para demultiplexarlas se utiliza una latch octal de tres estados

74LS373 con  $I_{OL} = 24 \text{ mA}$  e  $I_{OH} = -2.6 \text{ mA}$ , esta latch esta sincronizada a la linea de ALE (Adress Latch Enable) procedente del 8085, cuando esta señal esta en su estado inactivo las lineas A0-7 del 8085 contienen datos, cuando esta en estado activo dichas lineas contienen direcciones. Las lineas de direcciones A8-15 utilizan un 74LS244 como buffer, con  $I_{OL} = 24 \text{ mA}$  e  $I_{OH} = -2.6 \text{ mA}$ .

Las lineas de control son las siguientes:

1. MEMRD/ :

(Memory Read). Cuando esta señal esta en su estado activo el procesador trata de leer la memoria. Salida.

2. MEMWR/ :

(Memory Write). Cuando esta señal esta en su estado activo el procesador escribe en la memoria. Salida.

3. IORD/ :

(I/O Read). Cuando esta señal esta en su estado activo el procesador trata de leer el contenido de un registro I/O. Salida.

4. IOWR/ :

(I/O Write). Cuando esta señal esta en su estado activo el procesador trata de escribir información en algun registro I/O. Salida.

Estas cuatro señales anteriores son mutuamente exclusivas, es decir que no puede existir más de una activa a un mismo tiempo. Sus niveles de IOL e IOH son de 24 y -2.6 mA respectivamente.

5. SYSCLK :

(System Clock). Señal cuadrada de 3.072 MHz. IOL = 24 mA, IOH = -2.6 mA. Salida.

6. SYSRES :

(System Reset). Señal que esta activa durante los periodos en los que el 8085 es sometido a un Reset. Salida.

7. HOLD :

Cuando esta señal esta activa el DMAC esta solicitando al 8085 el control de SYSBUS. Entrada.

8. HLDA :

(Hold Acknowledge). Mientras esta señal esta activa el 8085 ha entregado el control de SYSBUS al DMAC. Esta señal se utiliza junto con HOLD para generar un Handshake. Salida.

9. AEN :

(Address Enable). Esta señal procedente del DMAC pone a las lineas de control, direcciones y datos procedentes del procesador en un estado de alta impedancia cuando esta activa. Es usada por el DMAC para retirar al procesador de SYSBUS una vez que ha adquirido el control de este mediante

HOLD. Entrada.

10. PFAIL :

(Power Fail). Esta señal procedente de la fuente de poder advierte al 8085 de que en un mínimo de 3 mS habrá una falla en el suministro de energía eléctrica, con el fin de que VTKBOS pueda proceder a una desconexión ordenada del sistema (en versiones posteriores se pretende guardar información de importancia en un área de memoria CMOS sostenida por una batería). Entrada.

11. VIRQ :

(Vertical Interrupt Request). Esta señal procedente del CRTIC advierte al 8085 de que se ha iniciado un regreso vertical, para que VTKBOS pueda reinicializar al CRTIC y llevar a cabo las rutinas de servicio del teclado y del puerto serie. Entrada.

12. RxDY :

(Receiver Ready). Esta señal procedente del SIO advierte al 8085 que se ha recibido alguna información procedente del computador anfitrión, a través del RS232C o del lazo de corriente de 20 mA. Entrada.

13. KBIRD :

(Key Board Interrupt Request). Esta señal procedente del controlador de teclado informa al 8085 de que alguna tecla ha sido presionada por el usuario. Entrada.

#### 14. SOD :

(Serial Output Data). Esta señal procedente del 8085 genera al cambiar de estado un clic: audible, se lleva a cabo dicho cambio de estado cada vez que una tecla es presionada.  $I_{OH} = -400 \mu A$ ,  $I_{OL} = -2.6 \text{ mA}$ . Salida.

Los buffers usados para estas señales son un 74LS125 buffer cuádruple, y un 74LS368 buffer inversor sextuplo.

El procesador recibe un Reset adecuado cada vez que se enciende la unidad, sin embargo se incluye, así mismo un interruptor de botón que puede usarse para generar un reset en cualquier momento en que se desea. Este botón tiene un filtro RC para eliminar las oscilaciones producidas, ya que el circuito mecánico del interruptor es subamortiguado, cuando el botón es oprimido se genera una constante de tiempo de aproximadamente 1 ms. Cuando el interruptor es liberado el tiempo para pasar de 0 a 5 v es de 40 ms aproximadamente, esto se hace para garantizar que el 8085 tenga los 10 ms que requiere para empezar a trabajar correctamente después de que se le haya aplicado Vcc.

### 3.3- MEMORIA:

La memoria de la terminal es de dos tipos, RAM y ROM, la RAM es de 32K x 8, implementada con 16 TMS4016NL de 2K x 8, por este motivo la memoria se dividió en paginas de 2K. La ROM de 8K x 8 se implemento con una EPROM 2764. En total la memoria consta de 20 paginas arregladas de acuerdo al siguiente mapa:

		núm.	
		de	
		pag.	
	dir.		
PAG	DCBA	FEDCBA9876543210	
de:	0	0000	0000000000000000   ROM PG0
a :	0	07FF	0000011111111111
	1	0800	0000100000000000   ROM PG1
	1	0FFF	0000111111111111
	2	1000	0001000000000000   ROM PG3
	2	17FF	0001011111111111
	3	1800	0001100000000000   ROM PG4
	3	1FFF	0001111111111111
	4	2000	0010000000000000   RAM PG0
	4	27FF	0010011111111111
	5	2800	0010100000000000   RAM PG1

5	2FFF	0010111111111111
6	3000	0011000000000000   RAM PG2
6	37FF	0011011111111111
7	3800	0011100000000000   RAM PG3
7	3FFF	0011111111111111
8	4000	0100000000000000   RAM PG4
8	47FF	0100011111111111
9	4800	0100100000000000   RAM PG5
9	4FFF	0100111111111111
10	5000	0101000000000000   RAM PG6
10	57FF	0101011111111111
11	5800	0101100000000000   RAM PG7
11	5FFF	0101111111111111
12	6000	0110000000000000   RAM PG8
12	67FF	0110011111111111
13	6800	0110100000000000   RAM PG9
13	6FFF	0110111111111111
14	7000	0111000000000000   RAM PG10
14	77FF	0111011111111111
15	7800	0111100000000000   RAM PG11
15	7FFF	0111111111111111
16	8000	1000000000000000   RAM PG12
16	87FF	1000011111111111
17	8800	1000100000000000   RAM PG13
17	8FFF	1000111111111111
18	9000	1001000000000000   RAM PG14
18	97FF	1001011111111111
19	9800	1001100000000000   RAM PG15

Para implementar la memoria ROM se requieren 16 circuitos TMS4016NL de 2k x 8, estos dispositivos son estaticos, por lo cual no se necesita ningun tipo de circuito de refresco. A cada uno de los TMS4016 y al 2764 se le a adio un capacitor de 0.01 uF  $\pm$  10% @ 10 V, del tipo ceramico de baja inductancia, como bypass entre Vcc y tierra, esto con la finalidad de que no se propage ningun transitorio de alta frecuencia a trav es de la linea de Vcc al habilitarse o deshabilitarse algun integrado, tambien se a ade un capacitor electrolitico de 22 uF  $\pm$  10% @ 10 V a la entrada del banco de memoria, con este mismo fin. (Plano 2).

### 3.3.1- Consideraciones de carga.

Una cantidad de memoria como la que estamos usando representa una carga más o menos alta para los buses de datos y de direcciones, tanto resistiva como capacitivamente, esta carga capacitiva puede degradar los tiempos de acceso de la memoria, para evitar esto se añadieron buffers a la memoria para aislarla del resto del sistema.

La carga representada por un circuito integrado se calcula mediante la siguiente ecuación:

$$I = I_i + Cdv/dt$$

Donde:

$I_i$  es la corriente resistiva de entrada al IC.  $C$  es la carga capacitiva.  $dv$  es el cambio de voltaje de nivel activo a inactivo.  $dt$  es el tiempo en que debe hacerse este cambio para no afectar el tiempo de ascenso y descenso de la señal.

Para el TMS4016NL:

$$C = 8 \text{ pF}$$

$$dv = 2 \text{ V}$$

$$dt = 10 \text{ nS}$$

$$I_i = 10 \text{ uA}$$

$$I = 1.61 \text{ mA}$$

Para el 2764:

$$C = 6 \text{ pF}$$

$$dv = 2 \text{ V}$$

$$dt = 20 \text{ nS}$$

$$I_i = 10 \text{ uA}$$

$$I = 610 \text{ uA}$$

Para 16 TMS4016, la corriente por motivos capacitivos puede llegar a ser de 25.76 mA (si seleccionaran los 16 a un tiempo, caso que no puede ocurrir).

La corriente de 1.61 mA asociada a la carga capacitiva de aproximadamente 70 pF (producida por los 7 dispositivos I/O) más la posible carga capacitiva del circuito impreso en el cual se monten los componentes (2.5 pF/in en epoxy) puede exceder la carga capacitiva que pueden soportar los buffers del VTKBP sin degradar sus tiempos.

Para evitar esto se utiliza un buffer 74LS245 como buffer de datos de la memoria, y un 74LS244 como buffer de direcciones. ( $I_{OH} = -15 \text{ mA}$ ,  $I_{OL} = 24 \text{ mA}$ ,  $t_{prop} = 12 \text{ nS}$ ). (Plano 2).

### 3.4- DISPOSITIVOS I/O.

La estructura de I/O del VTKBP mantiene a 7 dispositivos de I/O, que son los siguientes:

1. DMAC: (Direct Memory Access Controller; 8257). Este dispositivo permite que el CRTC accese la memoria imagen de video en forma directa, sin tener que usar al 8085 como intermediario.

2. SIO:

(Serial Input/Output; 8251). Este dispositivo conforma la parte central del DATACOMM (cap. V), se encarga de serializar la información recibida del 8085 y transmitirla en forma serie asincrónica al computador anfitrión, y de poner en paralelo la información recibida del computador anfitrión, para que pueda ser manejada por el 8085.

3. CRTC:

(Cathodic Ray Tube Controller; 8275). Este dispositivo se encarga de proveer la interfaz entre el VTKBP y el VG, y de leer la memoria imagen de video, mediante el DMAC, suministrando la información ASCII contenida en esta al VG que se encarga de serializarla y visualizarla.

#### 4. CTR:

(Counter; 8253). Este dispositivo contiene tres contadores programables de 16 bits, dos de los cuales son usados por el SIO para obtener su tasa de transmisión y de recepción de datos (Tx baud rate y Rx: baud rate), el tercero es utilizado por el generador de audio para conformar un tono programable. La base de tiempo para el CTR es obtenida de SYSCLK.

#### 5. COLOR:

(Graphics Color Latch; 74LS77). Este dispositivo es en realidad un arreglo de tres latches que contienen un código de color (0-7) depositado ahí por el VTKBP y que define el color del próximo pixel a ser visualizado.

#### 6. AUDIO:

(NE555). Este dispositivo es un multivibrador NE555 conectado en forma monoestable, que al ser seleccionado genera un pulso de 1.1 S de duración, utilizado para habilitar el generador de audio.

#### 7. KBC:

(Key Board Controller; 8279). Este dispositivo se encarga de realizar la exploración del teclado matricial de la terminal, y es la interfaz entre el teclado y el VTKBP.

Cada uno de estos dispositivos esta explicado con más profundidad en los capitulos que le corresponden, solo uno de ellos, el DMAC se explicara en el inciso 3.6 de este capitulo, ya que es una parte integrante de VTKBP, en tanto que los demas se consideran (funcionalmente hablando) como dispositivos perifericos que no forman parte del VTKBP, sino que estan conectados a la estructura de I/O de este, y son mencionados como dispositivos I/O en esta sección unicamente porque son direccionados directamente por el 8085 (es decir que desde el punto de vista del direccionamiento forman parte del VTKBP, pero, vuelvo a insistir, no desde el punto de vista funcional).

La estructura de I/O del VTKBP esta compuesta por 22 registros de I/O (R0-R22), que pueden ser de lectura, de escritura o de lectura/escritura. Cada uno de los dispositivos conectados a esta escritura hace uso de uno o más de estos registros, que no pueden ser compartidos, es decir que no pueden haber varios dispositivos que tengan un registro en comun.

Fisicamente estos registros estan contenidos en los mismos dispositivos conectados a la estructura I/O, es decir que no forman parte de esta, pero conceptualmente pertenecen a la estructura de I/O, ya que son direccionados por esta.

La habilitación de los registros de I/O se hace mediante los comandos de IDRD/ y IQWR/ que son generados por el VTKBP,

y la dirección del dispositivo I/O (el 8085 puede manejar hasta 255 de estos en el espacio de I/O) se expede en las líneas A0-7 del bus de direcciones de VTKBP, que son decodificadas por el decodificador de I/O.

El mapa de direcciones de la estructura I/O es el siguiente:

REG	DIR	I/O PAG III 76543210	
0	00	00000000	8251 DATA REG.
1	01	00000001	8251 CMD. REG.
2	20	00100000	8275 PAR. REG.
3	21	00100001	8275 CMD. REG
4	40	01000000	8279 DATA REG.
5	41	01000001	8279 CMD. REG.
6	60	01100000	8253 CTR. 0
7	61	01100001	8253 CTR. 1
8	62	01100010	8253 CTR. 2
9	63	01100011	8253 MODE WORD
10	80	10000000	8257 CHO DMA ADDR.
11	81	10000001	8257 CHO TC
12	82	10000010	8257 CH1 DMA ADDR.
13	83	10000011	8257 CH1 TC
14	84	10000100	8257 CH2 DMA ADDR.
15	85	10000101	8257 CH2 TC

16	B6	10000110	8257 CH3 DMA ADDR.
17	B7	10000111	8257 CH3 TC
18	BB	10001000	8257 STATUS (READ ONLY)
19	BB	10001000	8257 MODE (WRITE ONLY)
20	A0	10100000	74LS77 COLOR
21	C0	11000000	NE555 AUDIO

Como puede verse el espacio de I/O del 8085 se dividió en 8 paginas de 32 bytes cada una, estas paginas son seleccionadas por las direcciones A5-7. Con esta estructura podriamos tener hasta 8 dispositivos I/O (uno por pagina) con 32 registros cada uno. Estos registros de I/O comparten direcciones con ROM PGO, sin embargo no hay ningun problema de selección simultanea ya que los registros I/O son seleccionados por IORD/ e IDWR/, en tanto que la memoria es seleccionada por MEMRD/ y MEMWR/.

### 3.5- DECODIFICACION DE MEMORIA E I/O.

La decodificación de la estructura de memoria e I/O se hace a partir de un arreglo de cuatro decodificadores 3:8 del tipo 74LS138.

La memoria se dividió en 20 paginas de 2K x 8 (en total se ocupan solo 40 K del espacio de 64 K de memoria del 8085), se usaron tres 74LS138 para la decodificación de este espacio (3 decodificadores permiten seleccionar hasta 48 K de memoria en paginas de 2 K). La EPROM de 8 K x 8 ocupa 4 paginas, y se selecciona mediante el AND de las líneas ROM PG0, ROM PG1, ROM PG2 y ROM PG3 (ROM CS/ = ROM PG0/ \* ROM PG1/ \* ROM PG2/ \* ROM PG3/).

Los decodificadores de memoria son habilitados por la función G2B/ = MEMRD/ \* MEMWR/ es decir que la memoria solo sera habilitada si MEMRD/ o MEMWR/ estan activas. Esta misma señal de habilitación es la señal MEMEN/ utilizada para habilitar el buffer de datos (74LS245) de la memoria.

El decodificador de I/O es habilitado por la señal IOEN/ dada por la relación IOEN/ = IORD/ \* IOWR/, y por la señal AEN/, es decir que solo se habilita el decodificador despues de que el 8085 ha estabilizado la dirección de I/O en la parte baja del bus de direcciones (recuerdese que el 8085 multiplexa las direcciones con los datos en la parte baja del

bus de datos/direcciones).

Puede verse la estructura de decodificación en el plano

2.

### 3.6- CONTROLADOR DE DMA (DMAC).

El acceso directo a memoria (DMA) es un termino acuñado para designar un método por el cual un dispositivo periférico puede acceder la memoria central de un microcomputador sin que la información tenga que pasar a través del procesador.

Basicamente consiste en desconectar al procesador del bus del sistema durante un cierto periodo de tiempo (ciclo de DMA) en el cual otro dispositivo (DMAC) toma el control de este.

Existen varias formas en las cuales se puede hacer esto, en algunos procesadores con ciclo de instrucción constante (MC6800) el dispositivo periférico puede acceder el bus cuando el microprocesador esta en la parte de proceso interno del ciclo, y por ende, desconectado del bus del sistema. Otra forma, más comun, y que no requiere que el dispositivo periférico este sincronizado al procesador es deteniendo a este ultimo mediante una interrupción especialmente diseñada para este uso, y que desconecta al procesador del bus. En el 8085 esto puede hacerse mediante las líneas de HOLD y HLDA.

Cuando el DMAC (8257) quiere adquirir el control de SYSBUS lo unico que requiere hacer es generar un HRQ (HOLD Request) sobre la línea de HOLD del 8085, poniendo esta en su estado activo, al recibir esta señal de HOLD el 8085

terminara la instrucción que este ejecutando y se detendra, poniendo la señal de HLDA (Hold Acknowledge) en estado activo y todas sus señales de salida en un estado de alta impedancia esta señal de HLDA sera detectada por el 8257 y le indicara que el control del bus ya es efectivamente suyo, con lo cual el 8257 podra generar sus propias señales de MEMRD/ y MEMWR/, y utilizar el bus de direcciones para direccionar la memoria imagen de video y leerla a través del bus de datos, para luego pasar esta información al CRTIC. Este acceso a la memoria por parte del 8257 exige que el 8085 haya suministrado una cierta cantidad de información a este previamente, esta información consiste basicamente en la dirección de inicio del DMA y la longitud de la cadena a ser leida de la memoria (DMA Address y DMA Terminal Count). Esto se discute plenamente en el capitulo sobre VTKBOS V1.0-01.

El DMA es requerido al 8257 por el CRTIC (8275) cada vez que se vacia el buffer de renglon del CRTIC (el proceso se describe en el capitulo sobre arquitectura del CRTIC) mediante la linea de DRQ (DMA Request), y es garantizado por el 8257 al 8275 mediante la linea de DACK/ (DMA Acknowledge), mediante un proceso de Handshake similar al existente entre el 8085 y el 8257. (Plano 1).

#### IV. DISEÑO DE LA SECCIÓN DE VIDEO.

La sección de video de la terminal consta de dos elementos principales, el generador de video (VG) y el controlador de CRT (CRTC).

La función del VG es generar todas las señales de luminancia y crominancia requeridas por un monitor estandar RS170, a partir de la información contenida en ciertas áreas de RAM, llamadas imagenes de video (VII y VIO). Estas imagenes de video son continuamente actualizadas por el VTKBP, de acuerdo a la información recibida desde el computador anfitrión al cual esta conectada la terminal.

CRT son las siglas de la frase Cathodic Ray Tube (tubo de rayos catodicos), siglas que se han estandarizado como denominación para los monitores de video utilizados en el área de las computadoras, sera con este significado como las utilizaremos en el contexto de esta tesis.

Son dos las funciones del controlador (CRTC): generar las señales de sincronia y borrado requeridas por el monitor de video, y mediante el controlador de acceso directo a memoria (DMAC), transferir la información visualizable contenida en la memoria imagen de video al generador de video (VG). Para poder realizar esta ultima función, el controlador

ha de ser capaz, tambien, de informar al VG del modo de video en que esta operando el VTKB (alfanumerico, semigrafico o grafico); toda esta informacion se transfiere del CRTC al VG a través de un bus privado de video (PVB1).

El CRTC debe ser instruido por el VTKBP para poder realizar correctamente sus funciones, ya que aun cuando estas estan interconstruidas en el circuito, sus valores (tiempos para la señal de sincronia, etc.) son programables.

#### 4.1- DESCRIPCIÓN DE LAS SEÑALES REQUERIDAS POR UN MONITOR DE VIDEO EIA RS170.

Cualquier monitor de video requiere basicamente de dos grupos de señales: señales de sincronia y borrado; señales de luminancia y crominancia.

Una imagen de video puede definirse como una matriz de puntos apagados y encendidos, para determinar que puntos deben estar apagados y que puntos deben estar encendidos, el dispositivo que esta generando la imagen debe explorar esta punto por punto, en el estandar RS170 esta exploración se hace linea por linea hasta llenar una pantalla, cuando la pantalla esta llena se vuelve a empezar la exploración desde el principio.

La información de encendido-apagado (o más exactamente, la información sobre que tan brillante debe ser un punto de la pantalla) es llamada luminancia (Y), si además de brillo un punto debe tener color se debe generar una señal que represente esta información, esta señal se llama crominancia (I, Q).

Para que el monitor pueda saber en que puntos acaba una linea, es preciso que se le suministre esta información en forma de un pulso de sincronia horizontal. En realidad esta información la obtiene el monitor a partir de un generador de reloj interno, cuya frecuencia de oscilación (FH) es la misma

que la frecuencia de exploración horizontal del generador de video; el pulso de sincronía se utiliza unicamente para sincronizar estos dos osciladores.

De igual manera el monitor posee un oscilador cuya frecuencia (FV) es la misma que la frecuencia de exploración vertical del generador de video (número de cuadros por segundo), la señal de sincronía vertical tiene como unico objeto el mantener sincronizados ambos osciladores.

El objeto de la señal de borrado horizontal es mantener la señal de luminancia en el negro durante el regreso horizontal (HRTC), es decir durante el intervalo en el que el haz de exploración regresa del extremo derecho de la pantalla al extremo izquierdo.

El objeto de la señal de borrado vertical es mantener la señal de luminancia en el negro durante el regreso vertical (VRTC), es decir durante el intervalo en el que el haz de exploración regresa del final de la pantalla al principio de esta. Estas señales de borrado evitan que durante el regreso se trace una linea blanca en la pantalla.

#### 4.1.1- Señales de sincronía y borrado:

El estandar RS170 define cuatro señales de este tipo: sincronía horizontal, sincronía vertical, borrado horizontal y borrado vertical.

En el esquema 4.1.1 podemos observar la forma en que se construye una línea de video positivo compuesto, junto con sus niveles IRE, donde 140 IRE equivale a la amplitud total de la señal de video, y es 1 Vp-p para el estandar RS170.

1. Amplitud total: 140 IRE
2. Amplitud del pulso de sincronia: 40 IRE
3. Diferencia entre el negro y el pulso de borrado: 7.5 IRE
4. Amplitud de la señal de video: 92.5 IRE

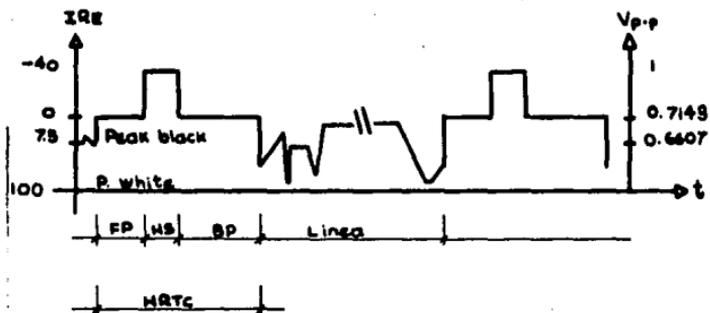
La frecuencia de exploracion horizontal (FH) es de 15,750 Hz, es decir que se escriben 15,750 líneas/seg.

H representa para nosotros la duración de una línea completa, incluyendo el HRTC, y tenemos que:

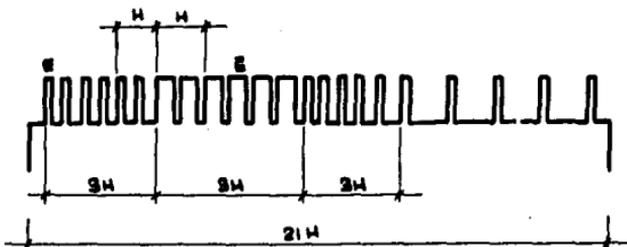
1.  $H = 63.5 \mu\text{s}$
2.  $\text{HRTC} = 9.5 - 11.5 \mu\text{s}$
3.  $\text{HSYNC} = 4.75 \pm 0.5 \mu\text{s}$
4.  $\text{Front porch} = 1.27 \mu\text{s (min)}$
5.  $\text{Back porch} = 3.61 \mu\text{s (min)}$
6.  $\text{Video} = 52 - 54 \mu\text{s}$

En una pantalla de video tenemos 525 líneas, es decir que su duración es  $525H$  (16.7 ms), estas 525 líneas se dividen en dos campos, llamados par e impar, el campo impar va de la línea 1 a la 241.5, y es el que se visualiza en primer lugar,

### F4.1.1



### F4.1.2



en segundo lugar se visualiza el campo par, que va de la línea 0.5 hasta la línea 504; en el campo par solo se incluyen líneas pares, y en el impar solo se incluyen líneas impares, esto se logra gracias al desfase de una línea entre ellos ( $\frac{1}{2}$  de línea al final del campo impar y  $\frac{1}{2}$  de línea al principio del par). Al final del campo par y del campo impar existe un VRTC de 21 líneas. Esto puede observarse en la figura 4.1.2. Este tipo de manejo de campos se conoce en inglés como interlace (entrelazado); y su razón es la de evitar que haya un cierto parpadeo en la imagen visualizada al doblar la frecuencia efectiva con que se actualiza esta.

De lo anterior se desprende que  $FV = \text{frecuencia de campo} = 2$   
\* frecuencia de pantalla.

1. Frecuencia de pantalla = 30 Hz
2. Frecuencia de campo = 60 Hz
3.  $V = 16.7 \text{ ms}$  (duración de un campo, incluye VRTC)
4.  $\text{VRTC} = 0.0008 - 0.0013 \text{ S}$
5.  $\text{VSYNC} = 27.35 \text{ uS}$
6.  $E = 2.54 \text{ uS}$
7.  $\text{SR} = 4.4 \text{ uS}$

Para que no se pierda la sincronía horizontal durante el pulso de VSYNC es preciso que se incluya alguna forma de conservarla durante VSYNC, esto se hace mediante seis indentaciones de 4.4 uS, llamadas serraciones, en VSYNC, y cuya frecuencia es  $2FV$  es decir, 31,500 Hz, así mismo se

añaden unos pulsos de 2.54 uS seis al principio y seis al final de VSYNC cuya frecuencia es también de 31,500 Hz, llamados pulsos de preequalización y de postequalización respectivamente, cuya finalidad es eliminar el desajuste de media línea existente entre el campo par y el impar. Es decir que para un campo par se tomarán en cuenta los pulsos de HSYNC cero, dos, cuatro, etc., y para un campo impar: uno, tres, cinco, etc., esto ocurre así porque el generador de barrido horizontal solo tomará en cuenta los pulsos de HSYNC cuya separación sea de aproximadamente una línea.

#### 4.1.2- Señales de luminancia y crominancia.

En el inciso anterior discutimos la necesidad y la forma de las señales de sincronía y borrado, dándonos cuenta de que estas señales no contienen ningún tipo de información acerca de la imagen que se está explorando, sino que su única utilidad estriba en suministrar información sobre la misma exploración en sí (puede considerarse que estas señales contienen información sobre la posición de un punto en la imagen), para que el generador de vídeo y el monitor estén en sincronía. Es por lo tanto evidente que se requieren otras señales para transmitir información sobre la imagen; estas señales son las señales de luminancia y de crominancia.

> Señal de luminancia (Y) :

La señal de luminancia contiene toda la información

necesaria sobre la luminosidad de un punto en la pantalla, en realidad esta señal es la señal de Blanco y Negro (B & W), esta señal tendrá una amplitud máxima de 92.5 IRE, con el blanco pico igual a 100 IRE y el negro pico igual a 7.5 IRE, la señal de borrado aparece como más negra que el negro, en 0 IRE, y el pulso de sincronía se extiende hasta -40 IRE, esto se aprecia en la figura 4.1.1.

En la figura 4.2.1 podemos ver dos señales de luminancia para dos imágenes construidas en blanco y negro puros. Si las imágenes contuvieran diversos tonos de gris, la amplitud de la señal de luminancia sería proporcional a cada tono de gris, siendo cercana a los 7.5 IRE para tonos oscuros y a 100 IRE para tonos claros. La señal de luminancia tal como la hemos representado hasta ahora corresponde a lo que se llama video compuesto positivo, el estándar RS170 especifica, sin embargo, video compuesto negativo; este video es equivalente al que hemos usado, pero invertido (figura 4.2.2). El término compuesto significa que la señal de luminancia aparece, sumada a las de sincronía y borrado en una sola señal de video, y no por separado en una señal independiente.

> Ancho de banda de la señal de luminancia:

El estándar RS170 especifica un ancho de banda máximo de 4.0 MHz para la señal de luminancia. Este ancho de banda limitado va a afectar directamente la cantidad de información que podemos representar mediante esta señal. En particular

esta información limitada va a afectar un parametro de la imagen de video llamado resolución: la información contenida en una línea ya no se va a representar en forma continua, sino en forma discreta mediante una serie de pequeños puntos, el número de estos puntos siendo directamente proporcional al ancho de banda de Y. La resolución depende directamente de cuantos puntos se utilizan para formar una imagen, a mayor número de puntos, mayor resolución.

