

2600 (Stella) Programmer's Guide

(Guía del Programador del Chip Stella)

by Steve Wright
12/03/1979

Fuentes:

- **STELLA PROGRAMMER'S GUIDE by Steve Wright 12/03/79 (Reconstructed by Charles Sinnett 6/11/93 Internet: cas@mentor.cc.purdue.edu).**

Stella.PDF

- **2600 (Stella) Programmer's Guide by Steve Wright 12/03/79 (Updated by Darryl May 07/01/88).**

2600_Guide.PDF

- **TIA 1A -TELEVISION INTERFACE ADAPTOR (MODEL 1A). Confidential.**

2600_TIA.PDF

INDICE

Sección I – Guía del Programador del Chip Stella

PROTOCOLO DE TELEVISIÓN.....	1
Diagrama 1 – Imagen de Tv Atari.....	2
El Chip TIA (como lo ve el programador).....	3
1.0 Descripción General.....	3
2.0 Registros.....	3
3.0 Sincronización.....	3
3.1 Temporización Horizontal.....	3
3.2 Sincronización del Microprocesador.....	4
3.3 Temporización Vertical.....	4
4.0 Color y Luminosidad.....	4
5.0 Playfield.....	4
6.0 Gráficos de los Objetos Móviles.....	5
6.1 Gráficos de los Misiles (M0 y M1).....	5
6.2 Gráficos de la Bola (BL).....	5
6.3 Gráficos de los Jugadores (P0, P1).....	5
7.0 Posicionamiento Horizontal.....	6
8.0 Movimiento Horizontal.....	6
9.0 Prioridades de Objetos.....	7
10.0 Colisiones.....	7
11.0 Sonido.....	7
11.1 Tono.....	7
11.2 Frecuencia.....	8
11.3 Volumen.....	8
12.0 Puertos de entrada.....	8
12.1 Puertos de Entrada Dumped (INPT0 hasta INPT3).....	8
12.2 Puertos de Entrada Latched (INPT4 hasta INPT5).....	8
El Chip PIA (6532).....	9
1.0 General.....	9
2.0 Timer de Intervalo.....	9
2.1 Manejo del Timer.....	9
2.2 Lectura del Timer.....	9
2.3 Cuando timer llega a cero.....	9
3.0 RAM.....	9
4.0 Puertos de Entrada/Salida.....	10
4.1 Puerto B - Switches de la Consola (sólo lectura).....	10
5.0 Puerto A - Joysticks (controladores manuales).....	10
5.1 Seteo para Entrada o para Salida.....	10
5.2 Entrada y Salida.....	10
5.3 Joysticks.....	10
5.4 Controladores Paddles (basados en potenciómetros).....	11
5.5 Controladores de Teclado.....	11
6.0 Tabla Sumarial de las Direcciones de Memoria.....	11
Conversiones a PAL/SECAM.....	12
PAL.....	12
SECAM.....	12

Sección II – Manual de Hardware del Chip TIA (Stella)

TIA 1A (el chip Stella) - Adaptador de Interfase de Televisión (modelo 1A).....	13
Descripción General.....	13
Descripción Detallada.....	13
1.0 Datos y Direccionamiento.....	13
2.0 Sincronización.....	14
2.1 Temporización Horizontal.....	14
2.2 Temporización Vertical.....	14
2.3 Sincronización Compuesta.....	14
2.4 Sincronización del Microprocesador.....	14
3.0 Registros Gráficos del Playfield.....	14
3.1 Descripción.....	14
3.2 Salida Serial Normal.....	15
3.3 Salida Serial Reflejada.....	15
3.4 Autocontrol.....	15
4.0 Contadores de Posición Horizontal.....	15
4.1 Descripción.....	15
4.2 Contador de Posición de la Bola.....	15
4.3 Contadores de Posición de los Jugadores.....	15
4.4 Contadores de Posición de Misiles.....	16
5.0 Registros de Movimiento Horizontal.....	16
5.1 Descripción General.....	16
5.2 Autocontrol.....	16
6.0 Registros Gráficos de los Objetos Móviles.....	16
6.1 Descripción General.....	16
6.2 Gráficos de Misiles.....	16
6.3 Gráficos de los Jugadores.....	16
6.4 Retardo Vertical (Vertical Delay).....	17
6.5 Gráficos de la Bola.....	17
7.0 Banderas de Detección de Colisión.....	17
7.1 Definiciones.....	17
7.2 Lectura de Colisiones.....	18
7.3 Reset.....	18
8.0 Puertos de Entrada.....	18
8.1 Descripción General.....	18
8.2 Puertos de Entrada Dumped (INPT0 a INPT3).....	18
8.3 Puertos de Entrada Latched (INPT4 e INPT5).....	18
8.4 Encoder de Prioridad.....	19
8.4.1 Propósito.....	19
8.4.2 Asignación de Prioridad.....	19
8.4.3 Control de Prioridad.....	19
9.0 Registros de Color y Luminosidad.....	19
9.1 Descripción.....	19
9.2 Multiplexación.....	20
10.0 Desplazador de Fase de Color (Color Phase Shifter).....	20
11.0 Circuitos de Audio.....	20
11.1 Selección de Frecuencia.....	20
11.2 Generador de Tono-Ruido.....	20
11.3 Selección del Volumen.....	20
Figura 1. Vertical Delay.....	21
Figura 2. Sincronización.....	22
Figura 3. Color-Luminosidad.....	22
Figura 4. Circuito de Movimiento Horizontal Típico.....	23
Figura 5. Gráficos del Playfield.....	24
Figura 6. Detección de Colisiones.....	25
Figura 7. Circuito de Audio.....	26
Figura 8. Puertos de Entrada.....	27
Figura 9. Sistema Completo.....	28
Funciones Detallas de las Direcciones de Escritura.....	29
WSYNC (wait for sync).....	29
RSYNC (reset sync).....	29
VSYNC.....	29
VBLANK.....	29
PF0 (PF1, PF2).....	29
Salida serial de los registros del Playfield.....	30
CTRLPF.....	30
NUSIZ0 (NUSIZ1).....	30

RESP0 (RESP1, RESM0, RESM1, RESBL).....	31
RESMP0 (RESMP1).....	31
HMOVE.....	31
HMCLR.....	31
HMP0 (HMP1, HMM0, HMM1, HMBL).....	32
ENAM0 (ENAM1, ENABL).....	32
GRP0 (GRP1).....	32
REFP0 (REFP1).....	32
VDELP0 (VDELP1, VDELBL).....	33
CXCLR.....	33
COLUP0 (COLUP1, COLUPF, COLUBK).....	33
AUDF0 (AUDF1).....	33
AUDC0 (AUDC1).....	34
AUDV0 (AUDV1).....	34
Sumario de las Direcciones de Escritura.....	35
Sumario de las Direcciones de Lectura.....	36
El chip Stella.....	37
TIA O0..02 AND LUM TIMING.....	38
TIA WRITE TIMING CHARACTERISTICS.....	39
TIA READ TIMING CHARACTERISTICS.....	40
TIA COMP-SYN AND READY TIMING.....	41
RSYNC, RES00, H01, H02, SHB, 02, 00.....	42
TIA RSYNC AND BLANK AND READY TIMING.....	43

PROTOCOLO DE TELEVISIÓN

(La imagen de televisión según Atari)

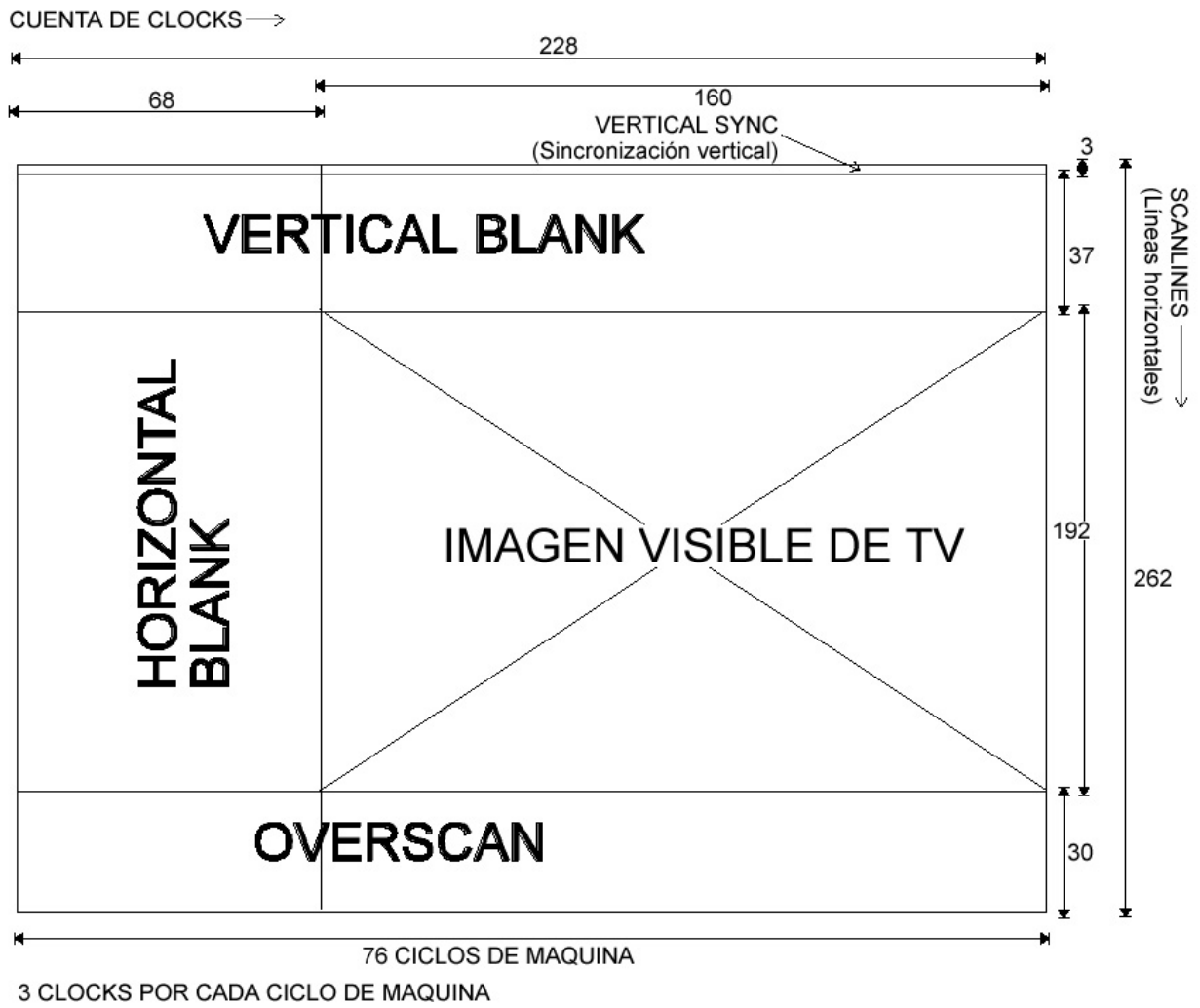
Para los propósitos de la programación del chip Stella, una pantalla simple de televisión consiste en 262 líneas horizontales, y cada línea está dividida en 228 ciclos de reloj (3.58 Mhz). La imagen de televisión es dibujada línea por línea desde arriba hacia abajo 60 veces por segundo, y en realidad apreciamos sólo una porción de la pantalla entera. Una imagen típica consistirá en tres líneas de sincronización vertical (Vertical sync lines - VSYNC), 37 líneas de margen vertical (VBLANK), 192 líneas de imagen de televisión, y treinta líneas finales (OVERSCAN). Según Atari este patrón funcionará en todos los tipos de televisores. Cada línea (SCANLINE), comienza con 68 ciclos de reloj de margen horizontal (Horizontal Blank - HBLANK - que no se ven en la imagen de televisión) seguidas de 160 ciclos de reloj para una línea completa de imagen. Cuando el haz de electrones del TRC llega al fin de una línea, en el lado izquierdo de la pantalla espera por 68 ciclos de reloj y procede a dibujar la próxima línea.

Toda la temporización horizontal es cuidada por el hardware, pero el microprocesador debe controlar manualmente la temporización vertical para indicar el comienzo de la nueva pantalla. Cuando la última línea de la pantalla es detectada, el microprocesador debe generar tres líneas de VSYNC, 37 líneas de VBLANK, 192 líneas de imagen de televisión o scanlines, y 30 líneas de OVERSCAN (finalización). Afortunadamente, ambas señales VSYNC y VBLANK pueden simplemente ser activadas o desactivadas en el tiempo apropiado, liberando al microprocesador para otras actividades durante su ejecución.

La imagen de televisión es dibujada de a una línea por vez, debiendo el microprocesador insertar los datos para cada línea dentro del adaptador de interfase de televisión (TIA, que es el chip Stella), el cual después convierte los datos en señales de vídeo. El TIA puede solo recibir los datos que pertenecen a la línea que se está dibujando en ese momento, así que el microprocesador debe estar un paso adelantado del haz de electrones que dibuja cada línea. Ya que un ciclo de máquina de microprocesador ocurre cada tres ciclos de reloj (del TIA), el programador tiene sólo 76 ciclos de máquinas por línea para construir la imagen (en realidad menos porque el microprocesador debe anticipar la imagen). Para permitir más tiempo para el software, es recomendable actualizar el TIA cada dos scanlines. La porción del programa que construye esta imagen de televisión es llamada KERNEL, y es la esencia del juego.

En general, las restantes 70 scanlines (3 para VSYNC, 37 para VBLANK, y 30 para OVERSCAN) proveen de 5320 ciclos de máquina para desarrollar la lógica de juego. Las actividades como cálculo de la nueva posición del jugador, actualizar el contador de puntos (SCORE) y verificar nuevas entradas se realizan típicamente durante este tiempo.

Diagrama 1 – Imagen de Tv Atari



El Chip TIA (como lo ve el programador)

1.0 Descripción General

El TIA es un circuito integrado desarrollado para crear la imagen de televisión y el sonido a partir de las instrucciones enviadas a él por el microprocesador. Convierte los ocho bits de datos paralelos desde el microprocesador en señales que son enviadas a los circuitos de modulación del vídeo, los cuales combinan y forman las señales compatibles con un televisor ordinario. Un campo de juego (PLAYFIELD) y cinco objetos móviles pueden ser creados y manipulados por el software.

Un Playfield puede ser paredes, nubes, barreras, y otros objetos móviles que pueden ser creados sobre un fondo de color. Los cinco objetos móviles pueden ser posicionados en cualquier parte, consisten en 2 jugadores (PLAYERS), 2 misiles (MISSILES), y 1 bola (BALL). El Playfield, jugadores, misiles, y bola son creadas y manipuladas por una serie de registros en el TIA que el microprocesador puede direccionar y escribir. Cada tipo de objeto tiene ciertas capacidades definidas. Por ejemplo, un jugador puede ser movido con una instrucción, pero el Playfield, debe ser re-dibujado para aparentar movimiento.

El color y brillo o luminosidad se pueden aplicar al fondo (BACKGROUND), Playfield, y los 5 objetos móviles. El sonido puede ser generado y controlado por volumen, altura y tipo de sonido. Los choques o colisiones entre varios objetos en la pantalla de televisión son detectados por el TIA y pueden ser leídos por el microprocesador. Los puertos de entradas pueden ser leídos para verificar el estado de los controladores manuales (Joysticks y switches de la consola).

