

20.
2ej



UNIVERSIDAD LA SALLE

ESCUELA DE INGENIERIA
INCORPORADA A LA U. N. A. M.

**DISEÑO DE UN SISTEMA GRAFICO
APLICADO A UN SISTEMA MULTIMAESTRO**

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A:

MARIA DE JESUS MIRELES LOPEZ

TESIS CON
FALLA DE ORIGEN

México, D. F.

1988.



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 esta 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

INTRODUCCION.....	9
CAPITULO I	
ANTECEDENTES DE HARDWARE GRAFICO.....	I.1
I.1 GENERALIDADES DE DISPOSITIVOS DE VISUALIZACION.....	I.3
I.2 MEMORIA VIDEO.....	I.4
I.3 CARACTERISTICAS DEL SISTEMA DISEÑADO.....	I.8
CAPITULO II	
II ARQUITECTURA SISTEMAS GRAFICOS.	
II.1 INTRODUCCION.....	II-1
II.2 UTILIZACION DE PROCESADORES GENERALES.....	II-2
II.2.1 VENTAJAS EN EL USO DE PROCESADORES GRALES.....	II-3
II.2.2 DESVENTAJAS EN EL USO DE PROCESADORES GRALES.....	II-3
II.3 UTILIZACION DE PROCESADORES ESPECIALIZADOS.....	II-4
II.3.1 ORGANIZACION GENERAL.....	II-4
II.3.2 CLASIFICACION DE PROCESADORES ESPECIALIZADOS.....	II-5
II.3.2.1 CONTROLADOR DE VIDEO.....	II-5
II.3.2.2 GENERACION DE FIGURAS GEOMETRICAS.....	II-5
II.3.2.3 EJEMPLOS DE PROCESADORES EN EL MERCADO.....	II-6

INDICE

II.3.3 VENTAJAS.....	II-6
II.3.4 DESVENTAJAS.....	II-7
II.4 UTILIZACION DE TABLAS DE COLORES.....	II-7
II.4.1 VENTAJAS.....	II-8
II.4.2 DESVENTAJAS.....	II-8

CAPITULO III

DISEÑO A BLOQUES DEL SISTEMA MAESTRO.

III ARQUITECTURA GENERAL.....	III-1
III.1 PROCESADOR GENERAL.....	III-1
III.1.1 UNIDAD DE INTERFAZ AL BUS.....	III-2
III.1.2 UNIDAD DE EJECUCION.....	III-2
III.1.2.1 COLA DE INSTRUCCIONES.....	III-4
III.1.3 ORGANIZACION DE LAS MEMORIAS.....	III-6
III.1.4 OPERACION AL BUS.....	III-7
III.1.5 DIRECCIONAMIENTO ENTRADA-SALIDA.....	III-8
III.1.6 ESTRUCTURA DE INTERRUPCION.....	III-8
III.1.7 DISEÑO A BLOQUES DEL PROCESADOR GENERAL.....	III-12
III.1.7.1 BLOQUE DE SINCRONIA.....	III-12
III.1.7.2 BLOQUE DE PROCESAMIENTO.....	III-13
III.1.7.2.1 LINEAS DE DATOS Y DIRECCIONES.....	III-14
III.1.7.2.2 LINEAS DE CONTROL DE ESTADOS.....	III-14
III.1.7.2.3 LINEAS DE POTENCIA Y SINCRONIA.....	III-15
III.1.7.3 DEMULTIPLEXORES.....	III-15
III.1.7.3.1 SISTEMA DE BIESTABLES TIPO D ("LATCHES").....	III-16
III.1.7.3.2 SISTEMA DE TRANSCPTORES.....	III-16
III.1.7.3.3 COPROCESADOR.....	III-17

INDICE

III.1.7.4 BLOQUE DEL CONTROLADOR.....	III-20
III.1.7.4.1 BLOQUE DE MEMORIA.....	III-21
III.1.7.4.2 MEMORIA PERMANENTE.....	III-21
III.1.7.4.3 MEMORIA VOLATIL.....	III-23
III.1.7.5 BLOQUE ENTRADA SALIDA.....	III-24
III.1.7.5.1 RECEPTOR TRANSMISOR UNIVERSAL MULTIFUNCIONES ASINCRONO.....	III-24
III.1.7.5.2 PROGRAMADOR DE INTERRUPCIONES (PIC).....	III-27

CAPITULO IV

CIRCUITERIA DE INTERFAZ ENTRE PROCESADOR, MEMORIA Y CONTROLADOR.

IV.1 CONTROLADOR GRAFICO (VSC) E INTERFAZ.....	IV-1
IV.1.1 DESCRIPCION GENERAL.....	IV-1
IV.1.1.1 INTERFAZ AL PROCESADOR.....	IV-2
IV.1.1.2 INTERFAZ A MEMORIA DE VIDEO.....	IV-5
IV.1.1.3 ARBITRO.....	IV-6
IV.1.2 DECODIFICACION GENERAL.....	IV-10
IV.1.2.1 DECODIFICACION DE LA MEMORIA DEL SISTEMA.....	IV-11
IV.1.2.2 DECODIFICACION DE FUNCIONES DEL VSC.....	IV-14
IV.1.2.3 DECODIFICACION DE PIXEL.....	IV-19
IV.2 MEMORIAS DE VIDEO.....	IV-21
IV.2.1 ACCESO ALEATORIO.....	IV-23
IV.2.2 ACCESO SECUENCIAL.....	IV-24
IV.3 PALETA DE COLORES.....	IV-25
IV.3.1 ARQUITECTURA INTERNA DE LA PALETA.....	IV-26
IV.3.2 OPERACION DE INTERFAZ MEMORIAS DE VIDEO-PALETA....	IV-27
IV 3.3 INTERFAZ AL MONITOR.....	IV-29

INDICE

CAPITULO V

V.1 PRUEBAS AL "HARDWARE".....	V-1
V.1.1 INICIALIZACION DEL CONTROLADOR	V-2
V.1.1.1 REGISTROS DE COMUNICACION CON EL PROCESADOR GENERAL.....	V-2
V.1.1.2 REGISTROS DE CONTROL DEL TRC	V-4
V.1.1.3 REGISTROS DE CONTROL DE ACCESO A MEMORIA.....	V-5
V.2 ALGORITMOS DE PROGRAMACION.....	V-7
V.2.1 INICIALIZACION DEL CONTROLADOR.....	V-7
V.3 APLICACIONES SIMPLES.....	V-10
V.3.1 GENERACION DE CARACTERES.....	V-10
V.3.2 TRAZADO DE LA PALETA DE COLORES.....	V-12
V.3.3 GENERACION DE LINEA RECTA EN EL PRIMER OCTANTE.....	V-14
CONCLUSIONES.....	107
GLOSARIO DE TERMINOS.....	109
REFERENCIAS.....	114
BIBLIOGRAFIA.....	117
APENDICE 1.....	120
APENDICE 2.....	127

I N T R O D U C C I O N

INTRODUCCION

En el Instituto de Investigaciones Eléctricas, dentro de su departamento de Electrónica se trabaja en sistemas de automatización de redes de distribución eléctrica en el área de Control Supervisorio el cual es uno de los sistemas más empleados en telemetría y telecontrol. El Control Supervisorio en distribución tiene las funciones de operar la red en forma confiable y segura, mantener la continuidad del servicio y optimizar la eficiencia del sistema. Este tipo de sistema tiene dos funciones básicas:

- 1.- Minimizar el tiempo de salida por falla.
- 2.- Hacer la distribución de energía eléctrica lo más económica posible.

Para lograr un desarrollo en el sistema de Control Supervisorio es necesario crear un sistema que ayude a la comunicación hombre-máquina por medio de un sistema gráfico, con el fin de utilizar las imágenes para el facilitar la operación del sistema, así como la comprensión de la información manipulada en la red por parte del operador, lo que repercutirá en la toma rápida de decisiones referentes a la red lo más fácil y acertadamente posible.

INTRODUCCION

El objetivo de este trabajo es el de diseñar y construir un Sistema Gráfico aplicado a un Sistema Multimaestro, es decir asociado a varios procesadores en paralelo, con el fin de desplegar información de campo (unifilares) para lograr la automatización de las redes de distribución del proyecto MORELIA.

Dentro del capítulo I de ésta tesis se tratará de manera simple lo que se refiere a generalidades de hardware Gráfico con el fin de dar una breve introducción a lo que es un sistema gráfico.

En el capítulo II se tratará la arquitectura de un sistema gráfico con el empleo de procesadores generales y procesadores especializados.

En el capítulo III se adaptará a partir de una infraestructura ya existente para el Sistema Multimaestro desarrollado en el Instituto de Investigaciones Eléctricas, la implementación para poder trabajar como tarjeta modo maestra gráfica.

En el capítulo IV se diseñará la parte gráfica del sistema en base a un controlador de video TM41061 de Texas Instruments.

En el capítulo V se tratarán la implementación y pruebas tanto de acceso general como de acceso a video, memoria de video, control de espacio total de video

INTRODUCCION

direccionable y control de acceso a la tabla de colores en forma general.

Finalmente exponemos nuestras conclusiones y posibilidades del presente trabajo para desarrollos futuros con el fin del hacer variaciones al trabajo presentado.

CAPITULO I

CAPITULO I

ANTECEDENTES DE HARDWARE GRAFICO.

Un sistema gráfico es un sistema compuesto por un sistema de procesamiento o computadora enlazado a un dispositivo para visualizar información. La computadora o microcomputador se comporta como el procesador central del sistema gráfico. Este sistema puede tener un micoprocesador propio, lo que haría una tarjeta autónoma o solamente constar con la parte gráfica y compartir el procesamiento con otros sistemas lo que sería un sistema gráfico dependiente.

Internamente la computadora maneja sus datos en forma binaria. Esto es, todos los números, letras, símbolos e instrucciones son representados como una secuencia de dos dígitos, unos y ceros. Una simple letra del alfabeto puede ser representada por ejemplo con 11010001, de esta misma

CAPITULO I

manera toda la información que es procesada se representa de la misma manera. Esto es llamado lenguaje máquina.

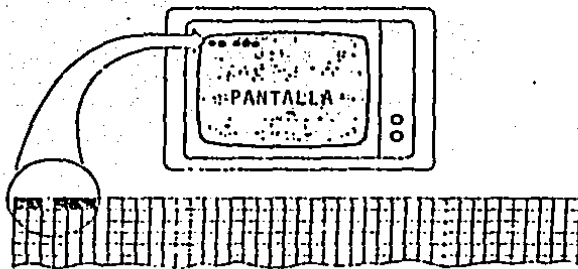
En el lenguaje máquina (binario), los ceros y unos son simplemente representados por circuitos en la computadora que están prendidos o apagados. Toda la información digital, es almacenada en una memoria para de ahí ser tomada en el momento en que se requiera, únicamente accediendo la dirección de la memoria donde la información fué guardada. Un espacio de memoria especial es necesaria para almacenar toda la información especial para video.

Las tarjetas gráficas mapean o ubican uno a uno los puntos de información, llamados "pixels", de una memoria video para ser entregada al monitor de visualización. (Ver figura 1.1)

La resolución de un sistema gráfico es la capacidad de direccionar o desplegar un número de "pixels" en la memoria video tanto en escritura como en lectura.

-Un sistema gráfico tiene baja resolución si su mapeo (1) en memoria video es igual o menor a 400X200 puntos o

(**)NOTA: Las palabras que se encuentran entre "comillas" están definidas en el glosario de términos al final de éste trabajo.



Localidades de memoria RAM

FIGURA 1.1 RELACION ENTRE MEMORIA Y PANTALLA.

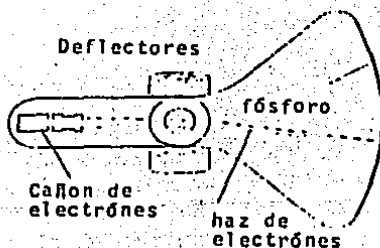


FIGURA 1.2 CONSTRUCCION BASICA DE UN CRT.

CAPITULO I

"pixels", de media resolución si es igual o inferior a 640x480 puntos; Si su mapeo es mayor a las anteriores se considera un sistema de alta resolución.

I.1 GENERALIDADES DE LOS DISPOSITIVOS DE VISUALIZACIÓN. (2) (3)

La Computación Gráfica demanda dispositivos de visualización que puedan cambiar rápidamente las imágenes. Los "displays" permiten a una imagen ser visualizada y modificada en tiempos cortos. Todas las pantallas usadas en los sistemas gráficos son tubos de rayos catódicos (TRC) (Ver figura 1.2). Existen varios tipos de TRC que son elegidos por el usuario dependiendo de la aplicación que se requiera. Los TRC principales son los de Refrescado y los de Rastreo.

Los TRC de Refrescado, también llamados de haz dirigido, de refrescado vectorial o de barrido aleatorio, usan el haz para trazar las líneas de la imagen, pintando y repintando cada línea de punto inicial a punto final.

Entre sus ventajas está la de poder animar la imagen, esto es rotarla, modificarla o trasladada en la pantalla sin necesidad de redibujar la imagen como en otros "displays".(Ver fig. 1.3)

Estos presentan como principal limitación que al trazar figuras complejas, la imagen parece parpadear debido al tiempo relativamente largo requerido por el sistema para

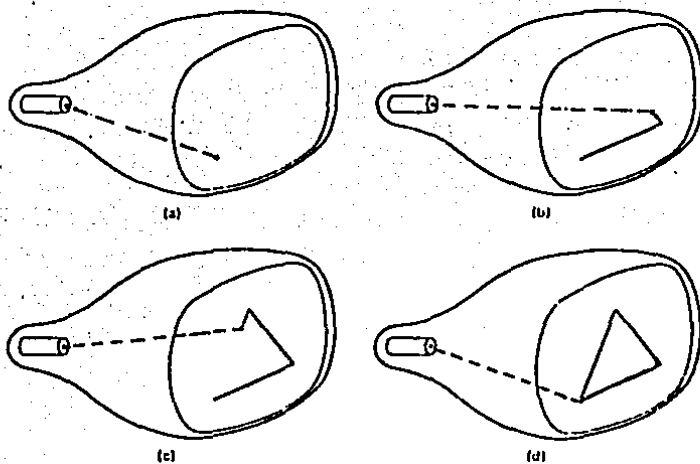


FIGURA 1.3 FIGURA TRAZADA POR BARRIDO ALEATORIO.

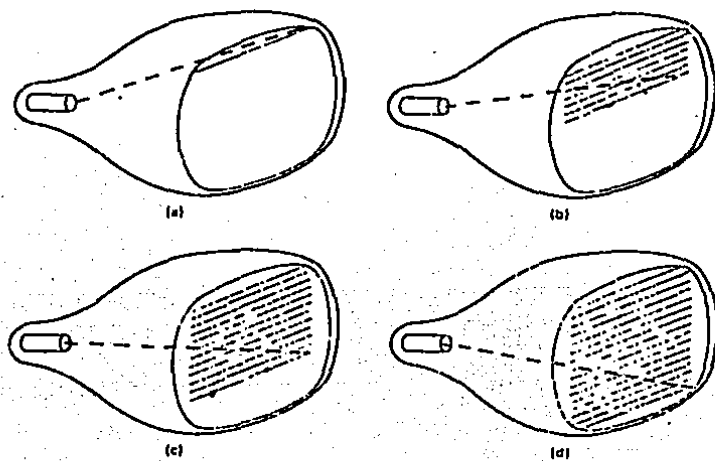


FIGURA 1.4 FIGURA TRAZADA POR BARRIDO DEFINIDO.

CAPITULO I

retrazar todas las líneas, otra desventaja es la gran cantidad de accesos aleatorios a memoria requeridos para almacenar la imagen mientras está siendo desplegada.

La pantalla de Rastreo Definido (elegida en nuestro caso), usa la forma tradicional que emplean las pantallas de TV caseras para trazar las imágenes. (Ver figura 1.4)

Crean la imagen a partir de una matriz de puntos llamados "pixels" almacenados en la memoria de la computadora especial para el video. El haz de electrones barre la pantalla de la parte superior a la inferior, iluminando cada "pixel" modulado en voltaje.

Debido a que el "display" es reescrito constantemente, las imágenes pueden ser animadas y manipuladas en tiempo real, además de ser más brillantes y poder desplegar colores. La principal limitación es la pobre calidad en la línea, debido a la naturaleza discreta de la matriz de puntos, hace que una línea inclinada se vea escalonada.

I.2.MEMORIA VIDEO. (4)

Hablar de pixels es hablar de información de visualización almacenada en memoria video. Existen dos formas de visualizar ésta información: Con niveles de gris y con ayuda de colores. Desde el punto de vista de manipulación y de almacenamiento de la información, estas

CAPITULO I

dos formas son equivalentes, solamente existe la diferencia al ser visualizadas en el monitor.

Si consideramos planos de un bit por pixel, tenemos dos posibilidades de visualización: cero o uno, blanco o negro si se trata de un "display" monocromático, o dos colores a escoger si se trata de un "display" de colores. Podemos decir que el hecho de aumentar un plano de memoria de video, equivale a doblar la posibilidad del número de colores. (Vease plano de memoria como un bit adicional).

Como conclusión a lo anterior podemos decir que si aumentamos el número de colores disponibles así como la resolución, aumentamos el tamaño físico de la memoria video necesaria para el manejo de nuestra información.

ARREGLOS DE MEMORIA VIDEO. (5)

La memoria de video antes mencionada, se encuentra configurada dentro de los sistemas gráficos en dos arreglos básicamente:

- a) Arreglo Planar.
- b) Arreglo "Packed Pixel".

En el arreglo Planar, la memoria de video se encuentra dividida en áreas iguales de memoria llamadas Planos. Por lo general se utilizan 4 planos, tres son para los colores ROJO, VERDE y AZUL y el otro se utiliza para atributos.

CAPITULO I

(Ver figura 1.5).

Para obtener la información del "pixel" en el arreglo Planar, es necesario acceder los cuatro bancos de memoria, ya que la información del puntito o "pixel" se encuentra localizada en 4 direcciones diferentes. Este tipo de arreglo es muy útil, cuando se requieren hacer llenado de áreas en forma rápida.

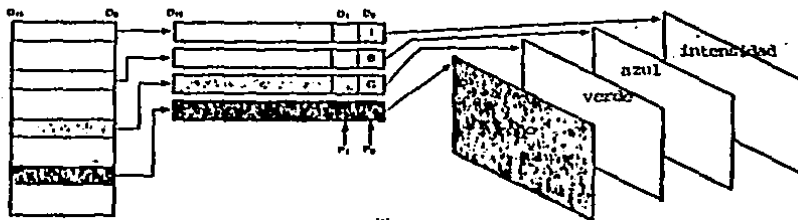
El arreglo "Packed Pixel" tiene la característica de que toda la información del "pixel, se encuentra junta, esto significa que los bits que forman el "pixel" se encuentran localizados en áreas contiguas de memoria. La principal ventaja es que al hacer un solo acceso se puede tener toda la información necesaria para desplegar el "pixel". Este arreglo facilita el manejo de pixeles individuales. (Ver figura 1.6).

El arreglo "Packed Pixel" fué el escogido para el desarrollo de nuestro sistema gráfico.

TIPOS DE MEMORIA EMPLEADAS MAS COMUNMENTE. (6)

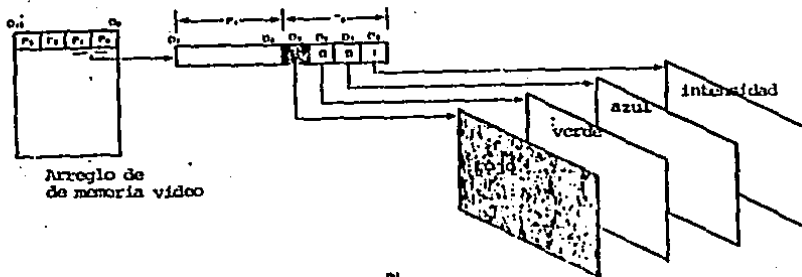
Los sistemas gráficos usan dos tipos de memoria principalmente:

a) Memorias Dinámicas.(DRAMS)



Arreglo de memoria video

FIGURA I.5 ARREGLO DE MEMORIA PLANAR.



Arreglo de memoria video

FIGURA I.6 ARREGLO DE MEMORIA "PACKET PIXEL".

CAPITULO I

b) Memorias Video.(VRAMS) (Memorias Dinámicas de Doble Puerto.)

Ambas memorias dinámicas son escogidas por:

- La facilidad con que son actualizadas, ya que basta con un pulso para refrescar la información.

- Por los cortos tiempos de acceso que aumentan la velocidad del sistema.

En el pasado, todos los sistemas gráficos de rastreo definido tenían el problema de de "efecto de cuello de botella" debido a la restricción de flujo de datos causados por el puerto simple de las memorias dinámicas convencionales. Típicamente el CPU podía acceder las memorias el 50% de las veces debido al problema de contención de datos provocados por accesos simultáneos a memoria para actualizar memoria, como para actualizar la información a desplegar.

Las memorias video VRAMS, son preferidas dentro de los sistemas gráficos, por la ventaja de su doble puerto, lo que elimina los cuellos de botella al facilitar la entrada y salida de información, con lo que el CPU tiene acceso ilimitado (94% mínimo) a memoria video para actualizarla.

CAPITULO I

Las memorias VRAMS fueron empleadas en el desarrollo de nuestra tarjeta. Esto se tratará con mas detalle dentro del capítulo IV.

I.3 CARACTERISTICAS DEL SISTEMA DISEÑADO.

Nuestro SISTEMA GRAFICO presenta las siguientes características:

Microprocesador 8086 INTEL.

Coprocesador 8087 INTEL.

Memoria ROM 64K

Memoria RAM 64K

Memoria Video 256K

Resolución 1024X512

Colores a Desplegar 16

Controlador Gráfico TMS34061 TEXAS INSTRUMENTS.

Memorias Video TM4161 TEXAS INSTRUMENTS 25.

Circuitos Periféricos de INTEL.

Arreglo de Memoria "Packed Pixel".

CAPITULO I

Sistema de Decodificación basado en EPLD EP310.