Segun Shannon, la capacidad maxima de un canal ideal, con un ancho de banda B, y transmitiendo una señal de n niveles es:

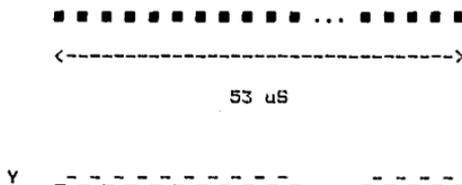
$$C = B \ln n / \ln 2$$

para B = 4.0 MHz y n = 2, C = 4.0 Mbit/s

Sin embargo es preciso tomar en cuenta que de estos 4.0 Mbit/s una parte debe utilizarse para las señales de sincronía y borrado que no contienen información sobre la imagen.

Para calcular el número máximo de puntos por línea que pueden acomodarse en este ancho de banda de 4.0 MHz, supondremos que la imagen que se esta reproduciendo esta formada por una sucesión de puntos blancos y negros, ya que el cambio de blanco a negro equivale a la mayor información posible (la excursión de amplitud de Y es la maxima posible,

de 100 IRE a 7.5 IRE):



Esta señal, por lo tanto ocupa el mayor ancho de banda posible, el ancho de banda sera la frecuencia de la señal cuadrada:

$$T = 2 * 53 \text{ uS} / n$$

donde n es el número de puntos en la señal y T es el periodo de esta, con  $T = 1/4 \text{ MHz}$ ,  $n = 424$  puntos por línea. Si el ancho de banda se reduce a  $3.2 \text{ MHz}$ , n se reduce a  $339.20$  puntos por línea.

> Señales de crominancia:

Para que el monitor pueda reproducir el color de una imagen no basta con que le demos información de sincronia y de luminancia, sino que es preciso darle alguna información que corresponda al color de cada uno de los puntos de la imagen, esta información se llama crominancia.

Cualquier color puede definirse como una combinación de

los tres colores basicos: rojo (red), verde (green) y azul (blue), entonces bastara con definir los porcentajes de cada uno de estos colores para cada uno de los puntos de la imagen para definir el color de este punto.

En el estandar del NTSC no se transmiten las señales R, G, B, sino que, para ahorrar ancho de banda de la señal de video, y para mantener la compatibilidad con los monitores de blanco y negro se prefiere utilizar la señal Y, en combinación con las señales I (In phase) y Q (Quadrature phase), obtenidas como combinaciones de las señales primarias R, G, B, de acuerdo a la siguiente matriz:

$$Y = 0.30R + 0.59G + 0.11B$$

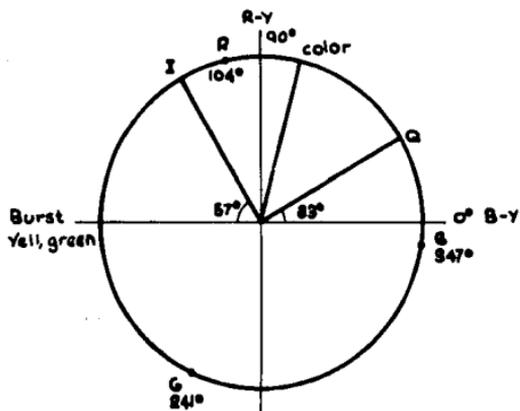
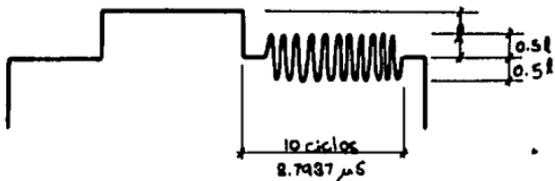
$$I = 0.60R - 0.28G - 0.32B$$

$$Q = 0.21R - 0.52G + 0.31B$$

Para que los receptores de TV a color sean compatibles con sus predecesores de blanco y negro es preciso que estas tres señales sigan ocupando el mismo ancho de banda de 4.0 MHz de la señal Y sola. Esto se logra gracias al hecho de que la señal Y no contiene información alrededor de las armónicas de 3.579545 MHz, existiendo una serie de claros alrededor de estas armónicas (figura 4.2.3). Claros que son ocupados por la señal de crominancia. Sin embargo esta señal no es una sola, sino que esta formada por dos señales I y Q que deben coexistir en el mismo espacio espectral. Para evitar que se mezclen y hacer posible su posterior recuperación, ambas

señales se modulan en contrafase antes de sumarse a la señal de video B & W, teniendo la señal I una fase de  $0^\circ$  y la señal Q una fase de  $90^\circ$ , de ahí sus nombres (In phase y Quadrature phase). La frecuencia de la señal moduladora para I y Q es de 3.579545 MHz, con lo cual se logra que las componentes de ambas señales aparezcan en un espacio libre alrededor de los 3.58 MHz, con una diferencia de fase de  $90^\circ$  entre ambas señales (figura 4.2.4). Es preciso, desde luego, limitar el ancho de banda de I y Q para evitar que se sobreimpongan estas señales sobre la señal Y; en el estandar NTSC se define un ancho de banda de 1.3 MHz para la señal I y uno de 0.5 MHz para la señal Q. Esta diferencia en los anchos de banda se debe a que el ojo humano tiene una mayor resolución para la señal I de un color naranja rojizo, que para la señal Q, de color púrpureo.

De lo anterior se desprende que la determinación de un color se va a hacer a partir de su ángulo de fase con respecto a la señal de 3.58 MHz que se utilizó para modular las señales I y Q. Es por lo tanto necesario que el monitor reciba del generador de video un fragmento de la señal de 3.58 MHz, que de ahora en adelante llamaremos subportadora de color, para así poner en fase con esta su propio oscilador interno. Este fragmento va a ser insertado por el generador en el portico trasero de la señal de borrado horizontal, y va a consistir en un grupo de 8 a 11 ciclos de la subportadora de color. Este grupo se conoce como color burst (figura 4.2.5) y tiene un color amarillo verdoso. En la figura 4.2.6



puede observarse un diagrama fasorial de las diferentes señales. En resumen:

1. Frecuencia subportadora de color: 3.579545 MHz
2. Color burst: de 8 a 11 ciclos de la subportadora.
3. Ancho de banda Y: 4.0 MHz
4. Ancho de banda I: 1.3 MHz
5. Ancho de banda Q: 0.5 MHz
6. Ancho de banda video compuesto: 4.5 MHz

Una vez se tienen las señales Y, I, Q, y SYNC, se suman todas ellas para dar origen a la señal de video colorplexed compuesto, cuya forma puede apreciarse en la figura 4.2.7.

#### 4.2- FORMATO DE PANTALLA DE LA TERMINAL. MODOS ALFABETICO, GRAFICO 1 Y GRAFICO 0.

Se decidió que la terminal poseyera tres modos de operación, que son:

1. Modo alfabetico (ALPHA).
2. Modo grafico 1 (GR1).
3. Modo grafico 0 (GR0).

Estos tres modos de operación le permiten a la terminal tener un funcionamiento sencillo y eficiente, tanto para proceso de textos, como para graficación a color.

ALPHA y GR1 son enteramente compatibles, y comparten la misma imagen de memoria (V11), así como los mismos circuitos de generación de video. GR0 requiere de un circuito de generación de video ligeramente distinto a de los otros dos modos, y posee una imagen de video propia (V10).

##### 4.2.1- Formato de la información contenida en la pantalla de video.

Para comprender la diferencia existente entre los diversos modos graficos es preciso comprender cual es el formato con el cual va a ser visualizada la información en la pantalla del monitor.

#### 4.2.2.1- Modo alfabetico y calculo de las señales requeridas.

En el modo alfabetico cada byte de la imagen de memoria contiene el codigo ASCII de un caracter a ser visualizado, el bit 7 de cada byte debe ser cero en este modo, pues de lo contrario el CRTC asumira que el codigo no es un caracter normal, sino un atributo visual o un codigo de comando. Esto se explica en profundidad en la sección de arquitectura del CRTC.

El caracter visualizable escogido es el de matriz de 7x5, encuadrado en un campo de 10x7:

	1	2	3	4	5	6	7	
01								
02		■	■	■	■	■		
03		■						
04		■						
05		■	■	■	■	■		caracter
06		■						
07		■						
08		■	■	■	■	■		
09		■	■	■	■	■		underline
10								

Antes de proseguir es preciso determinar cuales son las señales que deben ser generadas por el VG y por el CRTC para

alimentar al monitor de video. Estas señales son seis: SYNC, HRTC, VRTC, Y, I, y Q; las tres primeras son generadas por el CRTIC, en tanto que las tres ultimas son generadas por el VG. En la sección 4.1 ya se determinaron cuales son las características con las cuales deben cumplir estas señales, de acuerdo al estandar RS170. En la presente sección se determinaran las características reales de estas señales, tal como seran generadas por el CRTIC y por el VG.

Todas estas señales seran derivadas de una señal de 3.579545 MHz, llamada SQ3.58M. Esta señal es generada por el CRTIC, junto con la señal SIN3.58M, que es usada como subportadora de color. Esta señal se utiliza como base debido a que todas las señales del estandar RS170 pueden definirse con periodos que son multiples o submultiplos de ella.

Las diferentes señales que debon generarse son:

1. DOTCLK:

Esta señal tiene un periodo igual a la duración de un punto en la pantalla, se define como:

$$\text{DOTCLK} = 2 * \text{SQ3.58M}$$

Por lo tanto su frecuencia es de 7.15909 MHz. La duración de un punto es: 139.7 nS.

2. CCLK:

El periodo de esta señal es igual a la duración de un caracter (es decir: el tiempo necesario para escribir siete puntos en una línea de la pantalla):

$$CCLK = DOTCLK / 7$$

Por lo tanto su frecuencia es de 1.0227271 MHz y la duración de un caracter es de 977.78 nS.

### 3. HRTC:

Borrado horizontal. Segun el estandar RS170 debe estar entre 9.5 y 11.5 uS. Lo definiremos como el tiempo necesario para escribir 10 caracteres:

$$HRTC = 10 / CCLK$$

Por lo tanto su duración es de 9.78 uS.

### 4. HSYNC:

Sincronia horizontal. Segun el estandar RS170 debe valer  $4.75 \pm 0.5$  uS. Lo definiremos como el tiempo necesario para escribir 5 caracteres:

$$HSYNC = 5 / CCLK$$

Por lo tanto su duración es de 4.8889 uS.

### 5. FP:

Portico frontal. Segun el estandar RS170 debe tener una duraci3n de 1.27 uS m3nimo. Lo definiremos como el tiempo necesario para escribir 2 caracteres:

$$FP = 2 / CCLK$$

Por lo tanto su duraci3n es de 1.7556 uS.

#### 6. BP:

Portico trasero. Segun el estandar RS170 debe tener una duraci3n m3nima de 3.01 uS. Lo definiremos como el tiempo necesario para escribir 4 caracteres:

$$BP = 4 / CCLK$$

Por lo tanto su duraci3n es de 3.9111 uS.

Antes de seguir con las definiciones de tiempo vertical es preciso que definamos lo que sera un renglon de caracteres.

Con HRTC = 10 caracteres, H = 63.5 uS y CCLK = 1.0227271 MHz:

$$\begin{aligned} N &= 63.5 * 1.0227271 - 10 \\ &= 54.94 \\ &\approx 55 \end{aligned}$$

En una pantalla vamos a tener 241 lineas, si cada

caracter tiene diez líneas, podemos visualizar 24 renglones de caracteres por pantalla. Ya que cada renglon tiene 65 caracteres (55 visualizables y 10 de regreso), su duración es de  $65 * 10 / 1.02272714 = 635.56 \mu\text{s}$ .

Es interesante notar que no estamos utilizando video entrelazado, sino que estamos dejando en blanco uno de los dos campos que forman una pantalla, esto se hace con dos fines, en primer lugar para que la imagen sea menos susceptible al ruido de 60 Hz inducido por las líneas de potencia, y en segundo lugar para simplificar el circuito de generación de sincronía.

#### 7. VRTC:

Regreso vertical. Según el estándar RS170 esta entre 0.8 y 1.3 mS, lo definimos como el tiempo equivalente a dos renglones:

$$\text{VRTC} = 2 \text{ renglones}$$

Por lo tanto su duración es de  $2 * 635.56 \mu\text{s} = 1.2711 \text{ mS}$ .

#### 8. VSYNC:

Sincronía vertical. Según el estándar RS170 la duración de este pulso es igual a la duración de 3 líneas horizontales, es decir:

VSYNC = 3 \* H

Por lo tanto su duración es de 190.68 uS.

9. FH:

Frecuencia horizontal. Según el estándar RS170 debe ser de 15,734.263 Hz, en un monitor de color; la FH que nosotros estamos generando es de:

$$\begin{aligned} FH &= 1.0227271M / 65 \\ &= 15,734.263 \text{ Hz} \end{aligned}$$

Que es enteramente igual a la del estándar.

10. FV:

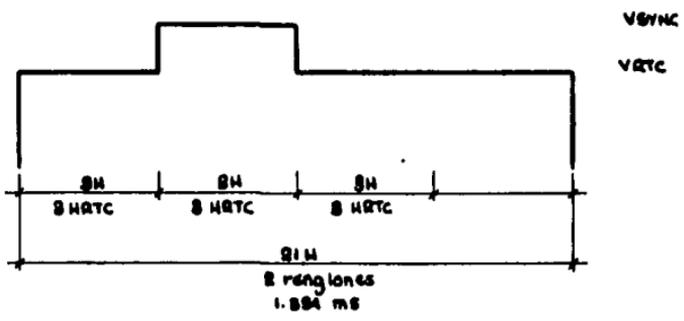
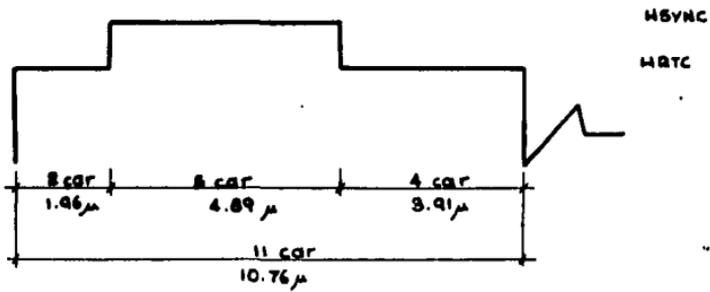
Frecuencia vertical. En el estándar RS170 es de 60 Hz, en nuestro sistema es de :

$$\begin{aligned} FV &= FH / 26 \\ &= 60.52 \text{ Hz} \end{aligned}$$

Lo cual es aceptable para un monitor RS170.

Todas las señales anteriores pueden verse en la figura 4.2.1.

De lo anterior se desprende que el número de caracteres por pantalla sera de  $55 * 24 = 1,320$ . Por lo que se requeriran 2,640 bytes (2.58 K) de memoria como imagen de video para el modo alfabetico. La imagen de memoria contiene



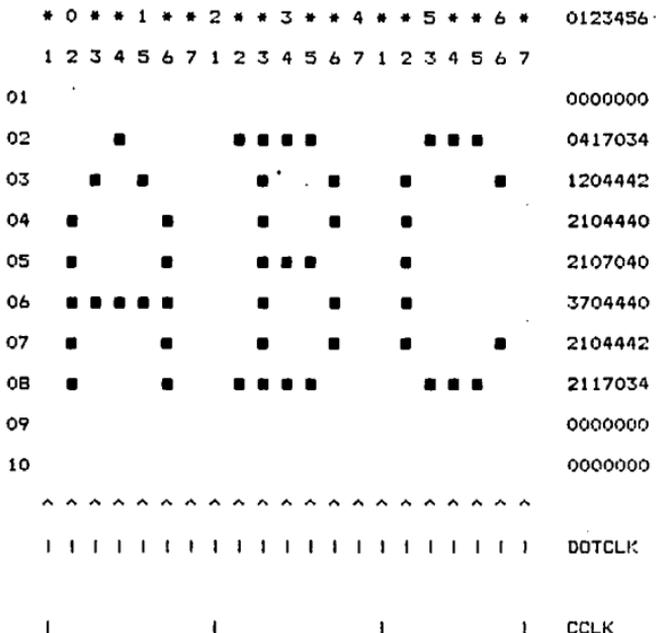
dos paginas de texto, por lo cual tiene un número de bytes igual al doble del número de caracteres que aparecen en la pantalla. En esta imagen de memoria cada byte contendrá el código ASCII del carácter que ocupa su misma posición en la pantalla. La única diferencia entre la imagen de video y la pantalla es que en esta última los caracteres aparecen ordenados en forma matricial, en tanto que en la imagen lo están en forma vectorial. Para encontrar la posición de un carácter de la pantalla en su imagen de video, lo único que debemos hacer es aplicar la siguiente ecuación:

$$\text{Posición} = \text{Loc. de inicio de VII} + X * Y + \text{PAGINA}$$

Donde X, Y son las coordenadas del carácter en la pantalla, tomando como origen el ángulo superior izquierdo de la misma (HOME).

El CRTC se encargará de leer la información contenida en VII, un renglón (55 bytes) a la vez y de determinar la posición de la pantalla en donde deben visualizarse los caracteres. Una vez se hayan leído los 55 bytes, la información se suministrará, carácter por carácter al VG, el cual contiene un generador de caracteres que se encarga de traducir el código ASCII de un carácter a la serie de puntos que lo forman en el monitor. Es evidente que debido a la exploración del monitor (línea por línea), los caracteres no se escriben completos de una sola vez, sino que se escribe primero la línea 1 de los 55 caracteres de un renglón, luego

la línea dos, y así sucesivamente hasta llegar a la línea siete:



En la anterior figura podemos ver un renglon de tres caracteres, junto con la representación en octal de cada línea, la forma en la que se transmite esta información desde el VG al monitor sería:

0000000.0417034.1204442.2104440.2107040.3704440.2104442.  
2117034.0000000.0000000.

En donde cada (.) representa un HRTC. Tambien se muestran los pulsos ascendentes de CCLK y de DOTCLK.

Los codigos ASCII de las letras A, B y C son respectivamente: 101, 102 y 103 en octal, por lo que la información contenida en la imagen de memoria seria:

1010

1020

1030

Y la información entregada por el CRTIC a el VG seria:

101 102 103.101 102 103.101 102 103.101 102 103.101 102 103.  
101 102 103.101 102 103.101 102 103.101 102 103.101 102 103.

En donde el (.) representa a un HRTC y los espacios entre codigos se intercalaron para facilitar su legibilidad.

Como puede verse la distancia entre el final de un HRTC y el principio de otro es de 3 caracteres, o lo que es lo mismo, de 21 puntos.

Advertase que un renglon real no consta de 3 caracteres, sino de 55 (385 puntos), aqui se utilizaron 3 solo por comodidad. La duración real del HRTC es de 10 caracteres (70 puntos).

Como se vio en el ejemplo anterior el CRTIC debe mandar la imagen de video al VG diez veces por caracter, si un renglon tiene 55 caracteres tendremos 550 transferencias en 53.78 uS, es decir 10.23 transferencias / uS. Si cada vez que hacemos una transferencia permitimos que el CRTIC accese la imagen de video el VTKBP no tendra tiempo para actualizar esta, por lo cual es preciso darle al CRTIC algun medio para que almacene temporalmente el renglon que haya leído de la imagen. Con esto solo se necesitara leer los 55 caracteres una sola vez, es decir que ocuparemos tan solo un 10.00% de tiempo de SYSBUS en relación con el procedimiento anterior, esto sin considerar que el hacer todo en un solo acceso consolidado optimiza el funcionamiento del sistema y disminuye la probabilidad de un error de memoria o de bus.

Los caracteres visualizables seran todos los caracteres visualizables del codigo USASCII (este codigo puede verse en el apendice A4).

A continuación se da una lista de estos caracteres junto con su codigo ASCII en hexadecimal, y la representación hexadecimal de cada linea:

000: 00H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■	■		3EH
03	■					■		22H
04	■					■		22H
05	■					■		22H
06	■					■		22H
07	■					■		22H
08	■	■	■	■	■	■		3EH
09								
10								

001: 01H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■	■		3EH
03	■							20H
04	■							20H
05	■							20H
06	■							20H
07	■							20H
08	■							20H
09								
10								

002: 02H

	1	2	3	4	5	6	7	
01								
02				■				0BH
03				■				0BH
04				■				0BH
05				■				0BH
06				■				0BH
07				■				0BH
08	■	■	■	■	■	■		3EH
09								
10								

003: 03H

	1	2	3	4	5	6	7	
01								
02				■				02H
03				■				02H
04				■				02H
05				■				02H
06				■				02H
07				■				02H
08	■	■	■	■	■	■		3EH
09								
10								

004: 04H

	1	2	3	4	5	6	7	
01								
02			■					10H
03				■				08H
04					■			04H
05			■	■	■	■		1EH
06				■				08H
07					■			04H
08						■		02H
09								
10								

005: 05H

	1	2	3	4	5	6	7	
01								
02		■	■	■	■	■		3EH
03		■					■	22H
04		■	■		■	■		33H
05		■		■		■		2AH
06		■	■		■	■		33H
07		■					■	22H
08		■	■	■	■	■		3EH
09								
10								

006: 06H

	1	2	3	4	5	6	7	
01								
02								00H
03					■			02H
04				■				04H
05	■		■					28H
06	■	■						30H
07	■							20H
08								00H
09								
10								

007: 07H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■					■		22H
04	■					■		22H
05	■	■	■	■	■			3EH
06		■		■				14H
07		■		■				14H
08	■	■		■	■			36H
09								
10								

008: 08H

	1	2	3	4	5	6	7	
01								
02				■				08H
03			■					10H
04	■	■	■	■	■			3CH
05		■				■		12H
06				■		■		0AH
07						■		02H
08						■		02H
09								
10								

009: 09H

	1	2	3	4	5	6	7	
01								
02								00H
03				■				08H
04					■			04H
05	■	■	■	■	■	■		3EH
06					■			04H
07			■					08H
08								00H
09								
10								

010: 0AH

	1	2	3	4	5	6	7
01							
02	■	■	■	■	■	■	
03							
04							
05	■	■	■	■	■	■	
06							
07							
08	■	■	■	■	■	■	
09							
10							

011: 0BH

	1	2	3	4	5	6	7	
01								
02								00H
03				■				08H
04				■				08H
05	■		■		■			2AH
06		■	■	■	■			1CH
07				■				08H
08								00H
09								
10								

012: 0CH

	1	2	3	4	5	6	7	
01								
02				■				0BH
03	■		■		■			2AH
04		■	■	■				1CH
05				■				0BH
06	■		■		■			2AH
07		■	■	■				1CH
08			■					0BH
09								
10								

013: 0DH

	1	2	3	4	5	6	7	
01								
02								00H
03				■				0BH
04			■					10H
05	■	■	■	■	■	■		3EH
06			■					10H
07				■				0BH
08								00H
09								
10								

014: 0EH

1 2 3 4 5 6 7

01

02     ■ ■ ■           1CH

03   ■           ■       22H

04   ■ ■     ■ ■       36H

05   ■   ■   ■       2AH

06   ■ ■     ■ ■       36H

07   ■           ■       22H

08     ■ ■ ■           1CH

09

10

015: 0FH

1 2 3 4 5 6 7

01

02     ■ ■ ■           1CH

03   ■           ■       22H

04   ■           ■       22H

05   ■   ■   ■       2AH

06   ■           ■       22H

07   ■           ■       22H

08     ■ ■ ■           1CH

09

10

016: 10H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■			3EH
03	■					■		22H
04	■					■		22H
05	■	■	■	■	■			3EH
06	■					■		22H
07	■					■		22H
08	■	■	■	■	■			3EH
09								
10								

017: 11H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■		■			■		2AH
04	■		■			■		2AH
05	■		■	■	■			2EH
06	■					■		22H
07	■					■		22H
08		■	■	■				1CH
09								
10								

018: 12H

1 2 3 4 5 6 7

01

02     ■ ■ ■     1CH

03   ■           ■     22H

04   ■           ■     22H

05   ■   ■ ■ ■     2EH

06   ■   ■   ■     2AH

07   ■   ■   ■     2AH

08       ■ ■ ■     1CH

09

10

019: 13H

1 2 3 4 5 6 7

01

02     ■ ■ ■     1CH

03   ■           ■     22H

04   ■           ■     22H

05   ■ ■ ■   ■     3AH

06   ■   ■   ■     2AH

07   ■   ■   ■     2AH

08       ■ ■ ■     1CH

09

10

020: 14H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■		■		■			2AH
04	■		■		■			2AH
05	■	■	■	■		■		3AH
06	■					■		22H
07	■					■		22H
08		■	■	■				1CH
09								
10								

021: 15H

	1	2	3	4	5	6	7	
01								
02								00H
03				■		■		0AH
04					■			04H
05	■		■		■		■	2AH
06	■	■						30H
07	■							20H
08								00H
09								
10								

022: 16H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03		■		■				14H
04		■		■				14H
05		■		■				14H
06		■		■				14H
07		■		■				14H
08	■	■		■	■			36H
09								
10								

023: 17H

	1	2	3	4	5	6	7	
01								
02					■			02H
03					■			02H
04					■			02H
05	■	■	■	■	■	■		3EH
06					■			02H
07					■			02H
08					■			02H
09								
10								

024: 18H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■			3EH
03	■					■		22H
04		■			■			14H
05			■					0BH
06		■			■			14H
07	■					■		22H
08	■	■	■	■	■	■		3EH
09								
10								

025: 19H

	1	2	3	4	5	6	7	
01								
02			■					0BH
03			■					0BH
04	■	■	■					1CH
05	■	■	■					1CH
06			■					0BH
07			■					0BH
08			■					0BH
09								
10								

026: 1AH

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■					■		22H
04	■							20H
05		■						10H
06			■					08H
07								00H
08			■					08H
09								
10								

027: 1BH

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■					■		22H
04	■					■		22H
05	■	■	■	■	■			3EH
06	■					■		22H
07	■					■		22H
08		■	■	■				1CH
09								

02B: 1CH

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■	■		3EH
03	■		■		■			2AH
04	■		■		■			2AH
05	■	■	■		■			3AH
06	■					■		22H
07	■					■		22H
08	■	■	■	■	■	■		3EH
09								
10								

029: 1DH

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■	■		3EH
03	■					■		22H
04	■					■		22H
05	■	■	■		■			3AH
06	■		■		■			2AH
07	■		■		■			2AH
08	■	■	■	■	■	■		3EH
09								
10								

030: 1EH

1 2 3 4 5 6 7

01

02 ■ ■ ■ ■ ■ 3EH

03 ■ ■ ■ ■ 22H

04 ■ ■ ■ ■ 22H

05 ■ ■ ■ ■ ■ 2EH

06 ■ ■ ■ ■ 2AH

07 ■ ■ ■ ■ 2AH

08 ■ ■ ■ ■ ■ 3EH

09

10

031: 1FH

1 2 3 4 5 6 7

01

02 ■ ■ ■ ■ ■ 3EH

03 ■ ■ ■ ■ 2AH

04 ■ ■ ■ ■ 2AH

05 ■ ■ ■ ■ ■ 2EH

06 ■ ■ ■ ■ 22H

07 ■ ■ ■ ■ 22H

08 ■ ■ ■ ■ ■ 3EH

09

10

032: 20H

1 2 3 4 5 6 7

01							
02							00H
03							00H
04							00H
05							00H
06							00H
07							00H
08							00H
09							
10							

033: 21H

1 2 3 4 5 6 7

01							
02		■					0BH
03		■					0BH
04		■					0BH
05		■					0BH
06		■					0BH
07							00H
08		■					0BH
09							
10							

034: 22H

	1	2	3	4	5	6	7	
01								
02		■		■				14H
03		■		■				14H
04		■		■				14H
05								00H
06								00H
07								00H
08								00H
09								
10								

035: 23H

	1	2	3	4	5	6	7	
01								
02		■		■				14H
03		■		■				14H
04		■	■	■	■	■	■	3EH
05		■		■				14H
06		■	■	■	■	■	■	3EH
07		■		■				14H
08		■		■				14H
09								
10								

036: 24H

	1	2	3	4	5	6	7	
01								
02				■				08H
03			■	■	■	■		1EH
04		■		■				2BH
05			■	■	■			1CH
06				■		■		0AH
07		■	■	■	■			3CH
08				■				0BH
09								
10								

037: 25H

	1	2	3	4	5	6	7	
01								
02		■	■					30H
03		■	■			■		32H
04					■			04H
05				■				0BH
06		■						10H
07		■			■	■		26H
08					■	■		06H
09								
10								

038: 26H

1 2 3 4 5 6 7

01							
02		■					10H
03	■		■				28H
04	■		■				28H
05		■					10H
06	■		■		■		2AH
07	■			■			24H
08		■	■		■		1AH
09							
10							

039: 27H

1 2 3 4 5 6 7

01							
02		■	■				18H
03		■	■				18H
04		■					10H
05	■						20H
06							00H
07							00H
08							00H
09							
10							

040: 28H

1 2 3 4 5 6 7

01							
02				■			04H
03			■				08H
04		■					10H
05		■					10H
06		■					10H
07			■				08H
08				■			04H
09							
10							

041: 29H

1 2 3 4 5 6 7

01							
02		■					10H
03			■				08H
04				■			04H
05				■			04H
06				■			04H
07			■				08H
08		■					10H
09							
10							

042: 2AH

	1	2	3	4	5	6	7	
01								
02				■				0BH
03		■		■		■		2AH
04			■	■	■			1CH
05		■	■	■	■	■		3EH
06			■	■	■			1CH
07		■		■		■		2AH
08				■				0BH
09								
10								

043: 2BH

	1	2	3	4	5	6	7	
01								
02								00H
03				■				0BH
04				■				0BH
05		■	■	■	■	■		3EH
06				■				0BH
07				■				0BH
08								00H
09								
10								

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

044: 2CH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04								00H
05			■	■				18H
06			■	■				18H
07			■					10H
08		■						20H
09								
10								

045: 2DH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04								00H
05		■	■	■	■	■		3EH
06								00H
07								00H
08								00H
09								
10								