2.0 Registros

Todas las instrucciones en el TIA son activadas direccionando y escribiendo en varios registros del chip. Algo que hay que recordar es que los datos escritos en los registros son retenidos hasta que cambian su valor mediante escritura por otra operación del microprocesador. Por ejemplo, si el registro de color de un jugador está seteado en rojo, así permanecerá hasta que el registro de color cambie. Todos los registros son direccionados por el microprocesador como parte de un espacio de memoria global RAM/ROM.

Todos los registros tienen direcciones fijas y nombre pre-asignado. Muchos registros no usan todos los ocho bits de datos, y algunos registros son usados como "STROBE" o registros gatillos, o sea que son instantáneos y los datos que se escriben en ellos son ignorados. Los únicos registros que puede leer el microprocesador son los registros de colisión o choque y los registros de puertos de entrada.

3.0 Sincronización

3.1 Temporización Horizontal

Cuando el haz de electrones cruza la pantalla de televisión y encuentra el lado derecho debe apagarse y volver al lado izquierdo de la pantalla para comenzar la próxima scanline. El TIA cuida esto automáticamente, independientemente del microprocesador. Un pulso de reloj generado por un oscilador de 3.58 MHz se llama "COLOR CLOCK" o pulso de color, el cual va conectado a un contador de pulso en el TIA. Este contador cuenta 160 color clocks hasta que el haz de electrones encuentra el lado derecho de la pantalla, y la entonces genera una señal de sincronización horizontal (HSYNC) para que el haz de electrones vuelva al lado izquierdo de la pantalla. También genera una señal para que la haz se apague (HBLANK) durante su regreso de 68 color clocks. El recorrido total del haz de electrones a través de una línea es $160+68=228$ color clocks. Todo esto es manejado por el TIA sin ayuda del microprocesador. (Ver diagrama 1).

3.2 Sincronización del Microprocesador

El reloj del microprocesador es el oscilador de 3.58 Mhz dividido por 3, por lo que un ciclo de máquina es equivalente a 3 color clocks. Entonces, una scanline completa de 228 color clocks es de sólo 76 ciclos de máquina. El microprocesador debe estar sincronizado con el TIA línea por línea, pero los ciclos de programa y los saltos del flujo de control toman impredecibles tiempos. Para resolver este problema de sincronización de software, el programador puede usar el registro strobe WSYNC (esperar por sincronización). Simplemente escribiendo en ese registro el procesador se detiene hasta que el haz de electrones encuentre nuevamente el lado derecho de la pantalla, entonces el microprocesador continúa sus operaciones en el comienzo de los 68 color clocks del HORIZONTAL BLANK. Ya que el TIA mantiene todas las instrucciones hasta que son alteradas por otra operación de escritura, podría ser actualizada cada dos o tres líneas. La ventaja es que el programador gana más tiempo para ejecutar sus instrucciones, pero al precio de tener una menor resolución en los gráficos.

Nota: WSYNC y todas las instrucciones siguientes están en el Manual de Hardware del Chip TIA.

3.3 Temporización Vertical

Cuando el haz de electrones pasó por las 262 líneas, se debe indicar al televisor que lo apague y posicione en el tope de la pantalla para comenzar una nueva imagen. Esta señal es llamada VERTICAL SYNC, y el TIA debe transmitir esta señal durante tres líneas. Esto se hace escribiendo un 1 en D1 del registro VSYNC para encender el haz de electrones, contar dos líneas restantes, entonces escribir un 0 en D1 del mismo registro para apagarlo.

Para que el haz de electrones no aparezca físicamente en la pantalla, el televisor necesita 37 scanlines de VERTICAL BLANK desde el TIA. Esto se hace escribiendo un 1 en D1 del registro VBLANK para comenzar el margen vertical, contar 37 scanlines y terminar el margen para comenzar a dibujar la nueva pantalla. Mientras, el microprocesador puede realizar otras operaciones durante ese tiempo.

4.0 Color y Luminosidad

El color y luminosidad pueden ser aplicados al fondo (BK), al Playfield (PF), a la bola (BL), jugador 0 (P0), jugador 1 (P1) y misiles 0 y 1 (M0 y M1). Sólo hay cuatro registros de color y luminosidad para estos siete objetos, según la siguiente tabla:

Registro	Objeto a colorear
COLUMP0	P0, M0 (jugador 0, misil 0)
COLUMP1	P1, M1 (jugador 1, misil 1)
COLUMPF	PF, BL (playfield, bola)
COLUMBK	BK (fondo)

Por ejemplo, si el registro COLUP0 se setea para el rojo brillante, los dos P0 y M0 serán de ese color.

Un registro de color y luminosidad consta de siete bits. Cuatro de los bits seleccionan uno de los dieciséis colores disponibles, y otros 3 bits seleccionan uno de los ocho niveles de luminosidad o brillo. El código específico requerido para el color y brillo determinado está listado en el Manual de Hardware del Chip TIA. Al igual que todos los registros, excepto los strobe, éstos conservan el valor hasta que son escritos nuevamente.

5.0 Playfield

El registro PF es usado para crear un Playfield de paredes, nubes, etcétera, que puede ser movido. Este registro de baja resolución es escrito para dibujar el lado izquierdo de la pantalla solamente. El lado derecho es una duplicación o reflejo del lado izquierdo según lo decida el programador.

El registro PF es de 20 bits, que se escriben en tres direcciones: PF0, PF1 y PF2. PF0 es de sólo 4 bits, y los dos restantes son de ocho bits. El registro PF es escaneado o barrido desde el lado izquierdo hacia el derecho, y donde hay un 1, el color es dibujado. Donde hay un 0, se ve el color del fondo. Si se escriben solos ceros en los registros PF, no se dibuja ninguna parte del Playfield.

Para hacer que el lado derecho sea una copia o duplicación del lado izquierdo, se escribe 0 en D0 del registro CTLPF (control del Playfield). Un 1 implica la reflexión del Playfield.

6.0 Gráficos de los Objetos Móviles

Todos los cinco objetos móviles (P0, M0, P1, M1, BL) pueden ser asignados a una posición horizontal en la pantalla y movidos hacia la izquierda o a la derecha en forma relativa de su posición. Las posiciones verticales son tratadas en manera muy diferente. En principio, los objetos aparecen en cualquier línea donde sus registros gráficos estén activados. Por ejemplo, asumamos que la bola está posicionada verticalmente en el centro de la pantalla. La pantalla tiene 192 scanlines y queremos que la bola sea 2 líneas de alto. Los gráficos de la bola deberían estar desactivados hasta la scanline 96, activados durante dos scanlines, y desactivados por el resto de la pantalla. Cada tipo de objeto (jugador, misil, y bola) tiene sus propias características y limitaciones.

6.1 Gráficos de los Misiles (M0 y M1)

Estos dos gráficos dibujan un misil en cualquier línea escribiendo un 1 en el bit de activación del registro de misiles (ENAM0, ENAM1).

Escribiendo un 0 en esos registros se desactivan sus gráficos. El lado izquierdo del misil se posiciona con un registro de posición horizontal, el lado derecho es en función del tamaño del misil. Su tamaño se controla escribiendo en los bits D4 y D5 de los registros de número-tamaño (NUSIZ0, NUSIZ1). Estos registros estirarán el misil 1,2,4 ó 8 color clocks (una scanline completa tiene 160 color clocks).

6.2 Gráficos de la Bola (BL)

Los registros gráficos de la bola trabajan igual que los registros de los misiles. Escribiendo un 1 en el registro de activación de la bola (ENABL), la bola se dibuja. La bola también puede ser estirada igual que los gráficos de misiles pero en el registro CTRLPF.

La bola también puede sufrir un “delay” o retardo de una scanline. Por ejemplo, si los gráficos de la bola están activados en la línea 95, puede ser retardada para no mostrarse en la pantalla hasta la línea 96 describiendo 1 en D0 del registro de retardo vertical (VDELBL). La razón de esto es que la mayoría de los programas actualizan el TIA cada dos líneas. Esto limita todos los movimientos verticales de los objetos asaltos de dos líneas. El uso del vertical delay permite que los objetos se muevan en una línea por vez.

6.3 Gráficos de los Jugadores (P0, P1)

Los gráficos de jugadores son más sofisticados que todos los demás objetos móviles. Tienen las capacidades de los misiles y de la bola, más tres capacidades de movimiento. Los jugadores pueden tomar la forma de hombres o de aeroplanos, etc., y pueden ser fácilmente volteados horizontalmente para mostrar una imagen refleja además de múltiples copias del mismo jugador.

Los gráficos de los jugadores son dibujados línea por línea como cualquier otro gráfico. La diferencia aquí es que cada línea de jugador tiene un tamaño de 8 bits, a diferencia de los misiles y la bola que sólo tienen un bit. Un jugador puede ser pensado como un gráfico en papel cuadriculado de ocho cuadros de ancho y tan largo como se quiera. Se pueden colorear los cuadros de este papel imaginario escribiendo en los registros gráficos de los jugadores (GP0, GP1). Estos registros de ocho bits son escaneados desde D7 hacia D0, y donde hay un 1 se dibuja el cuadro, donde hay un 0 no se dibuja. Para posicionar verticalmente un jugador se deja de escribir en sus registros gráficos hasta la línea donde queremos que aparezca el jugador, allí se escribe la información de 8 bits. Al terminar de dibujarse completamente el jugador, se vuelven a escribir 0 en sus registros gráficos para que no aparezca en la pantalla.

Para mostrar una imagen reflejada de la figura original, se describe un 1 en D3 del registro de reflexión (REFP0, REFP1). Un 0 mantiene la figura original.

Las múltiples copias de los jugadores y sus tamaños son controladas escribiendo tres bits (D0, D1, D2) en los registros de número-tamaño (NUSIZ0, NUSIZ1).

Estos tres bits seleccionan de una a tres copias del jugador, la separación entre copias, y el tamaño del jugador (el ancho de cada jugador puede ser de 1, 2, 4 u 8 color clocks). Cuando se seleccionan múltiples copias, el TIA automáticamente crea el mismo número de copias de misil para ese jugador. Otra vez, todas estas especificaciones están en el Manual de Hardware del Chip TIA.

El delay vertical de los jugadores funciona exactamente como el de la bola escribiendo un 1 en D0 de los registros VDELP0, VDELP1. Escribiendo un 0 se desactiva el delay vertical.

7.0 Posicionamiento Horizontal

La posición horizontal de cada objeto se maneja escribiendo en sus respectivos registros de reset (RESP0, RESP1, RESM0, RESM1, RESBL), los cuales son registros "strobe" (desarrollan su función tan pronto como son direccionados). Estos registros causan que el objeto se posicione donde el haz de electrones se encuentre en la pantalla. Por ejemplo, si el haz de electrones está a 60 color clocks en una scanline cuando se escribe en RESP0, significa que el jugador 0 se posicionará a 60 color clocks en la próxima scanline. Ya sea que se dibuje o no el jugador escribiendo en su registro gráfico GRP0. Si se resetean estos registros en cualquier lugar durante el horizontal blank, la posición de los objetos será el lado izquierdo de la pantalla (color clock 0). Ya que son tres color clocks por cada ciclo de máquina, y puede tomar cinco ciclos de máquina de escribir el registro, el programador está limitado a posicionar los objetos a intervalos de 15 color clocks a lo largo de la pantalla. Este posicionamiento se afina luego con el movimiento horizontal (HORIZONTAL MOTION), explicado en la sección 8.0.

Los misiles tienen un comando adicional de posicionamiento. Describiendo en D1 del registro de reseteo del misil al jugador (reset missile-to-player register), RESMP0 y RESMP1, se desactivan los gráficos de los misiles y se reposicionan horizontalmente en el centro de su jugador asociado. Hasta que un 0 se escriba en el registro, la posición horizontal de los misiles será en el centro de su jugador preparándose para ser disparado de nuevo.

8.0 Movimiento Horizontal

El movimiento horizontal permite al programador mover cualquiera de los cinco objetos relativamente de su actual posición horizontal. Cada objeto tiene un registro de movimiento horizontal (HMP0, HMP1, HMM0, HMM1, HMBL) de 4 bits que puede ser cargado con un valor en el rango de +7 a -8. Este movimiento no es ejecutado hasta que el registro HMOVE es escrito, en ese momento todos los registros de movimiento desplazan su objeto respectivo. Los objetos pueden ser movidos repetidamente tan solo ejecutando HMOVE. Cualquier objeto que no se quiera mover debe tener un 0 en su registro de movimiento. Con el comando de posicionamiento horizontal confinado a posicionar al objeto a intervalos de 15 color clocks, los registros de movimientos se mantienen moviendo objetos +7 a -8 color clocks. Los objetos no pueden ser colocados en cualquier posición de color clock a lo largo de la pantalla. Todos los cinco registros de movimiento pueden ser seteados a 0 simultáneamente escribiendo el registro de borrado de movimiento horizontal (HMCLR – HORIZONTAL MOTION CLEAR).

Hay consideraciones de tiempos para el comando HMOVE. El comando HMOVE debe seguir inmediatamente a un WSYNC para asegurarse que ocurre durante el horizontal blank. Esto es para permitir suficiente tiempo para que los registros de movimiento hagan lo suyo antes de que el haz de electrones comience a dibujar la próxima scanline. Además, por misteriosas consideraciones internas de hardware, los registros de movimiento no deberían ser modificados sino hasta que pasaron 24 ciclos de máquina desde el último comando HMOVE.

9.0 Prioridades de Objetos.

A cada objeto se le asignó una prioridad para que cuando los objetos están en la misma posición se vea el que tiene mayor prioridad. Para simplificar la lógica del hardware, los misiles tienen la misma prioridad que su jugador asociado, y la bola tiene la misma prioridad que el Playfield. El fondo, por supuesto, tiene la menor prioridad. La siguiente tabla muestra asignaciones de prioridad por defecto.

Prioridad	Objetos
1	P0, M0
2	P1, M1
3	BL, PF
4	BK

Esta asignación de prioridad significa que cada jugador y misiles se mueven en el frente del Playfield. Para hacer que los jugadores y misiles se muevan detrás del Playfield, un 1 se debe escribir en D1 del registro CTRLPF. La siguiente tabla ilustra prioridades afectadas:

Prioridad	Objetos
1	PF, BL
2	P0, M0
3	P1, M1
4	BK

Un control más de prioridad está disponible para usar cuando se muestran los puntajes o score. Cuando se escribe un 1 en D1 del registro CTRLPF, la mitad izquierda del Playfield adquiere el color del jugador 0, y la mitad derecha el color del jugador 1. El score del juego puede ser mostrado ahora usando los registros gráficos PF, teniendo los mismos colores que sus respectivos jugadores.

10.0 Colisiones

El TIA detecta las colisiones entre cualquiera de los seis objetos que genera (el Playfield y los cinco objetos móviles). Hay 15 posibles colisiones entre dos objetos que se almacenan en registros de un bit. Cada registro de colisión contiene dos de éstos, los cuales son leídos por el microprocesador en D6 y D7 del bus de datos para un acceso fácil. Un 1 en la línea de datos indica que la colisión ocurrió. Los registros de colisión pueden ser leídos en cualquier momento pero generalmente se lo hace durante el VERTICAL BLANK después que todas las posibles colisiones hayan ocurrido. Los registros de colisión se borran todos simultáneamente escribiendo en el registro CXCLR.

11.0 Sonido

Hay dos circuitos de audio para generar sonido que son los dos canales disponibles. Son idénticos pero completamente independiente y pueden ser operados simultáneamente para producir efectos de sonido a través del parlante del televisor. Cada circuito de audio posee tres registros que controlan un generador de tono-ruido (para la clase de sonido), un selector de frecuencia (sonido agudo o grave), y un control de volumen.