C A P I T U L O II

CAPITULO II

ARQUITECTURA SISTEMAS GRAFICOS

II.1 INTRODUCCION.

Como se mencionó en el capítulo I, los sistemas gráficos más versátiles para nuestra aplicación son los del tipo "raster" de colores. Sin embargo este tipo de sistemas necesita una memoria adicional que contenga las imágenes que van a ser visualizadas. Esta memoria se denomina Memoria de Video, Memoria de Imagen o "bit-map", como se dijo en el capítulo anterior. (7)

A su vez, la generación de objetos para ser desplegados, implica una escritura, para almacenar el dibujo, sobre la memoria de video. El tamaño de dicha memoria depende de la resolución o número de "pixels" visualizables que contenga el "display", así como del número de colores a desplegar.

CAPITULO II

La discusión se enfocará a los sistemas "raster". Primero analizaremos los sistemas en base a procesadores generales (el caso típico de un microprocesador) para después considerar el caso de procesadores especializados como serían los VLSI (very large scale integration) que agilizan las tareas propiamente gráficas del sistema.

II.2 UTILIZACION DE PROCESADORES GENERALES. (8)

Todo sistema gráfico consta de varias partes fundamentales que permitan que una información sea desplegada en un monitor. Una representación bastante simple de lo que sería un sistema gráfico, lo tenemos en la figura 2.1. Donde el calculador principal es el encargado de generar y manipular la información hasta obtener el punto final o "pixel", así como calcular la dirección de la información con que será almacenada en la memoria de video.

II.2.1 VENTAJAS EN LA UTILIZACION DE PROCESADORES GENERALES.

Al emplear microprocesadores se tiene como ventaja que la infraestructura a que se tiene acceso es muy amplia, lo que hace a un sistema muy versátil.

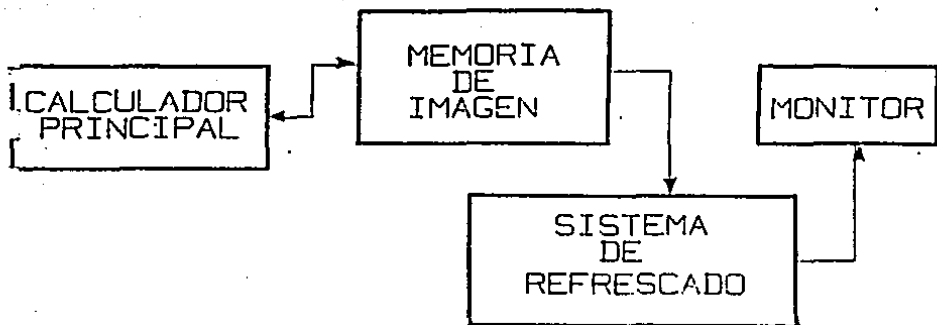


FIGURA 2.1 SISTEMA CON PROCESADOR GENERAL

CAPITULO II

II.2.2 DESVENTAJAS EN LA UTILIZACION DE PROCESADORES GENERALES.

-Carga excesiva de las tareas a desarrollar por el microprocesador, debido que éste es el que tiene que generar y manipular la información para obtener el punto final o "pixel", así como localizar el mismo en la pantalla.

-Existen problemas con la velocidad de procesamiento, ya que para obtener una nueva imagen, el calculador principal debe procesar completamente la información y transmitirla a la memoria de video.

-Complejidad en las rutinas que realizará el microprocesador, debido a que éste debe encargarse también de la generación de figuras primitivas.

-Lentitud en el procesamiento y despliegue de la información. En general cada nueva imagen debe ser recalculada desde un principio, lo cual repercute en la dificultad de hacer cambios en poco tiempo.

-Necesidad de utilizar circuitería adicional para manejo de señales de video y señales de memoria de imagen.

II.3 UTILIZACION DE PROCESADORES ESPECIALIZADOS.

II.3.1 ORGANIZACION GENERAL.

Una solución para resolver el problema de velocidad de

CAPITULO II

respuesta, es la incorporación de un procesador especializado para realizar tareas gráficas; a este tipo de procesador se le denomina más comunmente Procesador Gráfico. Un procesador de este tipo se encarga de transformar la información que proviene del procesador principal y generar la información que llenará la memoria de video, así como suministrar señales de control, tanto para las memorias como para la sincronización y dimesionamiento del TRC.

La arquitectura básica de un sistema que contiene un procesador gráfico especializado, es la mostrada en la figura 2.2.

II.3.2 CLASIFICACION DE PROCESADORES ESPECIALIZADOS.

II.3.2.1 CONTROLADOR DE VIDEO.(VSC). -

El VSC se ubica entre la computadora y el dispositivo de despliegue, recibiendo la información del computador principal para procesarla y enviarla a la memoria de video.

El VSC genera las señales necesarias para controlar las memorias de video, así como las señales de video de sincronización necesarias para controlar el TRC, genera automáticamente los ciclos de refrescado requeridos por la pantalla del TRC y ejecuta automáticamente la señal para mantener las memorias RAM dinámica habilitada.

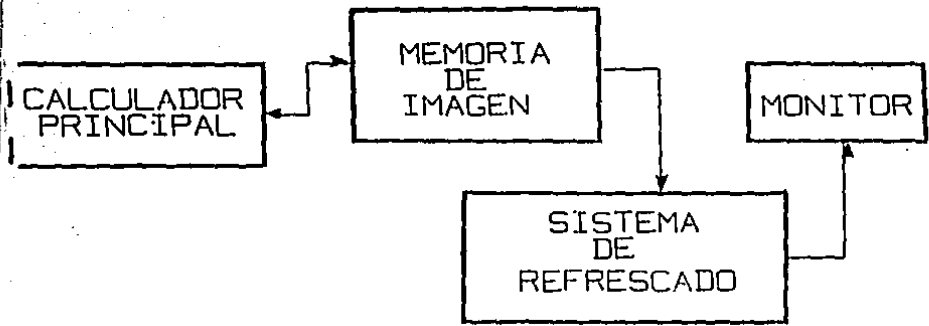


FIGURA 2.1 SISTEMA CON PROCESADOR GENERAL

CAPITULO II

El sistema que será utilizado en el diseño, es el mostrado en la figura 2.3.

II.3.2.2 GENERACION DE FIGURAS GEOMETRICAS. -

Existen procesadores especiales que dentro de sus comandos cuentan con instrucciones para la construcción de figuras geométricas o caracteres gráficos en la memoria de despliegue. Los comandos típicos para trazos incluyen los que despliegan líneas, arcos, círculos, rectángulos y áreas llenas, entre otros, permitiendo seleccionar estilos de líneas y patrones a desplegar.

II.3.2.3 EJEMPLOS DE PROCESADORES EN EL MERCADO. -(9)

Dentro de los ejemplos que podemos mencionar están desde circuitos simples de control de señales del TRC, hasta los que generan en tiempo próximo al real; tanto figuras simples a base de líneas, como figuras sofisticadas con mucho realismo.

Dentro del primer grupo tenemos: TM34061 de TEXAS INSTRUMENTS empleado en nuestro sistema que además permite manipulación de "pixel" individual. El 82720 de Intel, que aunque con capacidad de generación de figuras geométricas simples, está restringido en velocidad y resolución de memoria de video.

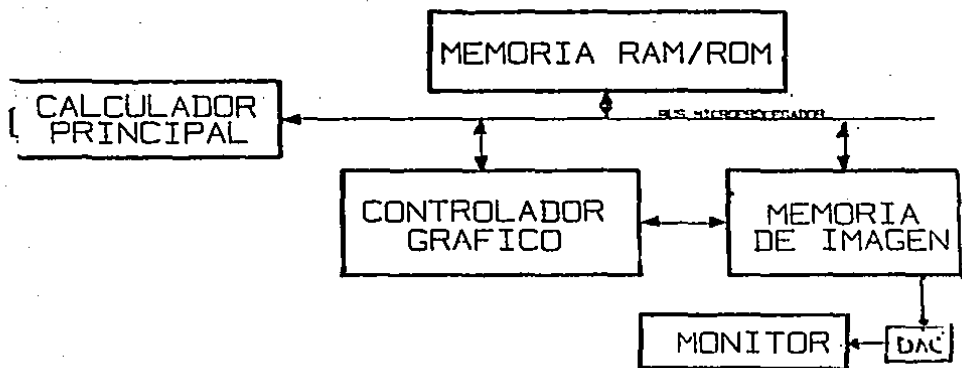


FIGURA 2.3 SISTEMA ELEGIDO EN NUESTRO CASO

CAPITULO II

Dentro del segundo grupo podemos mencionar: TMS34010 de TEXAS INSTRUMENTS para 32 bits que tiene capacidad de manejar operaciones aritméticas, lo que incrementa la velocidad del sistema al procesar información y 82786 de INTEL. Ambos procesadores compiten por el mercado, aunque cada uno presenta diferente filosofía.

Mientras que el 34010 de TEXAS es un verdadero procesador gráfico (esto incluye diferentes instrucciones que él puede ejecutar), el integrado de INTEL se comporta como un simple coprocesador, cuya interfaz con el procesador general se realiza por medio de una lista de visualización o lista de comandos que el coprocesador ejecuta uno a uno asincrónicamente con respecto al procesador general.

Las características principales de cada uno de los procesadores se muestra a continuación:

	82786	34010
TIPO DE INTEGRADO	COPROCESADOR	MICROPROCESADOR
MANEJO DE FIGURAS		
GEOMETRICAS	HARDWARE	MICROPROGRAMABLES
FLEXIBILIDAD	NULA	100%
MANEJO VENTANAS	HARDWARE	MICROPROGRAMABLES
TOLERANCIA MAX.		
MEMORIA VIDEO	4M BYTES	128M BYTES
MANEJO DE OPERACIONES		
CON BITS VIDEO	SI	SI

CAPITULO II

BITS POR PIXEL	1,2,4 Y 8	1,2,4,8 Y 16
ENCAPSULADO	88 PINS	68 PINS
ARQUITECTURA DE		
MEMORIA VIDEO	PACKED-PIXEL	PACKED-PIXEL

De las consideraciones se deben tomar en cuenta en la tabla anterior:

-El hecho que el procesador INTEL 82786 tenga todas sus funciones integradas en HARDWARE lo hace más veloz que el TMS34010.

- La facilidad de ventanas por HARDWARE del 82786 solamente es utilizable si se acopla a memorias RAM dinámicas.

- El hecho de que todas las funciones del 82786 se hagan por HARDWARE hace que el controlador de INTEL sea completamente rígido, lo que no sucede con el de TEXAS.

11.3.3 VENTAJAS EN LA UTILIZACION DE PROCESADORES ESPECIALIZADOS.

-Incrementa la velocidad de respuesta.

CAPITULO II

-El controlador de video releva de actividades al calculador principal; como sería el llenado de memorias de video, lo que hace que la carga de trabajo se vea repartida entre el controlador de video y el calculador principal.

-Logra un direccionamiento adecuado de la memoria de video, en la que se efectúa el mapeo de pixels.

-Al tener separados los sistemas de video y procesamiento general se permite que el sistema de video y el procesador principal corran asincrónicamente.

-Tener los bloques del calculador principal y de video separados, las posibilidades de errores se reducen.

-El uso de un controlador de video, facilita el manejo de ventanas, así como el aumento de información desplegada, que requiere un tratamiento especial.

II.3.4 DESVENTAJAS EN LA UTILIZACION DE PROCESADORES ESPECIALIZADOS.

El procesador principal puede tener acceso directo a la memoria de imagen al mismo tiempo que el controlador de video, lo que hace indispensable un sistema de arbitraje para el acceso a memoria video.

II.4 UTILIZACION DE TABLAS DE COLORES.

Se le llama tabla o paleta de colores, al dispositivo

CAPITULO II

que se encarga de hacer el enlace entre los datos digitales salidos de la memoria de video para ser entregados al Monitor de video; la paleta de colores que en nuestro caso será usada es la TMS34070 de TEXAS INSTRUMENTS, ésta contiene una tabla de colores que consta de dieciseis registros, los cuales son cargados según las necesidades del TRC para hacer la selección de los datos digitales de entrada y poder obtener la salida deseada.

La paleta tiene tres salidas de video TRC, que son los tres colores primarios en televisión (rojo, verde y azul), mediante los cuales se pueden generar todos los colores.

II.4.1 VENTAJAS.

-Permite manejar planos supuestos con únicamente cambiar el color de la paleta de colores, dando posibilidades para animación simple y cambios inmediatos en el despliegue.

-La tabla de colores permite que se pueda lograr la transformación de la señal digital a analógica sin la necesidad de utilizar dispositivos adicionales.

-La paleta es un circuito autónomo que no requiere de la intervención del procesador principal para su funcionamiento.

CAPITULO II

-Requiere de tiempos cortos de captura de entradas digitales, con lo que logra que los datos analógicos de salida sean más rápidos.

II.4.2 DESVENTAJAS.

-La cantidad de colores que se pueden almacenar simultáneamente para cada despliegue es limitado debido a que internamente se cuentan con pocos registros para este fin

-Al cambiar un registro de color, la información no mantiene el color inicial, sino que se actualiza en pantalla; lo que hace que aunque se puedan programar muchos colores solamente un número limitado de ellos puede ser desplegado.

C A P I T U L O . I I I

CAPITULO III

DISEÑO A BLOQUES DEL SISTEMA MAESTRO.

III. ARQUITECTURA GENERAL.

La arquitectura implementada se puede dividir para su análisis en tres bloques:

- a) Procesador General.
- b) Controlador Gráfico e Interfaz
- c) Interfaz al Bus General (Multimaestro). (Esta parte no se contempló en este trabajo).

III.1 PROCESADOR GENERAL. (10)

El microprocesador seleccionado fué el 8086-2 de INTEL. Esta selección se debe a la inversión que el INSTITUTO DE INVESTIGACIONES ELECTRICAS tiene en equipo de desarrollo de

CAPITULO III

la marca INTEL.

Las funciones internas del procesador 8086 se encuentran divididas en dos unidades de procesamiento. La primera es la Unidad de Interfaz al Bus (BIU); la segunda es la Unidad de Ejecución (EU), como se muestra en la figura 3.1.

III.1.1 UNIDAD DE INTERFAZ AL BUS.(BIU)

El BIU toma instrucciones y operandos alojados en la memoria del sistema y los coloca en una cola de ejecución de seis bytes de capacidad. La Unidad de Interfaz al Bus, posee interfaz lógica, registros de segmentos, lógica de direccionamiento de memoria y una serie de seis registros que forman una cola para alojar código de ejecución.

III.1.2 UNIDAD DE EJECUCION.

El EU recibe las instrucciones del BIU, a través de la cola de ejecución de manera secuencial, la información procesada es almacenada en una memoria general por medio de la Unidad de Interfaz al Bus. La Unidad de Ejecución contiene registros de datos y direcciones, unidades aritméticas y lógicas, más la unidad de control.

La Unidad de Ejecución y la de Interfaz al Bus operan en forma asíncrona. Cuando la Unidad de Ejecución está lista para ejecutar una nueva instrucción, la toma del primer registro que forma la cola de instrucciones de la Unidad de

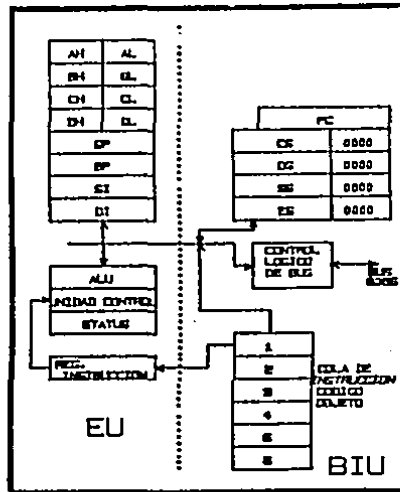


FIGURA 3.1 UNIDADES DE EJECUCION E INTERFAZ



FIGURA 3.2 CICLO DE BUSQUEDA 1ER. BYTE CODIGO OBJETO.

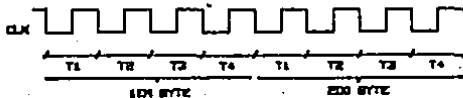


FIGURA 3.3

CAPITULO III

Interfaz al Bus, en seguida ejecuta la instrucción en un número determinado de ciclos, que no tienen relación con los ciclos del bus. Si la cola de instrucciones del código objeto está vacía, el BIU ejecuta un ciclo de lectura de código: El EU espera por la instrucción del código a ejecutarse, pero como la cola pocas veces estará vacía, esto hará que la Unidad de Ejecución ejecute más rápido las instrucciones leídas.

Si la memoria o un dispositivo entrada-salida es accesado en el momento que una instrucción se está ejecutando, el EU se lo informa al BIU, que ejecuta un ciclo externo en respuesta a la demanda del EU.

La Unidad de Interfaz al Bus es independiente de la de Ejecución. El BIU intenta mantener los seis bytes de código objeto llenos. Si dos o más de los seis bytes están vacíos, el BIU efectuará la lectura de más instrucciones, siempre y cuando el EU no tenga una petición del bus. Si el EU manda una petición de acceso al bus mientras el BIU está ejecutando una instrucción, éste terminará la lectura de la instrucción antes de aceptar la petición de acceso al bus.

III.1.2.1 COLA DE INSTRUCCIONES. -

Consideremos que pasa cuando una instrucción es ejecutada, empezando por el simple caso de que la cola de instrucciones de código objeto esté vacía y exista una solicitud de

CAPITULO III

información. El BIU tendrá que ejecutar un ciclo de lectura para el primer byte de instrucción de código objeto, (Ver figura 3.2) asumiendo que la instrucción requiere de dos bytes de código objeto. (Ver fig. 3.3.)

Ahora supondremos que ésta instrucción lee un dato de memoria y después ejecuta una operación usando el mismo. La instrucción requerirá de algunos ciclos de reloj para calcular la dirección de la localidad de memoria que será accesada. Algunos periodos adicionales son necesarios para efectuar la operación aritmética. Esta instrucción debe ser ejecutada en la siguiente secuencia de ciclos máquina, esto se muestra en la figura 3.4, donde:

- A.- Primer byte de lectura de código objeto.
- B.- Segundo byte de lectura de código objeto.
- C,D.- Datos de cálculo de direcciones.
- E.- Ciclo de lectura desde memoria.
- F,G.- Ejecución de operación aritmética en ciclos máquina normales y largos.
- H.- Comienzo de la ejecución de la instrucción siguiente con la lectura del código objeto.

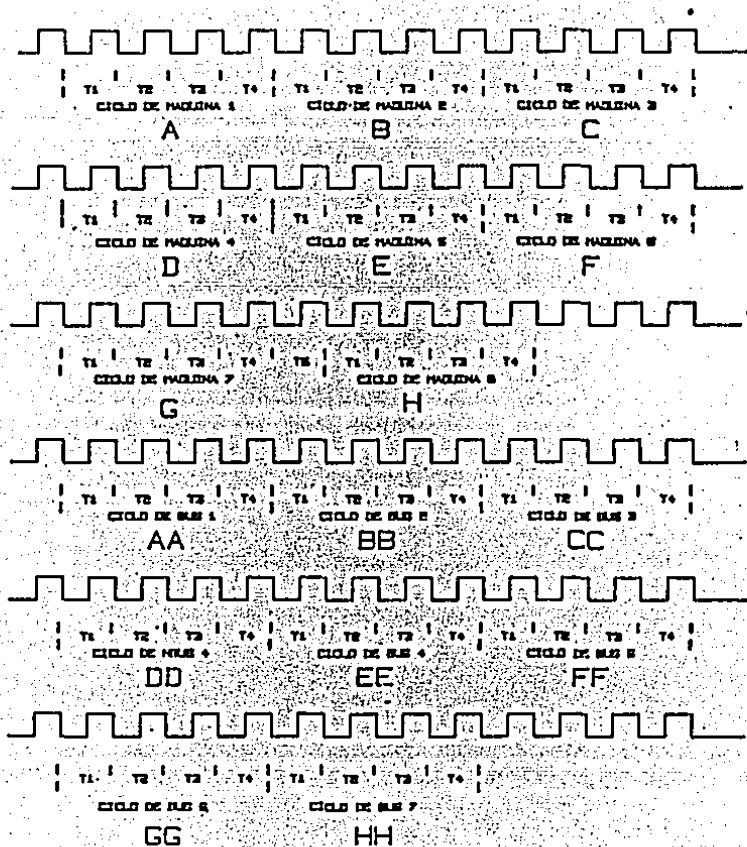


FIGURA 3,4

CAPITULO III

Siendo el CPU y La Unidad de Control de Bus asíncronos, usarán ciclos de reloj para la ejecución de la instrucción como sigue:

AA.- EU pregunta por la lectura del código objeto. El BIU lee el código objeto en un ciclo de bus.

BB.- La EU necesita un segundo byte de código objeto. BIU lee el segundo byte del código objeto en un ciclo de bus.

CC, DD.- El EU calcula la dirección de la memoria en siete ciclos de reloj. Al final del séptimo ciclo de reloj el CPU hace una petición de acceso al bus. Mientras el EU no haga una petición de acceso al bus, el BIU leerá los siguientes dos bytes de código objeto y los almacenará en la fila. Al final del cuarto ciclo de bus el EU hará una petición de acceso al bus, mientras la unidad de control de bus atiende al CPU.

EE.- El EU espera por la petición del dato que será leído por el BIU. El BIU hace la lectura de la localidad de memoria direccionada por el CPU.

FF, GG.- El EU utiliza nueve ciclos de reloj para efectuar la operación aritmética. El BIU continúa ejecutando ciclos de reloj para llenar la cola de instrucciones de código objeto.

CAPITULO III

HH.- El EU toma el siguiente byte del código objeto de la cola del BIU y comienza a ejecutar la siguiente instrucción.

III.1.3 ORGANIZACION DE LA MEMORIA.

El procesador consta de 20 líneas de direccionamiento a memoria. La memoria está organizada como un arreglo lineal de hasta un millón de bytes, direccionados de 0000(H) a FFFFF(H). La memoria se encuentra dividida en regiones o segmentos. La segmentación y mapeo de memoria, son hechas en base a las necesidades del programa de aplicación.