046: 2EH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04								00H
05								00H
06								00H
07			■	■				18H
08			■	■				18H
09								
10								

047: 2FH

	1	2	3	4	5	6	7	
01								
02								00H
03						■		02H
04					■			04H
05				■				08H
06			■					10H
07		■						20H
08								00H
09								
10								

04B: 30H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■					■		22H
04	■			■	■			26H
05	■		■		■			2AH
06	■	■				■		32H
07	■					■		22H
08		■	■	■				1CH
09								
10								

049: 31H

	1	2	3	4	5	6	7	
01								
02			■					0BH
03		■	■					1BH
04			■					0BH
05			■					0BH
06			■					0BH
07			■					0BH
08		■	■	■				1CH
09								
10								

050: 32H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■					■		22H
04						■		02H
05		■	■	■				1CH
06	■							20H
07	■							20H
08	■	■	■	■	■	■		3EH
09								
10								

051: 33H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■					■		22H
04						■		02H
05			■	■				0CH
06						■		02H
07	■					■		22H
08		■	■	■				1CH
09								
10								

052: 34H

	1	2	3	4	5	6	7	
01								
02				■				04H
03			■	■				0CH
04			■		■			14H
05		■			■			24H
06		■	■	■	■	■		3EH
07					■			04H
08					■			04H
09								
10								

053: 35H

	1	2	3	4	5	6	7	
01								
02		■	■	■	■	■		3EH
03		■						20H
04		■	■	■	■			3CH
05						■		02H
06						■		02H
07		■				■		22H
08			■	■	■			1CH
09								
10								

054: 36H

	1	2	3	4	5	6	7	
01								
02				■	■			0CH
03			■					10H
04		■						20H
05		■	■	■	■			3CH
06		■				■		22H
07		■				■		22H
08			■	■	■			1CH
09								
10								

055: 37H

	1	2	3	4	5	6	7	
01								
02		■	■	■	■	■		3EH
03						■		02H
04				■				04H
05				■				08H
06		■						10H
07		■						20H
08		■						20H
09								
10								

056: 38H

1 2 3 4 5 6 7

01

02     ■ ■ ■           1CH

03   ■           ■     22H

04   ■           ■     22H

05     ■ ■ ■           1CH

06   ■           ■     22H

07   ■           ■     22H

08     ■ ■ ■           1CH

09

10

057: 39H

1 2 3 4 5 6 7

01

02     ■ ■ ■           1CH

03   ■           ■     22H

04   ■           ■     22H

05     ■ ■ ■ ■         1EH

06                 ■     02H

07                 ■     04H

08     ■ ■           18H

09

10

058: 3AH

	1	2	3	4	5	6	7	
01								
02								00H
03			■	■				18H
04			■	■				18H
05								00H
06			■	■				18H
07			■	■				18H
08								00H
09								
10								

059: 3BH

	1	2	3	4	5	6	7	
01								
02			■	■				18H
03			■	■				18H
04								00H
05			■	■				18H
06			■	■				18H
07			■					10H
08		■						20H
09								
10								

060: 3CH

1 2 3 4 5 6 7

01							
02				■			04H
03			■				08H
04		■					10H
05	■						20H
06		■					10H
07			■				08H
08				■			04H
09							
10							

061: 3DH

1 2 3 4 5 6 7

01							
02							00H
03							00H
04	■	■	■	■	■	■	3EH
05							00H
06	■	■	■	■	■	■	3EH
07							00H
08							00H
09							
10							

062: 3EH

	1	2	3	4	5	6	7	
01								
02			■					10H
03				■				08H
04					■			04H
05						■		02H
06				■				04H
07				■				08H
08			■					10H
09								
10								

063: 3FH

	1	2	3	4	5	6	7	
01								
02			■	■	■			1CH
03		■				■		22H
04						■		02H
05					■			04H
06				■				08H
07								00H
08				■				08H
09								
10								

064: 40H

1 2 3 4 5 6 7

01

02     ■ ■ ■     1CH

03   ■         ■     22H

04         ■     02H

05     ■ ■ ■     1AH

06   ■   ■   ■     2AH

07   ■   ■   ■     2AH

08     ■ ■ ■     1CH

09

10

065: 41H

1 2 3 4 5 6 7

01

02         ■     0BH

03    ■       ■     14H

04   ■         ■     22H

05   ■         ■     22H

06   ■ ■ ■ ■ ■     3EH

07   ■         ■     22H

08   ■         ■     22H

09

10

066: 42H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■				3CH
03		■				■		12H
04		■				■		12H
05		■	■	■				1CH
06		■				■		12H
07		■				■		12H
08	■	■	■	■				3CH
09								
10								

067: 43H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03	■						■	22H
04	■							20H
05	■							20H
06	■							20H
07	■						■	22H
08		■	■	■				1CH
09								
10								

068: 44H

1 2 3 4 5 6 7

01

02 ■■■■■ 3CH

03 ■ ■ 12H

04 ■ ■ 12H

05 ■ ■ 12H

06 ■ ■ 12H

07 ■ ■ 12H

08 ■■■■■ 3CH

09

10

069: 45H

1 2 3 4 5 6 7

01

02 ■■■■■ 3EH

03 ■ 20H

04 ■ 20H

05 ■■■■■ 3CH

06 ■ 20H

07 ■ 20H

08 ■■■■■ 3EH

09

10

070: 46H

1 2 3 4 5 6 7

01							
02	■	■	■	■	■	■	3EH
03	■						20H
04	■						20H
05	■	■	■	■	■		3CH
06	■						20H
07	■						20H
08	■						20H
09							
10							

071: 47H

1 2 3 4 5 6 7

01							
02		■	■	■	■	■	1EH
03	■						20H
04	■						20H
05	■		■	■	■	■	2EH
06	■					■	22H
07	■					■	22H
08		■	■	■	■	■	1EH
09							
10							

072: 48H

	1	2	3	4	5	6	7	
01								
02	■					■		22H
03	■					■		22H
04	■					■		22H
05	■	■	■	■	■	■		3EH
06	■					■		22H
07	■					■		22H
08	■					■		22H
09								
10								

073: 49H

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03			■					08H
04			■					08H
05			■					08H
06			■					08H
07			■					08H
08		■	■	■				1CH
09								
10								

074: 4AH

	1	2	3	4	5	6	7	
01								
02					■			02H
03					■			02H
04					■			02H
05					■			02H
06	■					■		22H
07		■	■	■	■			1CH
08								
09								
10								

075: 4BH

	1	2	3	4	5	6	7	
01								
02	■					■		22H
03	■				■			24H
04	■		■					28H
05	■	■						30H
06	■			■				28H
07	■				■			24H
08	■					■		22H
09								
10								

07H

	1	2	3	4	5	6	7	
01								
02	■							20H
03	■							20H
04	■							20H
05	■							20H
06	■							20H
07	■							20H
08	■	■	■	■	■	■		3EH
09								
10								

077: 4DH

	1	2	3	4	5	6	7	
01								
02	■				■			22H
03	■	■		■	■			36H
04	■		■		■			2AH
05	■		■		■			2AH
06	■				■			22H
07	■				■			22H
08	■				■			22H
09								
10								

078: 4EH

	1	2	3	4	5	6	7	
01								
02		■				■		22H
03		■	■			■		32H
04		■		■		■		2AH
05		■			■	■		26H
06		■				■		22H
07		■				■		22H
08		■				■		22H
09								
10								

079: 4FH

	1	2	3	4	5	6	7	
01								
02			■	■	■			1CH
03		■				■		22H
04		■				■		22H
05		■				■		22H
06		■				■		22H
07		■				■		22H
08			■	■	■			1CH
09								
10								

080: 50H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■			3CH
03	■					■		22H
04	■					■		22H
05	■	■	■	■	■			3CH
06	■							20H
07	■							20H
08	■							20H
09								
10								

081: 51H

	1	2	3	4	5	6	7	
01								
02		■	■	■	■			1CH
03	■					■		22H
04	■					■		22H
05	■					■		22H
06	■	■	■	■				2AH
07	■			■				24H
08		■	■	■				1AH
09								
10								

082: 52H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■			1CH
03	■					■		22H
04	■					■		22H
05	■	■	■	■	■			1CH
06	■		■					28H
07	■				■			24H
08	■					■		22H
09								
10								

083: 53H

	1	2	3	4	5	6	7	
01								
02		■	■	■	■			1CH
03	■					■		22H
04	■							20H
05		■	■	■	■			1CH
06						■		02H
07	■					■		22H
08		■	■	■	■			1CH
09								
10								

084: 54H

	1	2	3	4	5	6	7	
01								
02	■	■	■	■	■	■		3EH
03			■					0BH
04			■					0BH
05			■					0BH
06			■					0BH
07			■					0BH
08			■					0BH
09								
10								

085: 55H

	1	2	3	4	5	6	7	
01								
02	■				■			22H
03	■				■			22H
04	■				■			22H
05	■				■			22H
06	■				■			22H
07	■				■			22H
08		■	■	■				1CH
09								
10								

086: 56H

	1	2	3	4	5	6	7	
01								
02	■					■		22H
03	■					■		22H
04	■					■		22H
05		■			■			14H
06		■			■			14H
07			■					08H
08			■					08H
09								
10								

087: 57H

	1	2	3	4	5	6	7	
01								
02	■					■		22H
03	■					■		22H
04	■					■		22H
05	■					■		22H
06	■	■			■			2AH
07	■	■			■	■		36H
08	■					■		22H
09								
10								

088: 58H

1 2 3 4 5 6 7

01

02 ■            ■            22H

03 ■            ■            22H

04    ■        ■            14H

05            ■            08H

06    ■        ■            14H

07 ■            ■            22H

08 ■            ■            22H

09

10

089: 59H

1 2 3 4 5 6 7

01

02 ■            ■            22H

03 ■            ■            22H

04    ■        ■            14H

05            ■            08H

06            ■            08H

07            ■            08H

08            ■            08H

09

10

090: 5AH

1 2 3 4 5 6 7

01

02 ■ ■ ■ ■ ■ 3EH

03 ■ 02H

04 ■ 04H

05 ■ 08H

06 ■ 10H

07 ■ 20H

08 ■ ■ ■ ■ ■ 3EH

09

10

091: 5BH

1 2 3 4 5 6 7

01

02 ■ ■ ■ 1CH

03 ■ 10H

04 ■ 10H

05 ■ 10H

06 ■ 10H

07 ■ 10H

08 ■ ■ ■ 1CH

09

10

092: 5CH

	1	2	3	4	5	6	7	
01								
02								00H
03		■						20H
04			■					10H
05				■				08H
06					■			04H
07						■		02H
08								00H
09								
10								

093: 5DH

	1	2	3	4	5	6	7	
01								
02		■	■	■				1CH
03				■				04H
04				■				04H
05				■				04H
06				■				04H
07				■				04H
08		■	■	■				1CH
09								
10								

094: 5EH

	1	2	3	4	5	6	7	
01								
02				■				0BH
03			■		■			14H
04		■				■		22H
05								00H
06								00H
07								00H
08								00H
09								
10								

095: 5FH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04								00H
05								00H
06								00H
07								00H
08		■	■	■	■	■	■	3EH
09								
10								

096: 60H

1 2 3 4 5 6 7

01

02            ■ ■            00H

03            ■ ■            00H

04            ■            1CH

05                ■            02H

06                       1EH

07                       22H

08                       1EH

09

10

097: 61H

1 2 3 4 5 6 7

01

02                       00H

03                       00H

04            ■ ■ ■            1CH

05                       ■            02H

06            ■ ■ ■ ■            1EH

07     ■                ■            22H

08            ■ ■ ■ ■            1EH

09

10

098: 62H

	1	2	3	4	5	6	7	
01								
02	■							20H
03	■							20H
04	■		■	■				2CH
05	■	■				■		32H
06	■					■		22H
07	■	■				■		32H
08	■		■	■				2CH
09								
10								

099: 63H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04			■	■	■			1CH
05	■					■		22H
06	■							20H
07	■					■		22H
08		■	■	■				1CH
09								
10								

100: 64H

	1	2	3	4	5	6	7	
01								
02					■			02H
03					■			02H
04		■	■		■			1AH
05	■			■	■			26H
06	■					■		22H
07	■			■	■			26H
08		■	■			■		1AH
09								
10								

101: 65H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04			■	■	■			1CH
05	■					■		22H
06	■	■	■	■	■	■		3EH
07	■							20H
08		■	■	■				1CH
09								
10								

102: 66H

	1	2	3	4	5	6	7	
01								
02				■				40H
03			■		■			0AH
04				■				0BH
05		■	■	■				1CH
06				■				0BH
07				■				0BH
08				■				0BH
09								
10								

103: 67H

	1	2	3	4	5	6	7	
01								
02		■	■		■			1AH
03		■			■	■		26H
04		■			■	■		26H
05			■	■		■		1AH
06						■		02H
07		■				■		22H
08			■	■	■			1CH
09								
10								

104: 68H

	1	2	3	4	5	6	7	
01								
02	■							20H
03	■							20H
04	■		■	■				20H
05	■	■			■			32H
06	■					■		22H
07	■					■		22H
08	■					■		22H
09								
10								

105: 69H

	1	2	3	4	5	6	7	
01								
02				■				08H
03								00H
04		■	■					18H
05				■				08H
06				■				08H
07				■				08H
08		■	■	■				1CH
09								
10								

106: 6AH

	1	2	3	4	5	6	7	
01								
02					■			02H
03								00H
04					■			02H
05					■			02H
06					■			02H
07	■					■		22H
08		■	■	■	■			1CH
09								
10								

107: 6BH

	1	2	3	4	5	6	7	
01								
02	■							20H
03	■							20H
04	■				■			24H
05	■			■				28H
06	■	■						30H
07	■			■				28H
08	■				■			24H
09								
10								

108: 6CH

	1	2	3	4	5	6	7	
01								
02		■	■					18H
03			■					08H
04			■					08H
05			■					08H
06			■					08H
07			■					08H
08		■	■	■				1CH
09								
10								

109: 6DH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04		■	■	■				34H
05		■		■		■		2AH
06		■		■		■		2AH
07		■		■		■		2AH
08		■		■		■		2AH
09								
10								

110: 6EH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04	■		■	■				2CH
05	■	■			■			32H
06	■					■		22H
07	■					■		22H
08	■					■		22H
09								
10								

111: 6FH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04			■	■	■			1CH
05	■					■		22H
06	■					■		22H
07	■					■		22H
08			■	■	■			1CH
09								
10								

112: 70H

	1	2	3	4	5	6	7	
01								
02	■		■	■				2CH
03	■	■			■			32H
04	■				■			22H
05	■	■			■			32H
06	■		■	■				2CH
07	■							20H
08	■							20H
09								
10								

113: 71H

	1	2	3	4	5	6	7	
01								
02			■	■		■		1AH
03	■				■	■		26H
04	■					■		22H
05	■			■	■			26H
06		■	■			■		1AH
07						■		02H
08					■			02H
09								
10								

114: 72H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04		■		■	■			2CH
05		■	■				■	32H
06		■						20H
07		■						20H
08		■						20H
09								
10								

115: 73H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04			■	■	■	■	■	1EH
05		■						20H
06			■	■	■			1CH
07							■	02H
08		■	■	■	■			3CH
09								
10								

116: 74H

	1	2	3	4	5	6	7	
01								
02				■				08H
03				■				08H
04		■	■	■				1CH
05				■				08H
06				■				08H
07				■		■		0AH
08					■			04H
09								
10								

117: 75H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04		■				■		22H
05		■				■		22H
06		■				■		22H
07		■			■	■		26H
08			■	■		■		1AH
09								
10								

118: 76H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04	■					■		22H
05	■					■		22H
06	■					■		22H
07		■			■			14H
08			■					08H
09								
10								

119: 77H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04	■					■		22H
05	■					■		22H
06	■	■			■			2AH
07	■	■			■			2AH
08		■			■			14H
09								
10								

120: 78H

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04		■				■		22H
05			■		■			14H
06				■				08H
07			■		■			14H
08		■				■		22H
09								
10								

121: 79H

	1	2	3	4	5	6	7	
01								
02		■				■		22H
03		■				■		22H
04		■				■		22H
05			■	■	■	■		1EH
06						■		02H
07		■				■		22H
08			■	■	■			1CH
09								
10								

122: 7AH

	1	2	3	4	5	6	7	
01								
02								00H
03								00H
04	■	■	■	■	■	■		3EH
05				■				04H
06				■				0BH
07			■					10H
08	■	■	■	■	■	■		3EH
09								
10								

123: 7BH

	1	2	3	4	5	6	7	
01								
02				■				04H
03				■				0BH
04				■				0BH
05			■					10H
06				■				0BH
07				■				0BH
08				■				04H
09								
10								

124: 7CH

	1	2	3	4	5	6	7	
01								
02			■					08H
03			■					08H
04			■					08H
05								
06			■					08H
07			■					08H
08			■					08H
09								
10								

125: 7DH

	1	2	3	4	5	6	7	
01								
02			■					10H
03				■				08H
04				■				08H
05					■			04H
06				■				08H
07				■				08H
08			■					10H
09								
10								

126: 7EH

	1	2	3	4	5	6	7	
01								
02		■						10H
03	■		■		■			2AH
04				■				04H
05								00H
06								00H
07								00H
08								00H
09								
10								

127: 7FH

	1	2	3	4	5	6	7	
01								
02		■		■				14H
03	■		■		■			2AH
04		■		■				14H
05	■		■		■			2AH
06		■		■				14H
07	■		■		■			2AH
08		■		■				14H
09								
10								

#### 4.2.2.2- Modo grafico 1.

El modo grafico 1 (GR1) hace uso del mismo formato de pantalla que el modo alfabetico. La pantalla se divide en 1320 bloques bicolors como el que se muestra:

```
      1 2 3 4 5 6 7
01 ■■■■■■■■ |
02 ■■■■■■■■ |
03 ■■■■■■■■ | Color X
04 ■■■■■■■■ |
05 ■■■■■■■■ |
06 * * * * * |
07 * * * * * |
08 * * * * * | Color Y
09 * * * * * |
10 * * * * * |
```

Como se ve cada bloque consta de 35 puntos de color organizados en una matriz de 5x7, en una pantalla tendremos 2640 bloques de color, definidos por 1320 bytes de memoria, cada byte contiene la especificación de dos bloques superpuestos:

```
      7 6 5 4 3 2 1 0
      X X X X Y Y Y Y
```

El nibble X especifica el color del bloque superior y el nibble Y especifica el color del bloque inferior. Como puede verse los dos bloques superpuestos ocupan el campo que en el modo alfabetico ocupa un carácter, por lo cual los contadores de caracteres (bloques) por renglon y renglones por pantalla del CRTC no necesitan ser redefinidos al pasar de un modo al otro, y pueden mezclarse bloques en modo GRA con bloques en modo GR1 con solo habilitar o deshabilitar el generador de caracteres del VG. La forma en la que puede hacerse esto se explicara al llegar a la sección de diseño de la arquitectura del CRTC y del VG.

La imagen de video para este modo es VII, al igual que para el modo alfabetico, y la posición de un par de bloques en la pantalla esta dada por la posición del su byte imagen en VII.

Son ocho los colores que se pueden manejar en este modo, y se definen como sigue:

C	R	G	B	NOMBRE	Y	I	Q	VY	VI	VQ
0	0	0	0	BK Negro	0.00	0.00	0.00	0.00	0.00	0.00
1	0	0	1	B Azul	0.11	-0.32	0.31	0.36	-1.36	1.02
2	0	1	0	G Verde	0.59	-0.28	-0.52	1.95	-0.92	-1.72
3	0	1	1	CY Cyan	0.70	-0.60	-0.21	2.31	-1.98	-0.69
4	1	0	0	R Rojo	0.30	0.60	0.21	0.99	1.98	0.69
5	1	0	1	MG Magenta	0.41	0.28	0.52	1.35	0.92	1.72

6	1 1 0	YW Amarillo	0.89	0.32	-0.31	2.94	1.06	-1.02
7	1 1 1	W Blanco	1.00	0.00	0.00	3.30	0.00	0.00

El formato de cualquier byte de VII debe ser:

7 6 5 4 3 2 1 0  
O R G B O R G B

En la tabla anterior se puede ver el número del color; su código RGB; su abreviatura; su nombre; sus valores Y, I, Q en una escala de -1 a +1; y el voltaje que se obtiene para él a la salida del generador de video, en una escala de -3.3 a +3.3 V.

Para calcular los valores de I, Q e Y se utilizaron las ecuaciones vistas en la sección 4.2, representadas por la siguiente matriz:

$$\begin{aligned} |Y| &= | 0.30 \quad 0.59 \quad 0.11 | |R| \\ |I| &= | 0.60 \quad -0.28 \quad -0.32 | |G| \\ |Q| &= | 0.21 \quad -0.52 \quad 0.31 | |B| \end{aligned}$$

En la sección sobre el generador de video puede observarse el circuito que nos permite obtener las señales I, Q, Y.

El VG tiene tres modos de funcionamiento: GRA, GR1 y GRO que corresponden al modo gráfico en el que opera la terminal.

La selección del modo en el cual se opera la hace el CRTC de acuerdo a las instrucciones que reciba de parte del VTKBP, estas instrucciones se dan en la forma de un atributo de campo para el CRTC. La definición de atributo de campo se dara en la sección de diseño de la arquitectura del CRTC, de momento solo diremos que es preciso que el bit 7 de cada byte de VII permanezca igual a cero para evitar seleccionar un FA o un CA, y que el primer byte del VII (que no es visualizable), sera 8CH, lo cual selecciona GRI:

```
7 6 5 4 3 2 1 0
1 0 0 0 1 1 0 0
```

Para seleccionar GRA es preciso que este byte sea 80H. Este primer byte se conoce como FAS (Field Attribute selector), y su utilización se explica a fondo dentro de la sección de arquitectura. Dentro de VII puede existir más de un FAS, y pueden encontrarse, así mismo, otros bytes llamados CAS (Character Attribute Selector). Cualquier byte cuyo bit 7 sea igual a cero se considerara como un FAS o CAS.

#### 4.2.2.3- Modo grafico 0:

El modo grafico 0 utiliza unos formatos distintos a los utilizados por GRI y ALPHA, sin embargo las señales de control y de video para este modo siguen siendo generadas por CRTC y VG. La señal de luminancia se obtiene de VIO, en tanto que el color es determinado con anterioridad por el CRTC de

acuerdo a un FAS. En este modo se instruye al CRTC para que un renglon siga teniendo 55 caracteres visibles, pero en lugar de que cada caracter se defina como una matriz de 10x7 se le define ahora como un vector de 1x7, es decir que un renglon tiene unicamente una linea. Esta definición nos permite utilizar la misma circuiteria para este modo que para los dos anteriores.

En este modo la pantalla se considera como una matriz de 240 renglones por 385 columnas. El total de puntos en esta matriz es de 92,400. Cada punto podra estar encendido o apagado, y en el primer caso podra presentar uno de cuatro colores, que son: +I, -I, +Q, -Q (naranja-rojizo, cyan, purpura, verde-amarillo).

El estado de encendido/apagado de un determinado punto se representara mediante un bit, pero solo se manejaran 7 puntos por byte, puesto que el bit 7 se utilizara para determinar que par de colores seran los utilizados en los puntos representados por ese byte (byte Q o byte I).

El color de un punto dependera de su posición en la pantalla, si el bit que lo representa en la imagen de memoria es 0, el punto sera negro, si ese bit es 1 y el bit 7 es uno, el color del punto sera Q si esta en una columna par y -Q si esta en una columna impar. Si el bit 7 es cero, los colores seran I, -I; para columnas pares e impares respectivamente. Es decir que en las columnas pares tendremos el color

primario, y en las impares su complemento.

El tamaño de VIO es de  $2 * 92,400 / 7 = 26,400$  bytes (25.78K).

La selección de GRO se hace también por medio de un FAS, cada vez que el CRTC lee un byte igual a 84H interpreta el código que sigue como código gráfico con la señal Q y -Q habilitadas. Si el FAS es igual a 8BH, se habilitan las señales I y -I. Estas definiciones permanecen hasta que el CRTC lea otro FAS que las cambie. Este modo de manejar los gráficos como atributos de campo nos permite intercambiarlos sin problema, aun cuando complica un poco el software del VTKBP, al no definir áreas específicas para VII y VIO.

#### 4.3- DISEÑO DEL GENERADOR DE VIDEO:

El generador de video (VG) tiene como función el generar las señales de video que representan la información a visualizar, formateandolas de acuerdo al estandar RS170. Estas señales son tres: Y, I y D ; y deben ser sumadas a las señales de sincronia y regreso procedentes del CRTC (SYNC, HRTC y VRTC).

El VG no tiene acceso al SYSEBUS, por lo cual su comunicación con el VTKBP se lleva a cabo por medio del CRTC. El canal de comunicación entre el CRTC y el VG es el bus privado de video 1 (PVB1). Tampoco tiene el VG generación interna de las diversas señales de tiempo que requiere, sino que estas las recibe del CRTC.

#### 4.3.1- Arquitectura del VG.

Con la palabra arquitectura nos referimos a la estructura lógica del VG, a los grupos que lo forman y a sus diversos registros, sin preocuparnos por la implementación real de estos. En una palabra, arquitectura del VG es el concepto del VG, su idea.

##### 4.3.1.1- Grupos del VG.

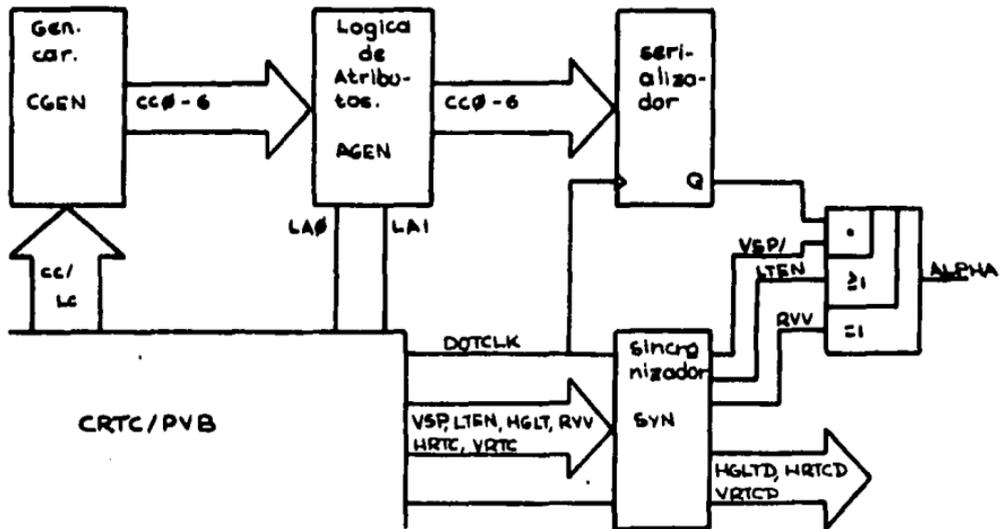
Con la palabra grupo nos referimos a una reunión de elementos que efectúan una función perfectamente determinada.

Los grupos del VG son:

1. Generador de caracteres (CGEN)
2. Sincronizador de video (VSYN)
3. Sincronizador de color (COLORSYN)
4. Generador de atributos (AGEN)
5. Generador de puntos (DOTGEN)
6. Multiplexor de video (VMUX)
7. Matriz de color (COLORMAT)
8. Sumador de video (VADD)
9. Inyector de color (CLBINJ)

Su relación puede verse en la figura 4.3.1/1.

Funciones:



#### 1. CGEN:

Su función es traducir el código ASCII que el VG recibe del CRTC en el modo ALPHA a una serie de puntos encendidos o apagados que forman el carácter tal como es visualizado. La información que entrega este grupo sigue estando en paralelo y es DOTGER quien la serializa. Figura 4.3.1/2

#### 2. VSYN:

Su función es sincronizar las diferentes señales de control generadas por el CRTC con las señales de video, para evitar que haya un desfase entre ellas debido al diferente tiempo requerido para sintetizarlas. Este grupo se encarga de que todos los cambios de estado que ocurren en el VG sean simultáneos, convirtiéndolo así en un circuito sincrónico. Las señales que se sincronizan aquí son VSP, LTEN, RVV, HBLT, SYNC, HRTC, VRTC, R, G y B. Figura 4.3.1/3

#### 4. AGEN:

Su función es generar los diferentes atributos visuales que pueden darse en el modo ALPHA, deshabilitando a CGER. Figura 4.3.1/2

#### 5. DOTGEN:

Su función es transformar la información de punto encendido/apagado que le suministra el CGER (modo ALPHA) de paralelo a serie. Figura 4.3.1/2

#### 6. VMUX:

Su función es canalizar una de las tres señales de video (ALPHA, GRI o GRO) hacia COLORMAT de acuerdo a la información que CRTC le suministra por las líneas GPAO y GPAI. Este grupo es el que permite la existencia de tres modos gráficos.

Figura 4.3.1/4

#### 7. COLORMAT:

Su función es transformar las señales digitales R, G y B que especifican el color de un punto entre ocho posibles a tres señales analógicas: I, Q e Y. Figura 4.3.1/4

#### 8. VADD:

Su función es sumar las señales I, Q, Y, SYNC, HRTC y VRTC en una sola señal de video compuesto RS170. Figura

4.3.1/4

#### 9. CLBINJ:

Su función es inyectar un fragmento de la señal SIN3.58M a la señal de video compuesto en los momentos en los que le es ordenado por la señal CLBDN procedente del CRTC.

#### 4.3.1.2- Canal de comunicaciones con el CRTC.

El VG recibe toda la información necesaria para su funcionamiento del CRTC, que la genera de acuerdo a las instrucciones provenientes del VTKBP.