11.1 Tono

El generador de tono-ruido (uno para cada canal) es controlado escribiendo en su registro de control de audio de 4 bits (AUDC0, AUDC1). Los valores escritos causan diferentes tipos de sonidos para ser generados. Algunos pueden ser puros como una flauta, otros tienen contenidos de ruido como un motor o explosión. En el Manual de Hardware del Chip TIA se listan los sonidos creados con cada valor, se necesita alguna experimentación para encontrar el sonido buscado.

11.2 Frecuencia

La selección de frecuencias es controlada escribiendo en el registro de frecuencia de audio de 5 bits (AUDF0, AUDF1). El valor de escrito se usa para dividir una frecuencia de referencia de 30 Hz creando un sonido más agudo o más grave sobre el tipo de sonido creado por el generador de tono-ruido. Combinando tonos puros disponibles en el generador de tono-ruido con la selección de frecuencia, un gran rango de tonos se pueden generar.

11.3 Volumen

El volumen es controlado escribiendo en el registro de volumen de 4 bits (AUDV0, AUDV1). Escribiendo ceros en esos registros el sonido se apaga completamente, y escribiendo un valor de hasta quince, se incrementa el volumen.

12.0 Puertos de entrada

Hay seis puertos de entrada, cuyos estados lógicos pueden ser leídos en D7 leyendo la dirección del puerto de entrada (INPT0, hasta INPT5). Se dividen en dos tipos, "dumped" y "latched".

12.1 Puertos de Entrada Dumped (INPT0 hasta INPT3)

Estos cuatro puertos se usan para leer hasta cuatro controladores paddle. Cada controlador paddle contiene potenciómetro ajustable en la manija del controlador. La salida del potenciómetro se usa para cargar un capacitor en la consola, y cuando el capacitor está cargado el puerto de entradas se pone en alta o HIGH. El microprocesador descarga este capacitor describiendo un 1 en D7 del VBLANK que es el tiempo que tarda detectar un 1 lógico en ese puerto. Esta información puede ser usada para posicionar objetos en la pantalla basado en la posición de la manija del controlador paddle.

12.2 Puertos de Entrada Latched (INPT4 hasta INPT5)

Estos dos puertos tienen registros que son activados escribiendo un 1 o desactivados describiendo un 0 en D6 del VBLANK. Cuando están desactivados, el microprocesador lee el nivel lógico del puerto directamente, cuando están activados el registro indica que el puerto está en baja o LOW. Cuando el puerto esta en LOW, el registro también lo está. Los botones de disparo de los controladores joysticks están conectados a este puerto.

El Chip PIA (6532)

1.0 General

El chip PIA es un adaptador de interfaz periférico modelo 6532, el cual tiene tres funciones: un timer o temporizador programable, 128 bytes de RAM y 2 puertos paralelos de entrada/salida de 8 bits.

2.0 Timer de Intervalo

El PIA usa el mismo clock que el microprocesador así que un ciclo PIA ocurre en cada ciclo de máquina. El PIA puede ser seteado en uno de cuatro diferentes intervalos, donde cada intervalo es un múltiplo del clock (y también del ciclo de máquina). Un valor de entre 1 y 255 se carga en el PIA, el cual se decrementa en 1 en cada intervalo. El timer puede ser leído por el microprocesador para determinar el tiempo transcurrido para varias operaciones de software y mantenerse sincronizado con el chip TIA.

2.1 Manejo del Timer

El timer se maneja escribiendo un valor (entre 1 y 255) en la dirección de memoria acorde con la siguiente tabla:

Dirección Hexa	Intervalo	Mnemónico
294	1 clock	TIM1T
295	8 clocks	TIM8T
296	64 clocks	TIM64T
297	1024 clocks	T1024T

Por ejemplo, si un valor de 100 se escribe en TIM64T (dirección de memoria hexadecimal 296) el timer decrementará hasta 0 en 6400 clocks (68 clocks por intervalo por 100 intervalos) los cuales también deberían ser 6400 ciclos máquina de microprocesador.

2.2 Lectura del Timer

El timer puede ser leído cualquier número de veces antes de ser cargado, pero al programador normalmente le interesa saber si el timer llegó a cero. El timer se lee en la dirección de memoria hexadecimal 284, o sea INTIM.

2.3 Cuando timer llega a cero

El PIA decrementa el valor cargado en el timer una vez en cada intervalo hasta llegar a 0. Además mantiene el 0 durante un solo intervalo, entonces el contador pasa a FF (valor hexadecimal) y comienza a decrementar nuevamente. El propósito de esta característica es permitir al programador determinar cuánto tiempo pasó desde que el timer llagó a 0.

3.0 RAM

El PIA tiene 128 bytes de memoria RAM alojados en el mapa de memoria del chip Stella desde la dirección 80 a FF (valor hexadecimal). El stack (porción de memoria) del microprocesador se aloja desde la dirección FF hacia abajo, y las variables normalmente se alojan desde la dirección 80 hacia arriba (esperando que no se encuentren nunca).

4.0 Puertos de Entrada/Salida

Estos dos puertos (Puerto A y Puerto B) son de ocho bits y pueden ser seteados tanto para entrada como para salida. El puerto A se usa como interfaz de los controladores joysticks, pero el puerto B maneja el estado de las llaves (switches) de la consola.

4.1 Puerto B - Switches de la Consola (sólo lectura)

El Puerto B está cableado para ser un puerto sólo de entrada pudiéndosele leer en la dirección de memoria SWCHB (HEX 282) para determinar el estado de todas las llaves de la consola según la siguiente tabla:

Bit de datos	Switch	Significado
D7	P1 dificultad	0 = fácil (B), 1 = difícil (A)
D6	P0 dificultad	0 = fácil (B), 1 = difícil (A)
D5/D4	(no se usa)	
D3	color - B/N	0 = B/N, 1 = color
D2	(no se usa)	
D1	game select	0 = switch presionado
D0	game reset	0 = switch presionado

5.0 Puerto A - Joysticks (controladores manuales)

El puerto A se controla totalmente por software para ser configurado como un puerto de entrada o como un puerto de salida. Puede ser usado para leer o controlar varios joysticks, dependiendo de la definición de sus bits de datos para cada tipo de controlador.

5.1 Seteo para Entrada o para Salida

El puerto A tiene un registro de dirección de datos (DDR) de ocho bits, que puede ser escrito en la dirección de memoria SWACNT (HEX 281) para setear cada pin individual del puerto para que sea de entrada o de salida. Los pines del puerto están numerados desde PA0 hasta PA7, y escribiendo un 0 en su correspondiente bit del DDR el pin será de entrada, si se escribe 1 el pin será de salida. Por ejemplo, si se escriben todos ceros en SWACNT quedan todos los pines como de entrada. Si se escriben todos unos, quedará un puerto totalmente de salida.

5.2 Entrada y Salida

Una vez que el DDR fue escrito para setear el puerto A para entrada o salida, puede ser leído o escrito en la dirección de memoria SWCHA (HEX 280).

5.3 Joysticks

Dos controladores manuales o joysticks pueden ser leídos configurando el puerto como entrada y leyendo en la dirección SWCHA según la siguiente tabla:

Bit de datos	Dirección	Jugador
D7	derecha	P0
D6	izquierda	P0
D5	abajo	P0
D4	arriba	P0
D3	derecha	P1
D2	izquierda	P1
D1	abajo	P1
D0	arriba	P1

(P0 = jugador izquierdo, P1 = jugador derecho)

Un 0 en el bit de datos indica que el joystick fue movido a esa posición. Todos 1 indican que el joystick no se ha movido.

5.4 Controladores Paddles (basados en potenciómetros)

El PIA sólo puede leer los botones del paddle. Los paddles leen en INP0 a INP3 del TIA. Se leen en SWCHA segun la siguiente tabla:

Bit de datos	Paddle número:
D7	P0
D6	P1
D5/D4	(no se usan)
D3	P2
D2	P3
D1/D0	(no se usan)

5.5 Controladores de Teclado

El controlador de teclado tiene doce botones arreglados en cuatro filas y tres columnas. Una señal debe ser enviada a una fila, entonces las columnas son chequeadas para ver si un botón está apretado, entonces la siguiente línea es señalada y todas las columnas censadas. El PIA envía señales hacia las filas, y las columnas son censadas leyendo en INPT0, INPT1 y INPT4 del TIA. Con el puerto A configurado como un puerto de salida, los bits de datos enviarán una señal hacia las filas del controlador de teclado según la siguiente tabla:

Bit de datos	Fila del teclado	Jugador
D7	cuarta	P0
D6	tercera	P0
D5	segunda	P0
D4	primera	P0
D3	cuarta	P1
D2	tercera	P1
D1	segunda	P1
D0	primera	P1

(P0 = jugador izquierdo; P1 = jugador derecho)

Nota: un delay o retardo de 400 microsegundos se necesitan entre escribir en este puerto y leer en el puerto de entrada del TIA.

6.0 Tabla Sumarial de las Direcciones de Memoria

Dirección Hexa	Mnemonico	Propósito
280	SWCHA	Puerto A; entrada salida (lectura o escritura)
281	SWACNT	Puerto A DDR, 0= entrada, 1=salida
282	SWCHB	Puerto B; switches de la consola (solo lectura)
283	SWBCNT	Puerto B DDR (cableado como entrada)
284	INTIM	Salida del Timer (solo lectura)
294	TIM1T	setear 1 clock intervalo (838 nseg/intervalo)
295	TIM8T	setear 8 clock intervalo (6.7 useg/intervalo)
296	TIM64T	setear 64 clock intervalo (53.6 useg/intervalo)
297	T1024T	set 1024 clock intervalo (858.2 useg/intervalo)

Nota: un clock es también un ciclo de máquina del microprocesador.

Conversiones a PAL/SECAM

PAL

El número de scanlines, y por consiguiente el tiempo de cada pantalla se incrementa de NTSC a PAL según la siguiente tabla:

	NTSC		PAL	
	Scan- lines	Micro- segundos	Scan- lines	Micro- segundos
VBLANK	40	2548	48	3085
KERNAL	192	12228	228	14656
OVERSCAN	30	1910	36	2314
FRAME	262	16686	312	20055

El sonido caerá un poco en frecuencia a causa de un clock (crystal clock) más lento. Algunos sonidos podrían necesitar retoques en AUDF0/AUDF1.

PAL opera a 50 hz comparado con NTSC que opera a 60 Hz. Es decir un 17% de reducción. Si la velocidad del juego está basada en pantallas por segundo, ésta decrecerá 17%, lo cual podría ser desastroso en juegos de acción. Si la versión NTSC está diseñada con técnicas de adición fraccional de 2 bytes (o cualquiera no basada en pantallas por segundo) para mover objetos, entonces la conversión a PAL puede ser tan simple como cambiar la tabla de fracciones.

SECAM

1. SECAM es poco diferente. Toma el software de PAL, pero el switch color/BN de la consola debe estar en BN. Entonces se lee la tabla blanco y negro de PAL en software y asigna un color fijo a cada luminosidad de blanco y negro, según la siguiente tabla:

<u>Luminosidad</u>	<u>Color</u>
0	negro
2	azul
4	rojo
6	magenta (rosa)
8	verde
A	cyan (celeste)
C	amarillo
E	blanco

Tener en cuenta: el manual es el mismo para NTSC, PAL Y SECAM. Esto significa que la descripciones de blanco y negro deben ser similares entre NTSC y PAL. Si usted hace mayores cambios al blanco y negro de PAL para obtener un mejor color SECAM, el blanco y negro NTSC debe ser similar.

2. El sonido de PAL funciona bien en SECAM con una sola excepción: cuando un sonido se apaga, debe ponerse un 0 en AUDV0/AUDV1, y no en AUDC0/AUDC1. De otra manera, se obtendría un mal sonido de fondo.

TIA 1A
(el chip Stella)

Adaptador de Interfase de Televisión (modelo 1A)

CONFIDENTIAL

Descripción General

El chip TIA es un circuito integrado MOS diseñado para ser interfase entre un microprocesador de 8 bit y un modulador de vídeo y para convertir datos paralelos de 8 bit en la salida serial de color, luminosidad, y sincronización compuesta requerida por el modulador de vídeo.

Este circuito opera línea por línea, siempre dando la misma salida de información en cada línea de televisión a menos que nuevos datos se escriban en él por medio del microprocesador.

Un contador sincrónico por hardware produce pulsos de sincronización horizontal independientes del microprocesador. La sincronización vertical se provee a este circuito por el microprocesador y se combina en sincronización compuesta (composite sync).

Se usan contadores de posición horizontal para generar la salida serial de cinco (5) objetos móviles horizontalmente: dos jugadores, dos misiles y una bola. El microprocesador puede sumar o restar en esos contadores de posición para mover los objetos a la izquierda o a la derecha.

El microprocesador determina todas las posiciones y movimientos verticales escribiendo 0 ó 1 en los registros de los objetos antes de cada línea horizontal donde vayan a ser dibujados.

Las paredes, nubes y otros objetos móviles se pueden producir con un registro de baja resolución llamado registro Playfield.

Un registro de colisión de 15 bits puede detectar todas las posibles colisiones entre dos de los seis objetos (los cinco objetos móviles y el Playfield). Este registro de colisión puede ser leído y reseteado por el microprocesador. También hay seis puertos de entrada en este chip que pueden ser leídos por el microprocesador. Estos puertos de entrada y el registro de colisión son las únicas direcciones que puede leer el microprocesador. Todas las otras direcciones son de sólo escritura.

Los registros de color y luminosidad pueden ser programados también por el microprocesador con 8 valores de luminosidad y 15 valores de color. Este chip incluye un desplazador de fase digital (phase shifter) que nos da una salida de color simple con 15 ángulos de fase.

Posee dos circuitos generadores de audio independientes, cada uno con registros de control de frecuencia, contenido de ruido, y volumen programables.

Descripción Detallada

1.0 Datos y Direccionamiento

Los registros en este chip son direccionados por el microprocesador en el espacio de memoria global RAM-ROM. Se incluye una tabla de direcciones de lectura y escritura que describen las funciones direccionables al final. No hay registros que sean de lectura y escritura a la vez. Aunque algunas direcciones son de lectura y escritura, la escritura se hace en un registro y la lectura de datos se hace por un registro diferente.

Si la línea de lectura-escritura está en la low, los datos indicados en esta tabla se escribirán en las locaciones de escritura direccionadas cuando el clock O2 (ver esquemas al final) pase de high a low. Algunos registros son de 8 bits, algunos de sólo 1 bit y los que no tienen bits se llaman registros strobe (no importa lo que se escriba en ellos) que realizan sólo funciones de control cuando se los escribe, como por ejemplo un reset.

Si la línea de la lectura-escritura está en high, las locaciones direccionadas pueden ser leídas por microprocesador en las líneas de datos 6 y 7 mientras el clock O2 está en high.

Las direcciones dadas en la tabla se refieren sólo a las seis verdaderas líneas de dirección. Si alguna de las cuatro líneas de selección del chip es usada para direccionar, las direcciones deben ser modificadas acordeamente.

2.0 Sincronización

2.1 Temporización Horizontal

Un contador hardware en este chip produce toda la temporización horizontal (sync, blank, burst) independientemente del microprocesador. Este contador está manejado por un oscilador externo a 3.58 MHz y puede llegar a una cuenta total de 228.

2.2 Temporización Vertical

Hay un registro direccionable de 1 bit para la sincronización y margen vertical (vertical sync y vertical blank). La temporización para estas tres funciones se establece por microprocesador escribiendo 0 ó 1 en esos bits (VSYNC, VBLANK).