Palabras de 16 bits pueden ser localizadas en direcciones pares y nones. Para direcciones y datos, el byte menos significativo de la palabra es almacenado en la parte más baja, y el byte más significativo es almacenado en la siguiente localidad.

Físicamente la palabra de datos se organiza como una parte alta (non)(D15-D8) y una parte baja (par)(D7-D0).

Ciertas localidades de memoria son reservadas para funciones del CPU, localidades de direccionamiento, de FFFF(H) a FFFFF(H) son reservadas para operaciones incluyendo saltos en la rutina que carga el programa inicial.

CAPITULO III

III.1.4 OPERACION DEL BUS.

EL 8086 tiene un bus donde se encuentran distribuidos en tiempo, datos y direcciones, compartiendo las mismas líneas físicas, lo cual optimiza el uso de sus 40 terminales o "pins". Este BUS LOCAL puede ser distribuido en espacio, en datos y direcciones por medio de biestables controlados por señales apropiadas.

Cada ciclo de procesamiento del BIU consta de cuatro ciclos de reloj, T1, T2, T3, T4. Las direcciones son transmitidas por el procesador en T1, los datos en T3 y T4, T2 es usado para cambiar la dirección del bus durante operaciones de lectura. Si durante el evento, algún periférico o memoria no está listo, son insertados ciclos de espera entre T3 y T4, cada ciclo de espera tiene la misma duración de un ciclo de reloj. Los bits de estado $\wedge S0$, $\wedge S1$ y $\wedge S2$ son utilizados para que el controlador del bus identifique tipo de intercambio realizado entre periféricos y procesador de acuerdo con la tabla 3.1

TABLA 3.1

$\wedge S2$	$\wedge S1$	$\wedge S0$	
0	0	0	Reconocimiento de interrupción.
0	0	1	Lectura entrada-salida.
0	1	0	Escritura entrada-salida.
0	1	1	Alto.

CAPITULO III

1	0	0	Instrucción de lectura de código.
1	0	1	Lectura de datos de memoria.
1	1	0	Escritura de datos a memoria.
1	1	1	Pasivo.

III.1.5 DIRECCIONAMIENTO ENTRADA-SALIDA.

En el Microprocesador 8086 las operaciones de entrada y salida pueden direccionar hasta un máximo de 64K en registros de bytes o 32K en registros de palabras. Los puertos de entrada salida son direccionados de la misma manera como las localidades de memoria. Direcciones pares en D7-D0 y direcciones nones en D15-D8.

Unicamente son utilizadas las líneas A15-A0 para puertos, A19-A16 no son utilizadas en operaciones de entrada-salida.

III.1.6 ESTRUCTURA DE INTERRUPCION.

Una interrupción es una forma de que el procesador detenga

* NOTA: El simbolo ^ antes de un comando, indica que la señal es activo bajo.

CAPITULO III

la ejecución de un programa, ya sea para atender la entrada de un puerto entrada-salida, efectuar cambios de contexto, cuando un error es detectado, etc. Por lo general el procesador ejecuta su trabajo, hasta que una interrupción ocurre, en cuyo momento salva su estado actual (apuntador de direcciones, segmento de código); ejecuta una rutina especial y regresa a lo que estaba haciendo antes de que ocurriera la interrupción.

La estructura de las interrupciones, se basa en una tabla de vectores de interrupción, almacenados en localidades de memoria (0000(H)-003FF(H)) como se ilustra en la figura 3.5.

Cada vector consta de 4 bytes, en 2 bytes se localiza el nuevo contenido del segmento de código y en los otros dos el apuntador de instrucciones.

La tabla de vectores de interrupción puede tener hasta 256 vectores, cada vector tiene asociado un número de interrupción. Las interrupciones pueden ser activadas vía "Software" o "Hardware".

Las interrupciones "Hardware" se activan mediante la línea de interrupción \wedge INTR del microprocesador, salvando las condiciones de sus registros, con un puntero en la dirección de retorno; cargando el segmento código y el puntero de instrucciones en la parte superior de la pila, para que el microprocesador pueda regresar a su tarea sin ningún

CAPITULO III

problema.

Existen interrupciones, predefinidas con un número, localizadas en la tabla de los vectores de interrupción, como se muestra a continuación:

INTERRUPCION 0.- DIVISION ENTRE CERO.

Se solicita automáticamente cuando al ejecutarse una instrucción de división el cociente excede el máximo valor permitido.

INTERRUPCION 1.- UN SOLO PASO.

Esta interrupción se emplea para ejecutar una sola instrucción a la vez. Después de que cada instrucción se ejecuta se genera una interrupción de este tipo.

INTERRUPCION 2.- NO MASCARABLE.

Esta es la interrupción de mayor prioridad vía "Hardware". Es disparada mediante una transición de bajo a alto en el "pin" denominado NMI del 8086. En el caso de que la señal que entra a esta terminal sea normalmente alta, deberá cambiar su estado a bajo, por lo menos por dos periodos de reloj antes de efectuar su transición de bajo a alto. Esta interrupción se utiliza para manejar situaciones de emergencia como errores no recuperables.

CAPITULO III

INTERRUPCION 3.- INTERRUPCIONES DE UN BYTE.

Esta es una interrupción via "Software". Dicha interrupción es generada por una petición de interrupción especial que ocupa un solo byte de código objeto. Es usada para detener el programa y efectuar un salto a una dirección fija.

INTERRUPCION 4.- INTERRUPCION POR DESBORDAMIENTO.

Esta petición de interrupción ocurre si la bandera de desbordamiento se establece en la palabra de estado, la instrucción INTO es ejecutada permitiendo la construcción de una rutina para manejar este tipo de excepción.

Si la señal de ^INTR es alta, el 8086 realiza una secuencia de reconocimiento de interrupción. Para garantizar la interrupción ^INTR debe permanecer alta hasta que la interrupción sea reconocida via ^S0, ^S1 y ^S2.

Si el BIU ejecuta un ciclo de bus y una interrupción ocurre, ésta debe aparecer dos ciclos de reloj antes de T4 para que sea válida. Si ocurre después, el ciclo de bus continúa con su tarea, hasta el proximo período, si ocurre simultaneamente con una petición de bus, ésta última tiene la mayor prioridad.

CAPITULO III

En modo máximo, al ser reconocida la interrupción por medio de ^S0, ^S1 Y ^S2, se provoca que la señal ^INTA del Controlador de Bus 8288 sea baja durante el ciclo de reconocimiento de interrupción. La señal LOCK (no ceder control del bus) será activa de T2 del primer ciclo a T2 del segundo ciclo, para prevenir una petición de bus y evitar que éste sea concedido durante la interrupción. Solamente interrupciones externas vía ^INTR pueden ser deshabilitadas, por lo tanto, éstas tienen menor prioridad.

Dentro de las interrupciones existen prioridades. Las de mayor son las predefinidas (Un solo paso, NMI, etc.) y las de menor prioridad son las vía "Software".

III.1.7 DISEÑO A BLOQUES DEL PROCESADOR GENERAL. (11) (12)

Para poder comprender en forma sencilla el diseño del procesador general se ha dividido en varios bloques como se muestra a continuación: (figura 3.6)

III.1.7.1 BLOQUE DE SINCRONIA. -

(VER PLANO (1,1)(*1) (8284)(Generador de Ciclos de Reloj).(U1).

El bloque de sincronía, es el que se encarga de generar

(*1)Consultar los planos en el apéndice I.

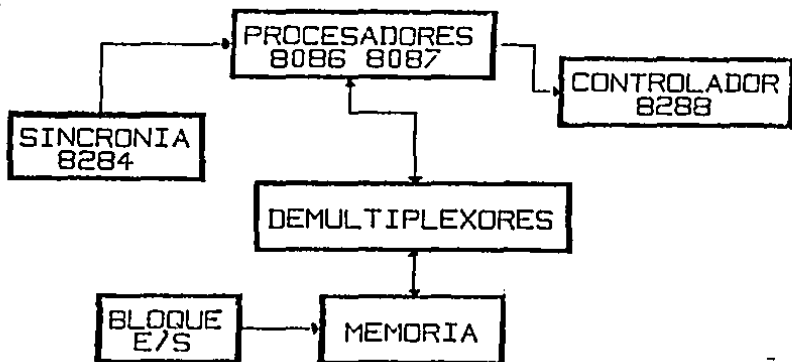


FIGURA 3.6

CAPITULO III

los sistemas de reloj, restablecimiento (RESET), así como la señal de READY cuya función es la de retrasar comandos para memoria y periféricos de baja velocidad, pudiéndose insertar desde un tiempo de espera hasta n ; el tiempo mínimo de espera es de un ciclo de reloj.

El sistema de reloj para los bloques de procesamiento y del controlador es de 6MHZ.. Para la comunicación entrada-salida, (con ayuda del MUART) (Ver III.1.6.1), la frecuencia es de 3MHz, ambas frecuencias son obtenidas en base a un cristal oscilador de 18MHz. El ciclo de trabajo del 8284 es del 33% del ciclo máquina.

Para que el restablecimiento o RESET obtenido del circuito 8284 sea reconocido, debe durar por lo menos 50 microsegundos.

III.1.7.2 BLOQUE DE PROCESAMIENTO. -

(VER PLANOS (1,1), (2,1) Y (3,1))

EL bloque de procesamiento, lo podemos representar de la forma que muestra la figura 3.8.

Las líneas de entrada proporcionan información al bloque de procesamiento, las líneas de salida transmiten información desde el bloque de procesamiento. Este bloque toma datos de la línea de entrada, manipula dichos datos para ejecutar instrucciones provenientes de la memoria del

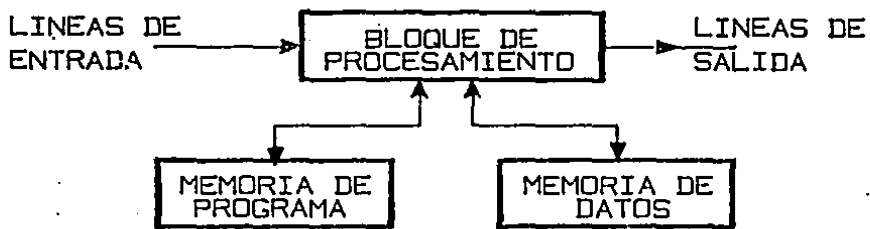


FIGURA 3.8

CAPITULO III

programa. Los resultados están disponibles via las líneas de salida, siendo almacenados estos datos variables en memoria.

Este bloque a su vez lo podemos dividir en tres partes que veremos posteriormente, lo que es el procesador propiamente dicho, el coprocesador y los demultiplexores.

El bloque de procesamiento, produce y recibe varios tipos de señales como las que se mencionan a continuación:

LINEAS DE DATOS Y DIRECCIONES.

-El sistema consta de 20 líneas de direcciones, las cuales se encuentran compartidas con otras señales, lo que permite direccionar hasta un megabyte de memoria.

-16 líneas de datos, que se encuentran compartidas con direcciones (AD0-AD15).

-Líneas que indican si se está direccionando una palabra o un byte.(A0 Y BHE).

LINEAS DE CONTROL Y ESTADOS.

-Entre las líneas de control estan las que manejan lectura, escritura, velocidad en los perifericos, interrupciones, restablecimiento y modo.

CAPITULO III

-Las líneas de estado se encuentran de la misma forma compartidas con otras señales, pero debido a que se está trabajando en modo máximo tenemos ^S0, ^S1 y ^S2. Estas señales alimentan al bloque del controlador.

-QS1 y QS2 son señales de estado de la cola de instrucciones, solo disponibles en modo máximo y son utilizadas para que un coprocesador se coordine con el 8086.Ver tabla 3.2

TABLA 3.2

QS0	QS1	
0	0	No operación.
0	1	El 1er byte de la instrucción se está ejecutando.
1	0	Vaciando la cola.
1	1	Una nueva instrucción está siendo tomada de la cola.

-Existen otras líneas que sirven para pedir, conceder y mantener el control del bus (LOCK, RQ/^GT). LINEAS DE POTENCIA Y SINCRONIA.

-CLK es la señal de reloj por la cual es sincronizada la lógica 8086. Esta señal proviene de el 8084(Generador de Ciclos de Reloj)(U1).

CAPITULO III

-VCC señal de alimentación de 5V +/- 10% para el 8086.

-GND el microprocesador tiene dos líneas de tierra.

III.1.7.3 DEMULTIPLEXORES. -

Como se mencionó anteriormente las líneas o bus de datos/direcciones comparten las mismas líneas físicas, para hacer la separación de ambas informaciones es necesario usar elementos lógicos que funcionen como separadores o demultiplexores, dichos dispositivos son "LATCHES" y "TRANSCPTORES".

III.1.7.3.1 SISTEMA DE BIESTABLES TIPO D ("LATCHES"). (VER PLANO - 2,1)(74LS373).(U4, U5 y U6).

Estos dispositivos de ocho bits permiten hacer la separación de las direcciones y de los datos, a la vez que sirven para reforzar la capacidad de corriente.

Para que el biestable pueda funcionar requiere de dos señales, una ALE, proveniente del controlador, cuya función es la de habilitar el biestable en el momento que las direcciones sean presentadas por el 8086 en el bus, para que éste las tome. La otra señal \overline{OE} habilita la salida del circuito que se encuentra en tercer estado, haciendola activa.

CAPITULO III

El bus de datos/direcciones está conectado a tres biestables. En el primero(U4) (AD0-AD7) obtenemos los bytes de direcciones de A0-A7, en el segundo(U5) (AD8-AD15) obtenemos A8-A15, en el tercero (U6) obtenemos A16-A19 y estados ^S0, ^S1 y ^S2 estables a lo largo del ciclo.

III.1.7.3.2 SISTEMA DE TRANSCPTORES. (VER PLANO 3,1)(245).(U8 Y U9) -

Estos dispositivos de ocho bits permiten hacer la selección de los datos y sirve de igual manera como reforzador de corriente. Para que el transceptor pueda funcionar requiere también de dos señales de control. T es la que controla la dirección, debido a que los dispositivos son bidireccionables, si T es alta, el transceptor se configura con A0-A7(*2) como entradas y B0-B7 como salidas. Si la señal T es baja, A0-A7(*2) son salidas y B0-B7 son entradas. La otra señal es ^OE que sirve para la habilitación de salidas. En la entrada de cada uno de los dos transceptores se encuentran los buses de direcciones/datos al primer transceptor(U8) obtenemos D0-D7, en el segundo transceptor(U9) obtenemos los datos de D8-D15.

La información obtenida de los biestables y los transceptores, alimenta las memorias de programa y de datos.

CAPITULO III

III.1.7.3.3 COPROCESADOR.(8087) -

(VER PLANO 1,1)

El bloque de procesamiento tiene un coprocesador numérico. Su función es la de servir como soporte del 8086, ya que puede ejecutar funciones trigonométricas, logarítmicas, exponenciales o punto flotante.

Al igual que el 8086, en el 8087(U3) datos y direcciones se encuentran compartiendo líneas físicas. Las líneas de estado determinan también el tipo de ciclo de bus.

Tabla 3.3

$\wedge S2$	$\wedge S1$	$\wedge S0$	
0	X	X	No se emplea.
1	0	0	No se emplea.
1	0	1	Lectura de memoria de datos.
1	1	0	Escritura en memoria de datos.
1	1	1	Pasivo.

(*2) NOTA: No confundir A0-A7 con direcciones es la nomenclatura que se usa en el circuito LS245 (Transceptor bidireccional).

CAPITULO III

Estas señales son empleadas por el controlador del bus, para generar las señales de control de acceso a memoria.

Debido a que el controlador es un dispositivo especializado, puede ejecutar sus funciones más rápidamente que un dispositivo de propósito general. El coprocesador requiere de ser conectado al bus local datos/direcciones del 8086, a las líneas de estados, de reloj, de READY, de restablecimiento (RESET) y de petición-concesión, lo que hace que éste tenga acceso tanto a memoria, como a los dispositivos entrada-salida del procesador.

El coprocesador es independiente de la configuración del sistema, pero trabaja junto con el procesador reconociendo instrucciones y ejecutando la función deseada. La operación del procesador y del coprocesador puede ser en forma paralela, sin que el primero se vea afectado, a menos que el coprocesador necesite referencia a memoria o a algún puerto entrada-salida.

Cuando el procesador cede el bus local al coprocesador, el primero puede continuar ejecutando su cola de instrucciones internas, deteniéndose cuando necesite el bus.

El procesador y el coprocesador deben estar sincronizados debido a que uno debe utilizar la información del otro. La instrucción de sincronización procesador-coprocesador WAIT

CAPITULO III

es manejada por la línea TEST del procesador. El coprocesador avisa al coprocesador que está ocupado vía esta línea. Siempre que el procesador ejecuta una instrucción de WAIT, éste detendrá la operación del programa mientras la entrada TEST sea activa.

El procesador tiene la responsabilidad de controlar todo el programa. La operación del coprocesador se inicializa por instrucciones especiales detectadas por el procesador, las cuales se llaman de ESCAPE. Hay 576 diferentes instrucciones de ESCAPE que permiten al coprocesador ejecutar diferentes instrucciones.

III.1.7.4 BLOQUE DEL CONTROLADOR.8288: -

(VER PLANO 1,2).

Una forma simple para representar el controlador es dividirlo en 2 partes: Decodificador y Generador de señales de control.

La función del Decodificador, es la determinación de la instrucción que se generará dependiendo de la combinación de las tres líneas de estados ^S0, ^S1 y ^S2.

tabla 3.4

^S2	^S1	^S0	
0	0	0	Reconocimiento de interrupción. ^INTA
0	0	1	Lectura entrada/salida. ^IORD
0	1	0	Escritura entrada/salida. ^IOWC

CAPITULO III

0	1	1	Halt.
1	0	0	Acceso a código. ^MRDC
1	0	1	Lectura de memoria. ^MRDC
1	1	0	Escritura en memoria. ^MWTC
1	1	1	Pasivo.

La segunda parte (Generador de señales de control), se encarga de generar salidas de control teniendo cuatro señales de entrada CLK (reloj), ^AEN (habilitación de direcciones), CEN (habilitación de comandos), IOB (modo del bus entrada-salida), obteniendo a la salida cuatro señales, que son: DT/^R (transmisión recepción de datos), DEN (habilitación de datos) ALE (línea de habilitación de biestables) MCE/^PDEN (habilitación maestra de interrupciones).

-La señal que habilita los biestables es ALE, la cual permite que estos capturen las direcciones y estados.

-DEN y DT/^R sirven como habilitación de los transceptores. el controlador puede ser visto como una máquina de estados, en la cual para poder pasar de un comando a otro, es necesario estar en PASIVO, excepto en el caso de HALT, en el que es necesaria una interrupción o RESET para salir del estado.

CAPITULO III

III.1.7.4.1 BLOQUE DE MEMORIA. -

(VER PLANOS 2,1 y 3,1).

El bloque de memoria lo podemos dividir en dos partes:

-Memoria Permanente .

-Memoria Volátil.

III.1.7.4.2 MEMORIA PERMANENTE. (27128)(U10 Y U11). -

En la memoria permanente se encuentra almacenado el programa, a esta memoria se le llama ROM (memoria de lectura) o memoria no volátil. Tiene un tiempo de acceso de 250 ns y una capacidad de direccionamiento de 16K por 8 bytes.

La memoria tiene diferentes líneas las cuales son:

-Líneas de Direcciones: Son 14 líneas, las cuales son alimentadas por dos biestables (U4,U5), de A1 A A15. A0 Tiene por finalidad de seleccionar si se direcciona byte par o non (en el caso de accesos a palabras completas de 16 bits en un ciclo, A0=BHE=0).

-Líneas de Datos: Son ocho líneas (D0-D7) de salida, las cuales obtienen información por medio de los transceptores (U8,U9); para la primera memoria 27128 (U10), las líneas son D0-D7, para la segunda memoria (U11) las líneas son D8-D15, esto se debe a que el procesador 8086 tiene un bus de datos

CAPITULO III

de 16 bits de longitud.

-Líneas de Habilitación: ^CS (habilitación del circuito) y ^OE (habilitación de salidas, mediante un comando de escritura.

-Línea de Programación: ^PGM por medio de la cual se efectua la programación.

Las memorias ROM 27128, tienen la facultad de que una vez programadas, puede ser restablecida mediante luz ultravioleta, teniendo cuidado de que esta exposición no sea por periodos muy largos, ya que se pueden ocasionar daños permanentes. Posteriormente las memorias pueden ser grabadas nuevamente.

III.1.7.4.3 MEMORIA VOLATIL. -

La memoria volátil o RAM (43256)(U12,U13) almacena datos variables que son generados durante el programa, por el microprocesador o código relocizable. En el circuito tenemos dos memorias RAM las cuales tienen un tiempo de acceso de 150 ns aproximadamente y una capacidad de direccionamiento de 32K por 8 cada una, las memorias tienen diferentes líneas.

-Líneas de Direcciones: Son 15 líneas, las cuales son alimentadas por dos biestables (U4,U5) con las direcciones A1-A14.

CAPITULO III

-Líneas de Datos: Son 8 (I/O0 I/O8) las cuales pueden ser tanto de entrada como de salida y funcionan capturando o enviando información por medio de los transceptores (U8,U9).

-Líneas de Habilitación: ^CS (habilitación del circuito), ^OE (habilitación de salida o lectura) y ^WE (habilitación de escritura).

III.1.7.5 BLOQUE ENTRADA-SALIDA -

III.1.7.5.1 RECEPTOR-TRANSMISOR UNIVERSAL MULTIFUNCIONES - ASINCRONO.

(MUART)(8256)(U14) (VER PLANO 3,1)

La primera parte del bloque ENTRADA/SALIDA está formada por el circuito llamado MUART. Este circuito está diseñado para ser usado en funciones como interfaz al 8086, para comunicación serie asíncrona, proporcionando soporte "hardware" para entradas-salidas paralelas, "timing", control de eventos e interrupciones. Todas estas funciones son totalmente programadas por medio de nueve registros internos, además los 5 contadores/"timers".

DESCRIPCION FUNCIONAL.

A) Comunicaciones Serie. La parte de comunicaciones posee un UART (Transmisor Receptor Universal Asíncrono), así como un generador de "baud rate" que permite la variación de

CAPITULO III

velocidades de comunicación sin necesidad de componentes externos. El UART puede ser programado por el CPU para la variación en los tamaño de los caracteres, generación y detección de paridad, detección de errores y bits de paro-inicio.