El canal de comunicación por el cual se transmite toda esta información se llama PVB1 (Private Video Bus 1), y esta formado por las siguientes 29 señales digitales (TTL):

1. GPA0:

General Purpose Attribute 0. Esta señal, generada por el CRTIC, se utiliza en conjunción con GPA1, para seleccionar (mediante VMUX) el modo de operación de VB (ALPHA, GRO o GR1).

2. GPA1:

General Purpose Attribute 1. Generado por el CRTIC.

3. CCO - CC6:

Estas siete señales provenientes del CRTIC son utilizadas en el modo GRA y en el modo GRO. En el primer caso contienen el código ASCII del carácter actualmente visualizado, código que será utilizado por el CGEN para generar la señal de video ALPHA. En el segundo caso las señales se utilizan por el VMUX para determinar el color del bloque a visualizar, de acuerdo al siguiente esquema:

CC	6	5	4	3	2	1	0
	R	G	B	X	R	G	B

CC6, 5 y 4 determinan el color del bloque superior y CC2, 1 y 0 el del bloque inferior. CC3 no es tomado en cuenta.

4. R1, G1 y B1:

Estas tres señales (Red, Green, Blue) procedentes del CRTC son utilizadas por COLORMAT para generar las señales de video en el modo GRI.

5. SYNC:

Señal de sincronía compuesta horizontal - vertical generada por el CRTC.

6. HRTC:

Señal de regreso horizontal generada por el CRTC.

7. VRTC:

Señal de regreso vertical generada por el CRTC.

8. CLBON:

Color Burst On. Esta señal, generada por el CRTC esta activa durante el periodo de insercción del color burst.

9. VSP:

Video Supression. Cuando esta señal, generada por el CRTC, esta en su estado activo, la señal Y es forzada a negro pico, a menos que la señal LTEN este activa.

10. LTEN:

Light Enable. Cuando esta señal esta en su estado activo la señal Y es forzada a blanco pico.

11. RVV:

Reverse Video. Cuando esta señal esta en su estado activo la señal Y es invertida.

12. HGLT:

High Light. Cuando esta señal esta en su estado activo la señal Y es forzada a blanco pico. Es igual a LTEN, pero su prioridad es menor a VSP.

13. LA0, LA1:

Line Attributes. Estas señales seleccionan los diferentes atributos de caracter que pueden definirse.

14. LCO - LC2:

Line Counter. Estas lineas son utilizadas por el CGEN para determinar la linea que esta siendo visualizada dentro de un determinado renglon, pudiendo asi generar la dirección correcta para la EPROM de caracteres.

15. SQ3.58M:

Square 3.58 MHz. Señal cuadrada de 3.58 MHz.

16. SIN3.58M:

Sinusoid 3.58 MHz. Señal sinusoidal de 3.58 MHz.

17. DOTCLK:

Dot Clock. Tren de pulsos de 7.16 MHz utilizada para

definir la duración de un determinado pixel de la imagen de video.

#### 18. CCLK:

Character Clock. Esta señal aproximadamente cuadrada de 1.0227271 MHz es utilizada por CGEN y DOTGEN. Cada periodo de esta señal marca la transición de un caracter a otro en la pantalla.

#### 19. QA, QB:

Señales obtenidas al mismo tiempo que CCLK y utilizadas por CGEN.

#### 4.3.1.3- Canal de comunicaciones con el CRT.

La salida del VG esta compuesta por una sola señal de video colorplexed llamada VIDEO. Esta señal es la suma analógica de las señales Y, I, Q, SYNC, VRTC y HRTC, y esta totalmente de acuerdo al estandar RS170. Figura 4.3.1/5

#### 4.3.1.4- Programación del VG.

El modo de operación del VG es parcialmente programable através de las líneas GPA0 y GPA1, de acuerdo a la siguiente tabla:

GPA1	GPA0	HRTCcount	Modo de operación.
------	------	-----------	--------------------

0	0	C<5	GR1. Bloque superior.
0	0	C>=5	GR1. Bloque inferior.
0	1	X	GR0
1	0	X	ALPHA
1	1	X	Ilegal.

Ninguna otra característica del VG puede ser programada via software. La programación del modo de operación que acabamos de mostrar unicamente determina cual de los tres grupos (ALPHA, GR1 y GR0) de tres señales (R, G y B) que llegan al VMUX es seleccionado como salida de este.

#### 4.3.2- Diseño del circuito del VG.

El VG fue implementado totalmente con circuitos SSI y MSI, tecnología LSTTL. El unico integrado LSI utilizado fue la EPROM MCM2716, tecnología NMOS, utilizado por el CGEN como tabla de traducción. Esto fue hecho de este modo debido a que en el momento de realizar este diseño no existia ningun circuito integrado LSI o VLSI que realizara las funciones buscadas.

El circuito no requiere más que una sola fuente de alimentación de +5v, y esta desarrollado totalmente de acuerdo a las especificaciones descritas en las secciones 4.1 y 4.2, cumpliendo totalmente con la arquitectura descrita en la sección 4.3.2.

##### 4.3.2.1- Diseño de CGEN, AGEN y DOTGEN.

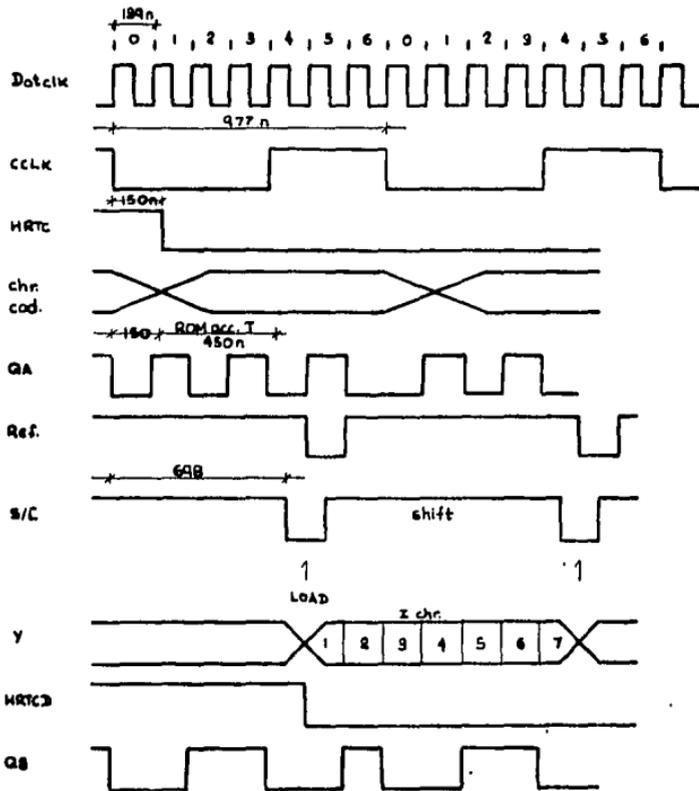
CGEN, AGEN y DOTGEN reciben de CRTC las siguientes señales: CC0-CC6, DOTCLK, CCLK, QB, QA, LA1, LA0 y LC0-LC2. De VSYN obtienen las señales VSPD, LTEND y RVVD. La señal neta obtenida de estos tres circuitos es llamada ALPHA y es la señal de luminancia utilizada en modo ALPHA. Figura 4.3.2/1.

En el 2716 se almacenan las representaciones de todos los caracteres visualizables en modo ALPHA, de acuerdo a la representación dada en la seccion 4.2. El tiempo de acceso de

esta memoria es de 450 nS (Figura 4.3.2/2), lo cual nos obliga a retardar las otras señales que vayan a ser utilizadas para la generación de la señal de video colorplexed y que no deban ser procesadas por CGEN, para evitar que los atributos de una determinada señal, su color, y su sincronía, llegen al CRT antes que su luminancia. El tiempo de retardo utilizado es igual a medio periodo de CCLK más un periodo de DOTCLK, es decir 628.56 nS aproximadamente, este tiempo toma en cuenta el retardo máximo en la 2716 (450 nS) más el retardo de las señales CC0-CC6 en el CRT con respecto a las demás señales de video (150 nS). La señal de salida de CGEN sigue estando fuera de sincronía con las demás señales de video, puesto que ahora puede aparecer antes de que lo hagan estas, pero esto no es de importancia, ya que DOTGEN, formado por el Shift Register de entrada en paralelo y salida en serie 74LS166 y por el FFD 74LS74 sincroniza la serie de puntos con DOTCLK, señal con la que están sincronizadas todas las otras señales de video mediante VSYN. El retraso introducido por AGEN no puede ser mayor a 20 nS, con lo cual aun quedan 8.56 nS de margen para nuestras señales (ver plano 1).

AGEN esta formado por una serie de compuertas que generan la serie de caracteres semigráficos mostrados en la sección sobre atributos de campo y de línea, y esta controlado por LA1 y LA0.

La señal ALPHA esta dada por la siguiente ecuación:



ALPHA = RVVD xor (LTEND or (QH and not VSPD))

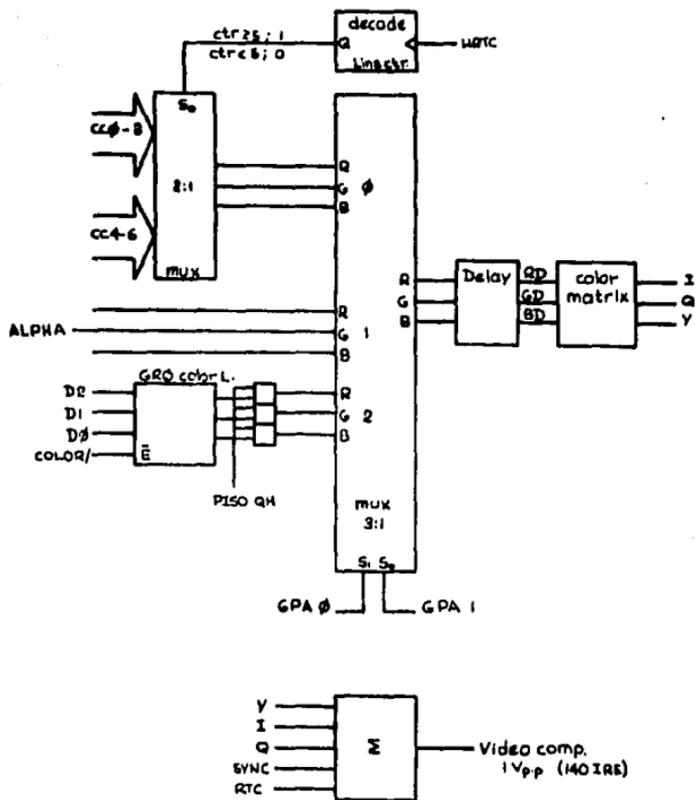
#### 4.3.2.2- Diseño de VSYN:

VSYN recibe del CRTIC las señales VSP, LTEN, RVV, HGLT, SYNC, HRTC, VRTC, R, G y B y las procesa obteniendo las señales VSPD, LTEND, RVVD, HGLTD, SYNCD, HRTCD, VRTCD, RD, GD y BD; que tienen un retardo de 628.56 nS con respecto a las originales y que están perfectamente en sincronía con DOTCLK. Esta sincronización es necesaria, pues todas las señales utilizadas por VG para generar VIDEO deben ocurrir al mismo tiempo. En la plano 1 vemos el circuito de VSYN, que está implementado por dos FFD octales 74LS374 y dos FFD cuádruples 74LS175. Como la señal ALPHA ya está previamente sincronizada por DOTGER, el retardo de color generado por los 74LS175 es deshabilitado mediante un multiplexor 74LS157 para evitar un doble retraso en este modo.

#### 4.3.2.3- Diseño de VMUX.

VMUX está totalmente implementado con circuitos SSI y MSI LSTTL. Su funcionamiento está descrito por la siguiente tabla:

GPA1	GPA0	HRTCCOUNT	STB	Canal seleccionado
0	0	C<5	1	1 GR1 bloque superior
0	0	C>=5	0	0 GR1 bloque inferior



0	1	X	X	2 GRO
1	0	X	X	3 ALFHA
1	1	X	X	Ilegal

La función de multiplexaje se lleva a cabo mediante dos multiplexores 74LS153 (multiplexor doble 4/1), las entradas de selección de canal en el multiplexor están rotuladas como A y B, las ecuaciones que determinan estas dos entradas se obtienen de la siguiente tabla de verdad:

GPA1	GPA0	STB	B A
0	0	1	0 1
0	0	0	0 0
0	1	X	1 0
1	0	X	1 1
1	1	X	Ilegal

$$A = \text{GPA0} / * (\text{GPA1} + \text{STB} * \text{GPA1} /)$$

$$B = \text{GPA1} + \text{GPA0}$$

GPA0 y GPA1 se obtienen de CRTIC, en tanto que STB es generado internamente en VMUX, mediante un contador de líneas, siendo STB = 1 para las cinco líneas superiores e igual a 0 para las cinco líneas inferiores, esto con objeto de poder realizar un cambio de color del bloque superior al bloque inferior en GR1, ya que este cambio no es controlado por el CRTIC (que de hecho no puede distinguir entre ambos bloques), sino por VMUX. Este circuito aparece en el plano 1.

#### 4.3.2.4- Diseño de COLORMAT, VADD y CLB INJ.

Estos circuitos tienen el objeto de convertir las señales digitales RD, GD, BD, SYNC, HRTCD y VRTCD en una sola señal analógica llamada VIDE, totalmente de acuerdo al estándar RS170. Estos tres circuitos se implementan con dos tipos de transistores, el 2N2222A y el 2N3251A, cuyas especificaciones se dan a continuación:

2N2222A: NPN

1. Capsula: TO-18
2.  $V_{cbo}$  (min): 75 V
3.  $V_{ceo}$  (min): 50 V
4.  $V_{ebo}$  (min): 6 V
5.  $I_{cbo}$  (max): 10 mA @ 60 V
6.  $h_{FE}$  (min): 50 @ 1 mA
7.  $V_{ce}$  (sat): 0.3 V @ 150 mA
8.  $f_T$  (min): 250 MHz @ 20 mA

Especificaciones más detalladas pueden encontrarse en el DISCRETE DATABOOK de National Semiconductor, Proceso 20.

2N3251A: PNP

1. Capsula: TO-18

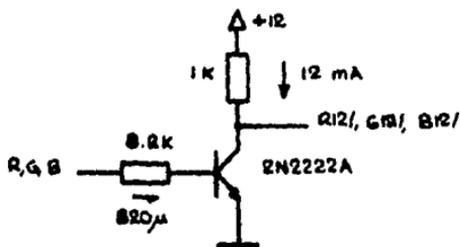
2.  $V_{cbo}$  (min): 60 V
3.  $V_{ceo}$  (min): 60 V
4.  $V_{ebo}$  (min): 5 V
5.  $I_{cbo}$  (max): no especificada.
6.  $h_{FE}$  (min): 90 @ 1 mA, 1 V
7.  $V_{ce}$  (sat): 0.25 @ 10 mA
8.  $f_T$  (min): 300 MHz @ 10 mA
9. NF (min) = 6 dB

Debemos obtener los siguientes resultados:

$$\begin{aligned} |Y| &= 10.30 & 0.59 & 0.11 & |R| \\ |I| &= 10.60 & -0.28 & -0.32 & |G| \\ |Q| &= 10.21 & -0.52 & 0.31 & |B| \end{aligned}$$

Como podemos ver en esta matriz existen varios coeficientes negativos, en las columnas pertenecientes a G y B, para obtener estos coeficientes generaremos dos nuevas señales,  $-G12/$  y  $-B12/$ , estas señales tienen un valor de -12 volts. Generaremos también otras tres nuevas señales  $R12/$ ,  $G12/$  y  $B12/$ , con un valor pico de +12 V. Los circuitos para obtener estas señales se analizan a continuación.

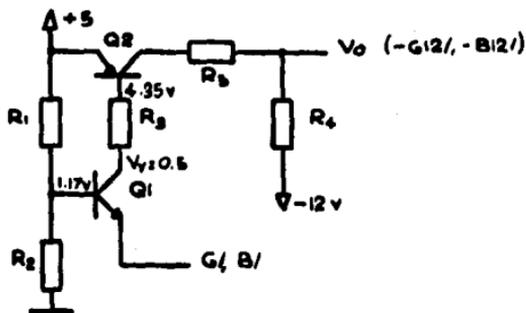
Para obtener las señales  $R12/$ ,  $G12/$  y  $B12/$  se usa un inversor RTL común:



Este inversor presenta una corriente de colector de saturación de 12 mA ( $I_{csat} = 12 \text{ V} / 1 \text{ K}$ ), que se escoge grande precisamente para que las matrices conectadas a la salida del inversor no lo cargen excesivamente.

$I_b = 12 \text{ mA} / 50 = 240 \text{ uA}$  con  $R_b = 8.2 \text{ K}$  se logra una  $I_b = 317.07 \text{ uA}$  y nos aseguramos de que el transistor este perfectamente saturado.

Para obtener las señales de -G12/ y -B12/ se utiliza el siguiente circuito:



Este circuito utiliza una entrada tipo TTL para conducir un inversor conectado a -12 V.

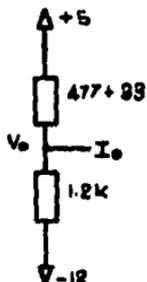
Cuando  $G, B = 3.3 \text{ V}$  Q1 debe estar en corte, escogiendo  $V_x = 1.16 \text{ V}$ :

$$R_1 = 33 \text{ K}, R_2 = 10 \text{ K}$$

y Q1 esta en corte (BE esta inversamente polarizado). Cuando Q1 esta en corte no hay corriente en la base de Q2, y por lo tanto este transistor tambien esta en corte, y  $V_o = -12 \text{ V}$ .

Cuando  $V_i = 0.25 \text{ V}$ , Q1 esta en saturación, y por lo tanto la union BE de Q2 esta directamente polarizada y Q2

esta en saturación, calculandose  $V_o$  por la siguiente relación:



$$V_o = 5 - R_5 I \quad (\text{si } I_o \ll I)$$

$$5 = (R_5 + R_4) I - 12$$

con  $V_o = 0$  para  $V_i = 3.3$  V:

$$I = 5/R_5$$

$$17 = 5(R_5 + R_4)/R_5$$

$$17R_5 = 5R_5 + 5R_4$$

$$R_5 = 0.4167R_4$$

$$R_4 = 1.2 \text{ K}$$

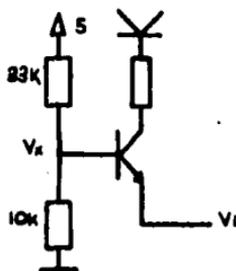
$$R_5 = 500 \text{ E } (470 + 33)$$

Cuando  $V_i = 0.7$ , D1 esta en saturación ( $V_x = 1.17 \text{ V} \pm 5\%$ ), y tambien D2:

$$I_{CQ2} = 5 / R_5 = 10 \text{ mA}$$

$$I_{BQ2} = 200 \text{ uA.}$$

Para lograr la saturación se necesita que  $V_y$  sea menor a  $V_x - 0.65 = 520 \text{ mV}$  para lograr que la unión BC este directamente polarizada (tal como lo esta la BE) y Q1 este efectivamente saturado. Con  $I_{CQ2} = 200 \text{ uA}$ :



$$(4.35 - 0.50) / R_3 = 200 \text{ u}$$

$$R_3 = 19.25 \text{ K}$$

$R_3 = 15 \text{ K}$  (valor comercial e inferior, por seguridad)

Evidentemente esta configuración no invierte las señales G,B, por lo que es preciso invertir las de antemano, mediante el inversor 74LS04.

El calculo de las resistencias para las diferentes

matrices se hace de acuerdo con la siguiente ecuación:

$$R_c = (1 - \kappa) R / \kappa$$

Seleccionando  $R = 3.9 \text{ K}$ , para que la corriente en las matrices sea mucho mayor que la  $I_b$  de los transistores a los que se conectan:

Y:

$$R = 3.9 \text{ K}$$

$$RR = 9.1 \text{ K} (8.2 \text{ K} + 1 \text{ K}) \quad 0.30 \text{ V}$$

$$RG = 2.7 \text{ K} \quad 0.59 \text{ V}$$

$$RB = 31.5 \text{ K} (27 \text{ K} + 4.7 \text{ K}) \quad 0.11 \text{ V}$$

I:

$$R = 3.9 \text{ K}$$

$$RR = 2.6 \text{ K} (2.7 \text{ K}) \quad 0.59 \text{ V}$$

$$RG = 10.03 \text{ K} (10 \text{ K}) \quad 0.28 \text{ V}$$

$$RB = 8.29 \text{ K} (8.2 \text{ K}) \quad 0.32 \text{ V}$$

O:

$$R = 3.9 \text{ K}$$

$$RR = 14.67 \text{ K} (15 \text{ K}) \quad 0.21 \text{ V}$$

$$RB = 3.6 \text{ K} (3.3 \text{ K} + 330 \text{ E}) \quad 0.52 \text{ V}$$

$$R_B = 8.68 \text{ K } (8.2 \text{ K} + 470 \text{ E}) \quad 0.31 \text{ V}$$

En la columna de la derecha se muestran los voltajes que serian obtenidos como salida de la matriz si 1 V se aplicara a la resistencia especificada, estando las otras dos resistencias conectadas a tierra.

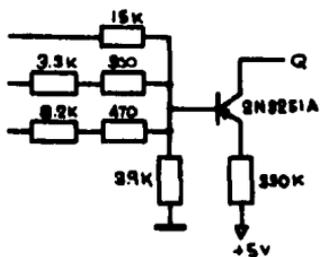
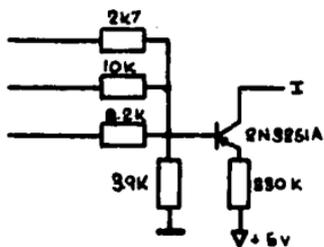
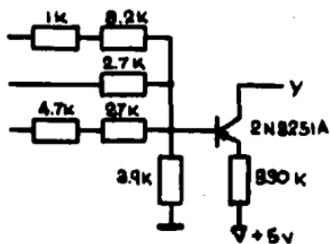
Los diferentes colores obtenidos son:

C	R	G	B	Nombre	Y	I	Q
0	0	0	0	Black	0.00	0.00	0.00
1	0	0	1	Blue	0.36	-1.06	1.02
2	0	1	0	Green	1.95	-0.92	-1.72
3	0	1	1	Cyan	2.31	-1.98	-0.69
4	1	0	0	Red	0.99	1.98	0.69
5	1	0	1	Magenta	1.35	0.92	1.72
6	1	1	0	Yellow	2.94	1.06	-1.02
7	1	1	1	White	3.33	0.00	0.00

En donde Y, I y Q estan dados en volts, y representan el voltaje que se obtiene de las matrices al suministrarles el codigo RGB correspondiente, siendo un 1 lógico igual a 3.3 V, y un 0 lógico igual a 0.25 V.

Estas matrices pueden verse en la figura 4.3.2/5.

Como puede verse en la figura 4.3.2/5, las matrices estan conectadas a los 2N3251A, los cuales funcionan como fuentes de corriente, eligiendose  $R_e = 330 \text{ K}$ , para que el



voltaje maximo de 3.3 V genere una corriente maxima de 7.88 uA.

Una vez que se tienen unas señales de corriente equivalentes a I, Q e Y, hay que sumar estas tres en una sola señal, junto con SYNC, HRTC, VRTC y Color Burst. Este proceso es llevado a cabo por VADD, que esta compuesto por un emisor-seguidor implementado mediante un 2N2222A, este EF tiene un bypass de 0.1 uF, destinado a aterrizar cualquier ruido de alta frecuencia que pueda ser generado. Un potenciómetro de 2 K proporciona un control de ganancia que nos permite controlar el brillo de la pantalla.

#### 4.4- CONTROLADOR DE CRT.

El controlador de CRT (CRTC) tiene como función generar las señales necesarias para que el VG pueda generar una señal de video colorplexed totalmente de acuerdo al estandar RS170.

El CRTC es totalmente programable y recibe del VTKBP toda la información necesaria para poder generar estas señales. Este circuito tiene, además, acceso directo a memoria por medio del DMAC, lo cual le permite leer la información depositada en las imagenes de video por el VTKBP, sin tener que recurrir a este para poder hacerlo.

#### 4.4.1 - Arquitectura del CRTC.

En esta sección se describe la estructura lógica del CRTC, sin recurrir a una explicación de su implementación física, sino solo desde el punto de vista funcional.

##### 4.4.1.1- Operación del CRTC.

La primera función del CRTC es generar las señales de tiempo utilizadas por el mismo y por el VG. Estas señales son cinco: SINS.58M, SQ3.58M, DOTCLK y CCLK.

De estas señales CCLK es la única que el CRTC requiere para su propia operación. Esta señal va a ser utilizada por el contador de caracteres del CRTC para contar cuantos caracteres han sido visualizados en el presente renglon.

A partir de estas señales el CRTC genera también las señales de VRTC, HRTC, SYNC y CLBON. La forma en que se hace esto puede verse en el diagrama de bloques de la figura 4.4.1/1.

El CRTC contiene una serie de contadores que le permiten conocer que caracter está siendo visualizado (Contador de caracteres), que línea del caracter está en pantalla (Contador de líneas), y que renglon de caracteres se está visualizando (Contador de renglones). Contiene también dos contadores que le permiten generar las señales de VRTC y



HRTC, el primero contando un determinado número (programable) de renglones por VRTC, y el segundo contando un determinado número de caracteres (también programable) por HRTC. Asimismo hay varios contadores, que apartir de HRTC y CCLK generan la señal de sincronía RS170 (SYNC). La señal de CLBON se obtiene a partir de SQ3,5BM y HSYNC.

La segunda función del CRTC es leer el contenido de las imagenes de video y transferirlo al VG. Esta lectura va a hacerse por DMA, a través del DMAC. El CRTC tiene dos buffers de renglon (Registro de desplazamiento de 80x8), mientras uno de los buffers esta siendo visualizado el otro se esta llenando con el contenido del siguiente renglon en la imagen de video. El CRTC puede requerir transferencias de DMA de 1 a 8 caracteres por transferencia, puede programarse, asi mismo, el intervalo entre dos grupos de transferencias, siendo este desde 0 a 55 ± 1 periodos de CCLK. En nuestro sistema las transferencias son de 8 caracteres con cero intervalo entre ellas.

El primer requerimiento de DMA de la pantalla ocurre un renglon antes de que finalice el VRTC de la anterior. Los requerimientos de DMA continuan como se haya programado hasta que el buffer de renglon se llena (en nuestro caso son 55 caracteres, lo cual implica  $55/8 = 6.88$  ciclos de transferencia encadenados. Puesto que nuestro DMAC requiere 4 ciclos de reloj -de 6.144 MHz- para transferir un byte, cada ciclo de transferencia tendra una duración de 5 uS, y un

renglon podra transferirse en aproximadamente 36 o 37 uS). Las transferencias sucesivas se activaran al principio de cada renglon. Es decir que mientras se esta visualizando el contenido de uno de los buffers, se esta llenado el otro. El tiempo requerido para visualizar un renglon es de 63.5 uS (incluyendo el HRTC), por lo cual tendremos 27.5 uS libres por renglon.

La información asi leida por el CRTC es transferida al VG mediante el PVBI, especificamente mediante las lineas CCO a CC6.

El CRTC tambien determina el color de un bloque en GR1 o de un pixel en GR0 mediante el contenido del registro de color, el cual especifica uno de ocho posibles colores a usar.

#### 4.4.1.2- Grupos del CRTC.

Los grupos que forman al CRTC son los siguientes (figura 4.4.1/1):

##### 1. CLKGEN:

Grupo de generación de señales de reloj. Este grupo tiene como función generar todas las señales de reloj que van a ser necesitadas por el CRTC y por el VG. Estas señales son SQ3.58M, SIN3.58M, DOTCLK, CCLK, QA y QB. Todas estas señales son digitales (TTL), excepto SIN3.58M que es una senoide.

SO3.58M es una señal fundamental que se utiliza para generar todas las otras señales de tiempo usadas por el circuito, y que van a ser múltiplos o submúltiplos de esta

## 2. SYNGEN:

Grupo de generación de señales de sincronía y borrado. Este grupo tiene como función generar las señales de sincronía y borrado necesitadas por el VG. Estas señales son tres: SYNC, HRTC y VRTC. SYNC es una suma lógica de las señales de sincronía horizontal y vertical, incluyendo los pulsos de equalización y las serraciones. HRTC y VRTC son pulsos digitales que están activos durante los regresos verticales y horizontales. Sus duraciones se especifican en la sección 4.1.

## 3. CODGEN:

Grupo de generación de códigos. Este grupo tiene como función generar los códigos necesarios para que el VG sepa que es lo que debe visualizar. Las señales generadas por este grupo son CC0-CC6; LC0-LC2 y LA0,1.

## 4. CTRLGEN:

Grupo de control. Este grupo genera las señales de control necesarias para que el VG sepa en que modo está operando, es decir, genera a GPA0 y GPA1.

## 5. SYSINT:

Grupo de interfaz con SYSDUS. Este grupo se encarga de

interconectar al CRTC con el SYSBUS.

#### 4.4.1.3- Registros del CRTC.

El CRTC tiene varios registros internos, de los cuales algunos están disponibles al sistema, y algunos no. La función de estos registros es mantener ciertas variables necesarias para el funcionamiento del CRTC, así como reflejar el estado del CRTC, de modo que el VTKBP pueda conocerlo con solo leer algunos de estos registros.