2.3 Sincronización Compuesta

La sincronización horizontal y la salida del bit de sincronización vertical se combinan para producir la sincronización compuesta (composite sync). Esta señal de sincronización maneja la salida del chip hacia una red de vídeo compuesto externa.

2.4 Sincronización del Microprocesador

El oscilador de 3.58 MHz controla un contador cuya salida ($1.19 \text{ MHz} = 3.58 / 3$) se utiliza para manejar una salida llamada O0. Esta provee la entrada de clock de fase cero para el microprocesador el cual produce el sistema de clock O2 (1.19 MHz).

Los ciclos de los programas toman distintas longitudes de tiempo para ejecutarse dependiendo de las decisiones de bifurcación hechas dentro del programa. La sincronización necesaria entre software y hardware se hace mediante un registro de 1 bit llamado WSYNC (wait for sync) cuando el microprocesador termina una rutina como puede ser la carga de registros para una línea horizontal, o como determinar nuevas posiciones verticales durante el vertical blank, puede utilizar WSYNC (poniéndolo en high) el cual maneja el registro RDY del microprocesador (registro de ready) escribiendo un 0 que provoca que el microprocesador se detenga y espere. Como se muestra en la figura 2, WSYNC se vuelve automáticamente a cero (low) en el momento de recibir la próxima señal de vertical blank, liberando la línea RDY y permitiendo que el microprocesador vuelva a calcular y escribir en los registros para esta nueva scanline o par de scanlines.

3.0 Registros Gráficos del Playfield

3.1 Descripción

Los objetos como paredes, nubes y score que no requieren movimiento, se forman con un registro de 20 bits llamado registro Playfield. Este registro (figura 5) recibe información del bus datos a través de tres registros de escritura separados (PF0, PF1 y PF2). El Playfield puede ser escrito en cualquier momento. Para borrarlo, se deben escribir 0 en los tres registros que lo componen.

3.2 Salida Serial Normal

El registro Playfield es escaneado automáticamente (y convertido a salida serial) por un registro de desplazamiento bi-direccional sincronizado de modo tal que los 20 bits se dibujan desde el comienzo hasta exactamente la mitad de la scanline o línea horizontal. Este escaneo o barrido se inicializa cuando termina el horizontal blank (lado izquierdo de la pantalla de televisión). Normalmente, los mismos se repiten en la mitad derecha de la scanline.

3.3 Salida Serial Reflejada

Se puede conseguir un Playfield reflejado escribiendo un 1 en D0 del registro de control (CTRLPF). En este caso el registro de desplazamiento escanea en la dirección opuesta durante la mitad derecha de la scanline, invirtiendo la secuencia de 20 bits.

3.4 Autocontrol

Los tres registros Playfield (PF0, PF1 y PF2) pueden ser escritos en cualquier momento, y si uno de ellos cambia sus valores mientras está siendo escaneado, parte del nuevo valor será mostrado en la scanline.

4.0 Contadores de Posición Horizontal

4.1 Descripción

El Playfield es un registro gráfico fijo, que siempre comienza a dibujarse en el comienzo de cada línea de televisión. El chip TIA también incluye cinco registros gráficos móviles, cuya salida serial es iniciada por cinco contadores de posición horizontal independientes cada vez que estos contadores llegan a 0. Estos contadores de posición son sincronizados continuamente durante la porción visible de cada línea de televisión y el tamaño de su cuenta es exactamente igual al número de clocks que se les da durante este tiempo. Cada registro llega a 0 siempre en el mismo momento durante una scanline y su salida no tendrá movimiento horizontal. Un contador horizontal típico se muestra en la figura 4.

De si se añaden clocks extras a los contadores (o se restan) la llegada a 0 ya no será al mismo momento que antes y el objeto será movido hacia la izquierda o hacia la derecha. Algunos de estos contadores funcionan de modo que pueden mostrar múltiples copias del mismo objeto durante una línea horizontal.

Todos los contadores de posición pueden ser reseteados a 0 por el microprocesador en cualquier momento, escribiendo una instrucción en la dirección de reset de cada objeto (RESBL, RESM0, RESM1, RESP0 y RESP1). Si ocurre un reset durante el horizontal blank, el objeto aparecerá en el lado izquierdo de la pantalla de televisión. Si se hace un reset en el momento apropiado el objeto puede dibujarse en cualquier posición horizontal que deseemos teniendo en cuenta los ciclos de microprocesador ejecutados hasta ese momento.

4.2 Contador de Posición de la Bola

Este contador de posición puede llegar a 0 sólo una vez y no puede mostrar múltiples copias de la bola.

4.3 Contadores de Posición de los Jugadores

Cada contador de posición del jugador puede llegar a 0 tres veces (posee tres decodes). Esto puede controlarse mediante los bits 0, 1 y 2 de los registros de control de número-tamaño (NUSIZ0, NUSIZ1), y mostrar 1, 2 ó 3 copias de un mismo jugador (a diferentes separaciones) a lo largo de la scanline. Estos mismos bits de control se usan para los contadores de posición de los misiles, mostrando la misma cantidad de misiles que de jugadores.

4.4 Contadores de Posición de Misiles

Los contadores de posición de misiles son idénticos a los de los jugadores excepto que tienen otro tipo de reseteo. Estas direcciones de reset extras (RESMP0, RESMP1) escriben un 1 en un registro de un bit cuya salida se utiliza para posicionar el misil (horizontalmente) directamente sobre su correspondiente jugador y para desactivar la salida serial del misil.

5.0 Registros de Movimiento Horizontal

5.1 Descripción General

Hay cinco registros de sólo escritura en este chip que contienen los valores de movimiento horizontal para cada uno de los cinco con objetos móviles. Un registro de movimiento horizontal típico se muestra en la figura 4. El bus de datos (bits 4 al 7) escribe en estos registros (HMP0, HMP1, HMM0, HMM1, HMBL) para cargarlos con los valores de movimiento. Estos registros añaden o restan clocks a los contadores de posición horizontal cuando el microprocesador lo indique utilizando la dirección HMOVE. Estos registros pueden ser seteados a 0 simultáneamente con el comando HMCLR desde el microprocesador, o también escribiendo 0 en cada uno los cinco registros. Cada registro tiene 4 bits y pueden ser cargados con valores positivos (movimiento hacia la izquierda), valores negativos (movimiento hacia la derecha) ó 0 para que no se muevan. Los valores negativos se representan en el formato de complemento de 2.

5.2 Autocontrol

Los cinco registros pueden ser cargados o borrados en casi cualquier momento. Los valores de movimiento que contengan se usan sólo cuando se ejecuta el comando HMOVE, y los cinco valores se usarán simultáneamente en los cinco contadores de posición horizontal a la vez. El único control en esta operación es el comando HMOVE. Este comando debe ser ejecutado por el microprocesador inmediatamente después de un comando WSYNC. Esto asegura que la operación comenzará inmediatamente al principio del horizontal blank, y tiene todo este tiempo para añadir o restar clocks extras a los contadores de posición horizontal. Estos registros no deberían ser modificados al menos durante 24 ciclos después del comando HMOVE.

6.0 Registros Gráficos de los Objetos Móviles

6.1 Descripción General

Hay cinco registros gráficos para los objetos móviles en este chip. El microprocesador carga estos registros en paralelo que son escaneados y convertidos a salida serial igual que el registro Playfield, a diferencia del cual los objetos móviles se comienzan a escanear cuando lo indiquen los contadores de posición horizontal. Un registro típico de gráficos se muestra en la figura 4.

6.2 Gráficos de Misiles

Los registros gráficos para ambos misiles son idénticos y muy simples. Cada uno consistente en un registro de un bit llamado "misile enable" (activación de misil – ENAM0, ENAM1). Estos gráficos se escanean cuando lo indiquen sus correspondientes contadores de posición. Hay cuatro bits de control (bits 4, 5 de NUSIZ0 y NUSIZ1) que pueden alargar estos gráficos para que sean mostrados su durante 1, 2, 4 u 8 clocks de scanline (una línea completa tiene 160 clocks de largo), o sea para que se vean más grandes.

6.3 Gráficos de los Jugadores

Los registros gráficos para ambos jugadores son idénticos y algo complejos. Consisten en un registro de 8 bits en paralelo (GRP0 y GRP1) y un contador de escaneo que convierte los datos paralelos en salida serial. Un registro de control de un bit (REFP0 y PEPF1) determina la dirección (reflexión) del escaneo

paralelo-a-serial, mostrando los bits desde D7 a D0 o desde D0 a D7. Esto permite obtener un reflejo horizontal de cada gráfico de los jugadores sin tener que invertir los datos del microprocesador.

El clock del contador de escaneo o barrido puede ser controlado (mediante 3 bits en NUSIZ0 y NUSIZ1) para provocar que el barrido sea más lento y obtener gráficos de 8, 16 ó 32 clocks de largo en una scanline, tal como pasaba con los misiles. Estos dos últimos registros también se usan para controlar las múltiples copias que pueden obtenerse de cada jugador.

6.4 Retardo Vertical (Vertical Delay)

Cada uno de los registros gráficos de los jugadores está compuesto de dos registros de 8 bits paralelos. El primero (GRP0 y GRP1) puede ser cargado por el bus de datos de 8 bit del microprocesador. El segundo (no le daremos un nombre especial) se carga automáticamente a partir de la salida del primero. La razón de esto es algo complejo llamado vertical delay. Para generar en la pantalla los jugadores, los misiles, la bola y el Playfield se requiere mucho tiempo de microprocesador para cargar los registros de este chip. Para la mayoría de los juegos este tiempo es demasiado largo como para caber en una sola scanline. De hecho, la mayoría de los juegos distribuyen este tiempo en dos scanlines (127 microsegundos). Por lo tanto, los registros gráficos se cargan cada dos líneas, y son usados dos veces por la salida serial entre cada carga. Este tipo de programación limita la resolución vertical de los objetos a múltiplos de dos líneas. Esto además limita la resolución de movimiento vertical a saltos de dos líneas. Para resolver el problema de la resolución vertical no podemos hacer nada; sin embargo sí podemos conseguir movimiento vertical con resolución de una sola línea utilizando el segundo registro gráfico que se carga automáticamente en forma paralela a partir de la salida del primero, una línea después de que primero fue cargado. La salida de este segundo registro gráfico se retrasa entonces verticalmente durante una línea. Un bit de control llamado vertical delay (VDEL0 y VDEL1) selecciona cual de los dos registros se utilizará en la salida serial de la próxima scanline. Si el microprocesador setea este bit de control entre cada imagen de televisión, el objeto se moverá una línea verticalmente la próxima pantalla (gracias al delay). En muchos programas los gráficos del jugador 0 y del jugador 1 se cargarán alternativamente como se muestra en la figura 1. Como se alternan las direcciones GRP0 y GRP1, éstas son retrasadas en una línea una de la otra. La dirección GRP0 puede usarse para cargar el registro gráfico retardado (delay) del jugador 1, y GRP1 puede usarse para cargar el registro gráfico retardado del jugador 0. Los dos bits de vertical delay seleccionan entonces los registros retardados o no retardados para cada jugador para usar en la salida serial.

6.5 Gráficos de la Bola

El registro gráfico de la bola es casi idéntico a los registros de los misiles. También consiste en un bit de activación (ENABL) cuya salida comienza cuando lo indica el contador de posición horizontal de la bola. Tiene dos bits de control (bits 4 y 5 del CTRPF) que pueden darle al gráfico de la bola un tamaño de 1, 2, 4 u 8 clocks de línea horizontal o scanline. A diferencia de los gráficos de los misiles, la bola también utiliza el vertical delay (VDELBL) tal como lo hacen los jugadores. A la salida del primer gráfico se la utiliza para cargar el segundo, una línea después de que primero fue cargado. El VDELBL selecciona cual de los dos gráficos se utiliza en la salida serial del chip. El primer registro gráfico (ENABL) debería ser cargado en el mismo momento que el jugador 0 (GRP0), ya que GRP1 se utiliza para cargar el segundo bit de activación desde la salida del primero en líneas alternadas.

7.0 Banderas de Detección de Colisión

7.1 Definiciones

La salida serial de todos los registros gráficos representan la ubicación horizontal en tiempo real de los objetos en la imagen de televisión. Si alguna de estas salidas ocurre en el mismo momento que otra, se verán superpuestas en la imagen (colisión). Los seis objetos que genera este chip pueden generar 15 posibles colisiones entre otros objetos. Estas son detectadas por 15 compuertas AND cuando ocurren, y son almacenadas en 15 registros individuales, como se muestra en la figura 6.

7.2 Lectura de Colisiones

En microprocesador puede leer 15 bits de colisiones en las líneas de datos 6 y 7 direccionando dos de los bits al mismo tiempo. Esto puede hacerse en cualquier momento pero generalmente se hace entre cada imagen de televisión (durante el vertical blank) luego de que todas las posibles colisiones hayan ocurrido.

7.3 Reset

Todos los bits de colisión pueden ser simultáneamente reseteados por el microprocesador usando la dirección CXCLR. Esto se hace generalmente cuando está por terminar el vertical blank cuando todas las colisiones han sido comprobadas.

8.0 Puertos de Entrada

8.1 Descripción General

Hay seis puertos de entrada en este chip cuyos estados lógicos se pueden leer en la línea de datos 7 utilizando las direcciones INPT0 a INPT5. Estos seis puertos se dividen en dos tipos, "dumped" (o de descarga, porque utilizan un capacitor) y "latched" (o de llaves, porque verifican el estado o nivel lógico). Ver la figura 8.

8.2 Puertos de Entrada Dumped (INPT0 a INPT3)

Estos cuatro puertos de entrada se usan normalmente para leer la posición de los controladores paddle desde un circuito externo compuesto de potenciómetro y capacitor. Para descargar estos capacitores cada uno de los puertos de entrada contiene un transistor, el cual se activa (conectando a masa los puertos) escribiendo en el bit 7 del registro VBLANK. Cuando este bit de control se limpia, los potenciómetros comienzan a cargar los capacitores y el microprocesador mide el tiempo requerido para detectar un 1 lógico en cada puerto de entrada (cada controlador paddle posee un potenciómetro en la palanca).

Cuando el bit 7 del VBLANK está en 0, los cuatro puertos quedan como puertos de entrada de alta impedancia para propósitos generales. Cuando este bit está en 1, los puertos quedan conectados a masa.

8.3 Puertos de Entrada Latched (INPT4 e INPT5)

Estos dos puertos de entrada consisten en llaves que pueden ser activadas o desactivadas escribiendo en el bit 6 del registro VBLANK.

Cuando están desactivadas, los puertos quedan como de entrada y el microprocesador les puede leer su estado lógico directamente.

Cuando están activadas, estas llaves almacenan señales negativas (niveles lógicos 0) en los dos puertos de entrada, y la dirección de los puertos nos dará la lectura de las llaves en lugar de la de los puertos mismos.

En un principio, cuando se activan estas llaves, mantienen el valor positivo tanto tiempo como lo hacen los puertos de entrada (valor lógico 1). Una señal de 0 en los puertos de entrada provocará que las llaves se pongan en 0, y así se quedarán hasta que se desactiven, incluso si los puertos vuelven a estar en 1. Ambas llaves pueden desactivarse simultáneamente escribiendo un 0 en el registro de 6 bits VBLANK.