B) Entradas-Salidas Paralelas. El MUART incluye 16 bits para el propósito de entrada-salida. Ocho bits (puerto 1) pueden ser configurados de manera individual como entrada, salida o funciones especiales de E/S. Los otro ocho bits (puerto 2) son programados como "nibbles" (grupos de 4 bits) o como bytes. Todos los bits del puerto 1 tienen funciones alternativas diferentes a entradas salidas como se muestra en la tabla 3.5

TABLA 3.5

SIMBOLO DE	FUNCION DE CONTROL
PATA	
P10	^ACK Señales de control para el puerto 2. ^STB
P11	^OBF Manejador de salidas de 8 bits. ^IBF
P12	Contador de eventos 2 Entrada de reloj.
P13	Contador de eventos 3 Entrada de reloj.
P14	"Baud rate" interno.
P15	"Timer" 5 disparador de entrada.

CAPITULO III

- P16 Entrada para detección de interrupciones.
P17 Entrada para interrupciones externas.

C) Contadores/"Timers". El MUART posee cinco contadores/"timers" de ocho bits. Los "timers" pueden ser programados para usar sistemas de reloj de 1 o 16 KHz. Cuatro de los contadores/"timers" pueden ser conectados en forma de cascada para formar dos contadores de 16 bits. El otro registro puede ser inicializado por una señal externa.

La diferencia entre contador y "timer" está determinada por la fuente del reloj, un "timer" mide el intervalo de tiempo absoluto y su frecuencia de entrada proviene del sistema de reloj del MUART. La entrada al contador es derivada de un pulso aplicado en una terminal externa. El contador es decrementado en cada flanco de subida de la señal.

D) Controlador de Interrupciones. En nuestro caso el manejo de las interrupciones se lleva a cabo por medio de un circuito especial programable para interrupciones PIC. Este circuito se tratará posteriormente.

INICIALIZACION. En general las funciones del MUART son independientes unas de otras y solo los registros y los bits con función especial asociada necesitan ser inicializados, no el circuito entero. La secuencia de los comandos es arbitraria ya que cada registro es directamente

CAPITULO III

direccionable.

INTERFAZ MUART-MICROPROCESADOR. La figura 3.10 muestra la interfaz entre el 8086 y el 8256.

El MUART se encuentra directamente conectado al bus multiplexado de datos y direcciones del microprocesador, las direcciones son separadas dentro del circuito durante el flanco de bajada de la señal ALE. Para sistemas de 16 bits son tomadas las líneas AD1-AD4 para seleccionar 4 bits de direcciones, con las cuales los 16 registros son direccionados. Señales de ALE, ^RD, ^WR y RESET son alimentadas desde el 8086 al circuito en modo mínimo, en modo máximo provienen del controlador de bus 8288.

III.1.7.5.2 PROGRAMADOR DE INTERRUPCIONES.(PIC). -

(8259)(U15).

La segunda parte del bloque de ENTRADA/SALIDA es el PIC (Controlador Programable de Interrupciones). El PIC tiene como función manejar el sistema de interrupciones. Este acepta peticiones del equipo periférico, determinando cual de las peticiones tiene mayor prioridad, la cual es atendida antes que las otras mandando una interrupción al CPU.

El PIC después de emitir la interrupción, debe de suministrar la información al CPU con la dirección de la rutina a ejecutar, predeterminada y almacenada en los

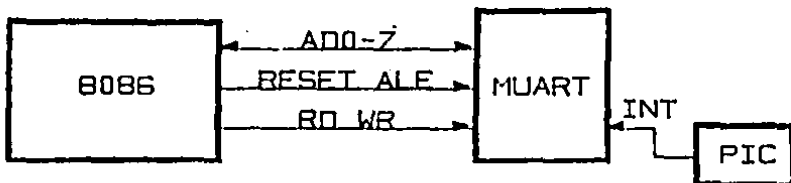


FIGURA 3,10 INTERFAZ AL MUART.

CAPITULO III

vectores de interrupción.

El PIC maneja ocho niveles o peticiones que pueden ser ampliadas, si se conectan a otros circuitos 8259, hasta un máximo de 64 niveles, cuyas prioridades son configurables.

El método de interrupciones se muestra en la figura 3.11

El PIC internamente puede dividirse en los módulos siguientes

1.- Registros. Donde se almacenan niveles de interrupción de periféricos, interrupciones en servicio y registros de interrupciones mascarables.

2.- Analizador de prioridades. Determina la prioridad de las interrupciones

3.- Líneas de control.

-INT (Interrupción). Va directamente a la línea de interrupción del CPU.

- $\overline{\text{INTA}}$. (Reconocimiento de Interrupciones) Los pulsos de $\overline{\text{INTA}}$, harán que el 8259 libere información de los vectores en el bus de datos e informará si se trata de una configuración sencilla o en cascada.

- $\overline{\text{CS}}$ (Selección del Circuito). Habilita el circuito para su programación inicial y para leer sus registros internos.

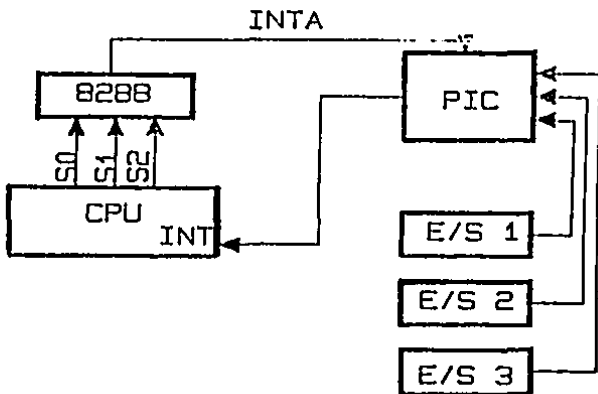


FIGURA 3.11 DIAGRAMA DE INTERFAZ AL PIC.

CAPITULO III

-[^]WR (Escritura). Habilita para escribir palabras de control.

-[^]RD (Lectura). Habilita el 8259 para mandar los niveles de interrupción en el bus de datos.

-AO se usa junto con [^]WR y [^]RD para escribir comandos en los registros así como para leer los mismos.

4.- "Buffer" del bus de datos (DBB). Este buffer bidireccional de ocho bits, sirve como interfaz entre el PIC y el bus de datos. Palabras de control e información son transferidos mediante el "Buffer" del bus.

SECUENCIA DE INTERRUPCION.

La secuencia normal de eventos de interrupción es la siguiente:

Primeramente una o más peticiones de interrupción son mandadas por medio de las líneas IR7-0, el 8259 evalúa estas peticiones y manda una señal de INT al procesador. El procesador reconoce la interrupción y manda una señal de [^]INTA por medio del 8288 en modo máximo.

Después de recibir un [^]INTA del CPU, el 8259 mandará un apuntador de ocho bits en el bus de datos, éste es leído por el CPU. La señal de interrupción permanecerá presente hasta que un comando apropiado sea indicado al final de la rutina de interrupción. (Comando de fin de interrupción).

C A P I T U L O I V

CAPITULO IV

CIRCUITERIA DE INTERFAZ ENTRE PROCESADOR, MEMORIA Y CONTROLADOR.

IV.1 CONTROLADOR GRAFICO (VSC) E INTERFAZ.

IV.1.1 DESCRIPCION GENERAL. (13)

Para nuestro diseño se empleo el controlador gráfico TMS34061 DE TEXAS INSTRUMENTS (U56)(PLANO 2,2). Dicho dispositivo se encarga de controlar tanto el refresco de la pantalla como el refrescado dinámico de las memorias dinámicas que sirven como memoria de video, también genera señales de "Blank" y sincronía necesarias para la interfaz con el TRC.

CAPITULO IV

El papel principal del VSC (Video System Controller) o controlador de video, es proveer a un procesador externo un acceso ilimitado a la memoria de video, eliminando así los retardos de actualización que ocurren cuando el procesador principal efectúa solo todo el trabajo. De esta manera dicho procesador es relevado de la carga de controlar el sistema de memoria, refrescado y cargado de la misma. El tipo de memorias usadas se verá en la parte IV.2.

EL VSC está formado de tres bloques como se muestra en la figura 4.1. A continuación detallaremos cada uno de estos bloques.

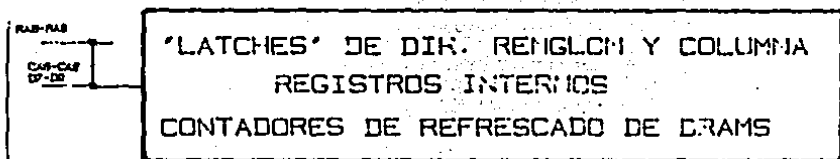
- a) La interfaz al procesador.
- b) Interfaz a la memoria de video.
- c) Arbitro.

IV.1.1.1 INTERFAZ AL PROCESADOR. -

El procesador suministra información al controlador de video. Esta información es procesada por el controlador para obtener las señales que alimentarán tanto a las memorias como al TRC.

La interfaz al procesador general consta de tres bloques funcionales básicos.

INTERFAZ-AL PROCESADOR



INTERFAZ A MEMORIA DE VIDEO

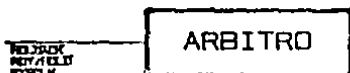
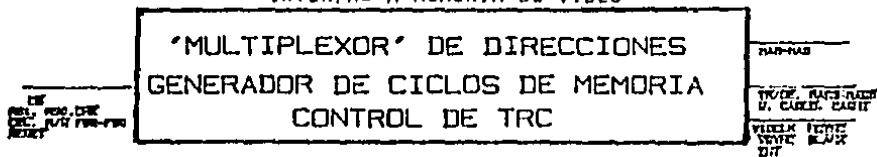


FIGURA 4.1 BLOQUES DEL CONTROLADOR.

CAPITULO IV

1) "Latches" de dirección de renglón y columna. Su función es que el controlador no necesite "latches" externos, sobre todo en sistemas en que se encuentra compartido el bus de datos y direcciones.

2) Registros Internos.

- Registros X-Y. La capacidad de direccionamiento X-Y es útil cuando el rango de direccionamiento lineal del procesador general se encuentra limitado para poder acceder todos los "pixels" de la pantalla. Este es el caso cuando se tiene un procesador de ocho bits de datos y solamente se cuenta con 16 bits para hacer el direccionamiento de memoria. El contenido de dichos registros reemplazan las salidas de renglón y columnas como fuente de direcciones de memoria. Un código de cuatro bits en las entradas CA4-CA1 del VSC determina el ajuste del direccionamiento indirecto (incrementos, decrementos, sin cambio, etc.). Los registros X-Y relevan al procesador principal de calcular la dirección del siguiente "pixel" a ser modificado.

- Registros de control 1 y 2. El VSC tiene dos registros de control internos programables formados por 16 bits cada uno, los cuales determinan la forma de operación del VSC. Cada registro puede ser leído o modificado por el procesador principal. Funciones de video como la frecuencia de refrescado de las RAMs dinámicas, ciclos de actualización de pantalla, la configuración de los tiempos de video,

CAPITULO IV

habilitación de las peticiones de interrupción y algunas señales de interfaz al procesador son manejadas por dichos registros.

- Registros de tiempo de video. Los tiempos de sincronía vertical y horizontal e intervalos de "blank", son controlados por 10 registros de tiempo de video. Cuatro registros y un contador horizontal generan las señales de tiempo horizontal, cuatro registros y un contador vertical generan la señal de tiempo vertical y los otros dos registros mandan interrupciones verticales al procesador principal cuando se traza una línea.

- Registros de actualización de despliegue. Estos registros programan las direcciones de salida durante cada ciclo de refrescado; dichos registros almacenan las direcciones de los datos a ser desplegados, determinan como son incrementadas y especifican la localización de la memoria a ser desplegada en la esquina superior izquierda de la pantalla, es decir, la dirección de inicio de despliegue.

- Contadores de refrescado de DRAMS. El VSC puede ser programado para ejecutar ciclos de refrescado de DRAMS, los contadores seleccionan el número de ciclos de refrescado de DRAMS necesarios por línea.

CAPITULO IV

IV.1.1.2 INTERFAZ A MEMORIA DE VIDEO. -

a) "Multiplexor" de Direcciones. El "multiplexor" de direcciones proporciona un arreglo de memoria de video en forma de renglones y columnas. Sus entradas provienen del procesador principal, del registro de direccionamiento indirecto X-Y, del registro de despliegue de direcciones y del contador de direcciones de refresco de la memoria de video. La fuente de las direcciones es controlada por un arbitro de memoria. Los nueve bits de direcciones de salida MA proporcionan 18 señales de direcciones "multiplexadas".

b) Generador de Ciclos de Memoria. Ejecuta varios ciclos de memoria dependiendo de como lo necesite el arbitro de memoria y proporciona un arreglo con las señales de manejo de memoria: ^RAS, ^CAS, ^TR/^QE y ^W. Los ciclos generados por el Generador de Ciclos de Memoria se listan a continuación

1.- Ciclos pedidos por el procesador

- Accesos directos e indirectos aleatorios a memoria.
- Transferencia memoria-registro de corrimiento.
- Transferencia registro de corrimiento a memoria.

2.- Ciclos de retraso requeridos por el procesador.

3.- Ciclos de transferencia requeridos internamente memoria-registro de corrimiento.

4.- Ciclos de refrescado requeridos internamente.

CAPITULO IV

c) Control del TRC. El VSC genera las señales de ^HSYNC, ^VSYNC y ^BLANK necesarias para el manejo del TRC. Estas señales son programadas a través de ocho registros accesibles por el procesador, los cuales son fácilmente configurados para adecuarse a diferentes tipos de "Displays" ya sea de barrido entrelazado o no. Dos registros adicionales mantienen la cuenta horizontal y vertical. Los valores de estos dos contadores son comparados con los valores de los ocho registros de tiempo de video, para determinar los límites de los intervalos de sincronía y "blank".

IV.1.1.3 ARBITRO. -

La función del arbitro es determinar cuando tendrán acceso a memoria el procesador, cuando se recargaran los registros de corrimiento internos o cuando ocurrirá el ciclo de refrescado de las memorias dinámicas. Debido a que las funciones de despliegue y refrescado utilizan normalmente menos del 6% de los ciclos de memoria disponible, el arbitro otorga el acceso al procesador cuando el VSC no ejecute sus diversas funciones. Cuando un conflicto ocurre, el arbitro otorga prioridades como sigue:

- 1.- Cualquier ciclo en proceso.
- 2.- Ciclos de actualización de despliegue.
- 3.- Ciclos de refrescado de DRAM que se han retrasado por más de media línea horizontal.

CAPITULO IV

4.- Cualquier ciclo requerido por el procesador.

5.- Ciclo de refrescado de DRAM

El VSC ejecuta diferentes tipos de ciclos de acceso en su triple papel de controlador de DRAM, de actualizador de despliegue y actualizador de los tiempos del TRC. Algunos ciclos son inicializados por el procesador principal, otros son generados automáticamente por el VSC, como se lista a continuación.

1.-Ciclos inicializados por el procesador principal.

- Acceso del microprocesador.
- X-Y indirecto.
- Transferencia registro de corrimiento.
- Acceso a registro.

2.-Ciclos generados internamente.

- Refrescado de DRAMs.
- Actualización de despliegue.

Los ciclos de actualización de despliegue y transferencia de registro de corrimiento, mueven datos entre el arreglo de las celdas de memoria y el registro de corrimiento interno de las memorias, sin embargo, un ciclo de actualización de datos de despliegue es iniciado automáticamente por el VSC, mientras que cada ciclo de lectura o escritura del registro de corrimiento es inicializado bajo un control explicito del procesador

CAPITULO IV

principal.

Un ciclo es seleccionado por el microprocesador via las tres entradas de código selector de funciones FS2, FS1 y FS0, mostrado en la tabla 4.1

TABLA 4.1

FS2	FS1	FS0	FUNCION
0	0	0	Acceso a registro.
0	0	1	Indirecto X-Y.
0	1	0	Acceso a registro.
0	1	1	Acceso directo del procesador.
1	0	0	Registro de corrimiento a memoria.
1	0	1	Memoria a registro de corrimiento.
1	1	0	Reservado.
1	1	1	Reservado.

Para decodificar cada uno de estos ciclos dentro del megabyte direccionable, se estableció el siguiente mapa de memoria. (figura 4.2)

Los primeros 64 Kb de memoria (00000H-0FFFFH) están destinados al uso de memoria tipo estático y su función es la de alojar los vectores de interrupción en el primer KB (0000H-0BFFH) y la parte restante se usa para almacenar código o programas ejecutables y tablas de datos. En el caso de usar una configuración multiprocesador, ésta memoria se usa para el intercambio de mensajes de los demás

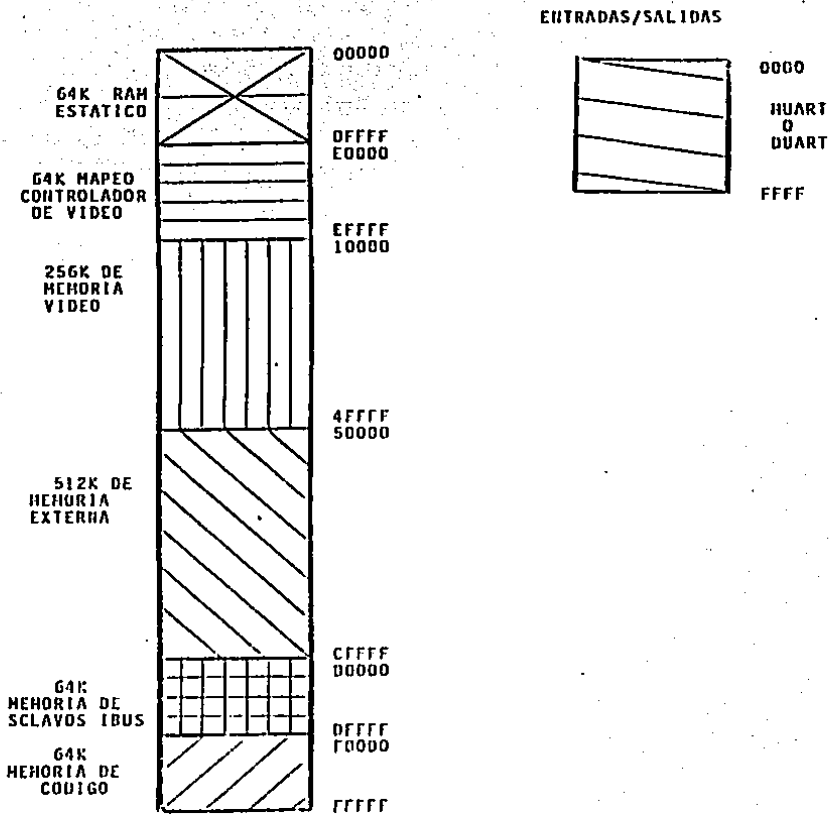


FIGURA A.2 MAPEO GENERAL DE LA TRAJETA.

CAPITULO IV

procesadores.

En la localidad 10000H-4FFFF se encuentra mapeada la memoria de video con un total de 256Kb de memoria visualizable, ésto implica que debido a que los "pixels" se encuentran formados de 4 bits por "pixel" se podrían desplegar hasta 512K "pixels" de información. En coordenadas cartesianas, ésto nos representa una memoria de video de 1024x512 pixels.

En la región mapeada de 50000H-DFFFFH se encuentra un área para alojar memoria externa a la tarjeta del procesador. Esta memoria es utilizada por todos los procesadores maestros que comparten el bus. Los maestros que no tengan capacidad gráfica, escriben aquí la información o procesos que deben ser visualizados para que la maestra gráfica los lea y despliegue.

La parte de memoria ubicada entre E0000H-EFFFFH, se utiliza para mapear los registros internos del TM41061, ésto se tratará posteriormente.

Por último la región entre F0000H y FFFFFH es área de código y la intención es la de almacenar funciones gráficas o primitivas que sean utilizadas de manera exhaustiva, así como un monitor o programa de utilería que nos permita ejecutar programas básicos como leer y escribir a memoria, buscar palabras o bytes específicos, mover bloque de información, ejecutar operaciones hexadecimales, etc.

CAPITULO IV

La parte de entrada y salida, que consta de 64000 bytes se utiliza en nuestro caso solamente para manejar dos circuitos perifericos que son, un circuito integrado multifunción MUART (U14) que tiene integrado un Transmisor-Receptor Universal Asíncrono (UART) que permite comunicación serie con cualquier terminal, dos puertos bidireccionales de ocho bits cada uno y un puerto para generar interrupciones.

El otro circuito integrado que comparte el área de entrada salida es un controlador de interrupciones programable (PIC)(U15) con capacidad para aceptar hasta ocho fuentes de interrupciones modo sencillo o 64 modo cascada. (Para manejar información ver el capítulo III)

IV.1.2 DECODIFICACION GENERAL.

La integración de un procesador general (en este caso el microprocesador 8086 de INTEL) junto con el controlador de video (TMS34061)(U56) y la memoria de video asociada (TMS4161) requiere de cierta lógica de interfaz entre una y otra. Este conjunto resultante es nuestro sistema gráfico deseado.

Como hemos podido observar, tanto el controlador como la memoria han sido diseñados para trabajar en concordancia; luego entonces su integración es bastante simple. En el caso del procesador general y el controlador de video, éstos

CAPITULO IV

aparte de ser de familias diferentes (INTEL Y TEXAS INSTRUMENTS) necesitan una pequeña interfaz. Esta interfaz tiene que ver tanto con el mapeo de memoria del procesador general como con las funciones de enlace con el controlador VSC.

En el caso de la interacción, procesador general-memoria de video, existe una interfaz para seleccionar bloques de "pixels" o simplemente "pixels" individuales, como se verá posteriormente.

IV.1.2.1 DECODIFICACION DE LA MEMORIA DEL SISTEMA.(EPLD) (14)

Para la decodificación general del sistema se usa un circuito llamado EPLD (Erasable Prograble Logic Device). Este circuito permite diseñar arreglos lógicos combinacionales. El uso de la lógica combinacional permite la implementación de circuitos de propósito específico en muy corto tiempo, en base a dos funciones lógicas Y/O (AND/OR), con la posibilidad de poder borrar los arreglos de la misma forma que se borra una memoria EPROM, y poder usarlo para otro arreglo diferente, ésto le da aún más versatilidad al circuito.