Los registros disponibles al sistema son cuatro:

- |                                   |                      |
|-----------------------------------|----------------------|
| 1. Registro de color (CLRR).      | Dirección: 0A0H (WR) |
| 2. Registro de comandos (CMDR).   | Dirección: 021H (WR) |
| 3. Registro de estatus (STATR).   | Dirección: 021H (RD) |
| 4. Registro de parámetros (PARR). | Dirección: 020H (WR) |

El registro de color es utilizado por el VG para determinar el color que va a tener un cierto punto en el modo BR1. Este es un registro de 3 bits, que especifica 1 de 8 colores, de acuerdo a lo mostrado en las tablas que aparecen en la sección sobre diseño de COLORMAT en el VG. Este registro está conectado directamente al SYSBUS, y su dirección es A0H. Es un registro de escritura únicamente.

El CMDR es utilizado por el VTKBP para ordenar al CRTC la ejecución de ciertas operaciones, mediante la escritura en

el de ciertos comandos. Cada comando requiere, además, la escritura de un cierto número de parámetros (0 a 4) en el PARR, inmediatamente después de la escritura del comando. Los comandos posibles son:

Comando	Parámetros
1. Reset	4
2. Start Display	0
3. Stop Display	0
4. Read Light Pen	2
5. Load Cursor	2
6. Enable Interrupt	0
7. Disable Interrupt	0
8. Preset Counters	0

A continuación se da el significado de cada comando.

#### 1. Reset:

	OPERACIÓN	AO	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Reset	00000000
Par	Write	0	Byte1	SHHHHHHH
Par	Write	0	Byte2	VVRRRRRR
Par	Write	0	Byte3	UUUUUUUU
Par	Write	0	Byte4	MFCCZZZZ

Acción: Después de escribir el comando de Reset, se detienen todas las peticiones de DMA, se deshabilitan las interrupciones, la señal VSP se pone en estado activo para borrar la pantalla. HRTC y VRTC siguen funcionando de acuerdo a sus valores anteriores. Tras un Power On estas dos señales tienen tiempos aleatorios.

A medida que los parámetros van siendo escritos, la composición de la pantalla va siendo definida.

Parámetro S: Renglones espaciados.

S=1; Renglones espaciados.

S=0; Renglones normales.

Parámetro HHHHHH: Caracteres por renglon.

HHHHHH	Car/renglon
000000	1
000001	2
.	.
.	.
.	.
1001111	80
1010000	Ilegal
.	.
.	.
.	.

1111111

Ilegal

Parametro VV: Renglonos por VRTC.

VV	Renglonos/VRTC
00	1
01	2
10	3
11	4

Parametro RRRRRR: Renglonos por pantalla.

RRRRRR	Renglonos por pantalla.
000000	1
000001	2
.	.
.	.
.	.
111111	64

Parametro UUUU: Posición del subrayado.

UUUU	Linea del subrayado.
0000	1
0001	2
.	.
.	.
.	.

Parametro LLLL: Número de líneas por renglon.

LLLL	Líneas/Renglon
0000	1
0001	2
.	.
.	.
.	.
1111	16

Parametro M: Modo del contador de línea.

M = 0; Modo 0 . No offset

M = 1; Modo 1 . Offset 1

Parametro F: Modo de los atributos de campo.

F = 0; FAS transparente.

F = 1; FAS visible.

Parametro CC: Formato del cursor.

CC	Cursor.
00	Bloque de video inverso parpadeante.
01	Subrayado parpadeante.
10	Bloque de video inverso.

11            Subrayado.

Parametro ZZZZ: Caracteres por HRTC.

ZZZZ	Caracteres/HRTC
0000	2
0001	4
0010	6
.	.
.	.
.	.
1111	32

## 2. Start Display:

	OPERACIÓN	A0	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Start Dsp.	001SSSBB

Sin Pars.

001: Start Display.

SSS: Núm. de CCLKs entre DMA requests.

SSS	
000	0
001	7
010	15

011	23
100	31
101	39
110	47
111	55

BB: Núm. de ciclos de DMA por acceso. (Bytes leídos por acceso).

BB	
00	1
01	2
10	4
11	8

Acción: Las interrupciones son habilitadas, empiezan los DMA requests, se deshabilita la señal VSP, se levantan las banderas de Interrupt Enable y Video Enable en la palabra de estatus.

### 3. Stop Display:

	OPERACIÓN	A0	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Stop Dsp.	01000000
Sin Pars.				

Acción: Deshabilita la señal de video (VSF activo), las interrupciones permanecen habilitadas, HRTC y VRTC no son afectados, la bandera de Video Enable es bajada.

#### 4. Read Light Pen:

	OPERACIÓN	AO	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Read LPEN	01100000
	Read	0	Char. number	Char. X coord.
Parametros	Read	0	Row number	Char. Y coord.

Acción: El CRTC suministra los contenidos de los registros de posición del bolígrafo de luz, en los próximos dos ciclos de lectura del registro de parametros.

#### 5. Load Cursor Position:

	OPERACIÓN	AO	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Load Cursor	10000000
Parametros	Write	0	Char. number	Char. X coord.
	Write	0	Row number	Char. Y coord.

Acción: El CRTC es condicionado para depositar los próximos dos parametros en los registros de posición del cursor. El estatus no es afectado.

#### 6. Enable Interrupt:

	OPERACIÓN	AO	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Enable Int.	10100000

Acción: La bandera de habilitación de interrupciones del estatus es levantada con lo cual el CRTC queda habilitado para generar el VRTC interrupt.

#### 7. Disable Interrupt:

	OPERACIÓN	AO	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Disable Int.	11000000

Acción: Se baja la bandera de habilitación de interrupciones. El CRTC no puede generar interrupciones hacia el VTKBP.

#### 8. Preset Counters:

	OPERACIÓN	AO	DESCRIPCIÓN	DATOS
				76543210
Comando	Write	1	Preset Count.	11100000

Acción: Los contadores X y Y del CRTC se fijan en HOME (es decir en el ángulo superior izquierdo de la pantalla).

El STATR contiene siete bits que al ser accedidos por el VTKBP le permiten conocer el estado del CRTC. Estos bits son:

7 6 5 4 3 2 1 0  
0 IE IR LP IC VE OU FO

IE: (Interrupt Enable). Un 1 habilita el VTRC interrupt. Es automáticamente levantado por el comando de Start Display, o por el Enable Interrupt. Es bajado por Disable Interrupt.

IR: (Interrupt Request). Esta bandera se levanta al principio del último renglón de la pantalla, siempre y cuando IE = 1. Se baja tras una operación de lectura del estatus.

LP: (Light Pen). Se levanta cuando LPEN = 1. Es bajado automáticamente por una lectura del STATR.

IC: (Improper Command). Se levanta cuando una cadena de parámetros es demasiado larga o corta para el comando que la precede. Una lectura de STATR la baja en forma automática.

VE: (Video Enable). Indica que la operación del CRTC está habilitada. Es levantada por un Start Display, y bajada por un Stop Display o un Reset.

DU: (DMA Underrun). Esta bandera es levantada cada vez que el CRTIC se queda sin datos durante un DMA. Tras la detección de un DU, la operación de DMA es detenida, y la pantalla es borrada hasta despues del VRTC. La bandera es bajada por cualquier lectura de STATR.

FD: (FIFO Overrun). Esta bandera es levantada cada vez que el FIFO se llena. Es bajada al leer STATR.

#### 4.4.2- Programación del CRTC.

El CRTC es programable a partir de una serie de comandos que fueron explicados en la sección anterior. Existen también una serie de códigos de control que cuando sean leídos desde la imagen de memoria por el CRTC provocarán una acción determinada, distinta a su visualización. Estos códigos se dividen en tres: FAS, CAS y códigos especiales.

##### 1. FAS: (Fiel Attribute Selector).

Estos códigos afectan los atributos visuales de un campo de caracteres. Empezando en el campo que sigue al carácter, y hasta el próximo FAS, o hasta el final de la pantalla.

Existen 4 atributos gráficos:

1. Blink. Los caracteres que siguen a este código parpadean, a causa de la activación de VSP con una frecuencia igual a la de refresco de la pantalla dividida entre 32 (1.88 Hz).

2. Highlight. Los caracteres que siguen a este código tienen un brillo mayor, lo cual se logra activando HGLT.

3. Reverse Video. Los caracteres que siguen a este código aparecen en video invertido, a causa de la activación de RVV.

4. Underline. Los caracteres que siguen a este código

aparecen subrayados, a causa de la activación de LTEN en la línea 9.

5, 6. General Purpose. Estos códigos nos permiten escoger el modo GR0 o GR1, de acuerdo a la tabla que aparece en la sección del VG.

FAS:

```
7 6 5 4 3 2 1 0
1 0 U R G G B H
```

H = 1; Highlight.

B = 1; Blink.

R = 1; Reverse Video.

U = 1; Underline.

GG = GPA1, GPA0.

Un mismo campo puede poseer más de un atributo de campo.

## 2. CAS: (Character Attribute Selector).

Los CAS son códigos que pueden ser utilizados para generar símbolos gráficos sin que estos estén programados en el generador de caracteres. Esto se logra activando selectivamente las señales LA0,1; VSP y LTEN. Los CAS pueden ser programados para parpadear o ser realizados en forma individual. La frecuencia de parpadeo es de 1.875 Hz. Los

distintos simbolos graficos que pueden ser obtenidos se muestran en la figura 4.4.1/2.

CAS:

7 6 5 4 3 2 1 0

1 1 C C C C B H

CCCC: Codigo grafico.

B: Parpadeo.

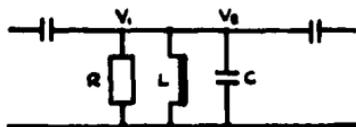
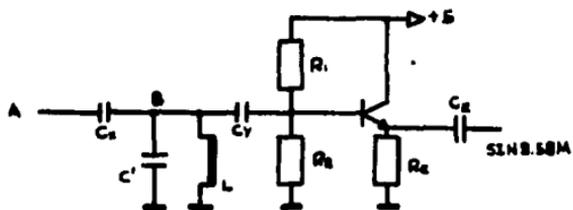
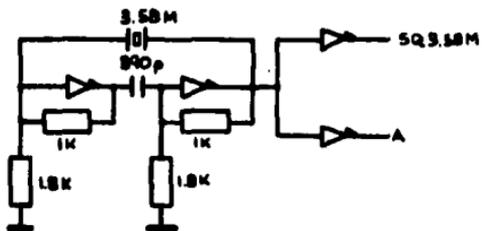
H: Realze.

#### 4.4.2- CIRCUITO DEL CRTC.

El CRTC esta construido alrededor de un controlador de CRT Intel 8275, el cual se encarga de soportar el protocolo de comunicación con el DMAC para leer la memoria imagen de video, de suministrar el código ASCII del carácter a visualizar al VG y de generar las señales de HRTC y VRTC, así como las señales de LAO-1, LTEN, VSP, HLGT y GPAO-1. El diagrama de bloques de este circuito se muestra en la figura 4.4.2/1.

Como se ve en esta figura el 8275 no genera la señal de CCLK sino que requiere que esta le sea entregada por un circuito externo. El circuito que genera  $SQ3.58M$  se muestra en el plano 1, y el generador de  $SIN3.58M$  aparece en el plano 5, como puede verse se trata de un multivibrador astable construido con tres inversores 74LS04 y un cristal de cuarzo corte AT, con una frecuencia resonante fundamental en serie de 3.579545 MHz, del tipo utilizado en los receptores de TV para fijar la frecuencia de la señal subportadora de color; para obtener la señal  $SIN3.58M$  a partir de la señal cuadrada, se pasa esta por un filtro pasabandas para eliminar las armónicas impares de 3.58 MHz que contiene la señal cuadrada; el cálculo del filtro resonante se muestra a continuación (figura 4.4.2/2):

La serie de Fourier de la señal cuadrada (sin componente dc) es la siguiente:



$$v = 4V_m (\sin \omega t + \sin 3\omega t/3 + \sin 5\omega t/5 + \dots) / \pi$$

Como vemos la serie solo presenta armónicas impares, la más cercana de las cuales a 3.58 MHz es 10.74 MHz (0.47712 décadas).

El filtro resonante está regido por las siguientes ecuaciones:

$$BW = 1 / (2\pi RC) \text{ Hz}$$

$$\theta_0 = 1 / \text{sqr}(LC) \text{ rad/s}$$

donde:

$$R = R_p || R_1 || R_2 || R_i$$

$$C = C_1 + C_{ob}$$

$$R_i = h_{ie} + (1 + h_{fe})R_e$$

$$h_{ie} = V_{Thfe} / I_c$$

Para el 2N2222A:

$$h_{FE}(\text{min}) = 75 @ 1 \text{ mA}$$

$$h_{FE}(\text{max}) = 200 @ 1 \text{ mA}$$

$$C_{ob}(\text{max}) = 8 \text{ pF}$$

$$V_T(\text{typ}) = 26 \text{ mV @ } 25^\circ\text{C}$$

Para polarizar el emisor seguidor en clase A (cuya

función es evitar que el filtro sea cargado por los circuitos a los que se conecte SIN3.58M y viceversa) en forma estable seguimos la siguiente relación, y escogemos  $I_c = 1 \text{ mA}$  aprox.:

$$(1 + hFE_{min})R_e > R_1 || R_2 > 5h_{ie_{max}}$$

Por lo tanto  $h_{ie(max)} = 4.85 \text{ K} @ I_c = 1.0714 \text{ mA}$ :

$$425.6 \text{ K} > R_1 || R_2 > 24.3 \text{ K}$$

$$R_1 = 68 \text{ K}$$

$$R_2 = 100 \text{ K}$$

y

$$R_i = 430 \text{ K}$$

$$R_1 || R_2 = 40.5 \text{ K}$$

Con estos valores y seleccionando el factor de calidad del inductor ( $Q_c$ ) igual a 100:

$$R_p = sLQ_c$$

Si escogemos  $C = 1 \text{ nF}$  (125 veces mayor a  $C_{ob(max)}$ ), con lo cual nos aseguramos que las variaciones en este factor no afectaran a nuestro filtro), entonces:

$$L = 1.9764 \text{ uF} (CL \ll 1 \text{ nF})$$

$$R_p = 4.4457 \text{ K}$$

$$B_W = 39.789 \text{ KHz}$$

$$Q = 89.975$$

Por lo tanto la magnitud de la función de transferencia del filtro en la frec. de resonancia es de:

$$M = 20 \log Q \text{ dB}$$

$$= 39.082 \text{ dB}$$

A la frecuencia de 10.74 MHz la señal se habra atenuado: 19.085 dB (-40 dB / dec) y además esta dividida entre tres, es decir que su magnitud estara 28.627 dB abajo de la magnitud de resonancia, es decir que sera de 10.455 dB. El valor absoluto de la atenuación es 27, por lo cual la señal resultante no sera exactamente una senoide, sino que tendra una cierta cantidad de distorsión armonica (3.7073%), sin embargo esto es suficientemente bajo para su aplicación como subportadora de color en la terminal de video.

En el plano 1 aparece el generador de DOTCLK, este circuito es un doblador de frecuencia que convierte la señal cuadrada de 3.58 MHz en un tren de pulsos de 7.15909 MHz.

Este doblador de frecuencia es un circuito bastante delicado, el cual requiere de un trazado muy limpio al llevar a cabo su implementación, para evitar retroalimentaciones parasitas que lo puedan hacer oscilar. La sencilla elegancia de este circuito merece que nos detengamos un poco a estudiarlo:

Basicamente lo que el circuito hace es detectar los cambios de estado de la señal de entrada y generar un pulso para cada cambio de estado, ya sea ascendente o descendente. Desde este punto de vista podemos considerar al circuito como un Edge Detector, totalmente digitalizado.

El circuito compara el estado actual ( $t_0$ ) de la señal de entrada (cuya frecuencia va a ser doblada), con el estado inmediatamente anterior de dicha señal ( $t-1$ ), si ambos estados son iguales se sigue repitiendo este proceso hasta que ocurre un cambio de estado (es decir, una transición positiva o negativa de la señal de entrada), momento en el cual se genera un pulso, cuya duración esta dada por la cadena de inversores en el lazo de retroalimentación, y debe ser menor al semiperiodo de la señal original.

La señal de CCLK se obtiene a partir de DOTCLK mediante

un contador hexadecimal sincrónico 74LS163, su frecuencia es de 1.0227271 MHz, o sea que es  $\text{DOTCLK} / 7$ , CCLK es asimétrico, con un ciclo de trabajo de  $3/7$  (33.33 %), lo cual significa que el pulso tiene una duración de 419.05 nS el ancho mínimo del pulso de reloj para el 74LS163 es de 20 nS, es decir que es compatible con DOTCLK. Este circuito puede verse en el plano 1.

La señal de sincronía (SYNC) se obtiene a partir de las señales de HRTC y VRTC, generadas por el 8275. Esta señal es el XOR de dos señales llamadas HSYNC (sincronía horizontal) y VSYNC (sincronía vertical).

HSYNC es generada por un arreglo de dos SIFO SR 74LS164 que actúan como un contador lineal de 4 bits (16 eventos), se podría usar un contador de 4 bits, pero esto nos obligaría a utilizar lógica combinatorial para decodificar las salidas del contador, lo cual sería más costoso (y también más incomprensible) que el utilizar los dos 74LS164. El pulso HSYNC se genera en forma continua, pero está cualificado por HRTC, de manera que es cero cuando HRTC es cero, y solo puede existir cuando HRTC es diferente de cero. El FP es de 2 caracteres, HSYNC tiene una duración de 5 caracteres y el BP tiene una duración de 4 caracteres (estas duraciones ya fueron previamente justificadas en la sección sobre señales generadas en el estándar RS170), la señal base para generar HSYNC es CCLK/, de lo cual se desprende que la unidad de medida natural de HSYNC es la duración de un carácter. El

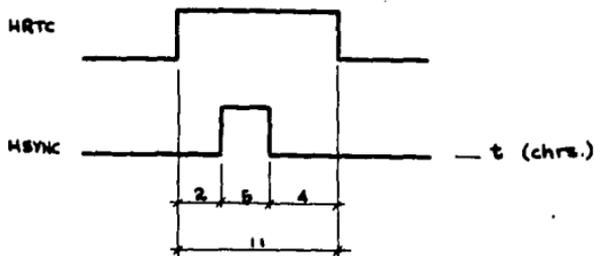
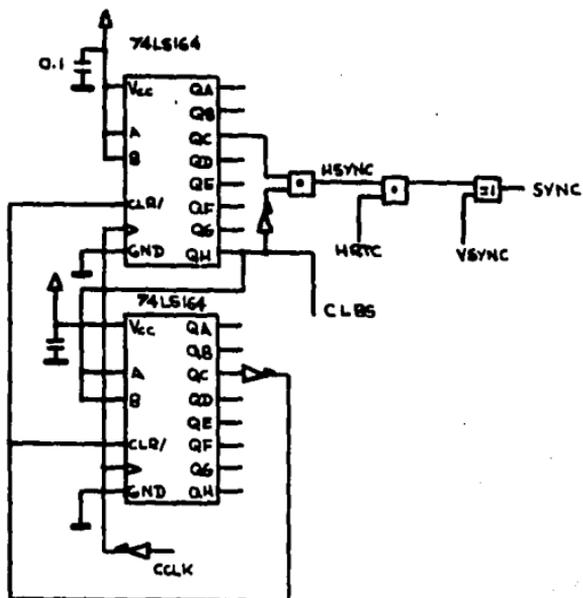
plano 1 muestra este circuito.

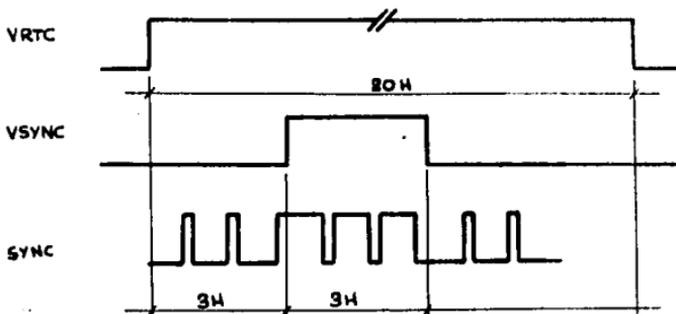
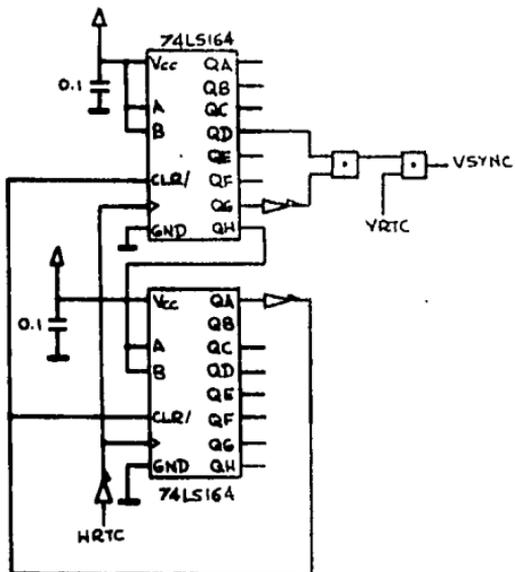
VSYNC se genera en forma similar a HSYNC, si bien utilizando a HRTC como señal generadora, de lo cual se desprende que su unidad de medida sera la duración de un renglon. VSYNC esta cualificada por VRTC, es decir que solo puede estar activa cuando VRTC lo esta. VRTC tiene una duración igual a la de tres renglones de video (incluyendo el HRTC) y ocurre tres renglones despues de haberse iniciado el VRTC, el cual tiene una duración total de 20 renglones.

El pulso de VSYNC es sumado mediante una OR exclusiva (74LS86) a el pulso de HSYNC para formar la señal de SYNC, esto hace que SYNC sea igual a HSYNC cuando VSYNC no esta activa, y a HSYNC/ cuando VSYNC esta activa, lo cual genera tres serraciones en el pulso de sincronia vertical. De esto se desprende que el sistema que estamos utilizando no es entrelazada, es decir que de dos campos que estan presentes por pantalla solo estamos utilizando uno, y el otro lo dejamos en negro. Este circuito aparece en el plano 1.

La señal de Color Burst On (CLBON) es utilizada para señalar al VG el intervalo de tiempo en el cual se debe insertar el Color Burst. CLBON es un pulso de aproximadamente 2.7937  $\mu$ S obtenido mediante un contador de decadas 74LS160A, el cual genera una ventana de 10 ciclos de la señal SQ3.58M (casi 3 caracteres). Esta ventana empieza a generarse a

### MSYNC y SYNC





partir del octavo caracter del HRTC, es decir, al terminar el pulso de HSYNC. Esto es señalado por la señal CLBS procedente del generador de HSYNC. Este circuito puede observarse en el plano 1.

## V. DISEÑO DE LA INTERFAZ DE COMUNICACIONES COMPUTADOR/TERMINAL

Como he mencionado en ocasiones anteriores la función de la terminal es visualizar información procedente de un computador anfitrión, y recibir información de un ser humano, transmitiéndola al mismo computador anfitrión, este capítulo trata del circuito que se encargara de establecer la comunicación computador anfitrión/ terminal.

Es lógico que el diseño de la interfaz de comunicaciones sea un factor de importancia en el diseño de la terminal, pues de él dependera que las comunicaciones sean confiables, la velocidad de proceso de la terminal y su compatibilidad con diversos computadores.

Existen estándares ya perfectamente definidos para la comunicación terminal/computador. Estos estándares especifican los protocolos de comunicación, tanto desde el punto de vista de que forma deben tener los datos transmitidos (Protocolos de datos), como desde el punto de vista de que características electricas deben tener las señales transmitidas, e incluso como deben ser los conectores usados, el tipo de conductor y la distancia maxima de transmisión, etc. (Protocolos de electrónica).

Estos estandares son distintos segun el tipo de aplicación y de ambiente en el que este trabajando la terminal, y es evidente que si se quiere asegurar que esta sea compatible con un número grande de computadores hay que seleccionar un tipo de interfaz que sea ampliamente usado en el tipo de ambiente en donde sera utilizada la terminal.

Para determinar el tipo de interfaz que se adapte a nuestra terminal es conveniente hacer un resumen de las características de la operación de esta. En primer lugar tenemos que no es preciso que su velocidad de transmisión de información sea muy alta, ya que la rapidez con que el usuario humano puede introducir información a la maquina desde el teclado, o leer información en el CTR no es alta. En segundo lugar sabemos que la distancia desde la terminal al computador central no sera muy grande en la mayor parte de los casos, y esperamos que la terminal sea utilizada en un ambiente normal de oficina, en donde no existan campos electromagneticos excesivamente fuertes que puedan provocar una gran cantidad de ruido, y por ende, de interferencia.

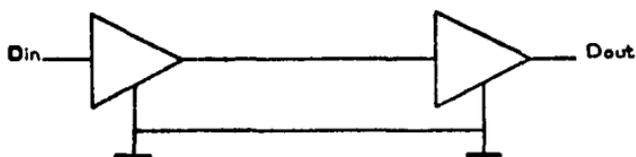
De la descripción anterior podemos determinar que la transmisión puede hacerse en forma serial, ya que la velocidad no necesita ser excesivamente alta, por lo que podemos soportar el retraso necesario para transmitir la información bit por bit, en lugar de transmitir varios bits en paralelo, lo cual es más caro, ya que requiere de varias

lineas de transmisión. También nos hace suponer que la transmisión puede ser desbalanceada, ya que no se requiere gran velocidad, ni transmitir a grandes distancias.

La discusión anterior nos hace pensar de inmediato en el estandar EIA RS232C, podrimos emprender un analisis mucho más profundo de los requerimientos de la terminal, y desarrollar nuestra propia interfaz, pero esto impediria que fuera compatible con ningun computador existente, con esto en mente decidi utilizar una interfaz RS232C, que es ampliamente utilizada en computadoras comerciales (otra opción era usar la interfaz HP-IL de Hewlett-Packard, que tiene la ventaja de permitir que hasta 30 terminales sean conectadas a un solo puerto, mediante un lazo en el cual todas reciben la información transmitida por las otras, en tanto que el RS232C exige la existencia de un puerto en el computador para cada terminal conectada. El HP-IL tiene, sin embargo, dos desventajas: en primer lugar la corta distancia a la cual puede situarse la terminal, en segundo lugar que no es usado por ningun fabricante de computadoras, excepto por Hewlett-Packard, y este solo lo utiliza en sus computadoras portatiles HP110, HP75, HP71 y HP41).

## 5.1- DISEÑO DE LA INTERFAZ EIA RS232C.

Este estandar es extremadamente popular, y es utilizado por una gran cantidad de fabricantes. Provee transmisión de información digital sobre una línea unidireccional (no reversible), desbalanceada y no terminada.



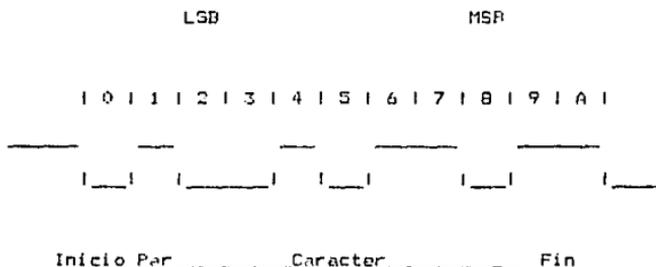
RS 232C

Sus principales características son:

1. Lógica positiva ( $\pm 5$  V min a  $\pm 15$  V max)
2. Protección contra fallas.
3. Control sobre la pendiente de los pulsos transmitidos (para controlar su contenido de altas frecuencias).
4. Distancia máxima de transmisión (recomendada en el estandar) de 15.24 m.
5. Tasa máxima de transmisión de 20,000 bits/s.

### 5.1.1- Protocolos de datos.

Utilizaremos transmisión serie, asincrónica, full duplex, código ASCII (CCITT alphabet Nº 5), usando 7 bits para el código, 1 bit de paridad (par, impar o ninguna, seleccionable mediante VTKBOS), 1 bit de principio (espacio) y 1.5 o 2 bits de final (marca):



Se puede transmitir con 10 velocidades estandar, a continuación damos una lista de estas junto con el divisor que hay que fijar en el 8253 para obtener la velocidad de transmisión a partir de un señal cuadrada de 1.536 MHz (es SYSCLK dividido entre dos mediante un flip-flop JK):

Velocidad bps	Divisor	Velocidad cps
19,200	80	1,745.45
9,600	160	872.73

4,800	320	436.36
2,400	640	218.18
1,200	1,280	109.09
600	2,560	54.55
300	5,120	27.27
150	10,240	13.64
110	13,964	10.00
75	20,480	6.82

La velocidad en cps se calculo suponiendo que se transmitan 11 bits por cada caracter. La velocidad de transmisión se selecciona mediante un comando de VTKDOS.

### 5.1.2- Protocolos de electrónica.

El protocolo EIA RS232C especifica una serie de características eléctricas y mecánicas que deben cumplirse, principiemos describiendo las señales presentes en el conector de 25 terminales usado por el estándar:

1. GND. Tierra física.
2. TXD.
3. RXD.
4. RTS/
5. CTS/
6. DSR/
7. SGND. Tierra de la señal.
8. Carrier Detect.
9. Sin usar.
10. Sin Usar.
11. Conectado a la terminal 19.
12. Secondary Carrier Detect.
13. Secondary Clear to Send.
14. Secondary Transmitted Data.
15. TxC/
16. Secondary Received Data.
17. Rx/C/
18. Sin usar.
19. Secondary RTS/
20. DTR/

21. Signal Quality Detect.
22. Ring Indicator.
23. Data Rate Select.
24. External Transmitter CLK.
25. Sin Usar.

Explicación de las diferentes señales:

1. GND.

Tierra física. Conectado al chasis de la terminal.