(Llaves = Latches)

8.4.0 Encoder de Prioridad

8.4.1 Propósito

Como ya dijimos en la sección de colisiones, salidas seriales simultáneas de los registros gráficos provocarán superposiciones de objetos en la pantalla de televisión. Para saber cuál de los objetos es el que será mostrado es necesario establecer prioridades entre los objetos para cuando se superponen. El encoder de prioridad se muestra en la figura 3.

8.4.2 Asignación de Prioridad

Si en la salida serial no se encuentra ningún objeto, lo que se muestra es el color y luminosidad del fondo. El fondo (BK) tiene la menor prioridad y sólo aparece cuando no hay objetos en la salida. Para simplificar la lógica, cada misil tiene la misma prioridad e igual valor de color y luminosidad que su correspondiente jugador (P0, M0) y la bola tiene la misma prioridad y valor de color y luminosidad que el Playfield (PF, BL).

Siguiente tabla ilustra la asignación normal de prioridades:

prioridad más alta	P0, M0
segunda prioridad	P1, M1
tercera prioridad	PF, BL
última prioridad	BK

Los objetos con más alta prioridad se mostrarán en frente de los objetos con menor prioridad. Los jugadores por lo tanto se moverán delante del Playfield (nubes, paredes, etc.)

8.4.3 Control de Prioridad

Hay dos bits de control del Playfield que afectan la prioridad, uno se llama prioridad de Playfield (PFP – bit 2 del CTRLPF) y el otro se llama score (bit 1 del CTRLPF). Cuando se escribe un 1 en el PFP se modifica la asignación de prioridad según la siguiente tabla:

prioridad más alta	PF, BL
segunda prioridad	P0, M0
tercera prioridad	P1, M1
última prioridad	BK

Los jugadores se moverán detrás del Playfield. Cuando se escribe un 1 en el bit de control score, el Playfield es forzado a tomar el color y luminosidad del jugador 0 en la mitad izquierda de la pantalla y del jugador 1 en la mitad derecha. Esto puede ser usado para mostrar puntuaciones e identificar a cuál de los jugadores corresponde a cada una. El encoder de prioridad produce cuatro líneas de selección de registros (mostradas en la figura 3) que son mutuamente exclusivas. Estas cuatro líneas seleccionan cada una al fondo, jugador 0, jugador 1 o Playfield y sólo una puede ser verdadera en un momento dado.

9.0 Registros de Color y Luminosidad

9.1 Descripción

Hay cuatro registros (mostrados en la figura 3) que controlan el color y la luminosidad. En cada uno de estos registros se debe escribir un código de cuatro bits para el color y un código de tres bits para la luminosidad (los registros son: COLUP0, COLUP1, COLUPF, CLUBK) Esto lo hace el microprocesador en cualquier momento. Estos códigos (que representan 16 valores de color y 8 valores de luminosidad) se muestran en la lista de Funciones Detallas de las Direcciones de Escritura.

9.2 Multiplexación

La salida serial de gráficos de los seis objetos es examinada por el encoder de prioridad, el cual activa una de las cuatro líneas de selección a través de un multiplexor 4 x 7. Este multiplexor (mostrado en la figura 3) seleccionará entonces uno de los cuatro registros color y luminosidad como una salida de 7 líneas. Tres de estas líneas codifican en binario la luminosidad y van directamente a la salida del chip. Las otras cuatro líneas van al desplazador de fase de color.

10.0 Desplazador de Fase de Color (Color Phase Shifter)

Esta parte del chip (mostrada en la figura 3) produce una salida de color de referencia (color burst) durante el horizontal blank y durante la parte visible de la pantalla produce una salida desplazada en fase con respecto a la de referencia. La cantidad de fase desplazada determina el color y es seleccionada por las cuatro líneas de código de color en el multiplexor de color-luminosidad. El código binario 0 (0000) significa que no hay color. El código 1 (0001) selecciona el color oro (igual fase que el color de la referencia). Utilizando un código desde 2 (0010) hasta 15 (1111) la fase se desplazará desde 0 hasta casi 360 grados permitiendo una selección de 15 colores en total sobre la paleta de color del televisor.

11.0 Circuitos de Audio

Este chip incorpora dos circuitos de audio. Son idénticos y completamente independientes, y sus salidas pueden ser combinadas externamente en un solo parlante (speaker). Cada circuito de audio consiste en tres partes que se describen a continuación, y en la figura 7.

11.1 Selección de Frecuencia

Un pulso de reloj (aproximadamente a 30 KHz) pasa desde el contador de sincronización horizontal a través de un circuito divisor por N, el cual se controla con el código de salida de un registro de frecuencia de 5 bits (AUDF). Este registro puede ser cargado por el microprocesador en cualquier momento, y provocar que el pulso de 30 KHz sea dividido por 1 (código 00000) hasta 32 (código 11111). Con esto podemos ajustar digitalmente el pulso desde aproximadamente 30 KHz a 1 KHz y usar esto en el reloj del generador de tono-ruido.

11.2 Generador de Tono-Ruido

Este circuito contiene un contador desplazador de 9 bits que puede ser controlado por el código de salida del registro de control de audio de 4 bits (AUDC), y su pulso es controlado por el circuito selector de frecuencia. El registro de control puede ser cargado por el microprocesador en cualquier momento, y selecciona diferentes golpes de retorno (feedback taps) del contador desplazador y distintos tamaños de ciclos para producir varias calidades de tono y de ruido.

11.3 Selección del Volumen

La salida del contador desplazador se usa para controlar la salida de audio a través de cuatro transistores graduados en distintos tamaños. Cada transistor es dos veces el anterior y se activa con un bit proveniente del registro del volumen del audio (AUDV). Este registro de volumen puede ser cargado por el microprocesador en cualquier momento. Se cargan códigos en binario desde 0 hasta 15 para provocar que los transistores se activen en una secuencia binaria. La salida del contador desplazador sin embargo puede bajar la salida de audio mediante 16 niveles de intensidad seleccionables.