La arquitectura interna de los EPLD se puede dividir en cinco bloques funcionales:

CAPITULO IV

- a) Entrada de datos.
- b) Entrada de reloj.
- c) Arreglos programables Y/O.
- d) Salidas.
- e) Control de funciones para retroalimentación.

Los diseños lógicos son programados en los EPLDs por medio de un paquete de programación especial de ALTERA. (15)

Los EPLDs son programados o "quemados" en base a ecuaciones booleanas predefinidas, las cuales dependen del tipo y características del arreglo que se quiera obtener. Nosotros deseamos decodificar de manera general el sistema dependiendo básicamente del mapeo que se hizo a memoria. (ver figura 4.2) El EPLD elegido es un EP310 de la familia ALTERA, que posee 10 entradas y 8 entradas/salidas internamente este circuito está dividido en ocho macroceldas, una sola macrocelda puede sumar lógicamente (OR) ocho productos de términos. Cada macrocelda tiene nueve términos para productos y un "buffer" de tercer estado que maneja las líneas de E/S. Un arreglo de 36 entradas Y (AND) son generadas con las 10 entradas y las 8 salidas. Los arreglos de las macroceldas no son independientes unos de otros, cada señal de entrada y de retroalimentación está disponible para los productos de términos.

CAPITULO IV

Con todo lo anterior podemos entender la forma en que se hace la programación, ahora procederemos a explicar la forma en que son establecidas las ecuaciones en base a la decodificación de memoria, lo que se puede mostrar con ayuda de la tabla 4.2.

TABLA 4.2

A19	A18	A17	A16	A15	Función	Dirección
0	0	0	0		Selecciona RAM	0XXXX
1	1	1	1		Selecciona ROM	FXXXX
0	0	0	1		Selecciona	1XXXX
0	0	1	0		memoria de Video	2XXXX
0	0	1	1			3XXXX
0	1	0	0			4XXXX
1	1	1	0		Mapeo de contro- lador de video	EXXXX
				0	Selecciona MUART	0XXX 7XXX
				1	Selecciona PIC	8XXX FXXX

Este circuito EP310 (U55)(PLANO 1,2) se aprovecho también para señales de habilitación del Bus Interno CENI, señales de interrupción, señales de lectura y escritura (referirse a la tabla 4.1) y señal de habilitación de datos.

CAPITULO IV

Las ecuaciones finales están dadas en el PROGRAMA A.
VER APENDICE 2. (16)

IV.1.2.2 DECODIFICACION DE FUNCIONES DEL VSC. -

Para lograr la decodificación adecuada de las funciones del VSC, se utilizó otro arreglo lógico programable (EP310)(U19)(PLANO 2,2) EPLD B. Las señales FS0, FS1 y FS2 se generan a través de las direcciones de entrada A9-A19, por medio de un conjunto de ecuaciones congruentes con el mapeo de memoria anteriormente establecido. Esto se describe a continuación de manera detallada.

Para mostrar en forma clara la relación entre las funciones, y su localización en memoria fué creada la tabla siguiente. (4.3)

TABLA 4.3

RANGO DE MEMORIA	FUNCION TMS3401	FS2	FS1	FS0
10000H-4FFFFH	Acceso directo del 8086 a memoria de video.	0	1	1
E03F0H-E03FFH	Modo control paleta no accesa TMS34061.	1	1	1
E0400H-E07FFH	Transferir registro de corrimiento a memoria.	1	0	0
E0800H-E0BFFH	Transferir memoria a registro de			

CAPITULO IV

	corrimiento.			
E0C00H-E00FFH	Registros internos delTMS34061.	0	0	0
E0E00H-E0EFFH	Acceso indirecto a memoria de video "pixel" par.	0	0	1
E1E00H-E1EFFH	Acceso indirecto a memoria de video "pixel" non.	0	0	1
E2E00H-E2EFFH	Acceso indirecto a memoria de video byte.	0	0	1

De esta tabla donde se muestran las combinaciones de FS0, FS1 y FS2 asociadas a las funciones y a los rangos de memoria deseados, las ecuaciones para el arreglo lógico, pueden ser obtenidas con facilidad.

Lo anterior se muestra en el cuadro siguiente:(tabla 4.4)

TABLA 4.4

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	FS0
1	1	1	0	0	0	0	0	0	1			E04XX-E07XX
1	1	1	0	0	0	0	0	1	1	0		E0CXX-E0DXX
												FS1
1	1	1	0	0	0	0	0	0	1			E04XX-E07XX

CAPITULO IV

1	1	1	0	0	0	0	0	1	0	E08XX-E0BXX	
1	1	1	0	0	0	0	0	1	1	0	E0BXX-E0DXX
1	1	1	0	0	0	0	0	1	1	1	E0DXX-E0FXX
1	1	1	0	0	0	0	1	1	1	1	E1DXX-E1FXX
1	1	1	0	0	0	1	0	1	1	1	E2EXX-E2FXX
											FS2
1	1	1	0	0	0	0	0	1	1	0	E0CXX-E0DXX
1	1	1	0	0	0	0	0	1	1	1	E0EXX-E0FXX
1	1	1	0	0	0	0	1	1	1	1	E1EXX-E1FXX
1	1	1	0	0	0	1	0	1	1	1	E2EXX-E2FXX
0	0	0	1								1XXXX
0	0	1	0								2XXXX
0	0	1	1								3XXXX
0	1	0	0								4XXXX
											PALETA
1	1	1	0	0	0	0	0	0	0	1	E02XX-E03XX
											IND0
1	1	1	0	0	0	0	0	1	1	1	E0EXX-E0FXX
1	1	1	0	0	0	1	0	1	1	1	E1EXX-E1FXX
											IND1
1	1	1	0	0	0	0	0	1	1	1	E0EXX-E0FXX
1	1	1	0	0	0	0	1	1	1	1	E1EXX-E1FXX

Las ecuaciones definitivas se muestran en el PROGRAMA B. VER APENDICE 2.

CAPITULO IV

La señal de video requerida para que el controlador de video acepte las señales de entrada RSO, RS1, RA0-RA3, FS0-FS2 como válidas se denomina ALEPAL. Cuando esta señal sufra una transición de alto a bajo, el controlador de video amarrará las señales de entrada por medio de un grupo de "latches" y dará comienzo un ciclo de máquina de video definida por cualquiera de las combinaciones válidas de FS0-FS2 (ver tabla 4.3).

ALEPAL se genera a partir de una ecuación determinada dentro de otro EPLD el EPLD C (U21)(PLANO 2,2) y se configura por 5 minterminos con las posibles combinaciones de FS0-FS2 y una señal de validación llamada CMD, esta última formada por dos términos que son escritura o lectura de memoria. La señal CMD también cumple con el cometido de establecer los tiempos de preestablecimiento y postestablecimiento con respecto a la señal de reloj del 8086 ya que se debe cumplir con los tiempos de sincronía establecidos por el fabricante y que se muestran en la siguiente figura (figura 4.3.)

CEH-CEL

Para complementar la interfaz Procesador-Controlador el EPLD genera las señales de CEH y CEL, las cuales deben ser sincrónicas con las señales de reloj del 8086 ya que sirven para establecer la longitud de la palabra que va a ser accesada. Dependiendo de la combinación de ambas señales

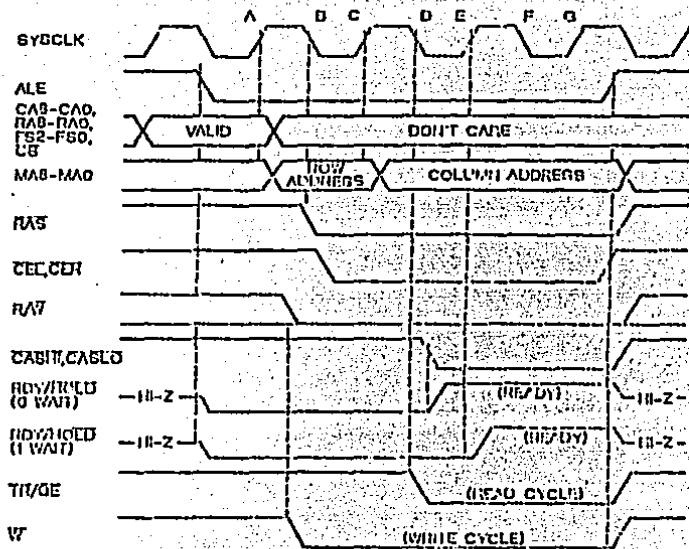


FIGURA 4.3 DIAGRAMA DE TIEMPOS DEL CONTROLADOR.

CAPITULO IV

será la selección, como se muestra en la tabla 4.5)

TABLA 4.5

CEH	CEL	
0	0	16 bits.
0	1	8 bits parte alta.
1	0	8 bits parte baja.
1	1	no acceso.

Este EPLD también genera A16PAL Y A17PAL que tienen la función de remapear las direcciones A16-A17 del "buffer" de video que comienza en la dirección 10000H a la cero lógica, para que el VSC vea las páginas 10, 20, 30, 40 como 0, 10, 20 y 30; en la tabla 4.6 se ve más claramente.

TABLA 4.6

A19	A18	A17	A16	A17PAL	A16PAL
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	1	1
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0

CAPITULO IV

Siguiendo el mismo procedimiento se pueden obtener las ecuaciones de las nuevas direcciones. Con toda la información anterior podemos establecer ya las ecuaciones del EPLD C que se muestran el PROGRAMA C. VER APENDICE 2.

IV.1.2.3 DECODIFICACION DE PIXEL.

Otra parte de la interfaz procesador-memoria la obtenemos del EPLD D (U20)(PLANO 2,2). Este EPLD se refiere a la selección de las memorias de video P0, P1, P2, P3, P4, P5, P6 Y P7 (ya sea en forma individual o en algún arreglo especial. (VER PLANO 3,2)

Todas la habilitaciones a memorias deben estar condicionadas a la señal de ALEPAL y por una señal llamada TREG, que indique que no hay transferencia de registro de corrimiento, siempre y cuando no se deseen seleccionar todas las memorias a un solo tiempo, en este caso, TREG quedará habilitada.

Existen dos señales para acceso adecuado a la o las memorias deseadas, que serian las señales ^IND0 e ^IND1, obtenidas previamente por el EPLD B dependiendo de la localidad de memoria. Las combinaciones de ambas señales manipulan los accesos a memoria como se define en la tabla 4.7.

CAPITULO IV

INDO	IND1	TIPO DE ACCESO
0	0	Individual indirecto por P0 P4 P2 P6
0	1	Byte o palabra Individual.
1	0	Individual indirecto non P1 P3 P5 P7
1	1	Directo.

Las señales de CASH y CASL dependiendo de que parte de la memoria se desee acceder alta o baja, son importantes para la selección de las memorias. Al indica si se está trabajando con la parte alta o baja de la palabra.

En nuestro sistema tenemos un arreglo de 8 memorias donde "P" significa cada una de ellas (ver plano 3.2) como se muestra en la figura 4.4. De acuerdo a ésto formularémos una tabla con las posibles combinaciones de seleccionamiento de las mismas como se indica a continuación en la tabla 4.8.

TABLA 4.8

	CAS	CAS	CAS	CAS	CAS	CAS	CAS	CAS	IND	IND	CAS	CAS	Al
	0	1	2	3	4	5	6	7	1	0	H	L	
P0+P4	0	1	1	1	0	1	1	1	0	0	1	0	X
P2+P6	1	1	0	1	1	1	0	1	0	0	0	1	X

CAPITULO IV

P1+P5	1	0	1	1	1	0	1	1	0	1	1	0	X
P3+P7	1	1	1	0	1	1	1	0	0	1	0	1	X
P0P1+P4P5	0	0	1	1	0	0	1	1	1	0	1	0	X
P2P3+P6P7	1	1	0	0	1	1	0	0	1	0	0	1	X
P0P1P2P3+													
P4P5P6P7	0	0	0	0	0	0	0	0	1	0	0	0	X
P0P1	0	0	1	1	1	1	1	1	1	1	1	0	0
P2P3	1	1	0	0	1	1	1	1	1	1	0	1	0
P4P5	1	1	1	1	0	0	1	1	1	1	1	0	1
P6P7	1	1	1	1	1	1	0	0	1	1	0	1	1
P0P1P2P3	0	0	0	0	1	1	1	1	1	1	0	0	0
P4P5P6P7	1	1	1	1	0	0	0	0	1	1	0	0	1
P0P1P2P3*													
P4P5P6P7	0	0	0	0	0	0	0	0	*	*	*	*	*

Una señal se ^ALEPAL se utiliza para asegurar que las salidas del EPLD D se encuentren flotadas mientras no se detecte una caída en dicha señal.

Las ecuaciones del EPLD D se muestran en el PROGRAMA D. VER APENDICE 2.

Después de generar las señales que habilitan las memorias, procederemos a la descripción de éstas.

IV.2 MEMORIAS DE VIDEO. (TMS4161)(U24-U31)(PLANO 3,2).

El fin de la memoria de video es el de almacenar la

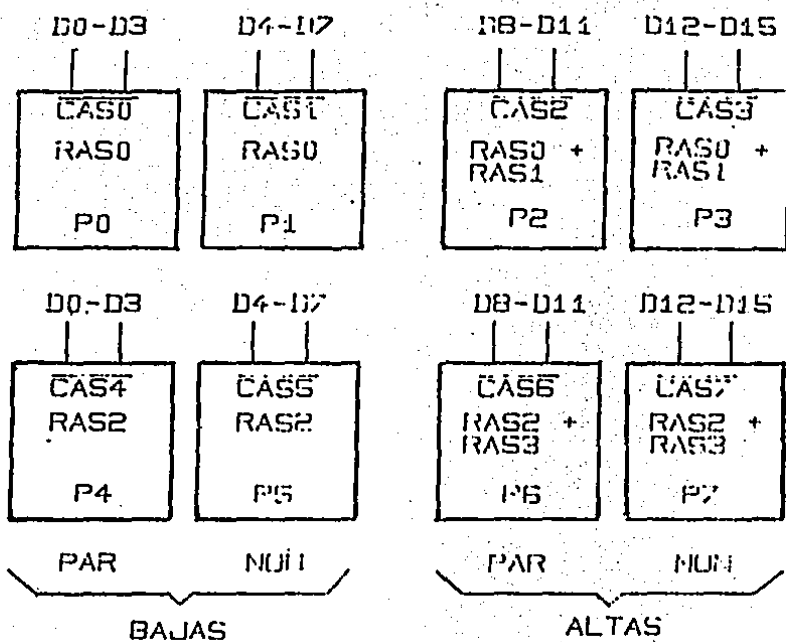


FIGURA 4.4 ARREGLO DE MEMORIA.

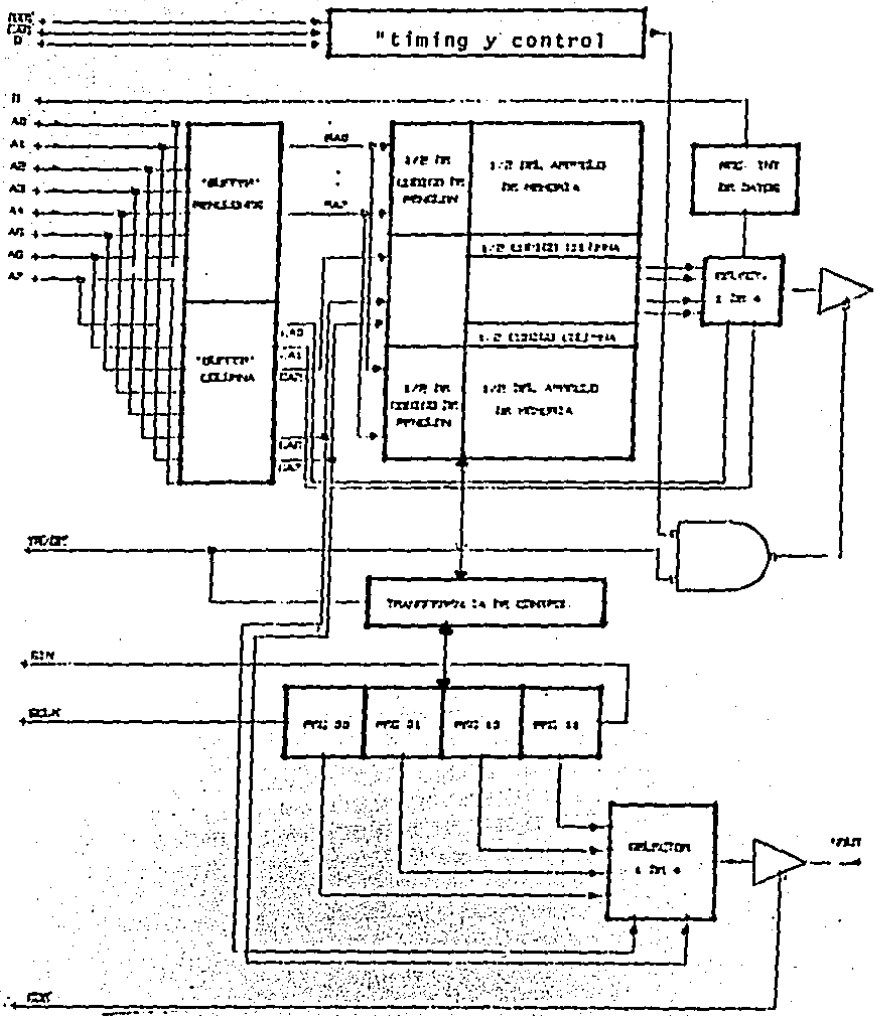


FIG. 4.5' ARQUITECTURA INTERNA DE LA MEMORIA VIDEO..

CAPITULO IV

información que va a ser desplegada en la pantalla. En nuestro caso la memoria de video se encuentra mapeada de la dirección 10000H a la 4FFFFH. Esta memoria consta de ocho tiras tipo TMS4161, dichas tiras son memorias dinámicas de alta velocidad, que pueden ser accedadas en forma aleatoria o secuencial, con capacidad de 64K cada una. Cada memoria posee cuatro registros de corrimiento internos de 64 bits conectados en cascada.

Los ciclos de refrescado tienen una duración de 4 mseg, durante la cual los 256 renglones contenidos en memoria, deben ser activados con la señal de $\bar{R}AS$ para retener el dato. $\bar{R}AS$ debe permanecer en estado alto durante el ciclo de refrescado.

Ver diagrama funcional a bloques de una memoria fig 4.5.

La memoria TMS4161 está diseñada con cada renglón dividido en cuatro secciones de 64 columnas, la primera sección de columnas para ser recorrida se selecciona con las dos direcciones superiores de las direcciones de columnas. Si los dos bits representan 00, del primero al cuarto registro serán recorridos en orden, si son 01 solamente del primer al tercer registro, si ambos bits representan un 10 solo serán corridos el primero y el segundo, y si son 11 solo el registro más significativo será recorrido. Seleccionando adecuadamente estos registros internos,

CAPITULO IV

podemos seleccionar también el inicio del despliegue de la memoria de video.

IV.2.1 ACCESO ALEATORIO.(ver Figura 4.6)

La línea $\overline{\text{TR}}/\overline{\text{QE}}$ posee dos funciones, una es la de seleccionar cualquier transferencia a registro y la otra es la de seleccionar acceso aleatorio.

Para usar la TMS4161 en modo de acceso aleatorio, la señal $\overline{\text{TR}}/\overline{\text{QE}}$ debe permanecer en estado alto cuando $\overline{\text{RAS}}$ cae, si dicha línea está en bajo cuando $\overline{\text{RAS}}$ cae, quedan conectados los registros de corrimiento y se indica que una transferencia ocurrirá entre un registro y un renglón del arreglo de memoria.

Para poder direccionar los 64K de memoria son necesarias 16 líneas, 8 direcciones de renglón (provenientes de MA0-MA7 del VSC) que son colocadas en las entradas A0-A7 y seleccionadas en el circuito por la señal $\overline{\text{RAS}}$, después las ocho direcciones de columnas son colocadas en las mismas entradas A0-A7 pero seleccionadas por la señal $\overline{\text{CAS}}$. La señal de $\overline{\text{W}}$ habilita los modos de ESCRITURA si la señal es baja, y de LECTURA si la señal es alta. La entrada de los datos está deshabilitada cuando el modo LECTURA se selecciona.

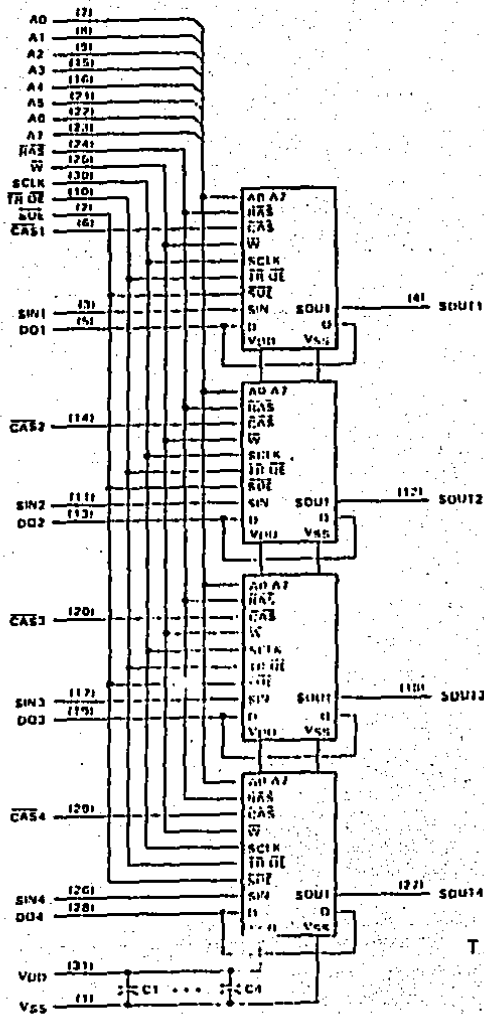


FIGURA 4.6
TIRA DE MEMORIA.