2. TxD. >

Transmisión de datos. Cuando no hay transmisión esta en marca.

3. RxD. <

Recepción de datos.

4. RTS/. >

Ready To Send. Esta señal se utiliza para señalar al computador anfitrión que existe una terminal conectada al puerto RS232C, y habilitarlo para transmitir. Es activada cada vez que la terminal es encendida.

5. CTS/. <

Clear To Send. Su función es permitir al computador anfitrión habilitar a la terminal como transmisor en Half duplex. En Full duplex el computador la mantiene siempre

activa. En nuestra implementación esta señal es ignorada.

6. DSR/. <

Data Send Ready. Esta señal es ignorada por la terminal.

7. SGND.

Tierra de la señal.

8. Carrier Detect. <

En Full duplex que la señal este activa significa que la línea esta OK. En Half duplex debe estar activa durante la transmisión. En nuestra implementación esta señal es ignorada.

9. Secondary RTS/. >

Esta señal es mantenida activa por la terminal.

10. Secondary Carrier Detect. <

Esta señal es ignorada por la terminal.

11. Secondary TxD. >

Esta señal no es utilizada en nuestra implementación.

12. TxCLK/. <

Trasmitter CLK. Esta señal es ignorada por la terminal.

13. Secondary RxD. <

Esta señal no es utilizada en nuestra implementación.

14. RxC/. <

Receiver CLK. Esta señal es ignorada en nuestra implementación.

15. DTR/. >

Data Transmitter Ready. Esta señal debe estar activa para que el modem pueda responder. En nuestra implementación siempre esta activa, excepto:

- i. Cuando la terminal esta apagada.
- ii. Cuando la terminal esta en local.
- iii. Durante un Hard Reset.

16. Signal Quality Detect. <

Esta señal es ignorada en nuestra implementación.

17. Ring Indicator. <

Esta señal es ignorada en nuestra implementación.

18. Data Rate Select. <

Esta señal es ignorada en nuestra implementación.

19. External TxC. <

Esta señal es ignorada en nuestra implementación.

El protocolo que se sigue para realizar una transmisión

en el estandar RS232C se basa en las señales DTR/, Carrier Detect, RTS/ y CTS/. En la discusión que sigue utilizaremos los nombres estandar, DTE, para la terminal (Data Terminal Equipment), y Modem para el dispositivo al cual esta conectado la terminal (que puede ser un modem FSK, PSK o ASK, si la información se va a transmitir por via telefonica o telegrafica, o el computador anfitrión).

En Full Duplex la transmisión y la recepción se hacen por líneas físicamente independientes, en tanto que en Half duplex se utiliza un esquema de TDM (Time Division Multiplexing) para transmitir y recibir sobre una misma línea.

Veamos lo que ocurre en Full Duplex: La DTE debe activar DTR/ para que el modem pueda responder, a continuación debe verificar el estado de Carrier Detect, si esta activo los datos recibidos son confiables, debido a que la línea esta en buen estado, en caso contrario la línea no esta limpia, por lo cual la información no es confiable. Durante el periodo en la que la DTE esta transmitiendo/recibiendo, debe mantener RTS/ activo para señalar al modem que esta recibiendo información, y el modem debe mantener CTS/ activo para que la DTE sepa que su información esta siendo recibida.

En Half Duplex DTR/ tambien debe ser mantenido activo por la DTE. Durante la recepción Carrier Detect debe estar

activo, y RTS/ debe activarse para habilitar al modem e indicarle que transmita hacia la DTE, durante la transmisión CTS/ debe ser mantenido activo por el modem, para indicar a la DTE que su información sera recibida.

En nuestra terminal solo se soporta el protocolo de comunicación Full Duplex, este es el motivo por el cual RTS/ esta siempre activo y CTS/, DSR/ son ignorados.

DTR/ esta siempre activo, excepto cuando la terminal esta apagada, o en local, o durante un Hard Reset.

Las señales que hemos descrito deben cumplir con una serie de características electricas y de frecuencia que son descritas a continuación:

PARAMETRO	CONDICION	MIN	TYP	MAX	U
VOH Driver out V	$R_l = \text{inf.}$			25	V
VOL open circ.		-25			V
VOH Driver out V	$3K \leq R_l \leq 7K$	5		15	V
VOL loaded		-15		-5	V
Ro Driver out R	$-2 \leq V_o \leq 2V$			300	E
power off					
Ios Driver out	$R_l = 0$	-500		500	mA
Short cir. I					
Driver out	% of unit	4			%
slew rate	interval				
Interchange				30	V/ $\mu$ S
circuits					
Control					
circuits		6			V/ $\mu$ S
Rate & time					
circuits		6			V/ $\mu$ S
Rin Receiver	$3 \leq V_{in} \leq 25V$	3		7	K
input R					

Receiver open				
cir. in bias V		-2	2	V
Receiver in				
treshold				
out = mark		-3		V
out = space			3	V

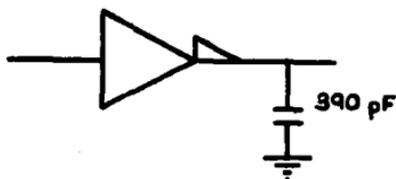
### 5.1.3- Diseño del circuito de la interfaz RS232C.

La parte lógica de esta interfaz se implementa con un 8251 USART de Intel, el cuál se encarga de las comunicaciones entre el RS232C y el 8085, pasando la información recibida del 8085 de paralelo a serie y la recibida del puerto de serie a paralelo, la tasa de transmisión se obtiene de SYSCLK que es dividido entre dos por un flip-flop JK 74LS112, y luego es dividido por los contadores 0 y 1 de un 8253 (contador programable) para obtener así el TxC y el RxC del 8251. TxC se obtiene de OUT1 del 8253, y RxC se obtiene de OUT0 del 8253, pudiendo ser distintos, con lo que se pueden lograr tasas distintas de transmisión y de recepción.

El 8251 tiene una terminal, llamada RxDY conectada al RST6.5 del 8085, cada vez que se recibe un carácter via RxD, la señal RxDY se activa, generando un Restart 6.5 para el 8085, el cual interrumpe su operación y ejecuta la rutina de servicio de lectura de datos del SIO. Esto se explica en el capítulo sobre el desarrollo del programa de control de la terminal.

Para implementar las características eléctricas del estándar EIA RS232C se utilizan dos circuitos integrados, el line driver MC1488 y el line receiver MC1489, estos dos integrados cumplen con todas las características del estándar (ver Motorola Linear Interface Integrated Circuits Data Book), excepto en el MC1488 cuyo slew rate (70 nS) es mayor

al de 30 V/uS especificado por el estandar, para eliminar este problema (el slew rate demasiado alto hace que el contenido de alta frecuencia sea demasiado alto para la linea de transmisi3n) se pone un capacitor en paralelo a la salida del 148B:



Para calcular el capacitor se utiliza la siguiente ecuaci3n:

$$C = I_{sc} \, dt/dv$$

Donde:

$I_{sc}$  = corriente de corto circuito (12 mA, segun las hojas de especificaciones del 148B).

$ft/dv = \text{reciproco del slew rate (1/30 uS/V)}$ .

$$C = 360 \text{ pF}$$

usando valores estandar:

$$C = 390 \text{ pF} \pm 10\%$$

El circuito utilizado puede verse en el plano 4, en donde tambien se incluye el circuito para el lazo de corriente de 20 mA, que se explica en la siguiente secci3n.

Los niveles logicos para nuestra implementaci3n del estandar RS232C son de  $V_{ol} = -10 \text{ V}$  y  $V_{oh} = +10 \text{ V}$ , lo cual se debe a que la fuente de alimentaci3n que usamos para el MC1488 es una fuente de  $\pm 12 \text{ V}$ , se utilizan dos diodos 1N4001 ( $I_{dc} = 1\text{A}$ ) en el MC1488 para prevenirnos contra cualquier inversi3n de voltaje.

## 5.2 - DISEÑO DE LA INTERFAZ DE LAZO DE CORRIENTE DE 20 mA.

La interfaz de lazo de corriente de 20 mA es una interfaz utilizada en computadores no muy recientes, y en los teletipos para transmisión de telex. Se incluye esta interfaz en la terminal porque le da una gama aun mayor de aplicaciones, y por que el costo de su implementación es bajo, ya que comparte los mismos circuitos que la RS232, excepto por dos optoacopladores TIL112, que convierten los niveles de +12 y -12 V del RS232 a un lazo de corriente en el que pueden haber 20 mA (mark) o 0 mA (space).

Esta interfaz no tiene ningun tipo de señal de control, y solo posee dos lazos, uno de salida (lazo activo, es decir que entrega la corriente) y uno de entrada (lazo pasivo, es decir, que recibe corriente), evidentemente el lazo pasivo debe ser conducido por un lazo activo en el computador anfitrión, y el lazo activo terminara en un lazo pasivo en el computador anfitrión.

El tipo de protocolo de datos que utilizamos en nuestra implementación del lazo de corriente de 20 mA es identico al utilizado en la interfaz RS232C, y se describe en la sección 5.1.1.

### 5.2.1- Diseño del circuito del lazo de corriente de 20 mA.

Para establecer un lazo de corriente es preciso tener un circuito cerrado en el que existan dos elementos, una fuente de corriente (elemento activo) y un sensor de corriente (elemento pasivo).

En nuestra terminal utilizamos dos optoacopladores TIL112 (ver Texas Instruments: The Optoelectronics Data Book) como elementos conversores de voltaje a corriente (en el extremo activo) y de corriente a voltaje (en el extremo pasivo).

La señal activa se obtiene de la salida QD (11) del MC1488, y se utiliza para polarizar el ILED del TIL112 activo, este ILED, a su vez, polariza al fototransistor, saturandolo, y por lo tanto estableciendo una corriente a través del lazo, que debe terminar en un ILED en serie con una resistencia de 47 E, en el computador anfitrión.

El extremo pasivo tiene como sensor de corriente a un ILED en serie con una resistencia de 47 E, que al ser establecida una corriente de 20 mA en el lazo, saturara al fototransistor del TIL112, el cual esta conectado a la entrada de RxD (3) del MC1489.

Las características del lazo de corriente se calculan mediante el siguiente circuito:

$$R = 1000/20$$

$$= 50 \text{ E}$$

seleccionando un valor comercial:

$$R = 47 \text{ E } \pm 5\% @ 1/4 \text{ W}$$

lo cual da:

$$I \text{ lazo} = 21.28 \text{ mA}$$

$$PDR = 21.28 \text{ mW}$$

$$PDR = 4.26 \text{ mW}$$

$$PDR = 25.54 \text{ mW}$$

En el lazo de entrada, con  $I_D = 21.28 \text{ mA}$ ,  $I_c = 4 \text{ mA}$  (aprox., este dato se obtiene de las curvas  $I_D/I_c$  que aparecen en las hojas de especificaciones del TIL 112) suponiendo un 20 % menos, por seguridad,  $I_c = 3.6 \text{ mA}$ , para que el transistor se sature necesitaremos, entonces:

$$R = 11,800/3.6$$

$$R = 3.3 \text{ K} \pm 5\% @ 1/4 \text{ W}$$

$$\text{PDR} = 42.48 \text{ mW}$$

El tipo de interfaz que utilizamos no sigue ningun estandar, excepto el que la corriente del lazo debe ser de aproximadamente 20 mA, el tipo de conector tampoco esta definido, por lo que estamos libres de escoger el que nosotros prefiramos, utilizaremos un conector telefonico de 4

hilos como conector.

## VI. DISEÑO DEL TECLADO

El teclado es una de las partes más importantes de la terminal, pues es el medio para que un ser humano pueda introducir información a esta. Un buen teclado debe ser ergonómico, debe tener teclas con acción positiva, grandes y fáciles de encontrar, y debe reunir un número suficiente de funciones para que no deban oprimirse más teclas que las estrictamente necesarias para realizar una cierta secuencia (es decir, deben definirse teclas que puedan generar una secuencia de ESC compleja, con una sola opresión, en lugar de tener que oprimir varias teclas para especificar la secuencia), es cómodo tener un grupo de teclas de edición que nos permitan realizar las secuencias de ESC más comunes, como mover el cursor y borrar partes de la pantalla, y es también útil la existencia de un grupo de teclas de función que puedan ser definidas por el programa del usuario.

No está dentro del alcance de esta tesis el diseñar la agrupación externa del teclado, etc., cosa que es trabajo de un experto en ergonomía, por lo cual escogeremos como modelo un teclado comercial diseñado por Hewlett-Packard para sus terminales y microcomputadoras, el HP46010AM. Este excelente teclado posee un arreglo que a mi me parece sumamente conveniente y cómodo, y una gran cantidad de teclas de función que permiten trabajar con gran rapidez y eficiencia,

por lo cual imitaremos su disposición externa (quiero dejar perfectamente asentado que no pretendo infringir ninguna patente de Hewlett-Packard, compañía a la cual pertenece el diseño de la configuración externa de este teclado, puesto que este es un trabajo exclusivamente académico, y no persigo ningún fin comercial, únicamente me limito a tomarlo como un excelente modelo).

Todos los circuitos relacionados con el teclado son, sin embargo, de diseño propio, y distintos a los utilizados por Hewlett-Packard.

## 6.1- ARQUITECTURA DEL TECLADO:

En esta sección se describen la configuración externa e interna del teclado a nivel de bloques.

### 6.1.1- Configuración externa:

El teclado consta de cinco grupos de teclas (ver figura 6.1.1/1), que son:

1. Grupo QWERTY. Es un teclado estandar de maquina de escribir, con 56 teclas .

2. Grupo de control. Formado por 6 teclas, implementa algunas de las funciones de control requeridas por la terminal: ESC/DELETE, PRINT/SEND, INTERRUPT/RESET, STOP, MENU, SYSTEM/USER.

3. Grupo de edición. Formado por 14 teclas: CLEAR TO END OF LINE, CLEAR SCREEN, INSERT LINE, DELETE LINE, INSERT CHR, DELETE CHR, CURSOR HOME, CURSOR UP, CURSOR LEFT, CURSOR RIGHT, CURSOR DOWN, SELECT, NEXT PAGE, PREVIOUS PAGE.

4. Grupo auxiliar. Formado por 22 teclas.

5. Grupo de funciones: Formado por 8 funciones definibles por

el usuario.

El teclado es una unidad separada del resto de la terminal (CRT/VTKBP/CRTC/VG/KBC), y conectada a esta por medio de un cable telefonico de 4 hilos, se escogio esta conformación separada por que nos da la posibilidad de mover el teclado para llevarlo hasta la posición en la que nos sea más comodo sin necesidad de mover toda la terminal.

#### 6.1.2- Configuración interna:

El teclado se organizo en forma matricial sobre una matriz de 16 x 16 (256 elementos).

Para poder tener el teclado separado del resto de la terminal es preciso proveer algun medio de comunicación entre estos, existen varios metodos posibles para esto. Es evidente que escogamos el metodo que escogamos el circuito que haga la exploración de la matriz: debiera estar en la misma unidad del teclado, pues de otra forma deberiamos desplazar 32 hilos entre el teclado y el resto de la terminal solo para las lineas de exploración.

Si emplazamos el circuito de exploración del teclado en el mismo teclado debemos transmitir el resultado de la exploración al VTKBP, esto se puede hacer en paralelo o en serie, ninguno de los dos procedimientos es muy comodo. Si hacemos la transmisión en paralelo debemos mandar 8 hilos, mas las lineas de control necesarias hasta el teclado, lo cual hace que este este demasiado rigidamente unido a la terminal y que sea dificil moverlo.

Si hacemos la transmisión en serie deberemos disponer de un SID en el teclado y de otro en la terminal, ademas deberiamos dotar al circuito de exploración de la inteligencia necesaria para controlar al SID, por lo cual deberiamos utilizar algun tipo de microcomputador para el

circuito de exploración (p.ej. un UPI 8741A de Intel).

Existe un procedimiento más sencillo (y mucho más ingenioso) para realizar esta función de comunicación, consiste sencillamente en poner un circuito de exploración en el teclado que cuando detecte una tecla oprimida genera una señal (STROBE) hacia la terminal, en esta se mantendrá un contador perfectamente sincronizado con el contador de exploración del teclado, que será detenido por el STROBE, su contenido será un reflejo del contenido de los contadores de exploración, y por lo tanto, de la posición en la matriz de la tecla recientemente oprimida. La señal de reloj que active a los contadores del teclado será la misma que active al contador imagen en el cuerpo de la terminal. Este contador imagen será sincronizado a ceros con el contador de exploración cada vez que ocurra un Hard Reset o una condición de Power On (figura 6.1.2/1).

Al circuito de exploración de teclado (que reside en el mismo teclado) lo conoceremos desde ahora como KBE (Key Board Explorer), y al circuito que se encarga de mantener la imagen de los contadores de exploración en la terminal, y de comunicarse con el VTKBP, lo conoceremos como KBC (Key Board Controller).

## 6.2- DISEÑO DEL CONTROLADOR Y DEL EXPLORADOR DE TECLADO.

### 6.2.1- Diseño del KBC.

El KBC se encargara, como ya dijimos, de mantener la imagen de los contadores de exploración del KBE y de interrumpir al VTKBP cada vez que se reciba una secuencia de teclas valida. El trabajo del KBC es bastante complejo, pues debera revisar toda una serie de condiciones que validen los datos recibidos del teclado, por lo cual sera necesario darle una cierta cantidad de inteligencia. Un circuito que puede encargarse a la perfección del trabajo es el Intel 8741A Universal Peripheral Controller. Este es un microcomputador dedicado que, entre otras, posee las siguientes características:

1. CPU de 8 bits.
2. 1024 x 8 EPROM. Memoria de programa.
3. 64 x 8 RAM. Memoria de datos.
4. 18 terminales I/O programables.
5. Un registro de estatus y dos registros de datos para implementar un sistema de comunicaciones maestro-esclavo asincrono, sin necesidad de tener un area dedicada de RAM para el paso de parametros.
6. Timer/contador de 8 bits.
7. Comunicaciones por DMA, Interrupción o cuestion.
8. Fuente de alimentación de +5 V.

Desde el punto de vista del Hardware las funciones del 8741A son las siguientes:

1. Sincronizarse al KBE tras un Hard Reset (es decir poner los contadores de exploración del KBE en ceros), mediante la señal de KBRESET/.
2. Generar un tren de pulsos (KBCLK) que serán contados por el contador de exploración del KBE, y le permitirán explorar todo el teclado, en forma secuencial.
3. Verificar la activación de la línea KEYIRQ, procedente del KBE y cuyo significado es que el KBE detecto una tecla oprimida.
4. Informar al VTKBP cuando se ha recibido un código de carácter válido, mediante la activación de la línea KBIRQ, conectada al RST5.5 del 8085AH.

Hay varias funciones más que deberán ser realizadas por el 8741A, pero que no tienen nada que ver con el Hardware, y que se refieren a la forma de interpretar la información procedente del KBE.

El KBC junto con el KBE puede verse en el correspondiente plano (Pl. 3). El 8741A está conectado a la estructura de I/D del VTKBP, en la página 2, de acuerdo al

siguiente mapa:

Dir.	RD/	WR/	Registro.
40	0	1	Lee DBBOUT
41	0	1	Lee STATUS
40	1	0	Escribe DBBIN (Datos)
41	1	0	Escribe DBBIN (Comandos)

La señal de reloj para el 8741A se obtiene de SYSCLK, por lo cual la frecuencia interna de operación del UPI es de  $3.072 \text{ M} / 15 = 204.80 \text{ KHz}$ , y el ciclo de instrucción tiene una duración de  $1 / 204.8 \text{ k} = 4.8828 \text{ uS}$ .

SYSCLK es conectado a X1 y X2 del 8741A en contrafase (mediante los dos 74LS04) y con dos resistencias de pull-up de 470  $\Omega$  para que el 1 lógico sobrepase el nivel de 3.8 V necesario por las entradas HMOS X1 y X2.

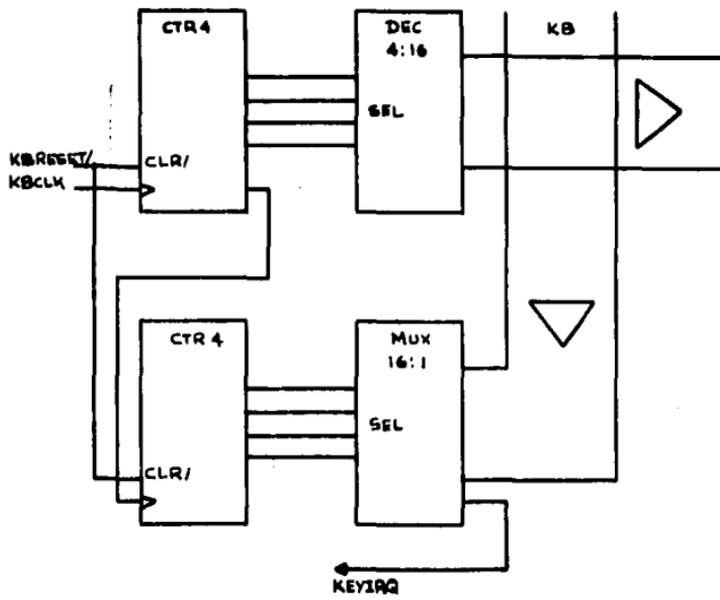
La señal de KEYIRQ esta conectada al 8741A en la terminal de TEST0. La señal de KBCLK se obtiene del puerto P10. La señal KBRESET/ se obtiene del puerto P11. La señal KBIRQ parte del puerto P24 (OBF interrupt).

#### 6.2.2- Diseño del KBE.

El explorador de teclado reside en la misma unidad que el teclado, esta enteramente realizado con circuitos integrados CMOS, los cuales fueron escogidos debido a su bajo consumo de potencia, a su alta inmunidad al ruido y a su magnifica histeresis (mayor a la de los circuitos TTL) que hace que sean menos sensibles al ruido mecanico de las teclas (el tiempo de debounce puede ser menor). El bajo consumo de potencia es preciso debido a que la alimentación debe hacerse a través de un cable telefonico, y un consumo alto podria provocar caidas en este.

El circuito consta, basicamente, de dos contadores sincronos de 4 bits (74C160) con clear asincrono, las terminales de CLR/ de ambos estan conectadas a la señal KBRESET/ proveniente del B741A, por lo que, cuando esta señal es llevada a cero, los contadores son puestos en cero.

Uno de los contadores recibe la señal de KBCLK en su terminal de CLK, este es el contador de renglones, y avanza un renglon por cada pulso recibido en la linea KBCLK. Cuando se terminan de contar los 16 renglones el contador genera un pulso en su salida de ripple carry (RC), este salida esta conectada al CLK del otro 74C160, que cuenta una columna por cada 16 renglones. De lo anterior se desprende que la exploración se hace columna por columna. Es decir, se selecciona la primera columna (columna 0) y se manda un cero



KB

por cada uno de los renglones en orden ascendente (0,1,...,15), luego se selecciona la siguiente columna y se repite el proceso, así sucesivamente hasta la columna 15.

El contador de renglones genera la entrada para un decodificador de 4 a 16 (74C154), que decodifica el conteo y va habilitando cada uno de los renglones. Inicialmente los renglones están en unos, el decodificador va desplazando un cero por cada uno de los renglones, habilitándolos secuencialmente.

El contador de columnas genera las entradas de selección de un multiplexor de 16 a 1 (74C150). Este multiplexor va habilitando todas las columnas en forma secuencial, si una tecla es oprimida en el renglón  $m$  y la columna  $n$  del teclado cuando el contador de columnas está en  $n$ , las entradas en la columna  $n$  serán canalizadas a KEYIRQ, cuando el renglón sea distinto a  $m$  KEYIRQ = 0, cuando el renglón sea  $m$  (estamos en la columna  $n$ ) KEYIRQ = 1. Nótese que el 74C150 invierte la señal de entrada ( $Q = \overline{EX}$ ), esta es la razón por la cual KEYIRQ = 1 cuando el 74C154 entrega un 0. (ver Pl. 3).

Los niveles lógicos de la familia CMOS 74Cxxx son enteramente compatibles con los de la familia TTL 74yxxxx, por lo cual no es preciso usar ningún tipo de pull-up en las líneas de comunicación.

A la entrada de Vcc en el circuito de exploración se

utiliza un capacitor de bypass de 470 uF, en paralelo con uno de .1 uF, su función es eliminar los transitorios en la línea de Vcc.

### 6.3- PROGRAMA DE CONTROL DEL KBC:

El teclado es controlado por el 8741A, que se encarga de realizar la exploración de este, de poner la información obtenida en un formato reconocible por el VTKBP y de transferirla a este.

Desde el punto de vista del software los detalles más importantes son:

1. El teclado es totalmente de exploración. Es decir: no existen líneas directas que no pertenezcan a la matriz de exploración.
2. Una entrada del teclado puede ser un código normal (NC) o un código especial (SC = ESC, CTRL, ALT, SHIFT1, SHIFT2).
3. Todos los códigos especiales pueden aparecer en una exploración válida, pero solo puede aparecer un código normal.
4. Una exploración válida es aquella en la que se detecta un solo código normal válido (es decir que duro más de 1.6 ms oprimida la tecla), o ese solo carácter más varios SC.
5. La exploración siempre es completa (es decir que se explora todo el teclado, no importa que ya se haya detectado

una tecla oprimida). Si la exploración no es válida, el resultado es un carácter nulo.

6. Una serie NC SC da como resultado solo el NC, a no ser que ocurra en la misma exploración, pero esta es tan rápida que entonces puede suponerse que el SC y el NC se oprimieron simultáneamente.

7. SC1 SC2 SC3 SC4 NC = SC4 SC3 SC2 SC1 NC. Esta es una forma particular de decir que no importa el orden de aparición de los SC siempre y cuando precedan al NC.

8. Para que un NC vuelva a ser reconocido tras haber aparecido una vez, es preciso que:

- i. Haya habido por lo menos un NC o SC+NC entre sus dos apariciones.
- ii. Haya habido uno o más caracteres nulos (exploraciones en las cuales no se detectó ninguna tecla oprimida, o se detectó un 2-Key Lockout) entre sus apariciones.
- iii. Hayan pasado 1.5 S entre apariciones. Repetición automática.

9. El NC es el terminador de una secuencia. Una secuencia consta de varios SC más un solo NC.

Es importante reconocer que se está implementando un sistema de 2-Key Lockout en el teclado, cuando se trata de

una tecla que no sea un código especial (CTRL, SHIFT, ESC, ALT), es decir que mientras estén dos o más teclas normales apretadas en forma simultánea en el teclado no se reconocera a ninguna de ellas. Sin embargo se pueden oprimir en forma simultánea todas las teclas de código especial, y todas ellas serán reconocidas, esto nos permite construir secuencias especiales de control (por ejemplo, para el Hard Reset, que se genera apretando CTRL SHIFT RESET simultáneamente) que no pueden ser activadas cuando se oprime una tecla en forma errónea (por ejemplo, tener una sola tecla para el Hard Reset sería en suma peligroso, pues podría accionarse por accidente, en tanto que utilizar un interruptor oculto sería muy incómodo, por lo tanto es muy superior utilizar secuencias como la anterior).

Este teclado no tiene ningún tipo de línea no matricial, es decir, conectada directamente a la terminal y que pueda ser activada independientemente del orden de exploración, esto le da un control muy superior al 8741 sobre el teclado.

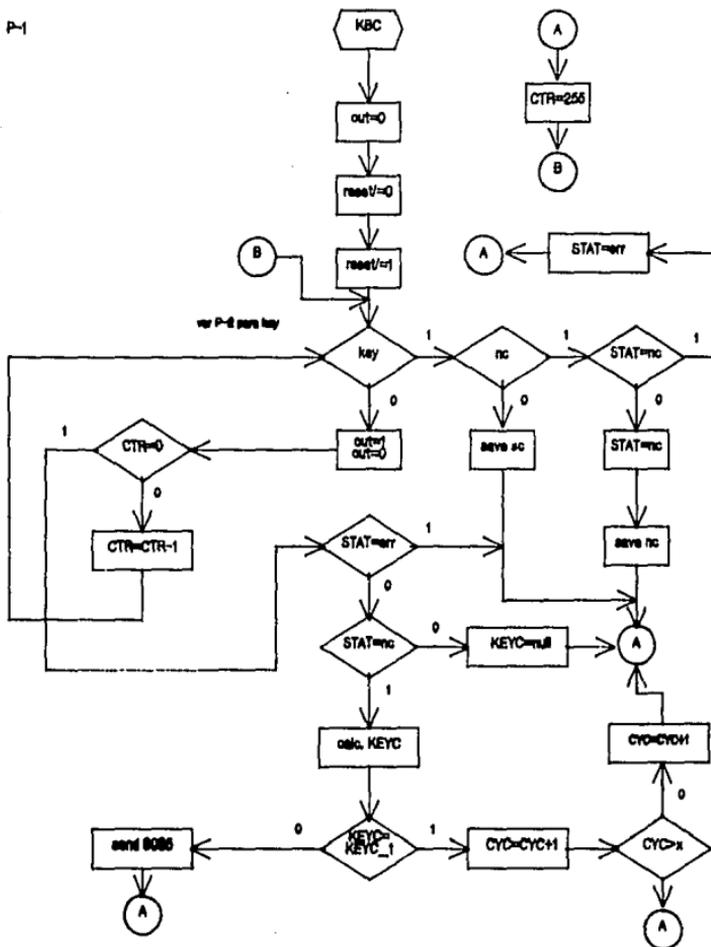
En la figura 6.3.0/1 puede verse el diagrama de flujo del programa que implementa la estructura anterior.