Figura 1. Vertical Delay

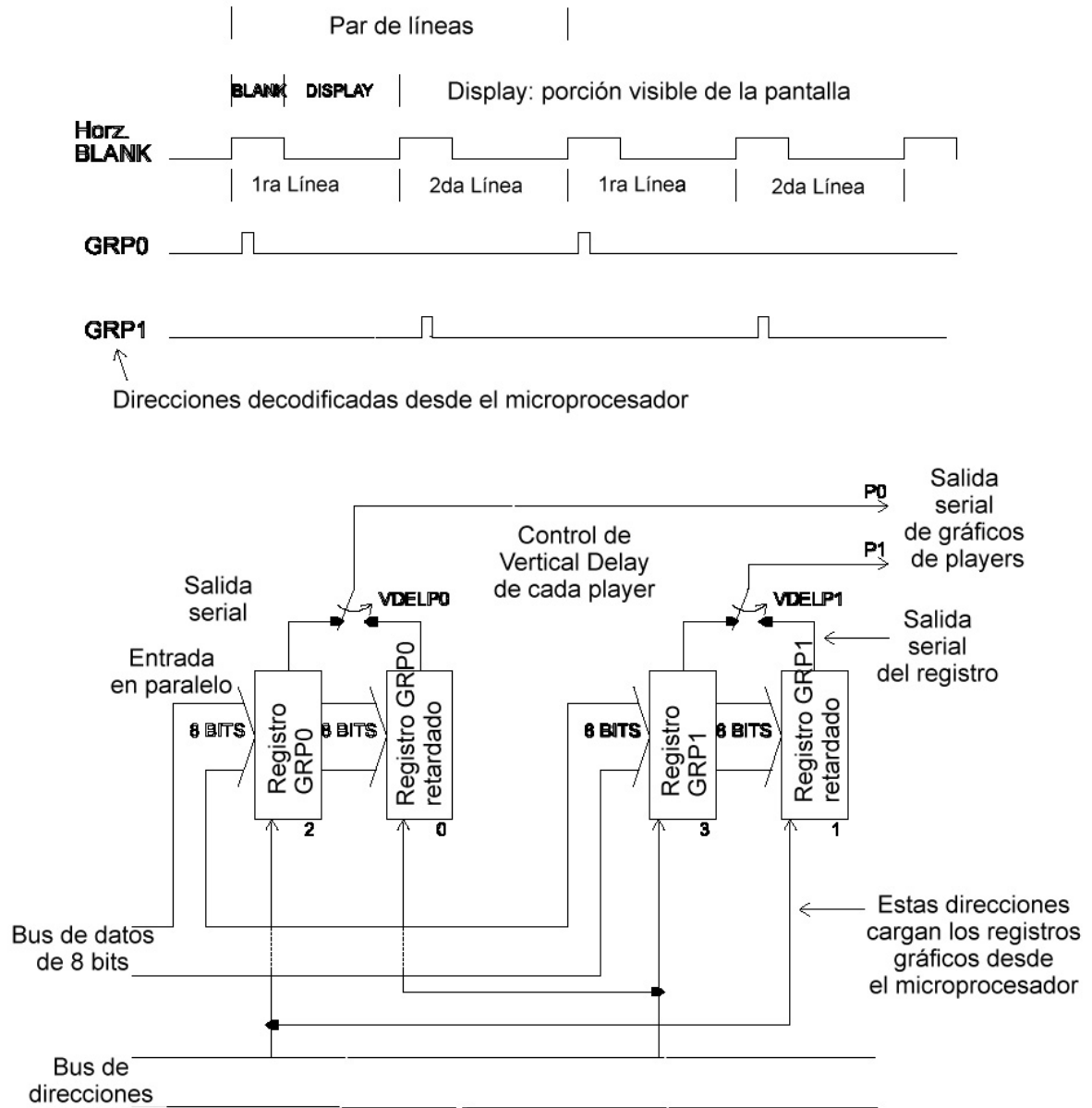


Figura 2. Sincronización

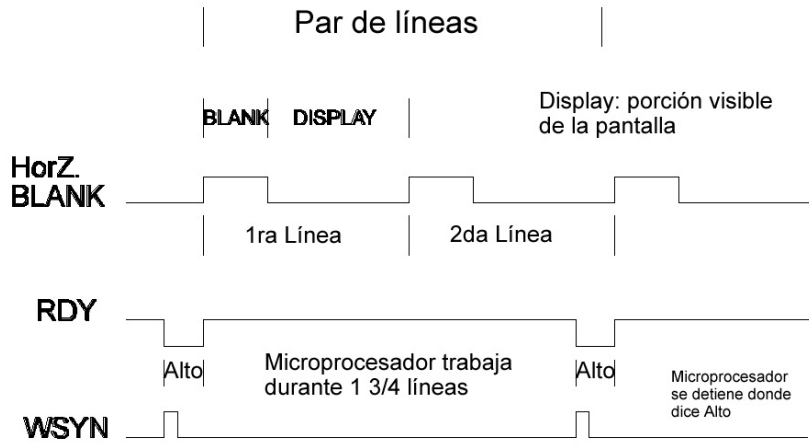


Figura 3. Color-Luminosidad

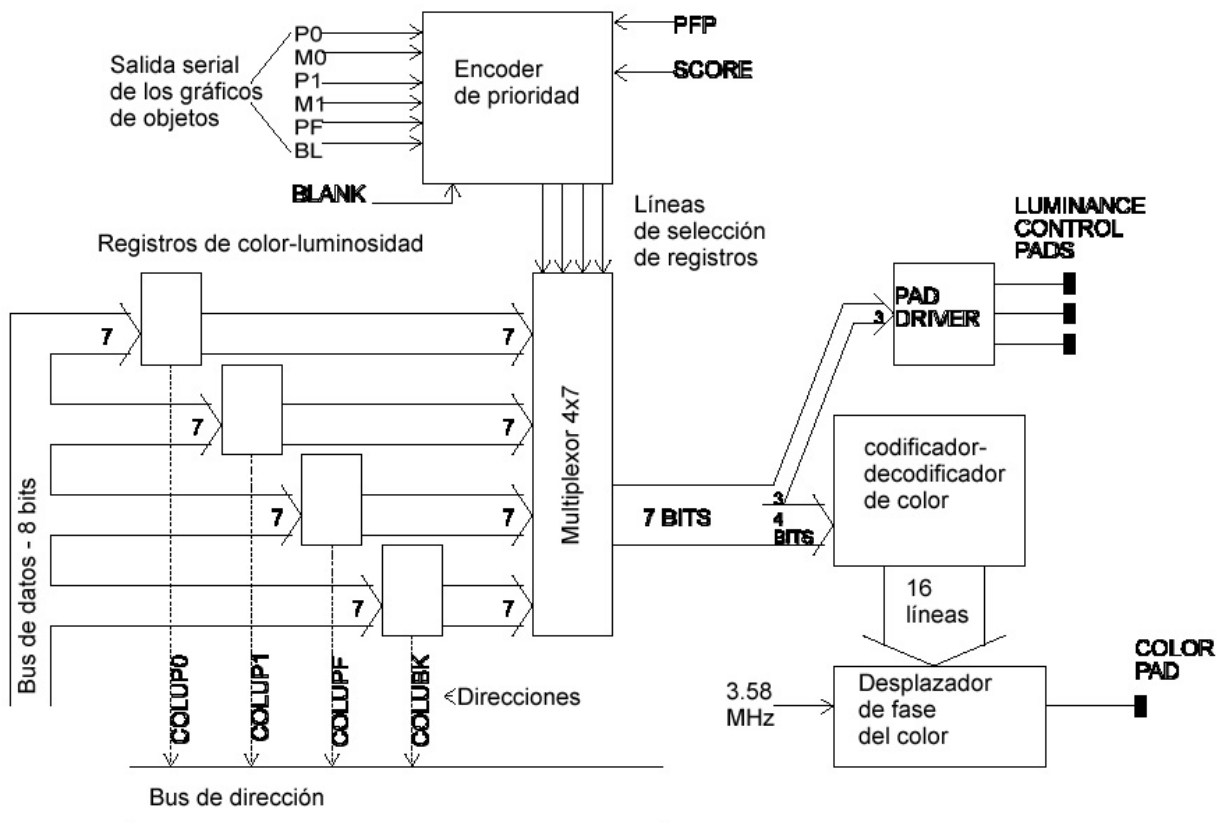


Figura 4. Circuito de Movimiento Horizontal Típico

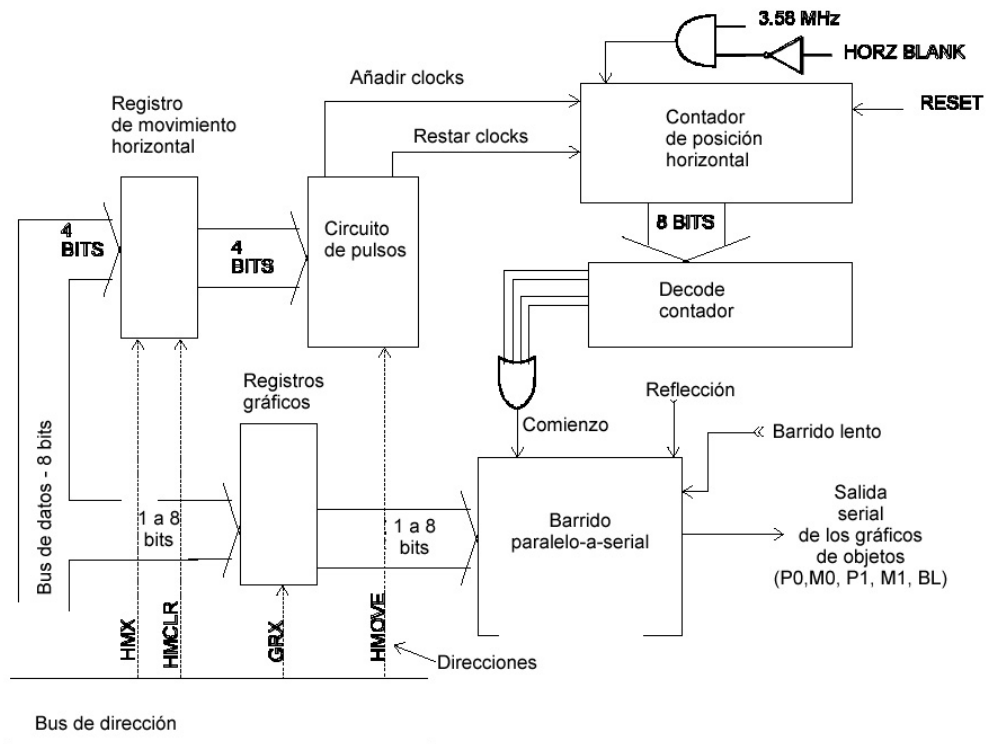


Figura 5. Gráficos del Playfield

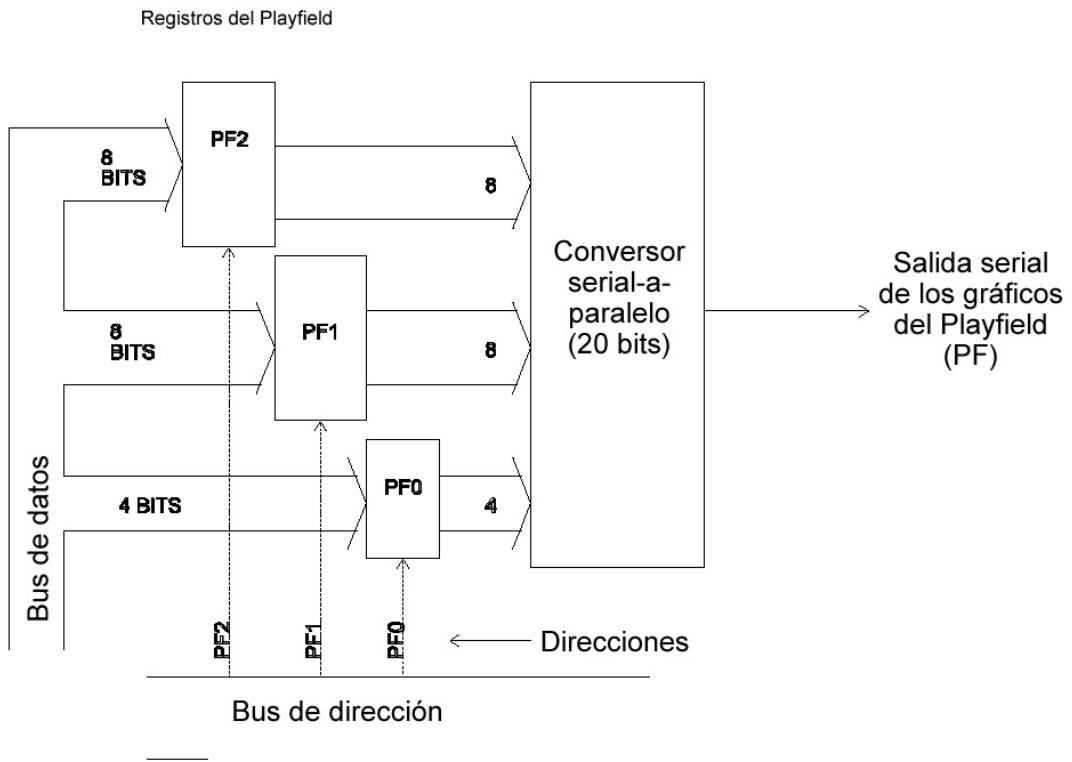


Figura 6. Detección de Colisiones

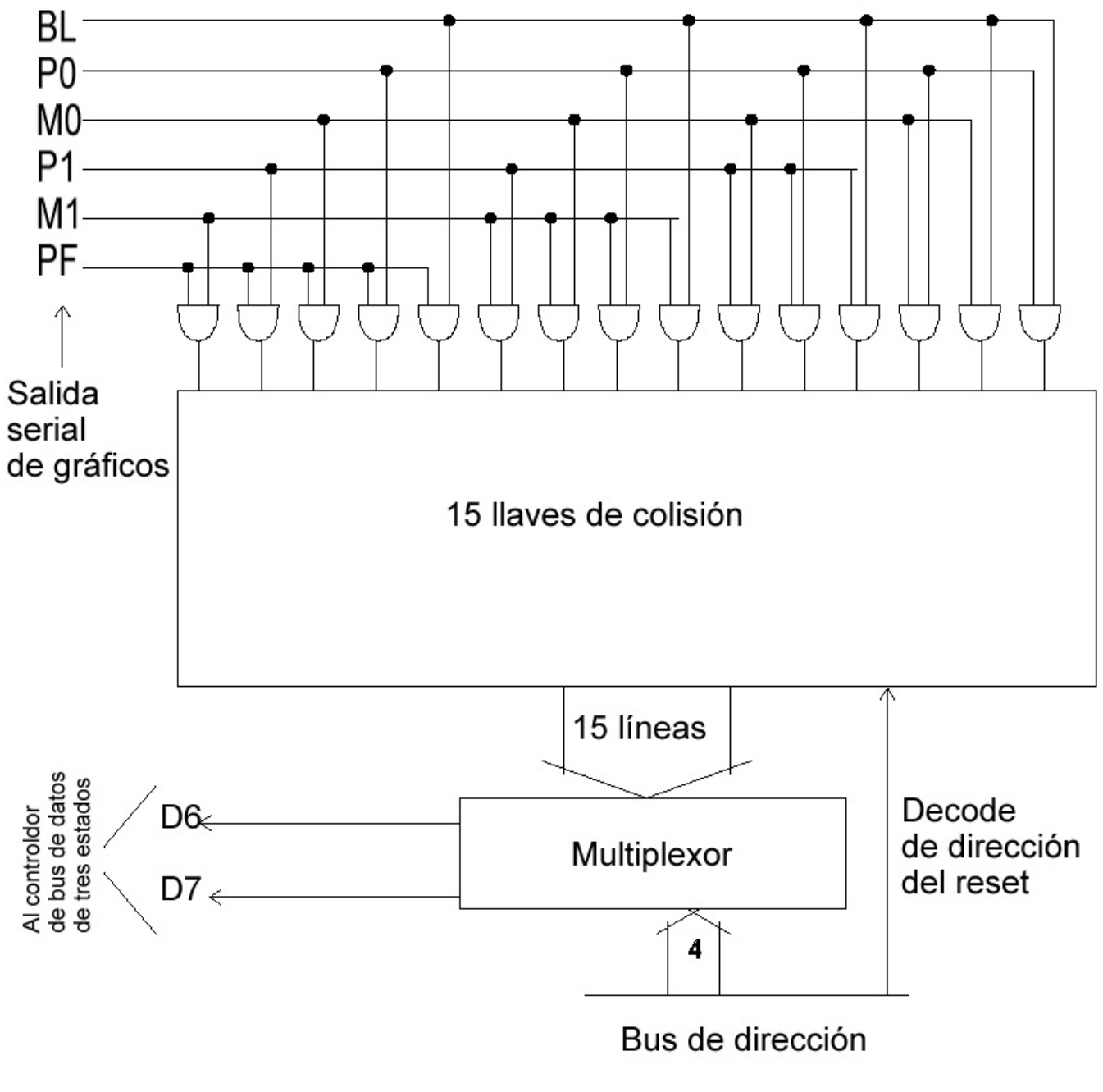


Figura 7. Circuito de Audio

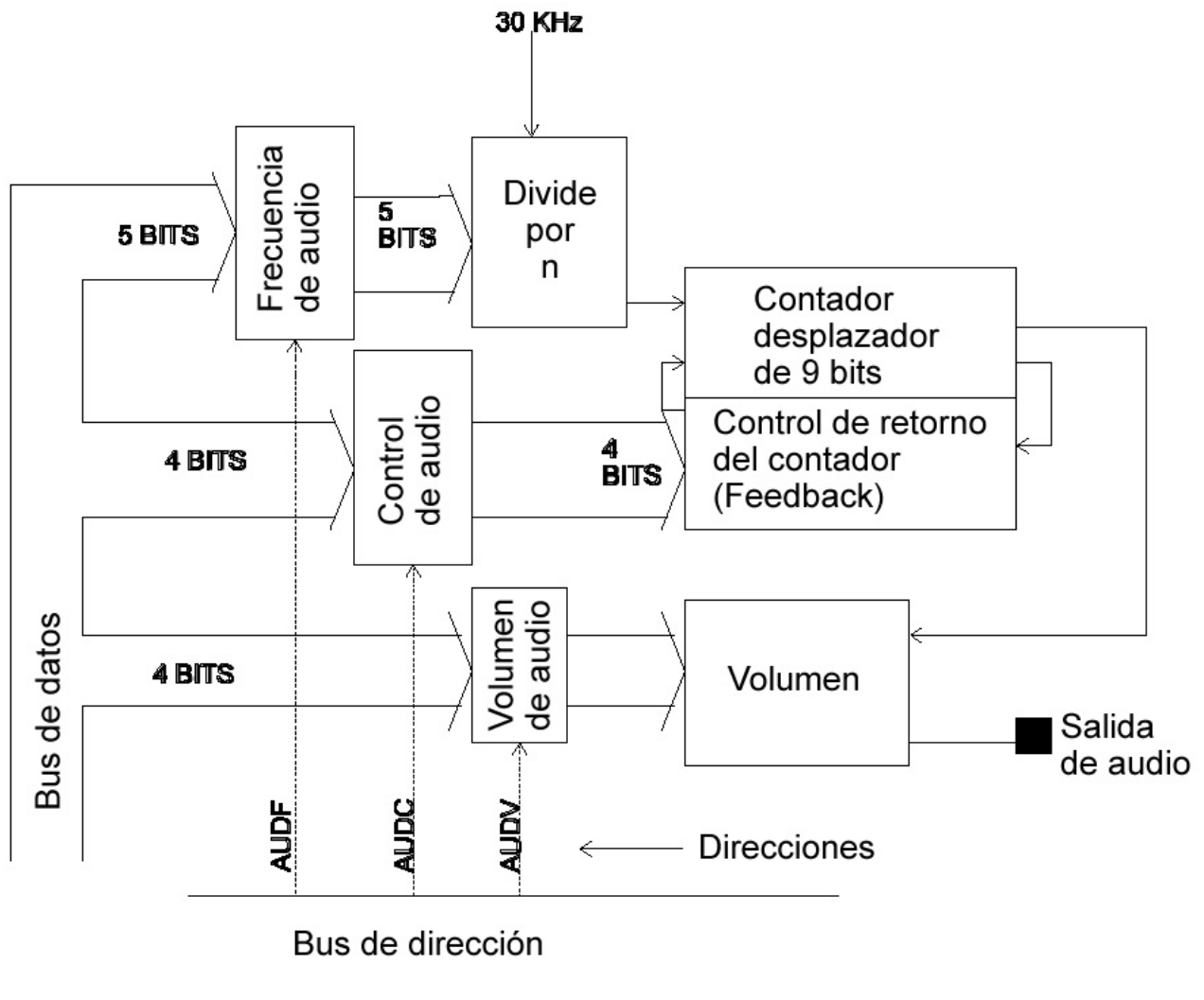


Figura 8. Puertos de Entrada

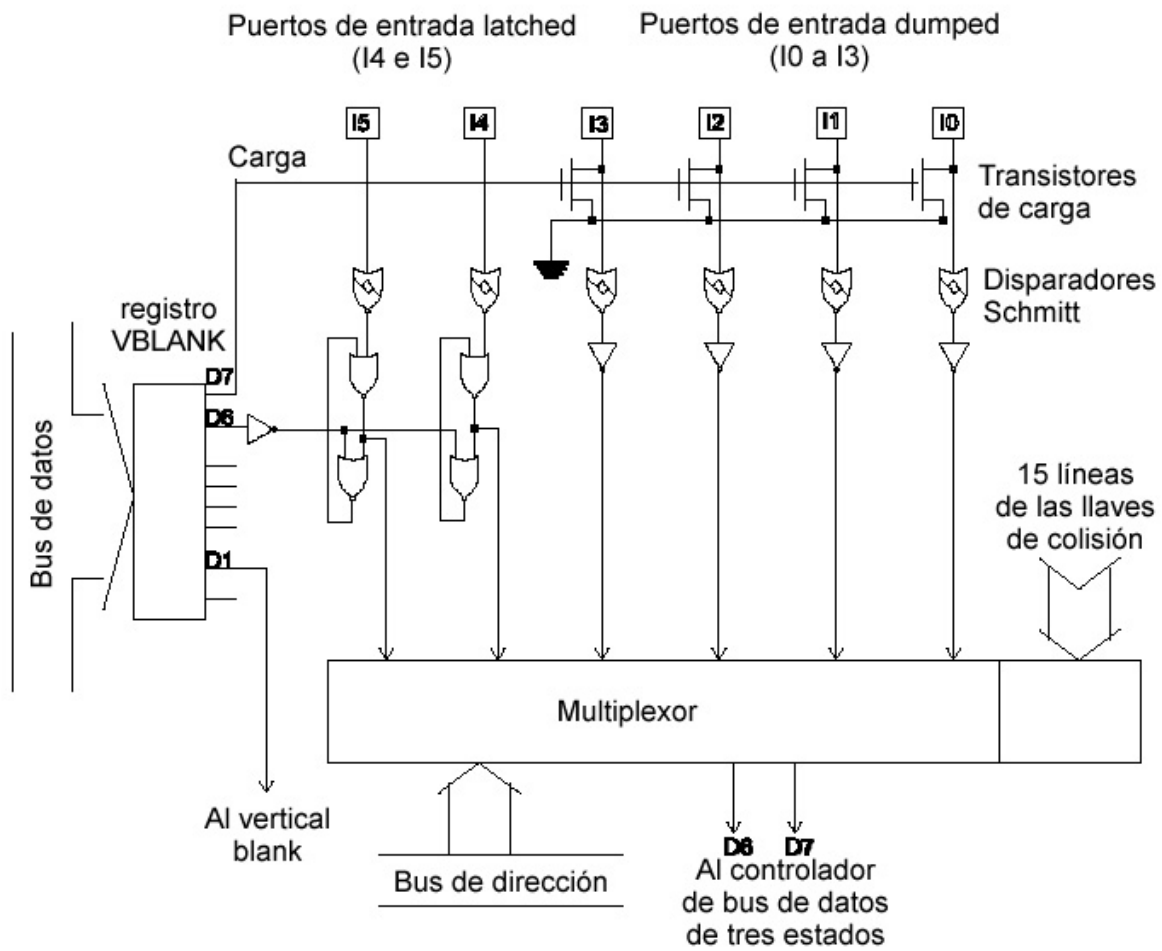
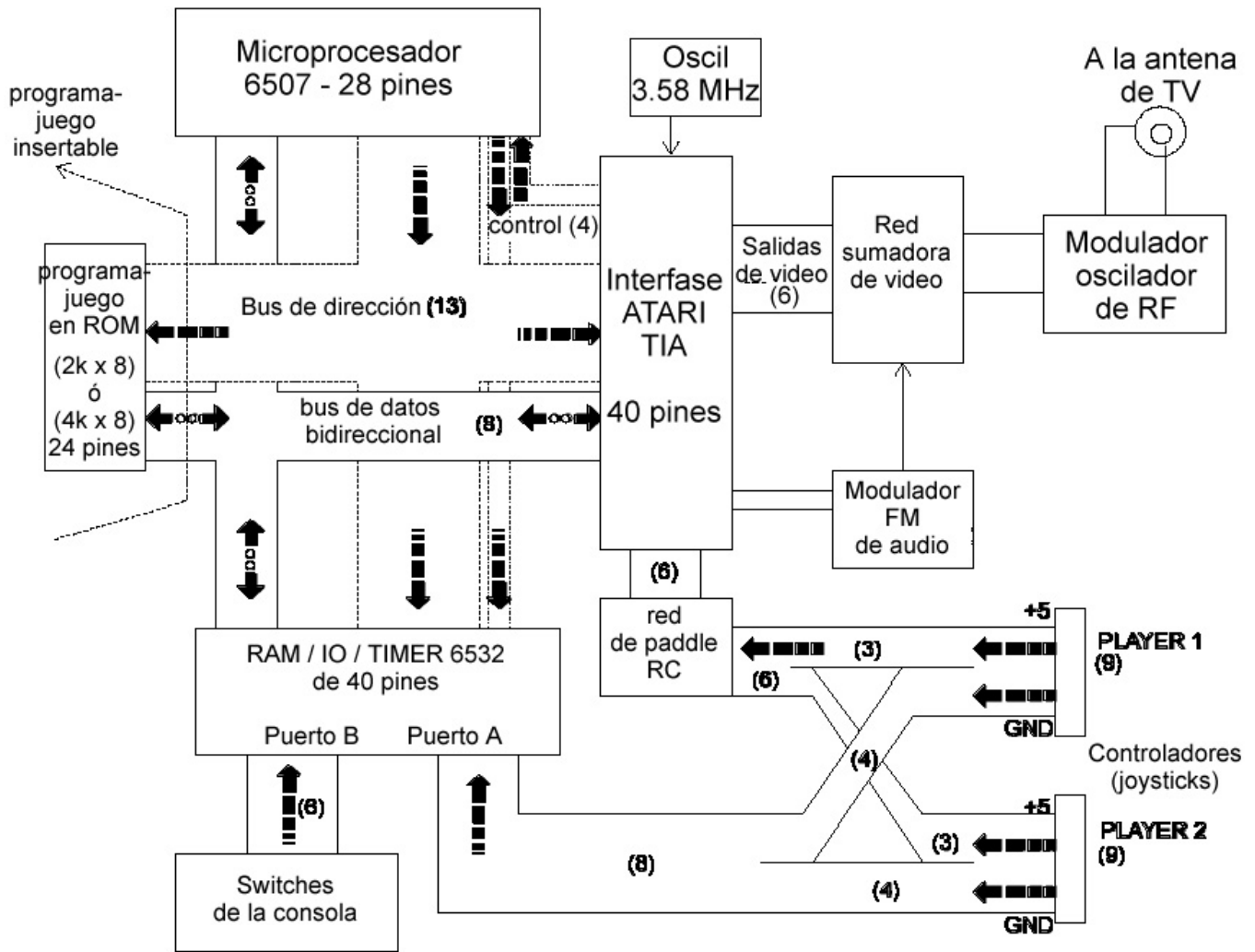