CAPITULO IV

Los datos son escritos durante el ciclo de escritura, la caída de $\overline{\text{CAS}}$ o $\overline{\text{W}}$ activa los datos en el "LATCH" de datos, si $\overline{\text{CAS}}$ y $\overline{\text{TR}}/\overline{\text{QE}}$ se mantienen en alto, el dato no aparecerá en la salida, hasta que ambas señales estén abajo.

En un ciclo de escritura y de transferencia de registro, las salidas permanecerán en alta impedancia. $\overline{\text{RAS}}$ evita cualquier salida durante el refrescado.

IV.2.2 ACCESO SECUENCIAL.

En el acceso secuencial, la línea $\overline{\text{W}}$ determina si se efectúa una transferencia desde los registros al arreglo de memoria o de memoria a registro, cuando $\overline{\text{W}}$ es baja y $\overline{\text{RAS}}$ cae la transferencia es registro a memoria si $\overline{\text{W}}$ es alta y $\overline{\text{RAS}}$ cae la transferencia es de memoria a registro. Por lo tanto lecturas y escrituras son siempre con respecto al arreglo de memoria.

Ocho bits de direcciones son necesarios para seleccionar uno de los 256 posibles renglones, $\overline{\text{A0-A7}}$, $\overline{\text{W}}$, $\overline{\text{TR}}/\overline{\text{QE}}$ son "latcheados" en cada $\overline{\text{RAS}}$, para poder seleccionar uno de los cuatro registros de corrimiento, las direcciones más altas ($\overline{\text{A6-A7}}$) de columnas, deben ser válidas cuando $\overline{\text{CAS}}$ cae. Las señales de $\overline{\text{CAS}}$ y las de las direcciones de registro no son necesarias todo el tiempo, únicamente cuando se desee un cambio o seleccionar uno de los cuatro registros internos.

CAPITULO IV

Los datos son recorridos durante el flanco de subida de SCLK, los registros de corrimiento son también elementos dinámicos de almacenamiento, el dato contenido en los registros se perderá a menos que SCLK suba para recorrerlo una posición, que exista una operación de transferencia de escritura o que un dato sea recargado desde el arreglo de memoria.

IV.3 PALETA DE COLORES. (17)

Es importante hacer un diagrama a bloques de la interfaz entre VSC y PALETA. Ver figura 4.7.

Las memorias de video se encuentran divididas en dos grupos de cuatro memorias cada uno, un grupo de pares y otro de impares, P0 Y P4 comparten líneas de datos D0-D3, P2 y P6 comparten las líneas D8-D11, P1 y P5 comparten D4-D7 y P3 y P7 comparten D12-D15. Las señales de direcciones, ^W, ^TR/^QE, provenientes del VSC son comunes a las ocho memorias. Señales de ^RAS también son obtenidas del VSC. Señales de ^CAS provienen del EPLD D como ya se dijo en la sección IV.1.2.3, cada una de estas señales es diferente para cada memoria.

Las memorias suministran información a ocho registros de corrimiento externos (74SN197)(U32-U39)(PLANO 3,3), que se encuentran ordenados en el mismo arreglo de las memorias, dichos registros transforman la información de paralelo a

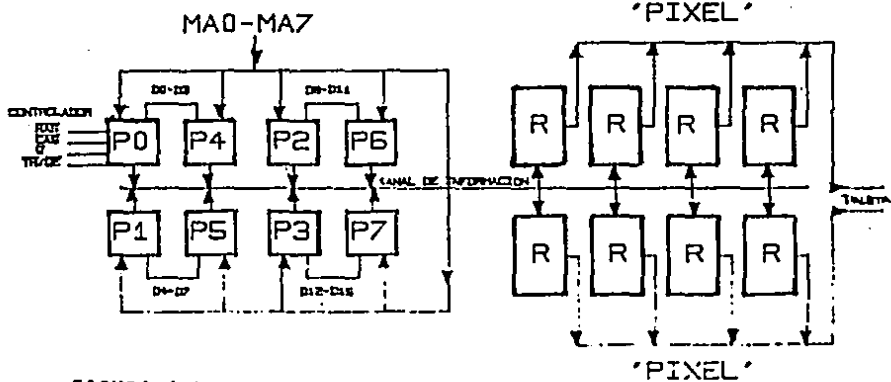


FIGURA 4.7 INTERFAZ CONTROLADOR-PALETA

CAPITULO IV

serie, la información de cuatro registros forman un "pixel" par, los otros cuatro registros generan un "pixel" non. La señal de frecuencia a la que la información es desplegada, es suministrada por la PALETA DE COLORES.

El circuito TMS34070 es un dispositivo, cuya función es la de transformar entradas de información digitales y obtener salidas de video RGB.

El monitor de despliegue usa las tres salidas analógicas de la PALETA para manejar los amplificadores conectados a las entradas (rojo, verde y azul). Cada salida analógica es manejada por un convertidor digital-analógico (DAC) contenido en el mismo circuito TMS34070. Cada DAC proporciona 16 niveles de intensidad en su respectivo color rojo, verde o azul lo que puede producir 4096 posibles tonalidades, aunque solo 16 son capaces de ser desplegados simultáneamente. Los registros de la tabla de colores son cargados directamente desde la memoria de video, dicha tabla posee 16 registros con 14 bits, 12 bits definen los colores y dos bits son atributos para casos especiales, los 12 primeros bits se dividen en cuatro bits para rojo, cuatro bits para azul y cuatro para verde.

IV.3.1 ARQUITECTURA INTERNA DE LA PALETA.

Dos grupos de 4 bits ("pixel") son introducidos en el dispositivo en la entradas DA3-DA0 y DB3-DB0, cada "pixel"

CAPITULO IV

es interpretado como un apuntador en los 16 registros de la tabla de colores, el contenido del registro seleccionado es usado para manejar los DAC de las salidas rojo, verde y azul.

Un reloj señal de "DOTCLK" es alimentado al circuito. El DOTCLK representa el tiempo durante el cual un solo "pixel" es desplegado en el monitor de video. La señal de "CLKOUT" de salida es una señal de la mitad del valor de frecuencia de DOTCLK, la cual sirve para alimentar circuitería adicional.

Cada par de valores de "pixel", representan dos puntos adyacentes en la pantalla del monitor. El "pixel" de una dirección par es típicamente DA3-DA0 y es desplegado a la izquierda del "pixel" DB3-DB0.

IV.3.2 OPERACION DE INTERFAZ MEMORIAS DE VIDEO-PALETA.

El circuito TMS34070 carga su tabla de colores automáticamente del área de memoria de video donde ha sido cargada previamente; ésto se hace sin necesidad de la intervención del procesador. La señal de entrada a la paleta DATEN, se usa como una señal de control empleada para habilitar la entrada del dato a DA0-3 y DB0-3. La señal DATEN es típicamente manejada por la señal de BLANK del sistema. En nuestro caso esta señal es manejada directamente por el controlador VSC. La señal de MODE

CAPITULO IV

configura el dispositivo para diversos tipos de operación en el cargado de información.

La paleta de colores tiene dos formas de realizar el cargado de la información: Cargado por Sistema y Cargado por Comando.

Cargado por Sistema. La tabla es cargada automáticamente antes de empezar cada trazo del monitor, en este caso, el dato cargado en la tabla, es almacenado al principio de la línea 0 de la localidad de memoria, dicha información no es desplegable. Las líneas de la 1 en adelante que son cargadas, contienen información de los "pixels".

Otra forma de cargar automáticamente por sistema es a nivel de línea por línea. Si usamos el método de la carga de línea, la tabla es cargada al principio de cada línea de información. En este caso el dato de la paleta es cargado pero no desplegado.

Cargado de comando: Cuando la aplicación requiera que los 16 registros de la tabla de colores sean cargados únicamente durante arreglos seleccionados o especiales, se utiliza el método de cargado por comando. después de dicho comando los colores permanecerán como están hasta otro nuevo comando de cargado.

CAPITULO IV

IV.3.3 INTERFAZ AL MONITOR.

Por último, las tres salidas analógicas del dispositivo son capaces de manejar directamente una carga de 75 ohms, sin embargo en algunos casos, las salidas analógicas no cumplen con el voltaje adecuado o la impedancia requerida, dicha salida puede ser adecuada a las necesidades de impedancia del monitor con un arreglo de resistencias en forma PI, como se muestra en el plano (3,3) donde las tres salidas (rojo, verde y azul) y la señal de sincronía BLANK requirieron de acoplamiento de impedancias.

C A P I T U L O V

CAPITULO V

PRUEBAS Y UTILIZACION DEL SISTEMA.

V.1 PRUEBAS AL "HARDWARE".

Después de tratar lo concerniente al diseño del sistema gráfico es interesante dar una breve introducción de las pruebas realizadas al sistema desde el punto de vista "software" y que nos validaron, en consecuencia el "hardware" asociado.

a) Inicialización del controlador.

b) Accesos a Memoria de Video.

c) Aplicaciones simples:

Trazo de caracteres.

Trazo de paleta de colores.

Trazo de línea.

CAPITULO V

V.1.1 INICIALIZACION DEL CONTROLADOR.

Como ya se mencionó en capitulos anteriores, el controlador tiene 18 registros programables de 16 bits que pueden ser escritos o leídos por el procesador principal. Existen también registros internos a los cuales el procesador principal no tiene acceso.

Los registros deben ser cargados, para que el controlador de video pueda cumplir con sus tareas. Los registros se dividen en 3 grupos dependiendo de su función, comunicación con procesador general, control del CRT y control memoria de video.

V.1.1.1 REGISTROS DE COMUNICACION CON EL PROCESADOR GENERAL. -

A) REGISTROS DE CONTROL 1 Y 2.

El VSC contiene dos registros de control que son programados por el procesador principal para configurar la operación del VSC. Ambos registros pueden ser leídos o escritos por el procesador principal.

Las funciones que son controladas por los registros son:

- Habilitar las peticiones de interrupción.
- Configurar los tiempos de las funciones de video.
- Establecer las señales de comunicación con el

CAPITULO V

procesador principal.

- Ciclos de refrescado de pantalla.
- Ciclos de refrescado de las DRAMs.

B) REGISTROS DE ESTADOS.

Estos registros contienen tres bits que determinan diferentes funciones como lo mostramos en la tabla 5.1

TABLA 5.1

BIT	FUNCION
B0	Interrupción vertical. Uno en este bit indica que una interrupción vertical ha ocurrido.
B1	Error en el despliegue. Este bit en uno indica que el VSC está inhabilitado para ejecutar ciclos de petición de actualización del monitor durante el intervalo de "Blank" horizontal.
B2	Error en el refrescado. Uno en este bit indica que el VSC está inhabilitado para la ejecución de ciclos de refrescado antes de que ocurra el siguiente "blank" horizontal.

Los registros de estados pueden ser leídos pero no escritos por el procesador principal.

CAPITULO V

C) REGISTROS DE INTERRUPCION VERTICAL.

Estos registros sirven para almacenar información que será comparada con un contador vertical para que el VSC mande peticiones de interrupción al procesador principal mediante la línea ^INT.

V.1.1.2 REGISTROS DE CONTROL DEL TRC. --

Sirven para programar los tiempos de las señales de sincronía horizontal y vertical del barrido de pantalla, además nos proporciona los números de punto por línea, número de líneas y direcciones de despliegue.

En nuestro caso contamos con un monitor MITSUBISHI C-3910 que presenta las siguientes características:(11)

Frecuencia horizontal 40-70 Hz.

Frecuencia vertical 15-18 KHz.

Ancho de banda 25 MHz.

Tiempo de "blank" horizontal 9 microseg.

Tiempo de "blank" vertical 0.7 mseg.

En base a estos datos se realizará la inicialización del controlador como se verá posteriormente.

CAPITULO V

A) REGISTROS DE ACTUALIZACION DEL MONITOR.

Los datos de video transmitidos al TRC para el trazo de una línea son transmitidos del arreglo de memoria al registro interno durante el periodo de "blank" horizontal que precede a la línea. La transferencia es ejecutada automáticamente por el VSC. La dirección de salida durante cada ciclo de actualización de memoria se establece mediante estos registros.

B) REGISTROS DE TIEMPOS DE VIDEO.

Tiempos de sincronía horizontal vertical y "blank" son generados con ayuda de un contador de líneas, para determinar límites en los periodos de sincronía horizontal, de principio y fin del "blank" horizontal, duración de trama horizontal total, final de la sincronía vertical, principio y fin de la señal de "blank" vertical, así como duración total de la sincronía vertical.

V.1.1.3 REGISTROS DE CONTROL DE ACCESO A MEMORIA. -

Estos son registros que agilizan el acceso a ciertas localidades de memoria. Desde el punto de vista del procesador general, los "pixels" de la memoria de video pueden ser accesados por dos métodos.

CAPITULO V

1.- Método directo. Mapeando la dirección X-Y del "pixel" a la dirección del "byte" correspondiente en el mapa de memoria

2.- Utilizar los registros especiales del controlador de video. Estos registros determinan un cierto ajuste en la dirección inicial programada. Por ejemplo existe un registro para incrementar la dirección en X, sin afectar la dirección en Y. (0001). Los tipos de ajustes serán mostrados posteriormente. (tabla 4.2.)

DIRECCIONAMIENTO X-Y.

El direccionamiento X-Y es útil en aplicaciones en las cuales el direccionamiento lineal del procesador principal es limitado para acceder de manera fácil todos los "pixels" en el área activa del monitor. Un ciclo de lectura o escritura a memoria que utiliza los registros X-Y se llama ciclo indirecto X-Y.

Durante el ciclo indirecto X-Y, 15 diferentes ajustes están disponibles

TABLA 4.2

ENTRADAS				FUNCION X-Y DE AJUSTE
CA4	CA3	CA2	CA1	
0	0	0	0	Sin ajuste
0	0	0	1	Incrementa X
0	0	1	0	Decrementa X

CAPITULO V

0	0	1	1	Limpia X	
0	1	0	0		Incrementa Y
0	1	0	1	Incrementa X	Incrementa Y
0	1	1	0	Decrementa X	Incrementa Y
0	1	1	1	Limpia X	Incrementa Y
1	0	0	0		Decrementa Y
1	0	0	1	Incrementa X	Decrementa Y
1	0	1	0	Decrementa X	Decrementa Y
1	0	1	1	Limpia X	Decrementa Y
1	1	0	0		Limpia Y
1	1	0	1	Incrementa X	Limpia Y
1	1	1	0	Decrementa X	Limpia Y
1	1	1	1	Limpia X	Limpia Y

Donde los contenidos de los registros X-Y son usados en lugar de direcciones de renglones y columnas sustituyendo los RA8-RA0 y CA8-CA0 por CA4-CA1. Los cuatro bits de código de entrada CA4-CA1 determinan la forma en la cual los registros X-Y son direccionados. Los ajustes se logran accediendo las direcciones correspondientes en el mapa de memoria.

V.2 ALGORITMOS DE PROGRAMACION.

V.2.1 INICIALIZACION DEL CONTROLADOR.

Para la inicialización del controlador es necesario:

CAPITULO V

- A) Programar registros de comunicación.
- B) Programar registros de estados.
- C) Programar registros de control del TRC
con respecto al monitor. (ver V.1.1.2) (18)

-Video no-entrelazado.

Se escogió video no-entrelazado ya que se desea evitar al máximo el parpadeo de la imagen que se presenta en el video entrelazado, ya que es molesto al observador y le cansa la vista.

-Frecuencia vertical escogida es de 50 Hz pues por lo general se utiliza en video una frecuencia de 50 ó 60 Hz.

Teniendo la frecuencia vertical determinaremos el tiempo vertical.

$$1/50\text{Hz.} = 20 \text{ mseg.}$$

-Frecuencia horizontal escogida es 18 KHz. debido a que se desea obtener una frecuencia determinada para el controlador de video que depende directamente de la frecuencia horizontal. En base a esta frecuencia se determinará el tiempo para cada línea horizontal.

CAPITULO V

$$1/18 \text{ KHz.} = 55 \text{ microseg.}$$

-El tiempo activo horizontal para cada línea se calcula restando el tiempo de barrido horizontal menos el tiempo de blanking horizontal dado en las especificaciones del monitor.

$$(55-9) \text{ microseg.} = 46 \text{ microseg.}$$

-El número de líneas se determina dividiendo la frecuencia horizontal entre la frecuencia vertical

$$18\text{K}/50 = 360$$

-Tiempo mínimo de duración de un "pixel" lo obtenemos tomando el máximo valor en el ancho de banda que es 25 MHz.

$$1/25\text{MHz.} = 40 \text{ nseg.}$$

-Número máximo de "pixels" por línea lo obtenemos dividiendo el tiempo de cada línea entre lo que tarda cada "pixel".

$$46\text{microseg.}/40\text{nseg.} = 1150 \text{ "pixels"}$$

CAPITULO V

-Debido a que nosotros escogimos un cristal oscilador que no genera 25MHz., sino 12.6MHz ya que a partir de él serán generadas todas las señales de reloj para la paleta(12.6MHz), los registros(6.3MHz.), las memorias de video y el controlador(1.57MHz.), es necesario recalcular el número de "pixels" horizontales que nosotros manejaremos.

Nuevo tiempo de duración de un "pixel".

$$1/12.6\text{MHz.} = 80\text{nseg.}$$

-El número de "pixels" por renglón será.

$$46\text{microseg}/80\text{nseg} = 560 \text{ "pixels"}$$

V.3 APLICACIONES SIMPLES.

Por el método indirecto podemos acceder un "pixel" determinado. Partiendo de éste método podemos generar pequeños programas de aplicación que nos muestren la capacidad del sistema. Tal es el caso de la generación de caracteres, del trazo de líneas y trazo de la paleta de colores.

V.3.1 GENERACION DE CARACTERES.

Para la generación de caracteres partiremos de la decodificación de cada caracter utilizando una matriz de puntos de 8x9 para generar todos los caracteres

CAPITULO V

alfanuméricos, los cuales pueden ser accedados en el momento requerido, ya que se encuentran almacenados en un espacio de memoria previamente asignada para este fin. como por ejemplo: Generemos la letra A, primeramente es necesario hacer la codificación de la letra o caracter elegido como se muestra a continuación.

```

- - X X X X X - ... 3E
- X - - - - - X ... 41
- X - - - - - X ... 41
- X - - - - - X ... 41
- X X X X X X X ... 7F
- X - - - - - X ... 41
- X - - - - - X ... 41
- X - - - - - X ... 41
- - - - - - - ... 00
  
```

El fin de que se codifique de ésta manera es que el lugar el caracter ocupa en la memoria sea de 9 "bytes" únicamente, si nosotros almacenáramos la información de cada punto por separado la cantidad de memoria utilizada sería de 56 "bytes". De ésta manera el espacio de memoria se optimiza.

Cada renglón de la matriz es almacenado en un "byte", por ejemplo:

CAPITULO V

Tomaremos el primer renglón de la matriz.

- - X X X X X -

Transformandolo a dígitos binarios sería:

0 0 1 1 1 1 1 0

Son ocho dígitos correspondientes a un "byte", si lo separamos en "nibbles" para su representación en sistema hexadecimal, obtenemos:

0 0 1 1 equivale a un 3

1 1 1 0 equivale a un E

así tendremos un 3E representando al primer grupo de ocho puntos.

Para entenderlo más facilmente diremos que cada lugar donde se encuentra una cruz será iluminado donde se encuentre una barra no será iluminado, lo que será el simbolo - nos dice no ilumines el "pixel" el simbolo X nos dice ilumina el "pixel".

V.3.2 TRAZADO DE LA PALETA DE COLORES.

Esta prueba consiste en visualizar los colores cargados en la paleta.

CAPITULO V

Para el trazado de la paleta de colores se hace de manera similar a lo que sería un caracter decodificado pero tomando como base una matriz de 15x15 puntos, como se muestra a continuación.

```
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
X X X X X X X X X X X X X X X X ...0000 o FFFF
```

Donde todos los puntos estarán ya sea iluminados o no. Este es un caracter en forma de un cuadro lleno, que será guardado y llamado igual que cualquier otro caracter. De esta manera se puede dibujar la paleta de colores, únicamente cambiando el color con que el caracter será pintado.

CAPITULO V

GENERACION DE UNA LINEA RECTA EN EL PRIMER OCTANTE.

Para la generación de una línea recta es necesario un algoritmo especial, ya que no es tan simple como los caracteres, debido a que una línea puede o no tener una pendiente y no es conveniente calcular cada punto de la línea basandonos en la ecuación de una recta.

$$X = mY + b$$

Donde m nos da la pendiente de la línea.

$$m = \text{DELTAY} / \text{DELTAX}$$

Con todo esto podemos observar que son necesarias muchas operaciones de multiplicación y división, pero en nuestro caso se hace necesario eliminar al máximo dichas operaciones ya que son muy costosas en tiempo haciendo muy tardado el sistema. Teniendo en cuenta que queremos reducir al mínimo el tiempo de cálculo, debemos efectuar únicamente operaciones de tipo de sumas, restas y comparaciones, lo que implica que tenemos que encontrar un algoritmo para la manipulación de dichas rectas.

Sabiendo que la pantalla es una rejilla de puntos bien definidos, es difícil trazar una recta inclinada exactamente; ya que los incrementos son hechos de uno en uno ("Pixel" a "Pixel"). En el caso real a veces es necesario hacer incrementos diferentes de uno. Debido a que es

CAPITULO V

imposible pintar un punto en coordenadas que estén entre las rejillas debemos aproximar la línea al punto más cercano a la línea real. (Ver figura 5.1)

Teniendo la ecuación general de una recta con origen en (0,0):

$$Y = mX$$

$$Y - mX = 0$$

$$Y = (\text{DELTAY}/\text{DELTAX})X$$

$$Y \text{ DELTAX} - X \text{ DELTAY} = 0$$

Lo ideal es que esto siempre se cumpla.

$$Y \text{ DELTAX} - X \text{ DELTAY} = E$$

Donde: E significa el error que debe ser lo más cercano a cero posible.