A continuación incluimos dicho programa en ensamblador 8741A.

diagrama de flujo

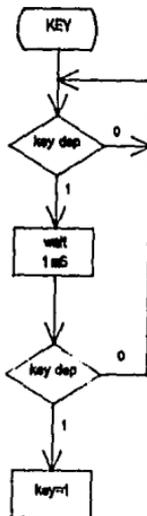
KBC-OS V1.0-01

P-1



P-2

bloque correspondiente  
a key en P-2



programa de control del KBC

\* KBC \*

R0 = CTR

R1 = STAT B0 = N, B1 = NC, B2 = ERROR

R2 = KEYC

R3 = SC

R4 = KEYA

R5 = CYCLE

R6 = KEY

F7 = CTRL

F6 = SHIFT

F5 = ALT

F4 = ESC

P10 = KBCLK

P11 = KBRESET/

TEST0 = KEYIRQ

P24 = KBIRQ

P17 = CLICK

EN FLAGS

; HABILITA UN DBF IRQ

DIS I

; DESHABILITA INTS.

DIS TCNT

; DES. IRQS. DEL TIM.

STOP TCNT

; DETIENE TIM.

SEL RBO

; SEL REG. BK. 0

```

ANL      P1,#0H          ; OUT, RES/ = 0
ORL      P1,#2H          ; OUT=0,RES/=1
MOV      R2,#0H          ; KEYC = NULL
MOV      R4,#0H          ; KEYA = NULL
;
;
;          * NEWSKAN *
;
;  RUTINA DE EXPLORACION. RASTREA TODO EL TECLADO EN
;  BUSCA DE UNA TECLA ORIMIDA
;
;
NEWSKAN:  MOV      R1,#0H          ; STAT = 0
          MOV      R0,#0FFH       ; CTR = 255
          MOV      R3,#0H          ; SC = NULL
          MOV      R6,#0H          ; KEY = NULL
SCAN:    CALL     KEY              ; SI HAY UNA TEC.
          JNC      NOKEY           ; VE A KEYIN
          CALL     KEYIN
NOKEY:   ORL      P1,#1H          ; OUT = 1
          ANL      P1,#0FEH       ; OUT = 0
          DJNZ    R0,SCAN         ; CTR=CTR-1 GO SCAN
;
;  TERMINA NEWSKAN
;
          MOV      A,R1           ; A = STAT
          JB2     NULL            ; ERROR, EXP. NULA
          JB1     NC              ; NORM. CODE
;

```

```

; NO SE OPRIMIO TECLA ALGUNA, O SE OPRIMIERON MAS DE 2
;
NULL:  MOV     R4,#0H           ; KEYA = NULL
        MOV     R2,#0H           ; KEYC = NULL
        JNP     NEWSKAN
;
; SE OPRIMIO UNA TECLA NORMAL
;
NC:     JOBF    NEWSKAN         ; OBF =1
        MOV     A,R6            ; A = KEY
        XRL     A,R2            ; KEY XOR KEYC
        JNZ     DIFF
        MOV     A,R4            ; SC XOR KEYA
        XRL     A,R3
        JNZ     DIFF
        CALL    CYCLE          ; ES LA MISMA TECLA
        JC      REP            ; REPEAT
        RET     ; QUE EN EXP - 1
;
; ES UNA TECLA DISTINTA
;
DIFF:   MOV     R5,#0H
REP:    MOV     A,R3            ; LEVANTA LOS FLGS.
        MOV     STS,A          ; DE SPECIAL CODE.
        MOV     R4,A           ; KEYA = SC
        MOV     A,R6           ; A = KEY
        MOV     R2,A           ; KEYC = KEY
        OUT     DBB,A          ; DBBOUT = KEY

```

JMP NEWSCAN

;  
;  
;  
;  
;  
;  
;  
;  
;  
;

\* KEYIN \*

DETERMINA QUE TIPO DE TECLA SE OPRIMIO. SI ES TECLA  
ESPECIAL LA DA DE ALTA EN SC, SIND LA GUARDA EN KEY  
EL CODIGO ES IGUAL AL NUMERO EN CRT.

```
KEYIN:  MOV     A,R0           ; A = CTR
        XRL     A,#CTRL      ; = CTRL
        JNZ     NEXT1        ; NO
        MOV     A,R3         ; SI: A = SC
        ORL     A,#B0H       ; F7 = 1
        RET
NEXT1:   MOV     A,R0
        XRL     A,#SHIFT     ; = SHIFT
        JNZ     NEXT2        ; NO
        MOV     A,R3         ; SI:
        ORL     A,#40H       ; F6 = 1
        RET
NEXT2:   MOV     A,R0
        XRL     A,#ALT       ; = ALT
        JNZ     NEXT3        ; NO
        MOV     A,R3         ; SI:
        ORL     A,#20H       ; F5 = 1
        RET
```



```

NDINK:  RET                ; DE TECLA
;
;
;          * WAIT *
;
;          T = 15 X 23 X 4.0028 uS = 1.6846 mS
;
WAIT:   MOV      A,#0FH    ; ESPERA 15 CICLOS
        DEC      A        ; 1 CIC = 4.8829 uS
        JNZ     WAIT      ; ESTA RUT. TIENE
        RET                ; 23 CICLOS
                          ; 1.6846 MS
;
;
;          * CYCLE *
;
;          ESTA RUTINA, LLAMADA TRAS UNA TECLA REPETIDA, DETERMINA
;          SI LA TECLA HA ESTADA OPRIMIDA DURANTE MAS DE 1.5 S,
;          CASO EN EL CUAL SE SUPONE QUE DEBE REPETIRSE.
;
CYCLE:  CLR      C
        MOV      A,R5      ; A = CYCLE
        CPL      A        ; A = COMPLEMENTO A
        ADD     A,#0FH    ; A = 0EFH - CYCLE
        JC      REPEAT    ; SI CY CYCLE > 0FEH
        INC     R5        ; NO: CYCLE = CYCLE+1
REPEAT: RET                ; REGRESA.
;

```



## VII. DISEÑO DE LA FUENTE DE ALIMENTACIÓN.

La fuente de potencia es uno de los elementos principales de cualquier dispositivo, una fuente mal diseñada puede comprometer en alto grado la confiabilidad del equipo, e incluso afectar sus características funcionales. Se ha calculado que hasta un tercio de las fallas más comunes en equipo de computación se deben a problemas con el suministro de energía.

La fuente de nuestra terminal debe suministrar tres diferentes voltajes:  $+5.0\text{ V} \pm 5\%$ ,  $+12.0\text{ V} \pm 10\%$  y  $-12.0\text{ V} \pm 10\%$ . Existen varios tipos de fuente que podrían ser utilizados, sin embargo el más recomendable para una terminal de video es, sin duda, la fuente de potencia de conmutación. En este tipo de fuente no se utiliza un gran transformador de 60 Hz para reducir la alimentación de ac, lo cual presenta dos ventajas:

1. Reducción del tamaño, peso y costo de la fuente. Logrado al eliminar el transformador de 60 Hz y sustituirlo por uno de más alta frecuencia, y por lo tanto más pequeño. Este tipo de fuente es, así mismo, mucho más eficiente, por lo que utiliza transistores más pequeños, y debo

disipar mucho menos calor.

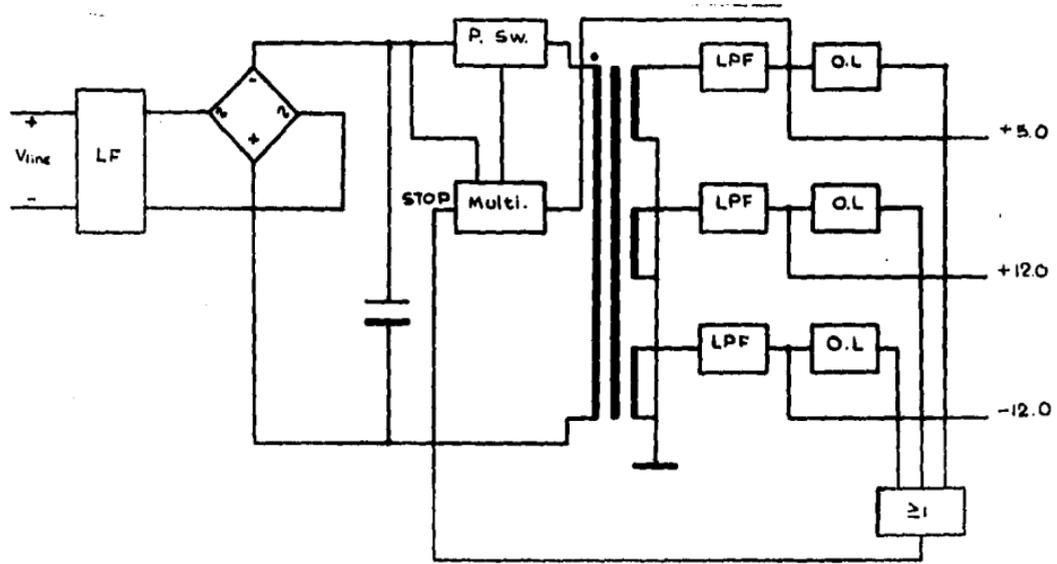
2. Eliminación de la posibilidad de interferencia entre las frecuencias de barrido vertical (60 Hz) y la energía radiada (inducida) por el transformador de 60 Hz.

## 7.1- ESTRUCTURA DE LA FUENTE:

El diagrama de bloques de la fuente puede verse en la figura 7.1.0/1.

Como puede verse esta formada por un bloque conversor AC/DC (rectificador más filtro RC) seguido por un conmutador de potencia, controlado (encendido/apagado) por una señal procedente de un multivibrador asimétrico cuyo ciclo de trabajo es proporcional a la retroalimentación tomada de la señal de +5.0 V, es decir que la fuente siempre tratara de mantener la salida de +5.0 V constante. Las salidas de +12.0 V y -12.0 V están dadas por un ratio fijo del transformador de salida, con relación a la línea de +5.0 V.

Cada salida posee un detector de sobrecorriente, al existir un corto circuito en alguna de ellas el multivibrador dejara de oscilar, regresando todas las salidas a cero.



PS

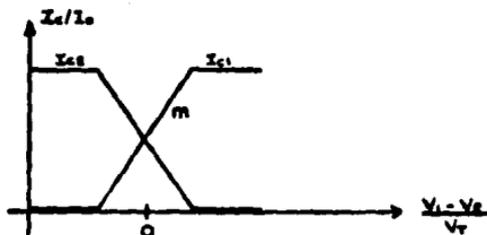
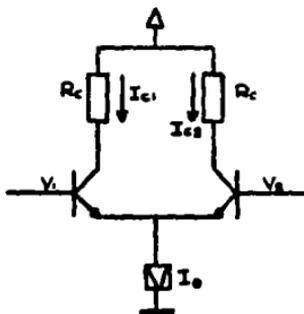
## 7.2- MULTIVIBRADOR ASIMETRICO Y FILTROS DE SALIDA.

El multivibrador asimétrico puede verse en la figura 7.2.0/1.

El ciclo de trabajo de este multivibrador es proporcional a la diferencia que existe entre una muestra del voltaje de +5.0 V y el voltaje  $V_r$  (voltaje de referencia de 2.50 V  $\pm$  25 mV, obtenido de un MC1403A).

Esta diferencia es amplificada mediante un amplificador diferencial, formado por Q1 y Q2, el par diferencial se saturará (positiva o negativamente) si la diferencia existente entre las dos señales es de más de 120 mV, por lo cual su acción de control será sumamente rápida, aumentando o disminuyendo el ciclo de trabajo del multivibrador con gran rapidez, para responder a un cambio de carga en la línea de +5.0 V.

Este primer par diferencial alimenta, a su vez, a un segundo par, formado por Q3 y Q4, que actúan como fuentes de corriente para el multivibrador, el tiempo de carga y de descarga de los capacitores C1 y C2 dependerá directamente de estas corrientes I3 e I4, es así como al variar la diferencia en el par 1-2 se produce un desbalance en el par 3-4 y por lo tanto un cambio en el ciclo de trabajo del multivibrador.



Las relaciones que rigen las corrientes existentes en un par diferencial como el anterior son:

$$I_{c1} = I_o / ( 1 + \exp [(V_2 - V_1) / V_t] )$$

$$I_{c2} = I_o - I_{c1}$$

donde  $I_o$  es constante.

En la region activa del par diferencial la pendiente es constante e igual a la transconductancia diferencial ( $g_{md}$ ):

$$g_{md} = dI_{c1} / d(V_1 - V_2)$$

$$g_{md} = I_o / 4V_t$$

El voltaje diferencial de entrada al segundo par diferencial es:

$$V_{d2} = V - R_c I_{c1} - (V - R_c I_{c2})$$

$$V_{d2} = R_c (I_o - I_{c1} - I_{c1})$$

$$V_{d2} = R_c (I_o - 2I_{c1})$$

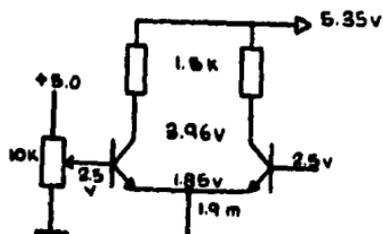
$$V_{d2} = R_c (I_o - 2g_{md1} V_{d1})$$

La corriente en el segundo par diferencial es:

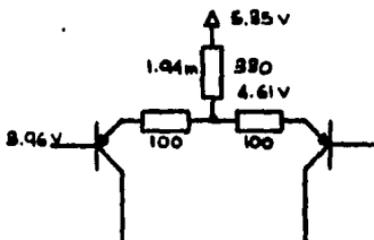
$$I'_{c1} = g_{md2} V_{d2}$$

$$I'_{c2} = I_o' - g_{md2} V_{d2}$$

Utilizando las anteriores relaciones para diseñar los dos pares diferenciales:



El voltaje de alimentación para el multivibrador se obtiene mediante un diodo zener de 4.7 V en serie con dos diodos 1N901, utilizados para compensar al zener contra variaciones de temperatura, y es de 5.35 V, aproximadamente. Con este voltaje de alimentación y escogiendo una  $I_0 = 1.85$  mA en el primer par diferencial se obtienen los valores mostrados en el dibujo. Se calibra el potenciómetro de 10 K de tal manera que cuando el voltaje de retroalimentación sea de 5.00 V el voltaje en la base de Q1 sea 2.50 V, y por lo tanto el par este balanceado. Q1 y Q2 son transistores BC548C.



En el segundo par diferencial se escoge una  $I_o$  de 1.9408 mA, y se incluyen dos resistencias de 100 E en los emisores de los dos transistores BC558C para proveer degeneración, y por lo tanto una mayor estabilidad del circuito.

Para  $V_o = 5.0$  V;  $I_{c1} = I_{c2} = 925$  uA,  $I_{c'1} = I_{c'2} = 970$  uA.

Para calcular el multivibrador nos basamos en el hecho de que la relación voltaje-corriente en un capacitor es:

$$C \, dV = i \, dt$$

si  $i$  es constante e igual a  $I$ , entonces:

$$V = It / C$$

El periodo de la señal generada por el multivibrador esta dado por la suma de los tiempos de carga y de descarga de los dos capacitores  $C_1$  y  $C_2$ . Si  $C_1 = C_2 = C$ :

$$T = t_1 + t_2$$

$$T = CV (1/Ic'1 + 1/Ic'2)$$

$$T = CV (Ic'1 + Ic'2) / (Ic'1 Ic'2)$$

$$f = 1 / T$$

Si escogemos  $C = 3.3 \text{ nF}$ , la frecuencia de operación del multivibrador sera:

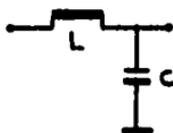
$$f_0 = 27.471 \text{ kHz @ } V_0 = +5.0 \text{ V}$$

El valor dc de una señal cuadrada con ciclo de trabajo  $\beta$  es  $\beta V_m$ , donde  $V_m$  es el valor maximo de la señal. El ancho de banda de la señal esta dado por la siguiente relación:

$$BW = 1 / \beta$$

Donde  $\sigma$  es el ancho de un pulso ( $\sigma = \beta T$ ), para  $\beta = \frac{1}{2}$ : BW  
=  $2 / T = 54.942$  KHz.

Por lo tanto el BW del filtro debe ser mucho menor a  
54.942 K para que solo la componente dc pase.



Si escogemos la frecuencia de corte del filtro 3 decadas  
abajo de 54.942 KHz, entonces  $f_c = 54.942$  Hz, y:

$$f_c = 1 / [2 * 3.1416 * \text{sqr}(LC)]$$

$$L = 10 \text{ mH}$$

$$C = 839.13 \text{ uF}$$

La  $f_c$  esta, por lo tanto, a 3 decadas de la primera  
armonica y la atenuación de esta sera de -120 dB (/

1,000,000).

Los filtros para +12.0 y -12.0 tienen los mismos valores.

$$V_o = \beta V_m$$

$$V_m(V_o = 5.0) = 10 \text{ V}$$

$$V_m(V_o = 12.0) = 24 \text{ V}$$

$$V_m(V_o = -12.0) = -24 \text{ V}$$

### 7.3- ETAPA DE ENTRADA.

En la etapa de entrada el primer elemento con el cual nos encontramos es el filtro de línea, el cual es un filtro pasivo que se encarga de atenuar transitorios de línea con una frecuencia mayor a 1 KHz, se utilizó un filtro de línea HP 9135-0038 el cual opera correctamente con corrientes hasta de 1 A, y voltajes de hasta 200 Vrms.

El siguiente elemento es un fusible de protección de 500 mA (la máxima corriente de entrada a la fuente es de 400 mA -en el peor caso-, como se muestra en la siguiente sección), de fusión rápida. Este fusible no es lo suficientemente rápido como para evitar que un sobrevoltaje destruya el transistor de conmutación (el elemento más vulnerable de la fuente), sin embargo si evita que la fuente de poder se incendie en el caso de que esto llegue a ocurrir, además, en esas circunstancias limita la falla a la destrucción del transistor, ya que los diodos rectificadores pueden soportar la corriente de cortocircuito del transistor durante unos 500 mS. Y la etapa de salida está aislada de la etapa de entrada por el transformador de 27 KHz (el cual también puede soportar la corriente de corto circuito durante el tiempo que el fusible requiere para destruirse -Para una descripción detallada del proceso de destrucción del fusible y de la protección que este aporta, véase el Motorola Silicon Rectifiers Data Manual, cap. 9-).

A continuación sigue el puente de diodos utilizado como rectificador, cuyo calculo se muestra en seguida (ver Pl. 4), cada uno de los diodos se pone en paralelo con un pequeño capacitor ceramico de 0.1 uF,  $\pm 10\%$ , 400 V con objeto de protegerlo contra los transitorios de alta frecuencia que no hayan sido atenuados suficientemente por el filtro de linea. Debido a que el transistor de conmutación esta trabajando casi en su  $V_{ceo}$  de ruptura (10% de margen de seguridad), es conveniente que esta fuente de poder sea usada con algun tipo de acondicionador de linea, que regule esta y recorte los transitorios de larga duración (que no seran eliminados por el filtro de linea, ni por los capacitores de protección) mediante el uso de un crow-bar o de un MOSV (metal-oxide varistor). La mayor causa de fallas en fuentes de poder de terminales y microcomputadoras comerciales es precisamente este tipo de transitorios en lineas no acondicionadas (el estandar en los E.E.U.U. para las lineas de potencia es de 115 V  $\pm 10\%$ , 60 Hz  $\pm 1\%$ , y esta fuente de poder esta perfectamente diseñada para operar bajo estos limites, usados en varios paises).

$$V_i = 115 \text{ V} \pm 20\% \text{ rms} \quad (92, 138 \text{ V})$$

$$V_{im} = 162.63 \text{ Vpk} \quad (130.10, 195.16 \text{ V})$$

$$V_I = 162.63 - 1.4 = 161.23 \text{ V}$$

$$\Delta V = V_{imT1} / T2$$

$$T2 = RC$$

$$T1 = 8.33 \text{ mS}$$

El rechazo al rizo de la etapa reguladora es de 120 dB a 55 KHz, a 120 Hz es de 6.7764, sin embargo gracias al conmutador de potencia el rizo casi no tiene importancia, ya que en el periodo de 8.33 mS se tomaran 228.83 muestras (a 27.4 KHz), y el rizo casi desaparecera.

Para que podamos calcular la R de carga del filtro es preciso conocer que cantidad de corriente consume nuestro circuito, esta cantidad puede obtenerse de la siguiente tabla, que muestra los consumos de corriente maximos de todos los circuitos integrados utilizados: (en mA).

Circuito.	Icc(max)	Icc(tot) 5
8085AH	120	120
8257	120	120
8212	130	130
8275	160	160
8741A	120	120

74LS164 x 4	27	108
74LS160	32	32
74LS02	5.4	5.4
74LS368	21	21
74LS373	40	40
74LS244 x 3	54	162
74LS245 x 2	95	190
2764	150 (50 STBY)	150
74LS166	38	38
74LS32 x 2	9.8	19.6
74LS08 x 3	8.8	26.4
74LS74	8.0	26.4
74LS163	32	32
74LS138 x 5	11	55
TMS4016NL x 16	90	1440
74LS374 x 2	40	80
B251	100	100
B253	140	140
74LS112	6.0	6.0
MC1488	25 (+12 V)	0
MC1488	23 (-12 V)	0
MC1489	26	26
TIL112 x 2	20	40
74LS153 x 2	10	20
74LS77	12	12
74LS162	32	32
74LS175 x 2	18	36
74LS157	11	11

NE555	12	12
74C163 x 2	300 u	600 u
74C150	300 u	300 u
74C154	300 u	300 u
TOTAL		3493.60 mA

Suponiendo un 30% de exceso (por seguridad y para proveer futuras expansiones):

$$I_{cc} = 4,541.68 \text{ mA}$$

$$I_{cc12} = 500 \text{ mA}$$

$$I_{cc-12} = 500 \text{ mA}$$

$$\text{Con } R = V / I$$

$$I = I_5 (N_5/N_p) + I_{12} (N_{12}/N_p) + I_{-12} (N_{-12}/N_p)$$

$$N_5 / N_p = 10 / 161.23 = 0.0620$$

$$N_{12} / N_p = N_{-12} / N_p = 24 / 161.23 = 0.1489$$

$$I = 390.79 \text{ mA}$$

$$R = 161.23 / 390.79 \text{ m} = 412.57 \text{ E}$$

Con  $C = 10,000 \text{ uF}$ ;

$$DV = 325.53 \text{ mV}$$

El rizo de salida sera, aproximadamente:

$$\text{rizo} = DV (NS/Np) / 22B$$

Para  $5 \text{ V}$ ; rizo =  $88.521 \text{ uV}$ , y con el filtro LC ( $-6.76 \text{ dB}$ ):

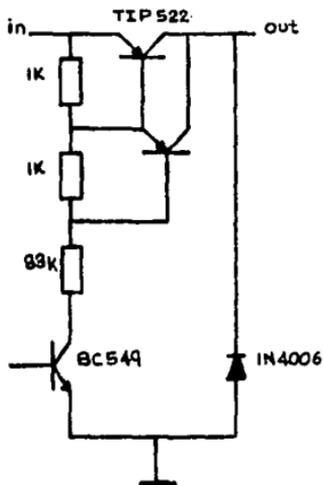
$$\text{rizo} = -87.835 \text{ dB} = 40.572 \text{ uV}$$

Es decir, que el rizo es de apenas el  $811.44 \text{ E-6\%}$ .

Los diodos rectificadores que se utilicen deben tener un PIV de  $195 \text{ V}$ , y soportar una  $I_{av} = 0.5 \text{ Iload} = 390 / 2 = 195 \text{ mA}$ , y una corriente pico de  $8 \text{ Iload}$ , es decir de  $3.120 \text{ A}$ .  
Unos diodos que cumplen perfectamente con estos requisitos son los  $1N4004$ , con un  $PIV = 400 \text{ V}$ ,  $I_{av} = 1.0 \text{ A}$  e  $I_{surge} = 30 \text{ A}$ .

#### 7.4 - CONMUTADOR DE POTENCIA.

Una parte esencial de la fuente es el conmutador, este esta formado por un par darlington usados en las regiones de saturación y de corte y un diodo de flyback:



Como puede verse la salida del multivibrador estable asimétrico pasa a un interruptor RTL, formado por el transistor Q3 y la resistencia R3, cuando la salida del multivibrador es 1 lógico el transistor Q3 esta en saturación, y por lo tanto también lo están los transistores Q2 y Q1.

El diodo de flyback es un IN4004. El transistor Q1 debe

soportar una corriente de 390 mA cuando esta en saturación, y un  $V_{ce0}$  de 195 V cuando esta en corte. Un transistor que puede ser utilizado es el TIP522, transistor PNP de potencia de silicio, con un  $V_{ce0} = 200$  V y un  $I_{cmax} = 2A$ .

La potencia disipada por este transistor sera:

$$PD = V_{cesat} I_c$$

donde  $I_c = I_{pk} \beta$

si  $\beta$  llegara a ser 1:  $I_c = I_{pk} = 390$  mA,  $V_{cesat\ max} = 1.5$  V, por lo tanto:

$$PD = 585\ mW$$

La resistencia termica de la union al aire libre de este transistor es  $R_{\theta jc} = 175$  C/W (JEDEC TO-39), por lo tanto:

$$PD = (T_j - T_a) / R_{\theta jc}$$

$$T_{jmax} = 200\ C$$

$$T_j - T_a = 175 * 0.585$$

$$T_j - T_a = 102.38\ C$$

$$T_a = T_j - 102.38$$

$$T_a = 97.625 \text{ C}$$

Es decir que el transistor de conmutación puede trabajar con una temperatura ambiente de hasta 97.625 C sin necesitar disipador.

La  $I_b$  del transistor de conmutación sera de aproximadamente  $390 \text{ m} / 20 = 19.5 \text{ mA}$ , que puede ser manejada sin ningun problema por el segundo TIP522.

Cuando el transistor Q3 esta saturado deben estarlo tambien Q2 y Q1, por lo tanto  $I_{c3} = 19.5 \text{ mA} / 20 = 975 \text{ uA}$  y  $V_{b2} = 162.63 - 1.4 = 161.23$ :

$$R_3 = 161.23 / 975 \text{ u} = 165.37 \text{ K} @ 157.20 \text{ mW}$$

Por lo tanto  $R_3 = 150 \text{ K} @ \frac{1}{4} \text{ W}$

$R_1$  se escoge para mantener correctamente polarizados a Q1 y a Q2 aun estando en corte Q3, si  $R_1 = R_2 = 1 \text{ K}$ ,  $I_1 = I_2 = 700 \text{ uA}$ . Q3 es un BC548C.

## 7.5- PROTECCION CONTRA CORTO CIRCUITO.

Cada una de las lineas de salida (+5.0, +12.0 y -12.0) tiene un circuito detector de sobrecorriente (ver Pl. 4).

Este circuito genera una señal que detiene al multivibrador cuando detecta un corto circuito en alguna de las salidas.

En la linea de +5.0 V el circuito de protección asume que cualquier corriente de más de 5.0 A es una corriente de corto circuito, como elemento de detección se utiliza una resistencia de 0.1 E @ 1 W en serie con la salida de +5.0 V, en caso de circular una corriente en exceso de 5 A por esta resistencia, se detectara una caída en exceso de 0.5 V en ella, que sera amplificada por un amplificador diferencial, y utilizada para generar la señal MSTOP que detiene al multivibrador. Para las lineas de +12 y -12 la corriente de corto circuito sera de 1 A, y producira una caída de 0.47 V en una resistencia de 0.47 E @ 1 W.

El voltaje de alimentación del circuito de detección se obtiene de la fuente de poder del multivibrador y es de 5.35 V, el voltaje de comparación se obtiene del voltaje de referencia de 2.50 V del multivibrador.

## 7.6- ESPECIFICACIONES GENERALES:

$$V_{in} = 115 \text{ V} \pm 10\%$$

$$f_{in} = 60 \text{ Hz} \pm 10\%$$

$$I_{in \text{ max}} = 400 \text{ mA}$$

$$V_o = +5 \text{ V} \pm 5\% @ 5 \text{ A}$$

$$+12 \text{ V} \pm 5\% @ 500 \text{ mA}$$

$$-12 \text{ V} \pm 5\% @ 500 \text{ mA}$$

$$T_{op} = 0 - 45 \text{ C (ambiente)}$$

La fuente de conmutación genera una gran cantidad de ruido de alta frecuencia, por lo tanto es preciso encapsularla en un blindaje de aluminio perfectamente aterrizado. Este blindaje sirve también como protección para el usuario, pues en caso de que algún conductor de esta se desprenda y entre en contacto con él, se generara un corto circuito a tierra y el fusible de entrada se fundiría, es por lo tanto totalmente preciso que exista una buena conexión entre tierra física y la terminal (todo el chasis metálico interno de la terminal está conectado a la tierra del contacto polarizado de la fuente, no al neutro, tal como es marcado por el NEC).

## VIII. PROGRAMA DE CONTROL DE LA TERMINAL

La terminal de video posee una electronica sumamente sofisticada y compleja que es controlada enteramente por el VTKBF, a partir de una serie de rutinas preprogramadas, las tareas de estas rutinas son:

1. Inicializar el sistema (tras un Hard Reset).
2. Comunicarse con el computador anfitrión.
3. Recibir información desde el teclado, e interpretarla para ponerla en una forma inteligible para el computador anfitrión.
4. Visualizar la información recibida desde el computador anfitrión.
5. Permitir la comunicación entre los diversos grupos que forman a la terminal.