Figura 9. Sistema Completo



Funciones Detallas de las Direcciones de Escritura

WSYNC (wait for sync - esperar sincronización)

Esta dirección congela al microprocesador seteando en 0 su línea RDY, la cual se vuelve a 1 cuando finaliza el horizontal blank.

No se usan datos

--	--	--	--	--	--	--	--

RSYNC (reset sync - resetear sincronización)

Esta dirección resetea el contador de sincronización horizontal para definir el comienzo del horizontal blank, y también es usado en pruebas del chip.

No se usan datos

--	--	--	--	--	--	--	--

VSYNC

Esta dirección controla la sincronización vertical escribiendo el bit D1.

						D1	
--	--	--	--	--	--	----	--

D1 [1= comenzar sincronización vertical; 0= detener sincronización vertical]

VBLANK

Controla la porción llamada vertical blank de la imagen y los transistores de los puertos de entrada escribiendo los bits D7, D6 y D1 del registro.

D7	D6					D1	
----	----	--	--	--	--	----	--

D1 [1= comenzar vert. Blank; 0= detener vert. Blank]

D6 [1= activar llaves (latches) I4 e I5; 0= desactivar llaves (latches) I4 e I5]

D7 [1= descargar I0, I1, I2 e I3 (a masa); 0= desconectarlos de masa]

Nota: desactivar las llaves (D6 = 0) también las resetea al valor lógico "true" (1 lógico)

PF0 (PF1, PF2)

Se utilizan para escribir en los registros gráficos del Playfield

PF0

D7	D6	D5	D4				
----	----	----	----	--	--	--	--

PF1

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

PF2

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

Salida serial de los registros del Playfield

Control de reflexión: REF=0

4	7	7	0	0	7	4	7	7	0	0	7
PF0	PF1		PF2		PF0	PF1		PF2			

Control de reflexión: REF=1

4	7	7	0	0	7	7	0	0	7	7	4
PF0	PF1		PF2		PF2		PF1		PF0		

Cada bit = 4 clocks

CTRLPF

Se utiliza para escribir en el registro de control del PF, un 1 lógico causa las acciones que se describen a continuación:

		D5	D4		D2	D1	D0
--	--	----	----	--	----	----	----

[PF= Playfield]

D0= REF (PF reflejo)

D1= SCORE (la mitad izquierda del PF toma el color del jugador 0, y la derecha toma el del jugador 1)

D2= PFP (el PF adquiere prioridad sobre los jugadores y éstos se mueven detrás del PF)

D4 y D5= Tamaño de la Bola:

D4	D5	Tamaño
0	0	1 clock
0	1	2 clocks
1	0	4 clocks
1	1	8 clocks

NUSIZ0 (NUSIZ1)

Se utiliza para controlar la cantidad y el tamaño de los jugadores y misiles.

		D5	D4		D2	D1	D0
--	--	----	----	--	----	----	----

Tamaño de los misiles:

D4	D5	Tamaño
0	0	1 clock
0	1	2 clocks
1	0	4 clocks
1	1	8 clocks

Cantidad de jugadores/misiles y tamaño de jugadores:

D2	D1	D0		Descripción
0	0	0		1 copia
0	0	1		2 copias cerca
0	1	0		2 copias a media distancia
0	1	1		3 copias cerca
1	0	0		2 copias alejadas
1	0	1		tamaño doble
1	1	0		3 copias a media distancia
1	1	1		tamaño cuádruple

RESP0 (RESP1, RESM0, RESM1, RESBL)

Estas direcciones sirven para resetear los jugadores, misiles y la bola. Cada objeto comenzará su gráfico serial en el lugar de la línea horizontal donde ocurrió el reset.

No se usan datos

--	--	--	--	--	--	--	--	--	--

RESMP0 (RESMP1)

Se utilizan para resetear la ubicación horizontal de un misil en el centro de su correspondiente jugador. Mientras este control almacene un 1, el misil tendrá siempre su posición horizontal en el centro de su jugador y su salida gráfica se desactiva, hasta que el bit cambie a 0 provocando que el misil se muestre y pueda ser movido.

						D1	
--	--	--	--	--	--	----	--

D1= RESMP (reset de misil/jugador)

HMOVE

Esta dirección causa que se accionen los registros de movimiento horizontal durante el horizontal blank en que ocurra. Debe ocurrir al principio del horizontal blank para que se puedan generar los pulsos de clocks extras en los contadores de posición horizontal, por lo cual debe utilizarse inmediatamente después de un comando WSYNC.

No se usan datos

--	--	--	--	--	--	--	--	--	--

HMCLR

Esta dirección limpia todos los registros de movimiento horizontal seteándolos a 0 (sin movimiento).

No se usan datos

--	--	--	--	--	--	--	--	--	--

HMP0 (HMP1, HMM0, HMM1, HMBL)

Estas direcciones escriben los valores de movimiento horizontal en los correspondientes registros. El movimiento horizontal se produce cuando el comando HMOVE es ejecutado.

D7	D6	D5	D4				
0	1	1	1	+7			
0	1	1	0	+6			
0	1	0	1	+5			
0	1	0	0	+4			
0	0	1	1	+3			
0	0	1	0	+2			
0	0	0	1	+1			
0	0	0	0	0			
1	1	1	1	-1			
1	1	1	0	-2			
1	1	0	1	-3			
1	1	0	0	-4			
1	0	1	1	-5			
1	0	1	0	-6			
1	0	0	1	-7			
1	0	0	0	-8			

Advertencia: Estos registros no debieran ser modificados sino hasta después de que pasaron al menos 24 ciclos desde que se ejecutó el último comando HMOVE. De otra forma, se obtendrán valores impredecibles.

ENAM0 (ENAM1, ENABL)

Estas direcciones escriben el bit D1 de los registros gráficos de los misiles y la bola.

						D1	
--	--	--	--	--	--	----	--

D1 [0= objeto habilitado; 1= objeto deshabilitado]

GRP0 (GRP1)

Estas direcciones escriben los registros gráficos de los jugadores.

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

Nota: la salida serial comienza en D7 a menos que REFP0 (REFP1) = 1.

REFP0 (REFP1)

Estas direcciones escriben el bit D3 del registro de reflexión de los jugadores.

				D3			
--	--	--	--	----	--	--	--

D3 [0= no reflejo, D7 del GRPx a la izquierda; 1= reflejo, D0 del GRPx a la izquierda]

VDELP0 (VDELP1, VDELBL)

Estas direcciones escriben el bit D0 en los registros de vertical delay de la bola o los jugadores para retardar su salida gráfica una línea.

							D0
--	--	--	--	--	--	--	----

D0 [0= sin delay; 1= con delay]

CXCLR

Esta dirección setea todos los 15 bits de colisión a 0 (sin colisión).

No se usan datos

--	--	--	--	--	--	--	--

COLUP0 (COLUP1, COLUPF, COLUBK)

Estas direcciones escriben en los registros de color y luminosidad de estos cuatro objetos.

COLOR	D7	D6	D5	D4	D3	D2	D1		LUM
Gris	0	0	0	0	0	0	0		Negro
Oro	0	0	0	1	0	0	1		Gris oscuro
Naranja	0	0	1	0	0	1	0		
Rojo/Naranja	0	0	1	1	0	1	1		Gris
Rosa	0	1	0	0	1	0	0		
Púrpura	0	1	0	1	1	0	1		
Púrpura/azul	0	1	1	0	1	1	0		Gris claro
Azul	0	1	1	1	1	1	1		Blanco
Azul	1	0	0	0					
Azul claro	1	0	0	1					
Turquesa	1	0	1	0					
Verde/azul	1	0	1	1					
Verde	1	1	0	0					
Amarillo/verde	1	1	0	1					
Naranja/verde	1	1	1	0					
Naranja claro	1	1	1	1					

AUDF0 (AUDF1)

Estas direcciones escriben en los registros divisores de la frecuencia de audio.

	D4	D3	D2	D1	D0	30 KHz divididos por:
	0	0	0	0	0	sin división
	0	0	0	0	1	divididos por 2
	0	0	0	1	0	divididos por 3

	1	1	1	1	0	divididos por 31
	1	1	1	1	1	divididos por 32

AUDC0 (AUDC1)

Estas direcciones escriben en los registros de control de audio para el contenido de ruido y división adicional de la salida de audio.

				D3	D2	D1	D0	Tipo de ruido o división:
				0	0	0	0	seteado a 1
				0	0	0	1	4 bit poli
				0	0	1	0	div 15 -> 4 bit poli
				0	0	1	1	5 bit poli -> 4 bit poli
				0	1	0	0	div 2: tono puro
				0	1	0	1	div 2: tono puro
				0	1	1	0	div 31: tono puro
				0	1	1	1	5 bit poli -> div 2
				1	0	0	0	9 bit poli (ruido blanco)
				1	0	0	1	5 bit poli
				1	0	1	0	div 31: tono puro
				1	0	1	1	setea últimos 4 bits a 1
				1	1	0	0	div 6: tono puro
				1	1	0	1	div 6: tono puro
				1	1	1	0	div 93: tono puro
				1	1	1	1	5 bit poli div 6

AUDV0 (AUDV1)

Estas direcciones escriben en los registros de control del volumen de audio que setean la bajada de impedancia del controlador de salida de audio.

				D3	D2	D1	D0	Salida de audio:
				0	0	0	0	sin sonido
				0	0	0	1	volumen más bajo
				0	0	1	0	...
			