En el caso de incrementos de X

$$Y \text{ DELTAX} - (X+1)\text{DELTAY} = E \text{ nueva}$$

$$Y \text{ DELTAX} - X \text{ DELTAY} - \text{DELTAY} = E \text{ nueva}$$

$$E \text{ nueva} = E - \text{DELTAY}$$

De lo anterior deducimos que si aumentamos el valor de X manteniendo la pendiente y el valor de Y constante, el error disminuirá.

En el caso de incrementos en Y

$$(Y+1)\text{DELTAX} - X \text{ DELTAY} = E \text{ nueva}$$

$$Y \text{ DELTAX} + \text{DELTAX} - X \text{ DELTAY} = E \text{ nueva}$$

$$E \text{ nueva} = E + \text{DELTAX}$$

CAPITULO V

Ahora, si aumentamos el valor de Y manteniendo la pendiente y el valor de X constante, el error aumentará.

Para trazar la línea de la figura 5.1 existen dos movimientos posibles:

Movimiento A. Incrementa X

Si el VALOR A = $-DELTA X$

E nueva = $E - DELTA X$

E nueva = $E + VALOR A$

E nueva disminuye.

Movimiento B. Incrementa X, Incrementa Y

$(Y+1)DELTA X + (X+1)DELTA Y = E nueva$

E nueva = $Y DELTA X + DELTA X$

$- X DELTA Y - DELTA Y$

E nueva = $E + DELTA X - DELTA Y \geq 0$

$DELTA X \geq DELTA Y$

E nuevo = $E - DELTA Y + DELTA X$

EL VALOR B = $DELTA X - DELTA Y$

E nuevo = $E + VALOR B$

E nuevo aumenta.

CAPITULO V

ALGORITMO SIMPLE PARA TRAZAR UNA LINEA EN EL PRIMER OCTANTE.

La idea es mantener el error lo más cercano a cero, y para esto contamos con dos movimientos A y B como vimos anteriormente.

Si el error es positivo debemos disminuirlo con un movimiento A, en el caso de que el error sea negativo debemos aumentarlo con un movimiento B.

$DELTA X = X_{final} - X_{inicial}.$

$DELTA Y = Y_{final} - Y_{inicial}.$

$VALOR A = -DELTA Y.$

$VALOR B = DELTA X - DELTA Y.$

$E = 0.$

Pinta (Xinicial, Yinicial)

---> Mientras X = a Xfinal hacer:

| Si E >= 0 entonces Movimiento A

| Si no Movimiento B

| Pinta (X, Y)

---- ir a Mientras.

De esta manera, teniendo la posibilidad de generar caracteres y líneas es factible crear un paquete muy básico para la generación de figuras de poca complejidad como serían unifilares simples.

CONCLUSIONES

CONCLUSIONES.

1. Mediante éste trabajo demostramos la posibilidad de diseñar sistemas gráficos que pueden ser empleados en los sistemas de adquisición de datos para control supervisorio, así como una tarjeta que puede ser adaptada para múltiples usos dentro del área de automatización de redes.
2. Debido al diseño modular de nuestro sistema gráfico, se tiene la opción de expandirla tanto a nivel "hardware" aumentando la capacidad de la memoria, como "software" elaborando más programas para la generación de diferentes trazos tan complejos como se deseen, así como hacer una variación simple, eliminando el bloque de procesamiento y agregando la interfaz al bus, para emplearla en computadoras personales. aplicación útil en el manejo de los monitores de control.
3. Dentro del marco de un sistema multimaestro, su característica de tarjeta maestra la hace muy versátil, debido a que es una tarjeta autónoma que tiene además la posibilidad de compartir recursos, así como de comunicarse de manera sencilla con las otras tarjetas maestras sin afectar sus funciones propiamente gráficas; ya que guarda

CONCLUSIONES.

ciertas semejanzas con las otras maestras como serían: El procesador y coprocesador, el circuito de comunicaciones y el mapeo de la memoria entre otras. Esto es sumamente importante debido a que no es fácil acoplar un sistema gráfico a un sistema multimaestro.

4. La tarjeta es capaz de manejar 1024x512 "pixels", considerada como de alta resolución dentro de los sistemas gráficos, aunque en nuestro caso la probamos usando un monitor de 360x580 "pixels", resolución suficiente para el despliegue de unifilares y caracteres.

Para utilizarla en cualquier otro monitor, solo es necesario hacer algunos cambios, tanto en el sistema de reloj de video como en la programación del controlador de video que maneja las señales de dimensionamiento de la pantalla.

5. Entre las expectativas a futuro para ser adicionadas al sistema, se encuentra la adaptación de dispositivos de interfaz como serían un "mouse" o un teclado, además de la expansión de la memoria de video, además de su posible variación para ser utilizada dentro de una computadora personal, lo que hace a nuestro sistema atractivo, sobre todo si se trata de aplicarlo en sistemas de control supervisorio para automatización de las redes de distribución.

G L O S A R I O D E T E R M I N O S

GLOSARIO DE TERMINOS:

ACCESO: Operación que se realiza para obtener un dato de un dispositivo de almacenamiento o de una unidad periférica.

"BAUD RATE": Velocidad de transmisión que generalmente corresponde a el número de bits/seg.

BIT: Una contracción de dígito binario. Uno de dos caracteres (1 o 0).

"BIT MAP": Representación digital de una imagen cuyos bits son mapeados en "pixels" o puntos. Bloques de memoria usados para mantener imágenes trazadas en un formato específico.

"BLANK": Señales de pulsos usados para evitar el trazo del cañon durante los periodos de retrazo verticales u horizontales.

"BUFFER": Circuito aislador interpuesto entre dos circuitos para minimizar la reacción de la salida en la entrada de un circuito. Componente que sirve como almacenador auxiliar.

GLOSARIO DE TERMINOS.

"BUS": Conductor usado para transmitir señales de una o más fuentes, a uno o más destinos. Un solo alambre no se considera un bus.

"BYTE": Grupo de ocho dígitos binarios que funcionan como una unidad.

COMPUERTAS Y (AND) O (OR): Elemento lógico que opera con dígitos binarios que provee una señal de salida dependiendo de las señales de entrada aplicadas.

DIRECCION: Identificación por medio de un número de una localidad de memoria, registro u otra fuente o destino del dato.

"DISPLAY": Nombre con el que se le conoce al TRC cuya función es la de desplegar información de manera visual.

"EFECTO CUELLO DE BOTELLA": Efecto que se produce al entrar y salir información por la misma puerta de acceso a memoria.

ERROR: Cualquier condición en el cual un resultado actual difiera de un resultado correcto.

"HARDWARE": Parte física de un sistema como circuitos impresos, componentes, cableado, etc.

GLOSARIO DE TERMINOS.

HEXADECIMAL: Notación de números con base 16.

INICIALIZACION: Proceso llevado a cabo al principio de un programa para cargar todos los indicadores y constantes a valores determinados.

INTERFAZ: Enlace entre sistemas o dispositivos, representado por los canales o circuitos de control asociados realizando las conexiones entre procesador central y periféricos, o con otra unidad.

"LATCH": Biestable que sirve para separar y mantener direcciones de un bus multiplexado de datos y direcciones.

MEMORIA VOLATIL: Memoria que no puede retener los datos cuando el sistema se apaga.

MICROPROCESADOR: Comunmente usado como un sinónimo de microcomputadora, término que se emplea para reconocer al procesador central.

"MUART": Circuito Transmisor-Receptor Multifunciones Universal Asíncrono.

"MULTIPLEXOR": Unidad funcional que permite a varias fuentes de datos utilizar simultáneamente medios comunes de transmisión, asegurando a cada fuente su propia vía independiente.

GLOSARIO DE TERMINOS.

"NIBBLE": Grupo de dígitos binarios que consta de 4 elementos o bits.

"PIN": Nombre que se le da a las patitas o terminales de los circuitos integrados.

"PACKED PIXEL": Nombre con el que se conoce a un arreglo de memoria video en el cual toda la información del punto a desplegar se encuentra en localidades de memoria contiguas.

"PIXEL": El punto más pequeño controlable de luz en una pantalla del TRC.

RAM: Memoria de Acceso Aleatorio.

"RASTER": Patrón definido como el barrido de una pantalla de izquierda a derecha de la parte superior a la inferior de una pantalla.

RESET: Regresar un registro a un valor cero o inicial.

REGISTRO: Area de almacenamiento temporal para datos digitales.

REGISTRO DE CORRIMIENTO: Un registro en el cual datos almacenados pueden ser movidos de izquierda a derecha, o viceversa.

GLOSARIO DE TERMINOS.

RELOJ: Dispositivo capaz de generar señales periódicas.

ROM: Memoria para lectura solamente, también llamada memoria permanente.

SINCRONIA VERTICAL. Señal de sincronización que habilita retrazos verticales del cañon del TRC.

"SOFTWARE": Todos los programas escritos para ser ejecutados en el hardware, incluyendo sistemas operativos, programas de utilería y programas de aplicación para cumplir con los requerimientos de los usuarios.

"TIMER": Medidor de tiempo que no depende de señales externas de reloj par llevar su cuenta.

"UART": Transmisor Receptor Asíncrono que permite la conversión de transmisión paralelo a serie o serie a paralelo.

REFERENCIAS

REFERENCIAS.

- (1) IEEE COMPUTER GRAPHICS AND APPLICATIONS.
THE THIRD GENERATION OF PC GRAPHICS CONTROLLERS
J Michael James.

- (2) PRINCIPLES OF INTERACTIVE COMPUTER
GRAPHICS.
William M. Newman
Mc. Graw-Hill
1982

- (3) COMPUTER GRAPHICS
Donald Hearn
M. Pauline Baker
PRENTICE-HALL
MAYO 86

- (4) COMPUTER-AIDED DESIGN
COMPUTER-AIDED MANUFACTURING.
John K. Krouse.
Marcel Dekker, Inc.
1979.

- (5) SPECIAL REPORT
GRAPHICS ENGINES
Margery Conner.

REFERENCIAS.

- (6) TEXAS INSTRUMENTS GRAPHICS
VIDEO SYSTEM CONTROLLER PRODUCT DESCRIPTION
1985
- (7) COMPUTER DESIGN
Bit Map Architecture Realizes Raster Display
Potential.
Robert J. Gray
July 1980
- (8) ELECTRONICS
Computer Graphics.
Tom Manuel.
June 18 1984
- (9) INTEL GRAPHICS
A COLLECTION OF ARTICLE REPRINTS.
80286 y 386
1986
- (10) 8086/8088
16 BITS MICROPROCESSOR PRIMER
Christopher L. Morgan
Mitchell Waite
Mc. Graw-Hill 1982

REFERENCIAS.

- (11) MICROSYSTEM COMPONENTS HANDBOOK.
Microprocessors and Peripherals vol. I y II.
INTEL 1985.

- (12) THE TTL DATA BOOK FOR DESIGN ENGINEERS
TEXAS INSTRUMENTS 1980

- (13) TMS43061 USER'S GUIDE (PRELIMINARY)
VIDEO SYSTEM CONTROLER
TEXAS INSTRUMENTS 1986.

- (14) COMPUTER DESIGN
PLDs Slow advance of Gate Arrays in Low-End
Designs.
Ken Marrin.
February 1, 1986.

- (15) ALTERA PROGRAMMABLE LOGIC
HANDBOOK
ALTERA 1985

- (16) ALTERA PROGRAMMABLE LOGIC USER SYSTEM
USER GUIDE
ALTERA 1985

REFERENCIAS.

- (17) TMS34070 USERS' GUIDE (PRELIMINARY)
MANUAL UPDATE
TEXAS INSTRUMENTS 1986.

- (18) MITSUBISHI
Color Displays
Manual de Usuario
1894.

BIBLIOGRAFIA

BIBLIOGRAFIA

1.- FUNDAMENTALS OF INTERACTIVE COMPUTER GRAPHICS.

Foley-Van Dam

Addison-Wesley Publishing Company. 1983

2.- MICROSYSTEM COMPONENTS HANDBOOK.

Microprocesors and Peripherals Vol. I y II

INTEL 1985.

3.- BASIC TELEVISION PRINCIPLES AND SERVICE

Bernard Grob.

Mc Graw-Hill Kogakusha. 1975

4.- THE TTL DATA BOOK FOR DESIGN ENGINEERS

Texas Instruments Incorporated. 1980

5.- THE PENGUIN DICTIONARY OF MICROPROCESSORS.

Antony Chandor.

Penguin Books. 1980

BIBLIOGRAFIA

6.- THE NEW PENGUIN DICTIONARY OF ELECTRONICS.

E. C. Young.

Penguin Books. 1979

7.- MANUAL UPDATE

TMS34070 USER' S GUIDE (PRELIMINARY)

Texas Instruments 1986

8.- TM34061 USER' S GUIDE (PRELIMINARY)

VIDEO SISTEM CONTROLER

4 Texas Instruments 1986

9.- 8086/8088

16 BITS MICROPROCESSOR PRIMER

Cristopher L Morgan

Mitchell Waite

Mc. graw Hill 1982

10.- TEORIA DE CONMUTACION Y DISEÑO LOGICO.

Frederick J. Hill

Gerald R. Peterson

LIMUSA 1980.

11.- PAL PROGRAMABLE ARRAY LOGIC

Handbook

Monolithic memories 1983

BIBLIOGRAFIA

- 12.- ALTERA PROGRAMABLE LOGIC
HANDBOOK
ALTERA 1985
- 13.- ALTERA PROGRAMMABLE LOGIC USER SYSTEM
USER GUIDE
ALTERA 1985
- 14.- IEEE COMPUTER GRAPHICS AND
APPLICATIONS
VLSI for Graphics.
Vol. 6 Number 10
October 1986
- 15.- IEEE COMPUTER GRAPHICS AND APPLICATIONS.
Computer Graphics Hardware.
Vol. 6 Number 1.
January 1986.
- 16.- COMPUTER AND GRAPHICS
Memory Design For Raster Graphics Display
Whitton-Whitland
1984.
- 17.- COMPUTER
Real-Time Graphics in Command and Control.
Vol. 15 Number 4. 1982.

BIBLIOGRAFIA

18.- COMPUTER-AIDED DESIGN AND
COMPUTER-AIDED MANUFACTURING.

John K. Krouse

Marcel Dekker Inc.

1982.

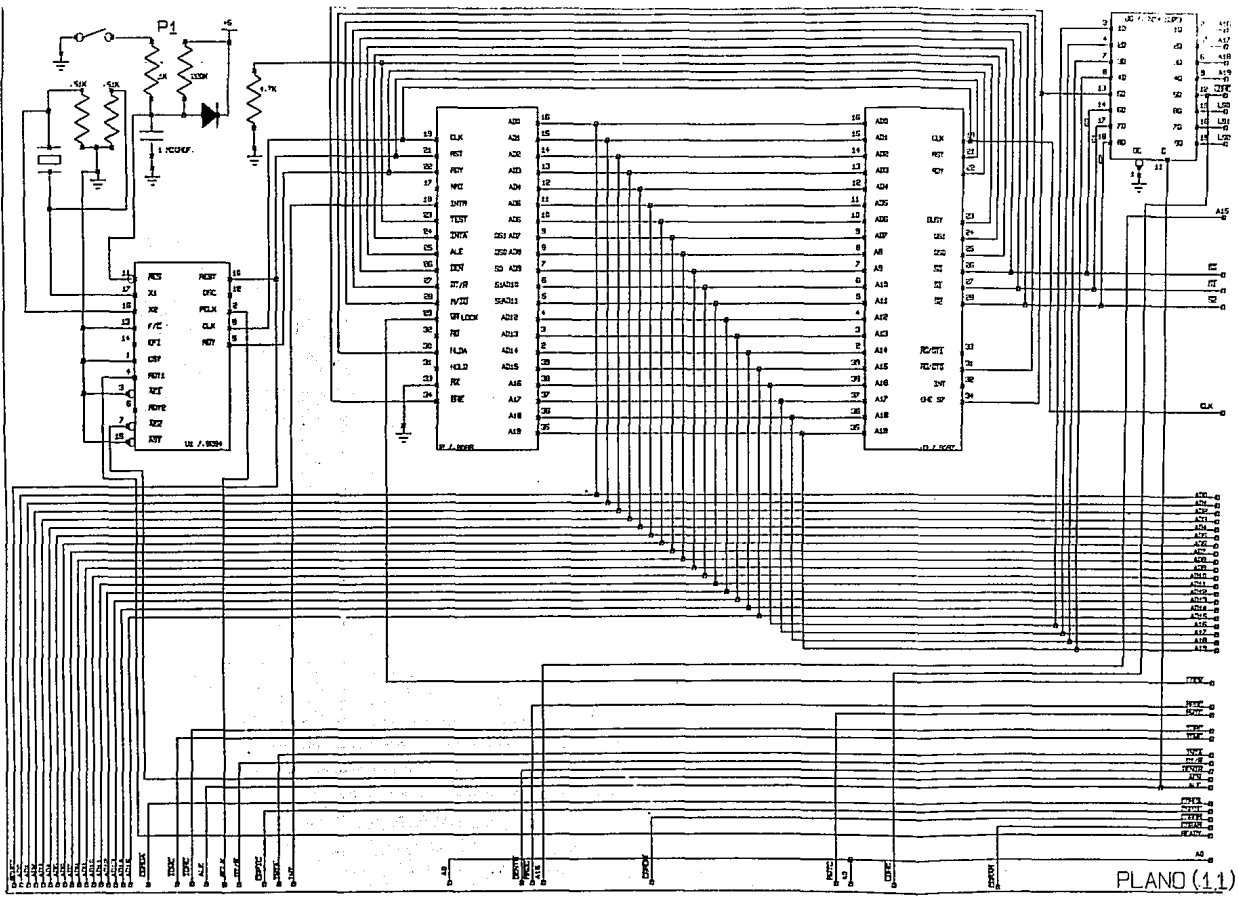
19.- PRINCIPLES OF INTERACTIVE COMPUTER
GRAPHICS

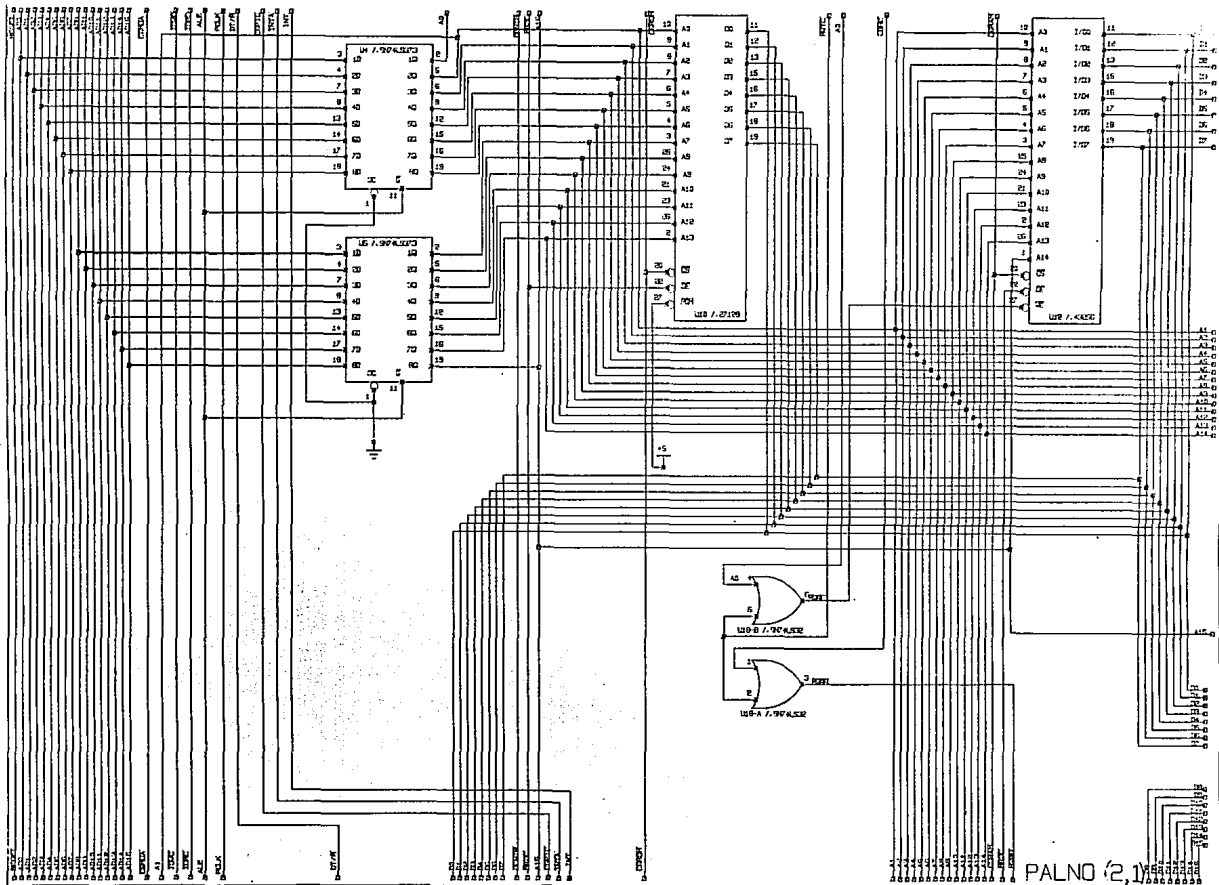
William M. Newman

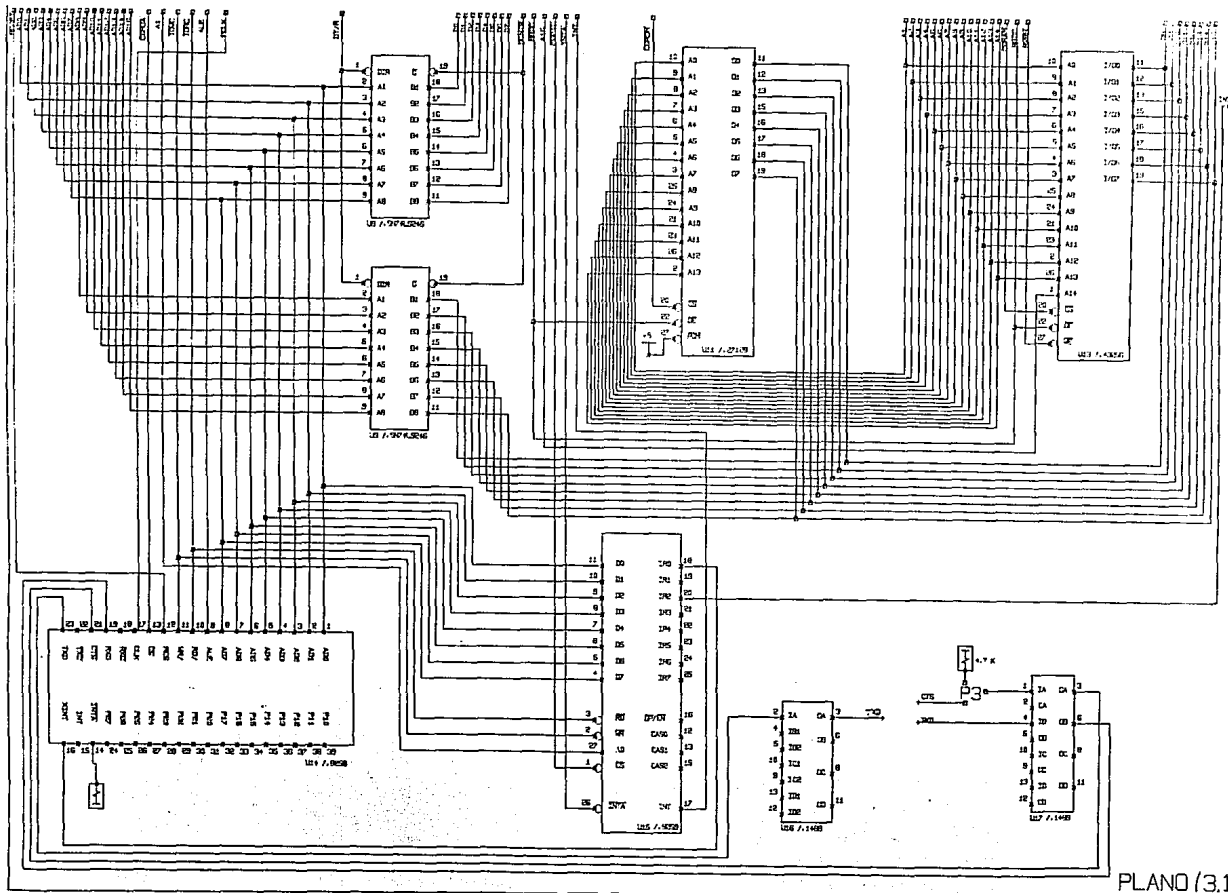
Mc. Graw-Hill

1979.