Para lograr estas tareas se utilizan las siguientes rutinas:

- |                |       |
|----------------|-------|
| 1. Cursor Up   | ESC[A |
| 2. Cursor Left | ESC[D |

3. Cursor Right	ESC[C
4. Cursor Down	ESC[B
5. Row Up	
6. Row Down	
7. Column Left	
8. Column Right	
9. Home	ESC[H
10. Clear Screen	
11. Clear to end of row	ESC[K
12. Clear to start of row	ESC[W
13. Clear to end of screen	ESC[J
14. Clear to start of screen	ESC[Z
15. Backspace	CTRL H
16. Line Feed	CTRL J
17. Carriage Return	CTRL M
18. Clear line	
19. Fill screen	
20. Enter local mode	ESC[l
21. Enter remote mode	ESC[r
22. XON	CTRL Q
23. XOFF	CTRL S
24. Enter graphics 0	ESC[0g
25. Enter graphics 1	ESC[1g
26. Enter alpha mode	ESC[A
27. Cursor move	ESC[ <i>x</i> , <i>yc</i>
28. Reverse video	ESC[i
29. Normal Video	ESC[n
30. Blinking Video	ESC[b

### 31. Display

Estas subrutinas mencionadas son en realidad una lista de las funciones basicas que son ejecutadas por la terminal.

## **8.1 - DIAGRAMA DE FLUJO. MODO REMOTO.**

El proceso que el VTKB sigue para establecer la comunicación entre un ser humano y el computador esta dado por el digrama de la figura 8.1/0/1.

## **8.2 - DESCRIPCIÓN DE LA PROGRAMACIÓN DE LA SECCIÓN DE VIDEO.**

La forma en la que la terminal funciona depende del modo en el que este operando (remoto o local; alfabetico, grafico 1 o grafico 0). El software que fue desarrollado en esta tesis solo incluye el caso en que la terminal esta trabajando en modo remoto (es decir, conectada a un computador anfitrión) y alfabetico. Existe, sin embargo, suficiente espacio en el EPROM 2764 que contiene a VTKBOS V0.0-01 como para incluir el software que nos permita esos otros modos de aplicación.

### **8.2.1 - Características de la operación en modo remoto alfabetico. Paginas de video, estatus.**

Para que el 8085 tenga un control total sobre la operación de la terminal es preciso que conozca el estado de esta en todo momento. Este estado es reflejado por un byte de

memoria llamado Status, cuya forma es la siguiente:

b0 = 1 :secuencia de ESC en progreso.

= 0 :operación normal.

b1 = 1 :XOFF

= 0 :XON

b2 = 1 :ERROR

= 0 :NO ERROR

b3 = 1 :input buffer lleno

= 0 :input buffer sin llenar.

El bit 0 indica si se esta ejecutando una secuencia de ESC, y es utilizado por la subrutina que procesa estas secuencias. El bit 1 refleja el estado del flag XOFF/XON, cuando el computador anfitrión esta disponible manda un XON (transmitter on) a la terminal, cuando no manda un XOFF a esta. El bit 2 nos indica si existe o no una condición de error en la terminal. El b3 nos indica si podemos seguir o no recibiendo datos desde el computador anfitrión, cuando b3 = 1 la VTIB genera un XOFF hacia el computador anfitrión, y cuando b3 vuelve a ser 0, se le manda un XON al computador.

La terminal esta diseñada de tal modo que la representación (ASCII en el modo alfabetico) de lo que se esta visualizando esta guardada en un area llamada memoria imagen de video (VIO RAM), es decir, que para cada caracter en la pantalla su representación ASCII esta guardada en la memoria RAM de la maquina, en una posición proporcional a la

que ocupa en la pantalla.

Pueden guardarse dos paginas de video en su memoria, en el modo en el que se este (aunque el modo alfabetico y el modo grafico 0 comparten la misma VIO). En modo alfabetico la pantalla del CRT es una ventana a la memoria imagen de video (VIO), y puede deslizarse hacia arriba y hacia abajo de esta memoria, viendose siempre una pagina en la pantalla. Esto implica que las paginas no estan fijadas en la memoria, es decir, que no existe un bloque de memoria que contenga siempre a la pagina 0 y otro que contenga siempre a la pagina 1.

A medida que vamos llenando la VIO RAM la pantalla se va desplazando hacia abajo sobre esta, cuando la VIO RAM se llena totalmente el ultimo renglon de la pantalla pasa a ser el primer renglon de la VIO RAM, que es borrado antes de ser visualizado. De hecho la VIO puede considerarse circular y la pantalla puede considerarse como una ventana, tambien circular, sobre esta VIO.

La posición del cursor dentro de la pantalla esta dada por las variables CURSOX y CURSOY, que son la posición x, y de este. El inicio de la pagina visualizada en la pantalla esta dado por la variable TOPFAG (Top of page), que contiene la dirección del primer caracter del primer renglon de la pagina. Para nuestro programa se supone que la VIO RAM se inicia en la dirección 0H de memoria, en realidad para que la

VIO pueda iniciarse en cualquier otro lugar su dirección de inicio esta dada por la constante OFFSET (2000H) esto se hace para que el programa sea independiente de la posición real de VIO en la memoria de la terminal, por lo cual si hay que hacer un cambio en la posición de la VIO en alguna nueva revisión de la terminal el unico cambio que deba hacerse al programa sea un cambio en el EQU que define a OFFSET.

El inicio de la pagina visualizada dentro de la VIO esta dado por:

TOPPAG + OFFSET

Un renglon de la pantalla contiene 55 caracteres (37H caracteres), y una pantalla contiene 24 (18H) renglones, es decir que hay 1320 (528H) caracteres por pagina:

PAGINA	DE	A
0	0H	527H
1	528H	A4FH

Otra variable que se mantiene en la memoria es ROW, esta variable contiene la dirección del primer caracter del renglon en el que esta el cursor (relativa a TOPPAG), esta variable no es igual a CURSDY, ya que CURSDY es un byte que

nos indica el número de renglon, en tanto que ROW son dos bytes que nos dan la dirección del primer caracter del renglon.

La posición de un caracter en la VIO RAM no puede darse mediante una relación sencilla, ya que si esta posición es mayor a 0A4FH hay que restarle esta misma cantidad (para hacer la VIO circular). La forma en la que se calcularia esta posición es la siguiente:

**begin**

```
MEMOCURSOR = CURSOR + ROW + TOPFAG + OFFSET
```

```
if MEMOCURSOR > 0A4FH then
```

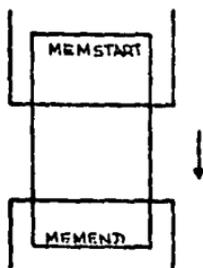
```
MEMOCURSOR = MEMOCURSOR - 0A4FH
```

**end**

Esta función es utilizada por la rutina DISPLA en el programa de control.

Cuando se empieza a escribir un texto, la pagina en la que se este se va llenando, cuando se llene toda la pagina (1320 caracteres) se habra llegado al ultimo caracter visualizado en la pantalla, en este momento se ejecutara un SCROLL, esta subrutina desplaza toda la pantalla un renglon hacia arriba, desapareciendo un renglon de la ventana por la parte de arriba, y apareciendo un nuevo renglon en blanco por

abajo, esto no significa que el renglon que desaparecio por la parte de arriba haya sido perdido, simplemente se salio del area cubierta por la ventana, pero aun permaneco en la VIO, cuando ya se han ejecutado scrolls hasta llenar completamente la VIO, el proximo scroll destruira el contenido del primer renglon de VIO, y empezara a escribir información sobre este primer renglon, el efecto total es como si hubieramos roto la ventana, dejando 23 renglones en el final de la VIO, en tanto que pusimos el renglon 24 sobre el primer renglon de la VIO.



A medida que se va escribiendo el cursor siempre ocupa la posición en la que se va a insertar el siguiente caracter, es decir, que al escribir un caracter automaticamente se

actualiza el cursor y avanza un caracter.

Existen cuatro tipos de desplazamiento que pueden efectuar el cursor cuando no se esta escribiendo, son:

1. Cursor a la derecha (CURIGH): cuando el cursor se desplaza a la derecha y llega al final del renglon, avanza al primer caracter del siguiente renglon. Para moverse hacia abajo ejecuta CURDWN, rutina que toma en cuenta la posibilidad de estar en el ultimo renglon de VIO.

2. Cursor a la izquierda (CURLEF): cuando el cursor se desplaza hacia la izquierda y llega al principio de renglon, avanza al ultimo caracter del renglon precedente. Para moverse hacia arriba ejecuta CURSUP.

3. Cursor hacia arriba (CURSUP): cuando el cursor se desplaza hacia arriba y llega al inicio de la ventana, mueve la ventana hacia arriba, junto con el, perdiendo renglones por la parte inferior, y manteniendose siempre en el primer renglon. Si llega al principio de VIO pasa el inicio de pagina (TOPFAG) al final de la VIO (0A4FH), empezando de nuevo el ciclo pero desde la parte de abajo de la imagen de video.

4. Cursor hacia abajo (CURDWN): cuando el cursor se desplaza hacia abajo y llega al final de la ventana desplaza a esta con el, perdiendo renglones por la parte de arriba. Si llega

al final de VIO pasa al primer renglon de esta.

El diagrama de flujo que describe todo lo anterior puede verse en la figura B.2.0/1.

En realidad el programa de control, escrito en ensamblador 8085 es autoexplicativo, debido a la forma estructurada (por subrutinas) en la que esta escrito, y ala gran cantidad de comentarios que se le añadieron.

El CRTC se inicializa para generar renglones no espaciados (24 por pantalla, con un VRTC = 2 renglones), 55 caracteres por renglon, HRTC = 10 caracteres, 10 lineas de video por renglon, underline en la linea 9, FAS transparentes, cursor de bloque y un DMA burst de 8 bytes, sin intervalo entre los DMA burst, hasta acabar de llenar el FIFO de 80 x 8 del 8275.

El DMAC es inicializado en modo Autoload con TC STOP deshabilitado, en el canal 3 de DMA se almacenan, los parametros de reinicialización del 8257 (dirección de inicio del DMA = 2000H, Terminal count = 0A4FH) que lo hacen empezar un nuevo ciclo al princpio de la VIO. Al principio de cada VRTC el CRTC genera una interrupción (RST7.5) que hace que el 8085 ejecute la rutina de servicio del DMAC, esta rutina hace la dirección de inicio del DMA igual a TOPPAG + OFFSET, es decir, al inicio de la pagina que se esta visualizando, y el TC se hace igual a la diferencia entre el final de VIO y la

posición en la que esta. Esto significa que:

1. En cada VRTC se actualiza la posición de inicio de la ventana (los scrolls, etc. se ejecutan solo durante el VRTC).
2. Cuando el cursor llega al final de la VIO se alcanza el TC, por lo que el 8257 ejecuta un Autoload y regresa el cursor al inicio de VIO (esto efectivamente implementa el VIO circular, sin que deba haber ningún tipo de intervención de parte del software de la terminal).

La pagina de video visualizada empieza en la dirección de inicio del DMA y sigue hasta que se visualizan 24 renglones (al final de cada HRTC el CRTIC genera un DRQ para el DMAC y lee 55 caracteres en 6 bursts de 8 bytes y un burst de 7 bytes, el DMAC se encarga de incrementar automaticamente la dirección de la cual se lee el caracter). Si antes de haber visualizado los 24 renglones se llega al final de la VIO se habra alcanzado el TC del DMAC, y este automaticamente reinicializara su canal 2 con los datos del canal 3 (dirección de inicio del DMA = dirección del primer caracter en la VIO, longitud del TC = 0A4FH = longitud del VIO). Cuando se visualicen los 24 renglones se generara un VRTC y el CH2 del DMAC volvera a ser inicializado en TOPPAG, con un TC igual a la distancia entre TOPPAG y el final de VIO.

Para más detalles acerca de la operación del DMAC es

interesante leer las hojas de especificaciones de Intel sobre el 8257, las cuales pueden encontrarse en el Intel Component Data Catalog.

### B.2.2- Tiempo utilizado para refrescar el CRTC.

El CRTC solicita un burst de DMA al principio de cada renglon, este burst debe transferir 55 bytes de VIO al CRTC, el DMAC requiere 4 ciclos de reloj para transferir un byte, y cada ciclo tiene una duración de  $1 / 3.072 \text{ M} = 325.52 \text{ nS}$ , es decir que 1 byte se transfiere en  $4 * 325.52 \text{ n} = 1.3021 \text{ uS}$ , y 55 bytes en  $71.615 \text{ uS}$ , una línea tiene una duración de  $63.5 \text{ uS}$ , y hay 10 líneas en un renglon, por lo tanto 1 renglon dura  $635 \text{ uS}$ , de los cuales  $635 - 71.615 = 563.39 \text{ uS}$  están disponibles para el 8085. En una pantalla tenemos 24 renglones y 2 de VRTC, de los dos renglones de VRTC el primero ( $635 \text{ uS}$ ) está disponible completamente, puesto que no se lee el VIO en esta línea, durante el segundo renglon de VRTC se lee el primer renglon activo de la pantalla y solo tenemos disponibles  $563.39 \text{ uS}$ , durante el último renglon activo de la pantalla (renglon 23) tampoco se lee la VIO, ya que en el siguiente renglon nada se visualizara (es el primer renglon de VRTC), por lo tanto el tiempo durante el cual el 8085 puede acceder a SYSBUS es:

$$T(8085) = 563.39 * 24 + 2 * 635.00$$

$$T(8085) = 14.791 \text{ mS por pantalla}$$

Puesto que el tiempo que dura una pantalla (incluyendo el VRTC) es de  $1 / 60 = 16.667 \text{ mS}$ , el 8085 tiene acceso al SYSBUS durante el 88.748% de la duración de una pantalla (es

decir, durante 887.48 mS por segundo).

Esto nos da una idea de lo eficiente que resulta el sistema de acceso directo a memoria, así como los dos FIFOs del 8257. Mientras se esta visualizando el contenido de un FIFO (en nuestro caso 55 caracteres) el otro se llena rapidamente con el contenido del siguiente renglon, si no existieran estos FIFOs deberian hacerse 10 accesos a VIO por caracter (uno por cada linea) dando un total de 550 accesos a VIO lo cual tomaria (aun con DMA) 716 uS por renglon, que es más de lo que dura el renglon.

### B.2.3 - Prioridad de las interrupciones.

Existen cuatro interrupciones cuya prioridad esta de acuerdo a la velocidad del dispositivo que las genera, siendo el más rapido el que tiene mayor prioridad. Estas interrupciones son:

Prioridad	Nombre	Linea
0	PFail	TRAP
1	VIrq	RST7.5
2	SIOIRQ	RST6.5
3	KBIRQ	RST5.5

Siendo el 0 la prioridad maxima.

1. PFAIL (Power Fail). Esta interrupción es generada por la fuente de potencia 3 mS antes de una falla en el suministro de energia.

2. VIrq (Vertical Interrupt Request). Generado por el CRTc al principio de cada VRtc.

3. SIOIRQ (SIO Interrupt Request). Generado por el SIO al recibir un dato desde el anfitrión.

4. KBIRQ (Key Board Interrupt Request). Generado por el KBC

cada vez que se detecta una secuencia valida de teclas  
oprimidas en el teclado.

### B.3 - PROGRAMACIÓN DEL DATACOMM.

El puerto I/O es programado para cumplir con las especificaciones que se dan en la sección sobre el DATACOMM: 9600 bps, TxEN = RxEN = 1, RTS/ = DTR/ = 0, ASCII 7 bits, 2 bits de final, 1 de arranque, paridad par.

Cada vez que se recibe un dato procedente del computador anfitrión por algun puerto, se genera un RST6.5 hacia el 8085, el cual ejecuta la rutina de servicio de entrada de datos por el puerto SID (Serial I/O). El byte recibido se almacena en un buffer de 10 caracteres hasta que el 8085 tiene tiempo de atender al RST6.5, si el buffer se llena se hace el bit de XON/XOFF del estatus igual XOFF, y se transmite un XOFF hacia el anfitrión. Si llega un nuevo caracter se genera una señal de error (ERROR) en el Status. Cuando el 8085 quiere transmitir hacia el computador anfitrión solo debe verificar que XONE este activo.

El baud rate generator (CTR0 y CTR1; 8253) es inicializado para dividir SYSCLK entre 10 y generar TxCLK y RxCLK (el 8251 divide la señal entre 16, a su vez).

El generador de audio del 8253 es programado para dividir SYSCLK entre 0E00H, con lo cual se obtiene una frecuencia de audio de aproximadamente 800 Hz.

#### 8.4 - PROGRAMACIÓN DEL KBC.

El programa de control del KBC se simplifica debido a que el 8741A efectúa la mayor parte de las verificaciones, etc. requeridas, librando al 8085 de esta tarea.

Cuando el 8741A ha detectado una secuencia valida procedente del teclado lo primero que hace es generar un codigo (KEYC) que represente a la tecla oprimida (este codigo no es el codigo ASCII) y otro codigo (KEYA) que representa los codigos especiales (atributos) que fueron detectados junto con esa tecla. KEYC es depositado en el DBB OUT del 8741A y KEYA en el Status. Tras esto el 8741 genera un KBIRQ (mediante el RST5.5) y espera a que sean leidos estos dos codigos. Se supone improbable que se oprima otra tecla antes de que el 8085 lea el KBC, ya que el 8085 ejecutara la rutina de servicio tan pronto como haya terminado la rutina que este ejecutando (el 8085 nunca dejara de ejecutar una rutina para iniciar otra, ya que los interrupts estan deshabilitados por software por todas las rutinas, y solo podran ser ejecutados cuando el 8085 haya terminado una rutina, siendo habilitadas las interrupciones por esta, que colocara al 8085 en estado de WAIT. Esto se hace con la finalidad de que en posteriores versiones del VTKBDS el 8085 pueda ejecutar rutinas del usuario, que sean interruptibles, durante los periodos en los que ahora el 8085 esta en WAIT).

Una vez que el 8085 ha leído KEYC y KEYA desde el 8741A, se procede a interpretar el tipo de carácter que se ha leído. El código puede ser de 5 tipos:

1. Código normal. Caso en el cual el 8085 busca su equivalente ASCII en una tabla y lo manda al anfitrión, de estar en modo remoto. De estar en modo local se genera un eco al CRT.

2. Código de control. Caso en el cual el 8085 busca su equivalente ASCII en una tabla y lo manda al anfitrión, de estar en modo remoto. De estar en modo local se genera un eco al CRT.

3,4,5. Iniciador de secuencia de ESC, carácter válido de secuencia de ESC o terminador de secuencia de ESC. Caso en el cual el 8085 sigue leyendo caracteres hasta encontrar el final de secuencia de ESC, estos caracteres van siendo almacenados en un buffer especial para secuencias de escape. En este momento pueden ocurrir tres cosas:

- i. De estar en Local se ejecuta la secuencia de ESC.
- ii. De estar en modo remoto y ser la secuencia ESCII se entra en modo local.
- iii. De estar en modo remoto y no ser la secuencia ESCII se manda esta al anfitrión.

Una secuencia de ESC valida es la que empieza con un iniciador de secuencia de ESC, prosigue con caracteres validos de secuencia de ESC y termina con un terminador valido de secuencia de ESC.

1. Iniciador de secuencia de ESC: ESCI
2. Caracter valido de secuencia de ESC: cualquier número o una secuencia de estos separados por comas o puntos y coma.
3. Terminador valido de secuencia de ESC: cualquier letra mayuscula o minuscula.

A continuación se incluyen los diagramas de flujo que describen el funcionamiento de la terminal a nivel software, así como el programa de control de esta en ensamblador 8085.

## AA-2. Principales abreviaturas y terminos.

1. VTKD. Video Terminal and Keyboard: Termino usado a lo largo de esta tesis para referirnos al conjunto de la terminal; monitor, circuiteria y teclado.
2. VIO. Video Image 0: Area de RAM en la cual se guarda el contenido de la pantalla de graficas 0.
3. VI1. Video Image 1: Area de RAM en la cual se guarda el contenido de la pantalla de graficas 1.
4. CRT. Catode Ray Tube: Tubo de rayos catodicos. Monitor de video.
5. VTKDOS. Video Terminal and Keyboard Operating system: Conjunto de instrucciones ejecutables por el 8085AH o por el 8741A que controlan la terminal.
6. VTKDP. Video Terminal and Keyboard Processor: Procesador central de la terminal.
7. KB. Keyboard: Teclado.
8. SYNC: Señal de sincronia.

9. HSYNC: Sincronia horizontal.
10. VSYNC: Sincronia vertical.
11. HRTC. Horizontal Retrace: Borrado horizontal.
12. VRTC. Vertical Retrace: Borrado vertical.
13. Terminal de video: Dispositivo utilizado para comunicar a un ser humano con un computador, consta de un teclado, mediante el cual el humano introduce información al computador, y de un CRT mediante el cual el humano puede recibir información visual desde el computador.
14. CRTC. Cathodic Ray Tube Controller: Circuito que genera las señales de tiempo necesarias para interconectar un computador con un monitor de video.
15. VG. Video Generator: Generador de Video.
16. Software: La parte de un computador formada por las instrucciones y programas que lo controlan.
17. Hardware: La parte mecánica y electrónica de un computador.
18. Firmware: Software permanentemente grabado en la electrónica de un computador.

19. 8085AH: Microprocesador de 8 bits fabricado por Intel y muy popular como controlador.

20. 8741N: Microcomputadora de 8 bits fabricada por Intel.

21. RAM. Random Access Memory: Memoria en la que se pueden guardar y de la que se pueden recuperar datos en forma no secuencial.

22. ROM. Read Only Memory: Memoria preprogramada, no se puede escribir en ella más que una vez.

23. VNUX. Video Multiplexer: Multiplexor de Video.

24. ASCII. American Standard Code for Information Interchange. Código muy popular entre los fabricantes de computadoras, usado para representar información alfanumérica.

25. SYSBUS. System bus: Bus al cual están conectados todos los componentes de la terminal.

26. Murphy, Leyes de: Leyes fundamentales que explican porque el mundo no parece funcionar como debiera, sino exactamente al revés de como debiera. Fueron enunciadas hace tiempo por Murphy. Entre las más conocidas está la cuarta ley:

Cuando no necesitamos un objeto cualquiera (A), totalmente inofensivo e inutil, lo encontramos con una media de 28.437 apariciones por minuto, interponiéndose en nuestro paso y estorbándonos lo más posible. Si el objeto A se vuelve repentinamente necesario y util solo lo hallaremos tras una búsqueda de unas 8.7 horas, aunque acabe de interponerse en nuestro camino hace apenas 3 minutos.

27. Bus: Término con el cual nos referimos a un canal de comunicaciones (compuesto por más de una señal) entre varios elementos de una computadora. Medio de transporte.

Esta lista no es extensiva, solo se dan algunos terminos de importancia, si el lector encuentra algun otro termino que no comprende le recomendamos buscarlo en algun manual de referencia sobre terminos especiales de electrónica y computación. Si aun asi sus dudas no quedan resueltas, le pedire entrar en contacto con el autor.

**AA-3.** Resumen de las especificaciones de la terminal.

1. Controlador: 8085AH, 3.072 MHz.

2. Controlador del teclado: 8741A, 204.8 KHz.

3. RAM: 32 K x 8, estatica.

4. ROM: 8 K x 8 (EPROM).

5. Sistema operativo: VTKBDS V0.0 - 01

6. Requerimientos de potencia:

+5 V  $\pm$  5% @ 5 A. 25 W

+12 V  $\pm$  10% @ 500 mA. 6 W

-12 V  $\pm$  10% @ 500 mA. 6 W

Potencia total: 37 W

7. Video:

i. Modo alfabetico (ALPHA):

\* 24 renglones x 55 caracteres

\* 1,320 caracteres por pagina, dos paginas en memoria

(2,640 caracteres)

- \* Blanco y negro.
- \* Caracteres de 7 x 5, en un campo de 10 x 7.

ii. Modo grafico 1 (GR1):

- \* Matriz de 48 x 55 bloques de 8 colores.
- \* 2,640 bloques por pagina.
- \* Dos paginas en memoria (5,280 bloques)
- \* Colores: blanco, negro, rojo, verde, azul, cyan, magenta.

iii. Modo grafico 0 (GR0):

- \* Matriz de 240 x 385 pixeles.
- \* 92,400 pixeles por pagina.
- \* Dos paginas en memoria (184,000 pixeles)
- \* Colores: negro, violeta, purpura, cyan y magenta.

Señales de video conforme al estandar RS170:

B. Comunicaciones:

- \* Puerto EIA RS232C, velocidad maxima de transmisión:  
19,600 bps, programable por software.
- \* Lazo de corriente de 20 mA.
- \* Protocolo de comunicación de datos: ASCII, 7 bits por  
caracter, 1 bit de paridad par, 1 bit de arranque

(space) y dos bits de final (mark).

A-4. Código ASCII.

DEC.	HEX.	
0	0	Null
1	1	SOH
2	2	STX
3	3	ETX CTRL/C
4	4	EOT
5	5	END
6	6	ACK
7	7	BELL
8	8	Backspace
9	9	HT
10	A	LF
11	B	VT
12	C	FF
13	D	CR
14	E	SO
15	F	SI CTRL/D
16	10	DLE
17	11	DC1 CTRL/D XON
18	12	DC2
19	13	DC3 CTRL/S XOFF
20	14	DC4
21	15	NAK CTRL/U

22	16	SYN	
23	17	ETB	
24	18	CAN	
25	19	EM	
26	1A	SUB	CTRL/Z
27	1B	ESC	
28	1C	FS	
29	1D	GS	
30	1E	RS	
31	1F	US	
32	20	SP	Space
33	21	!	
34	22	"	
35	23	#	
36	24	\$	
37	25	%	
38	26	&	
39	27	'	Apostrofe
40	28	(	
41	29	)	
42	2A	*	
43	2B	+	
44	2C	,	Coma
45	2D	-	
46	2E	.	Punto
47	2F	/	
48	30	0	
49	31	1	

50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M

78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	'
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i

106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~
127	7F	DELETE

## BIBLIOGRAFIA

### B1- Referencias de electrónica:

1. Millman, Jacob; Halkias, Christos. Integrated Electronics: Analog and Digital Circuits and Systems. International Student Edition. McGraw-Hill. Tokyo 1983.
2. Millman, Jacob. Microelectronics: Digital and Analog Circuits and Systems. International Student Edition. McGraw-Hill. Tokyo 1984.
3. Peatman, John B. . Digital Hardware Design. International Student Edition. McGraw-Hill. Tokyo 1983.
4. Peatman, John B. . The Design of Digital Systems. International Student Edition. McGraw-Hill. Tokyo 1981.
5. Gunturia, Stephen D.; Wedlock, Bruce D. . Electronic Circuits and Applications. John Wiley & Sons. New York 1975.
6. Schilling, Donald L., Belove, Charles. Electronic Circuits, Discrete and Integrated. International Student Edition. McGraw-Hill. Tokyo 1979.

7. Kuo, Franklin F. . Network Analysis and Synthesis. Wiley International Edition. John Wiley & Sons. New York 1966.

8. Schwartz, Mischa. Information Transmission, Modulation, and Noise. International Student Edition. McGraw-Hill. Tokyo 1981.

9. Texas Instruments. Transistor Circuit Design. International Student Edition. McGraw-Hill. Tokyo 1981.

10. Daryanani, Gobind. Principles of Active Network Synthesis and Design. John Wiley & Sons. New York 1976.

**B2-** Referencias para los principios de video:

1. Grob, Bernard. Basic Television and Video Systems. International Student Edition. McGraw-Hill. Singapore 1964.

2. Kennedy, George. Electronic Communication Systems. International Student Edition. McGraw-Hill. Tokyo 1977.

**B3-** Referencias para la sección del microcomputador:

1. Motorola Semiconductors (editor). Understanding Microprocessors. Unwin Brothers Ltd. Old Woking 1980.
2. Wakerly, John F.. Microcomputer Architecture and Programming. John Wiley & Sons. New York 1981.
3. Intel. MCS-80/85 IM, Family Users Manual. Intel co. Santa Clara 1979.
4. Intel. The 8006 Family User's Manual. Intel co. Santa Clara 1979.
5. Intel. iAPX 88 Book. Reston Publishing Company. Reston 1983.
6. Intel. Memory Design Handbook. Intel co. Santa Clara 1981.

**B4-** Referencias para la sección de comunicaciones:

1. Kane, Gerry; Harper, Steve; Ushijima, David. The HP-IL System: An Introductory Guide to the Hewlett-Packard Interface Loop. Osborne/McGraw-Hill. Berkeley 1982.

2. National Semiconductor. Summary of Electrical Characteristics of Some Well Known Digital Interface Standards. National Semiconductor. Santa Clara 1978.

**B5-** Referencias para la sección de video:

1. Intel. Peripheral Design Handbook. Santa Clara 1979.
2. Brilliot, Al. A Low Component Count Video Data Terminal Using the DP8350 CRT Controller and the IN8000 CPU. Application Note 199. National Semiconductor. Santa Clara 1978.
3. Morales, Arnold J. A Low cost Terminal Using the MC6801. Application Note 798. Motorola Semiconductor. Austin 1980.

**B6-** Manuales:

1. National Semiconductor. Interface Databook 1978.
2. Texas Instruments. The MOS Memory Data Book: 1980.
3. Texas Instruments. The Optoelectronics Data Book 1978.
4. Texas Instruments. The Power Data Book for Design Engineers 1980.

5. Texas Instruments. The TTL Data Book for Design Engineers 1980.
6. Texas Instruments. 1981 Supplement to The TTL Data Book.
7. Motorola. Microprocessors Data Manual 1980.
8. Motorola. Linear Circuits 1978.
9. Motorola. Linear Interface Circuits 1980.
10. Motorola. Memory Data Manual 1980.
11. National Semiconductor. CMOS Databook 1981.
12. National Semiconductor. Discrete Databook 1978.
13. Intel. Component Data Catalog 1981.