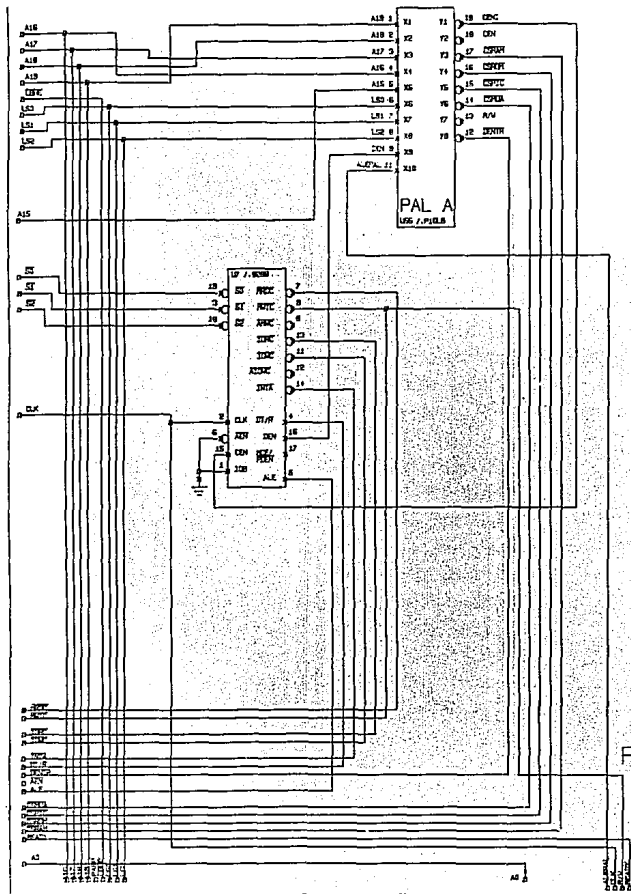
A P E N D I C E 1





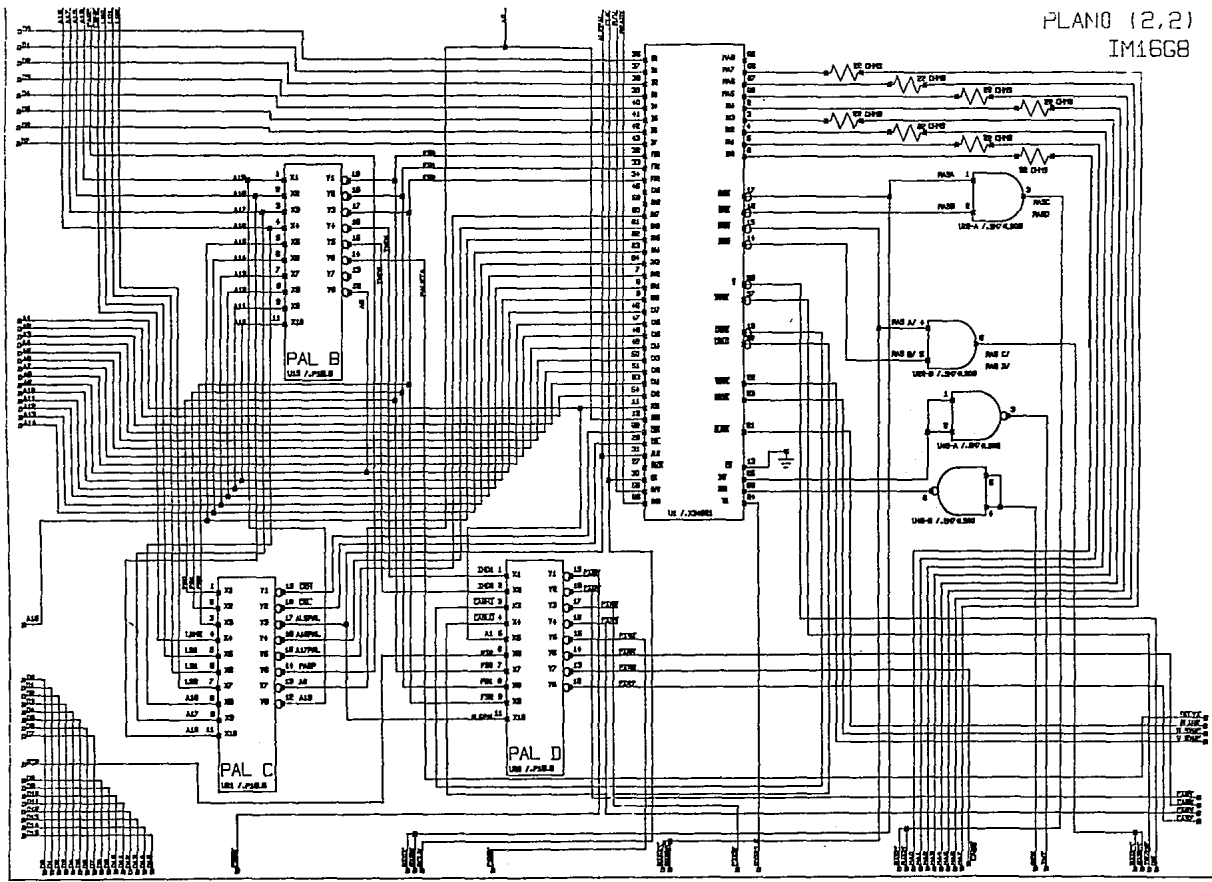


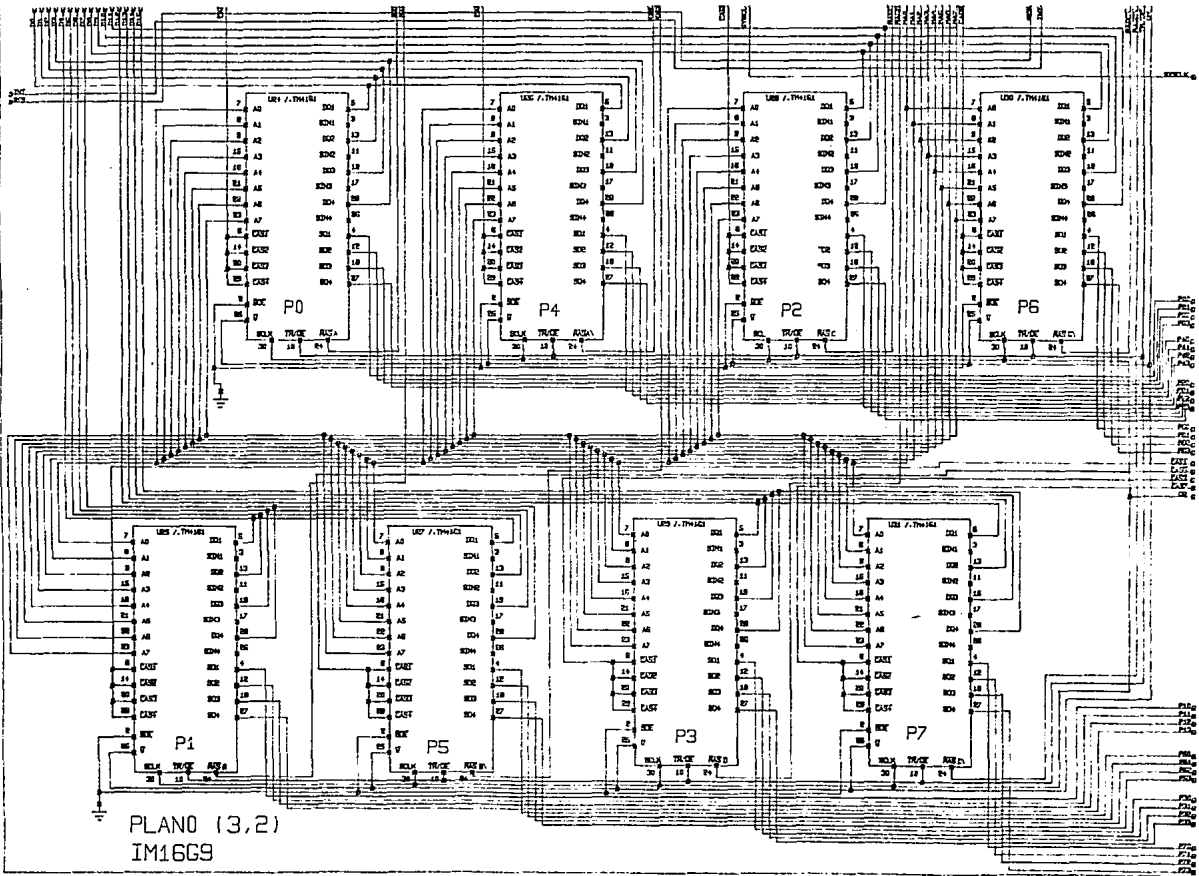
PLANO (3.1)



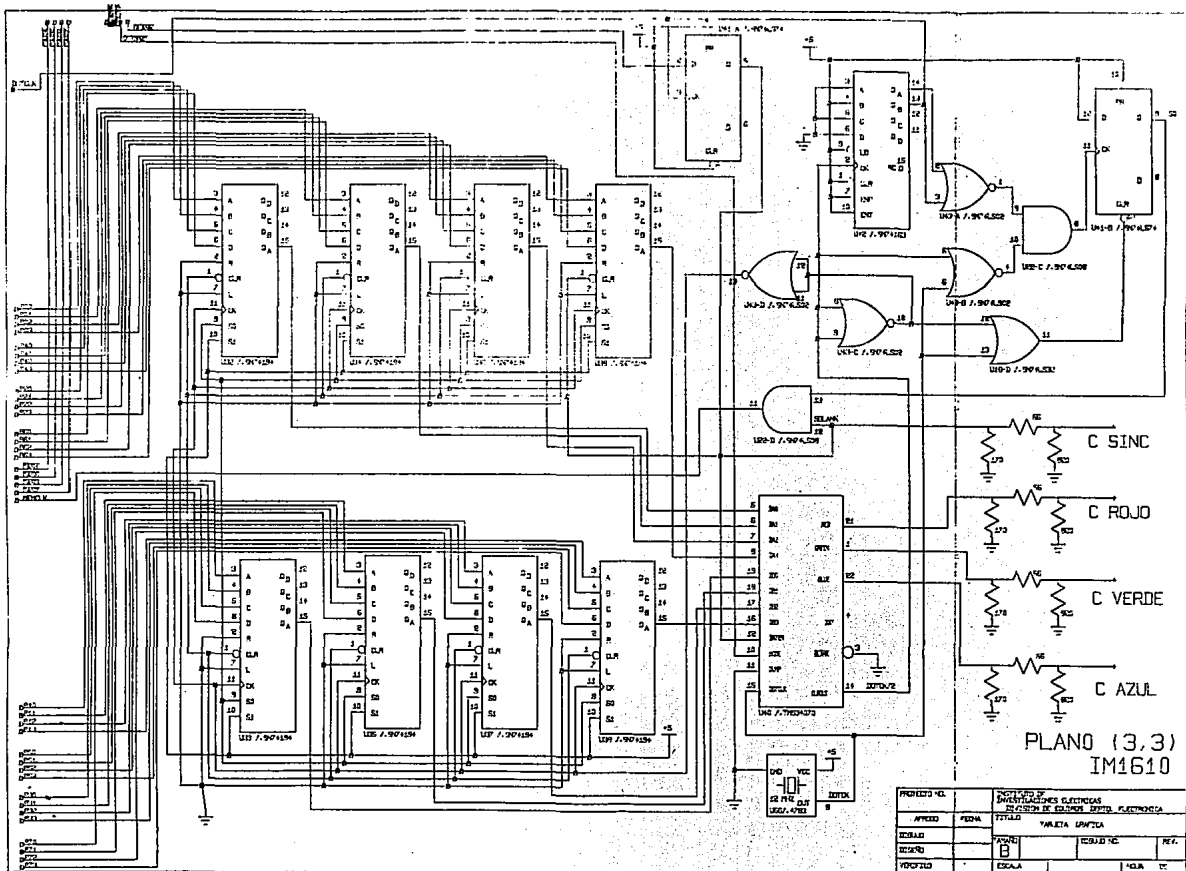
PLANO (1,2)
IM16G11

PLANG (2,2)
IM16G8





PLANO (3,2)
IM16GS



PLANO (3,3)
IM1610

PROYECTO 44	DISEÑO DE SISTEMAS DIGITALES		
APROBADO	FECHA	TÍTULO	
REVISADO		VARIAZ	UNIDAD
DISEÑO	B	ESQUEMA	REV.
VERIFICADO	ESCALA		CM. DC

A P E N D I C E 2

MIRELES
IIE
17/11/86
1.00

A
EP310
TARJETA GRAFICA
OPTIONS: TURBO=OFF
PART: EP310
INPUTS: A1701, A1B22, A1703, A1604, A1505, L8006, L8107, L8208,
DENI09, ALE011
OUTPUTS: CENI019, CENE018, CSRAM017, CSROM016, CSPIC015, CSMUA014,
RW013, DENTR012

NETWORK:

A19 = INP(A19)
A18 = INP(A18)
A17 = INP(A17)
A16 = INP(A16)
A15 = INP(A15)
L80 = INP(L80)
LS1 = INP(LS1)
L82 = INP(L82)
DENI = INP(DENI)
ALE = INP(ALE)
CENI, CENI = COEF(CENIc,)
CENE = CONF(CENEC,)
CSRAM = CONF(CSRAMc,)
CSROM = CONF(CSROMc,)
CSMUA = CONF(CSMUAc,)
CSPIC = CONF(CSPICc,)
RW = CONF(RWc,)
DENTR = CONF(DENTRc,)

EQUATIONS:

CSRAMc = $/(A19*/A18*/A17*/A16*L82)$;
CSROMc = $/(A19*A18*A17*A16*L82)$;
MCV = $(A19*A18*A17*/A16*L82)$;
FRBUFF = $/(A19*/A18*A16*L82) + (A19*/A18*A17*/A16*L82) + (A19*A18*/A17*/A16*L82)$;
INTER = $(L80*/LS1*/L82)$;
CSMUAc = $/(L82*/A15*/INTER)$;
CSPICc = $/(L82*A15*/INTER)$;
CENIc = $(A19*/A18*/A17*/A16*L82) + (A19*A18*A17*A16*L82)$
+ $(L82*/A15) + (L82*A15) + INTER + MCV + FRBUFF$;
CENEC = /CENI;
DENTRc = $(L82*/L81*/INTER*DENI)$;
RWc = $(L82*/L81*L80*CENI*/ALE) + (L82*LS1*/L80*CENI*/ALE)$;
END*

```

MIRELES
IIE
02/02/87
1.00
B
TARJETA GRAFICA
OPTIONS:TURBO=OFF
PART:EP310
INPUTS:A1901, A1802, A1703, A1604, A1505, A1406, A1307, A1208, A1109,
      A1011, A912, LB2013
OUTPUTS:PALETA014, INDO015, IND1016, FB2017, FB1018, FB0019
NETWORK:
A19 = INP(A19)
A18 = INP(A18)
A17 = INP(A17)
A16 = INP(A16)
A15 = INP(A15)
A14 = INP(A14)
A13 = INP(A13)
A12 = INP(A12)
A11 = INP(A11)
A10 = INP(A10)
A9 = INP(A9)
LB2 = INP(LB2)
INDO = CONF(INDOC, )
IND1 = CONF(IND1C, )
PALETA = CONF(PALETAC, )
FB2 = CONF(FB2C, )
FB1 = CONF(FB1C, )
FB0 = CONF(FB0C, )
EQUATIONS:
FS0C = /((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*LB2)
+((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)))
FB1C = /((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*LB2)
+((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*/A10*LB2)
+(A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+(A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+(A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)))
FB2C = /((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+(A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+(A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+((A19*/A18*/A16)*/((A19*/A18*/A17*/A16)*/((A19*A18*/A17*/A16))))
PALETAC = /((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*/A10*/A9*LB2)))
INDOC = /((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)))
IND1C = /((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)
+((A19*A18*A17*/A16*/A15*/A14*/A13*/A12*/A11*A10*/A9*LB2)))
END$

```

```

MIRELES
IIE
2/02/87
1.00
C
EP310
TARJETA GRAFICA
OPTIONS;TURBO=OFF
PART;EP310
INPUTS;FS01, FS102, FS203, LBHE04, MWR05, MRD06, LB207, A1608,
A1709, A18011, A19012, A0013
OUTPUTS;PASP014, A17PAL015, A16PAL016, ALE017, CEL018, CEH019
NETWORK;
FS2 = INP (FS2)
FS1 = INP (FS1)
FS0 = INP (FS0)
MWR = INP (MWR)
MRD = INP (MRD)
LB2 = INP (LB2)
A19 = INP (A19)
A18 = INP (A18)
A17 = INP (A17)
A16 = INP (A16)
LBHE = INP (LBHE)
AO = INP (AO)
CEL = CONF (CELc, )
CEH = CONF (CEHc, )
PASP = CONF (PASPc, )
ALE, ALE = COCF (ALEc, )
A16PAL = CONF (A16PALc, )
A17PAL = CONF (A17PALc, )
EQUATIONS;
VALIDA = /FS2*/FS1*FS0+ /FS2*FS1*FS0+FS2*/FS1*/FS0+FS2*/FS1*FS0;
A17PALC = /((/A19*/A18*/A17*/A16*LB2)+(/A19*/A18*/A17*/A16*LB2))*VALIDA;
A16PALC = /((/A19*/A18*/A17*/A16*LB2)+(/A19*/A18*/A17*/A16*LB2))*VALIDA;
PASPc = /A19*A18*A17*/A16*LB2;
CMD = /MWR + /MRD;
ALEc = /((CMD*/FS2*/FS1*/FS0)+(CMD*/FS2*/FS1*FS0)+(CMD*/FS2*FS1*FS0)
+/(CMD*FS2*/FS1*/FS0)+(CMD*FS2*/FS1*FS0));
CEHc = /(/LBHE*/ALE);
CELc = /(/AO*/ALE);
ENDS

```

MIRELES
11E
02/02/87
D

EP310
TARJETA GRAFICA
OPTIONS,TURBO-OFF
PARTI EP310

INPUTS:IND101, INDO02, CASH03, CASL04, A105, FS007, FB108, FB209,
PIP06, ALE011
OUTPUTS:CAS012, CAS6013, CAS5014, CAS4015, CAS3016, CAS2017, CAS1018,
CAS019

NETWORK:

IND1 = INP(IND1)
INDO = INP(INDO)
CASL = INP(CASL)
CASH = INP(CASH)
A1 = INP(A1)
PIP = INP(PIP)
FS2 = INP(FS2)
FS1 = INP(FS1)
FS0 = INP(FS0)
ALE = INP(ALE)

CAS0 = CONF(CAS0C, ALEN)
CAS1 = CONF(CAS1C, ALEN)
CAS2 = CONF(CAS2C, ALEN)
CAS3 = CONF(CAS3C, ALEN)
CAS4 = CONF(CAS4C, ALEN)
CAS5 = CONF(CAS5C, ALEN)
CAS6 = CONF(CAS6C, ALEN)
CAS7 = CONF(CAS7C, ALEN)
EQUATIONS:

TREG = ((FS2*/FS1*/FS0)+(FS2*/FS1*FS0));
ALE = /ALE

CAS0C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS1C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS2C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS3C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS4C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS5C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS6C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS7C = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

1

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS6c = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

CAS7c = /((TREG*/IND1*/INDO*/CASH*/CASL)+ (TREG*IND1*/INDO*/CASH*/CASL)

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);

+((TREG*/IND1*/INDO*/CASH*/CASL)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)
+((TREG*/IND1*/INDO*/CASH*/CASL*/A1)+TREG);