				1	1	1	0	...
				1	1	1	1	volumen más alto

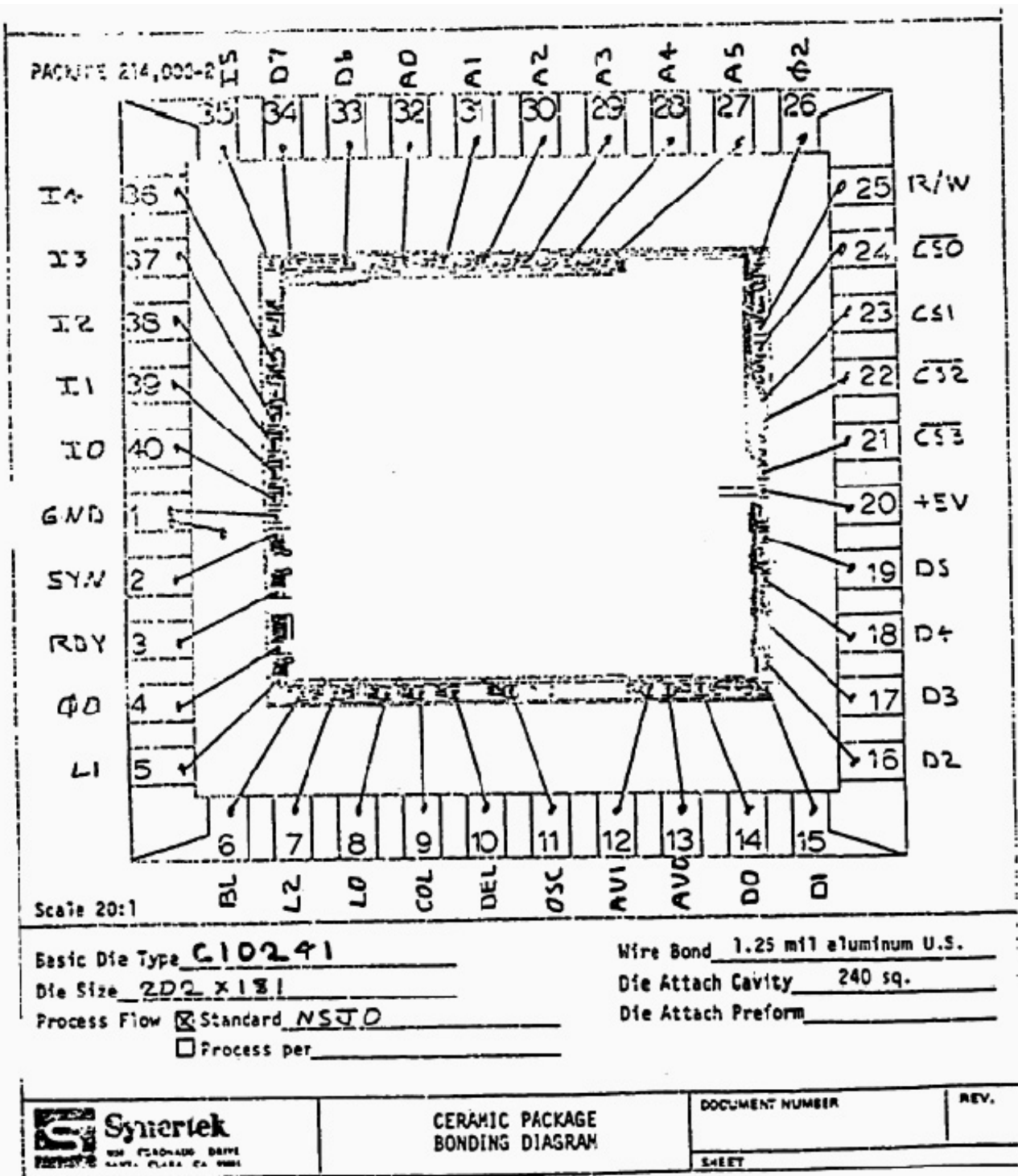
Sumario de las Direcciones de Escritura

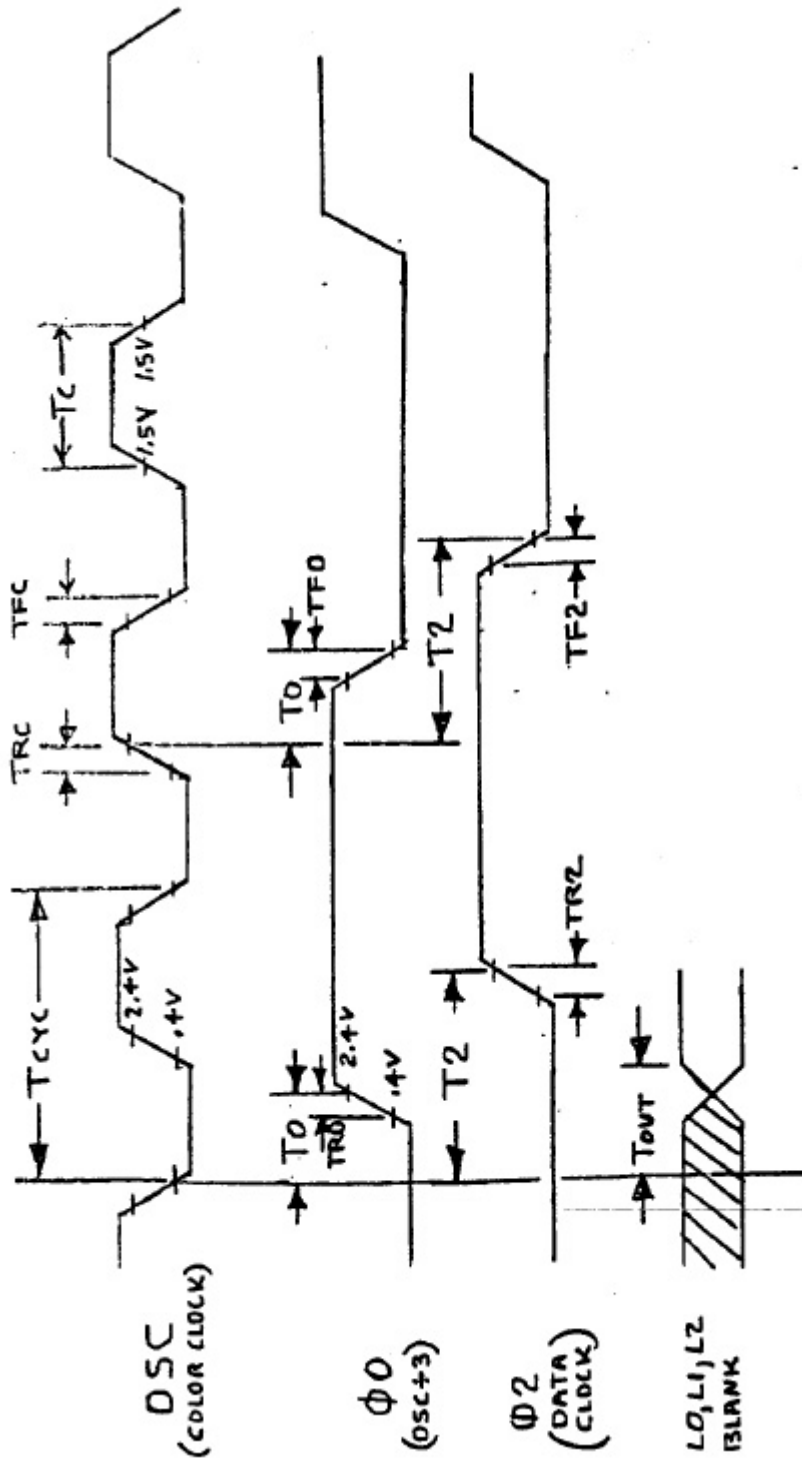
Dirección de 6 bits	Nombre de la dirección	7	6	5	4	3	2	1	0	Función
00	VSYNC							1		Setea el vertical sync
01	VBANK		1	1				1		Setea el vertical blank
02	WSYNC		s	t	r	o	b	e		Esperar comienzo de horizontal blank
03	RSYNC		s	t	r	o	b	e		Resetear contador de sincronización horizontal
04	NUSIZ0			1	1	1	1	1	1	Número-tamaño de jugador-misil 0
05	NUSIZ1			1	1	1	1	1	1	Número-tamaño de jugador-misil 1
06	COLUP0	1	1	1	1	1	1	1		Color-luminosidad del jugador 0
07	COLUP1	1	1	1	1	1	1	1		Color-luminosidad del jugador 1
08	COLUPF	1	1	1	1	1	1	1		Color-luminosidad del Playfield
09	COLUBK	1	1	1	1	1	1	1		Color-luminosidad del fondo
0A	CTRLPF			1	1		1	1	1	Control de Playfield, tamaño de la bola y colisiones
0B	REFP0					1				Jugador 0 reflejo
0C	REFP1					1				Jugador 1 reflejo
0D	PF0	1	1	1	1					Byte 0 del registro Playfield
0E	PF1	1	1	1	1	1	1	1	1	Byte 1 del registro Playfield
0F	PF2	1	1	1	1	1	1	1	1	Byte 2 del registro Playfield
10	RESP0		s	t	r	o	b	e		Reset jugador 0
11	RESP1		s	t	r	o	b	e		Reset jugador 1
12	RESM0		s	t	r	o	b	e		Reset misil 0
13	RESM1		s	t	r	o	b	e		Reset misil 1
14	RESBL		s	t	r	o	b	e		Reset bola
15	AUDC0					1	1	1	1	Control de audio 0
16	AUDC1				1	1	1	1	1	Control de audio 1
17	AUDF0				1	1	1	1	1	Frecuencia de audio 0
18	AUDF1					1	1	1	1	Frecuencia de audio 1
19	AUDV0					1	1	1	1	Volumen de audio 0
1A	AUDV1					1	1	1	1	Volumen de audio 1
1B	GRP0	1	1	1	1	1	1	1	1	Gráficos del jugador 0
1C	GRP1	1	1	1	1	1	1	1	1	Gráficos del jugador 1
1D	ENAM0							1		Gráficos del misil 0 (habilitación)
1E	ENAM1							1		Gráficos del misil 1 (habilitación)
1F	ENABL							1		Gráficos de la bola (habilitación)
20	HMP0	1	1	1	1					Movimiento horizontal del jugador 0
21	HMP1	1	1	1	1					Movimiento horizontal del jugador 1
22	HMM0	1	1	1	1					Movimiento horizontal del misil 0
23	HMM1	1	1	1	1					Movimiento horizontal del misil 1
24	HMBL	1	1	1	1					Movimiento horizontal de la bola
25	VDELP0								1	Vertical delay del jugador 0
26	VDEL01								1	Vertical delay del jugador 1
27	VDELBL								1	Vertical delay de la bola
28	RESMP0							1		Resetea misil 0 al jugador 0
29	RESMP1							1		Resetea misil 1 al jugador 1
2A	HMOVE		s	t	r	o	b	e		Accionar movimiento horizontal
2B	HMCLR		s	t	r	o	b	e		Limpiar registros de movimiento horizontal
2C	CXCLR		s	t	r	o	b	e		Limpiar registros de colisiones

Sumario de las Direcciones de Lectura

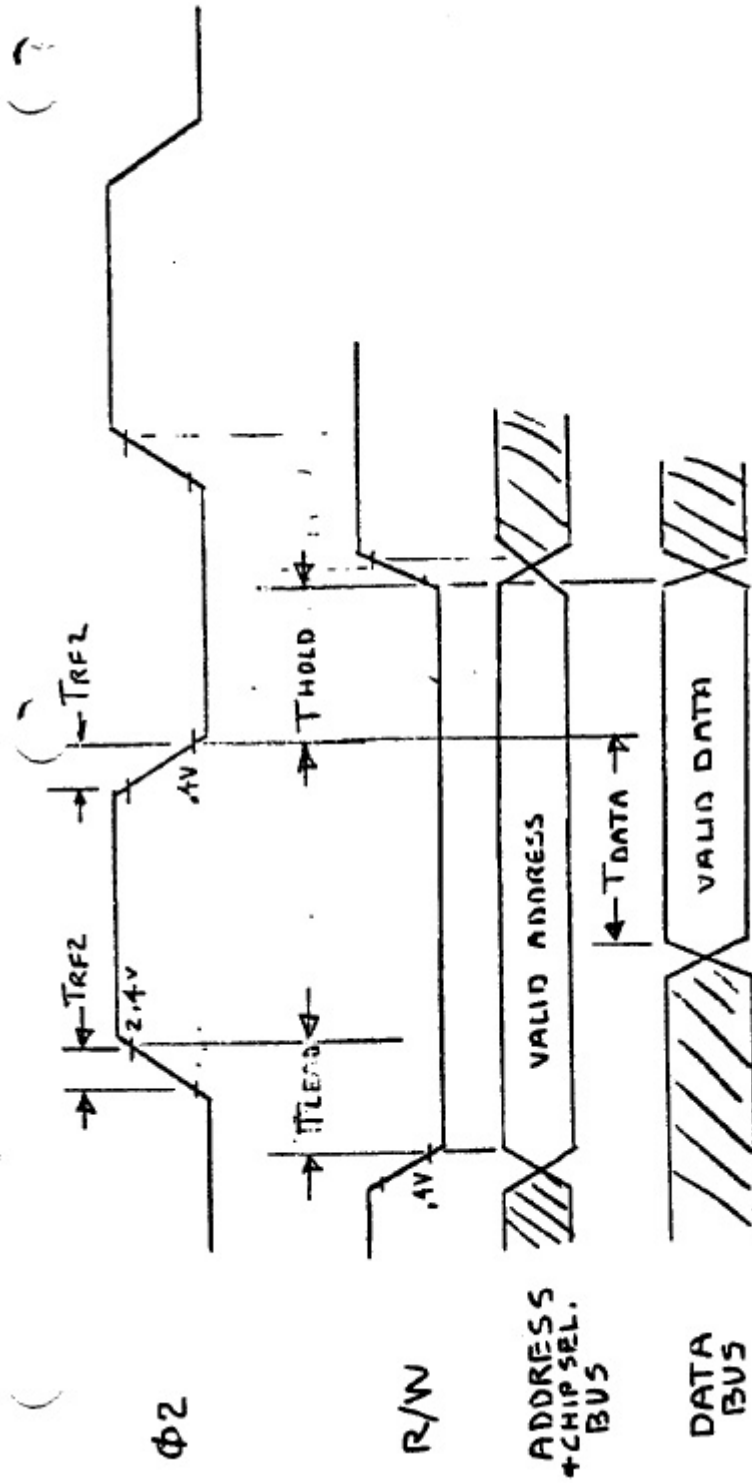
Dirección de 6 bits	Nombre de la dirección	7	6	5	4	3	2	1	0	Función	D7	D6
0	CXM0P	1	1							Lectura de colisión	M0 P1	M0 P0
1	CXM1P	1	1							Lectura de colisión	M1 P0	M1 P1
2	CXP0FB	1	1							Lectura de colisión	P0 PF	P0 BL
3	CXP1FB	1	1							Lectura de colisión	P1 PF	P1 BL
4	CXM0FB	1	1							Lectura de colisión	M0 PF	M0 BL
5	CXM1FB	1	1							Lectura de colisión	M1 PF	M1 BL
6	CXBLPF	1								Lectura de colisión	BL PF	NADA
7	CXPPMM	1	1							Lectura de colisión	P0 P1	M0 M1
8	INPT0	1								Lectura de potenciómetro en el puerto		
9	INPT1	1								Lectura de potenciómetro en el puerto		
A	INPT2	1								Lectura de potenciómetro en el puerto		
B	INPT3	1								Lectura de potenciómetro en el puerto		
C	INPT4	1								Lectura de entrada		
D	INPT5	1								Lectura de entrada		

El chip STELLA

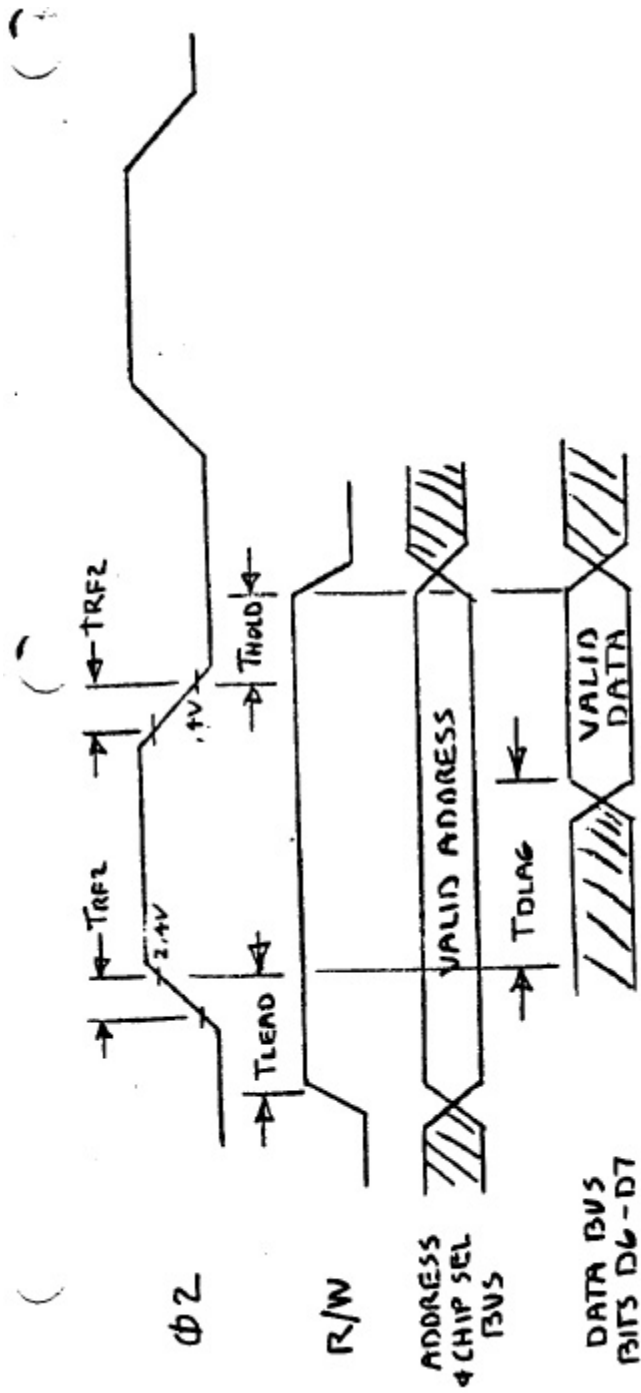




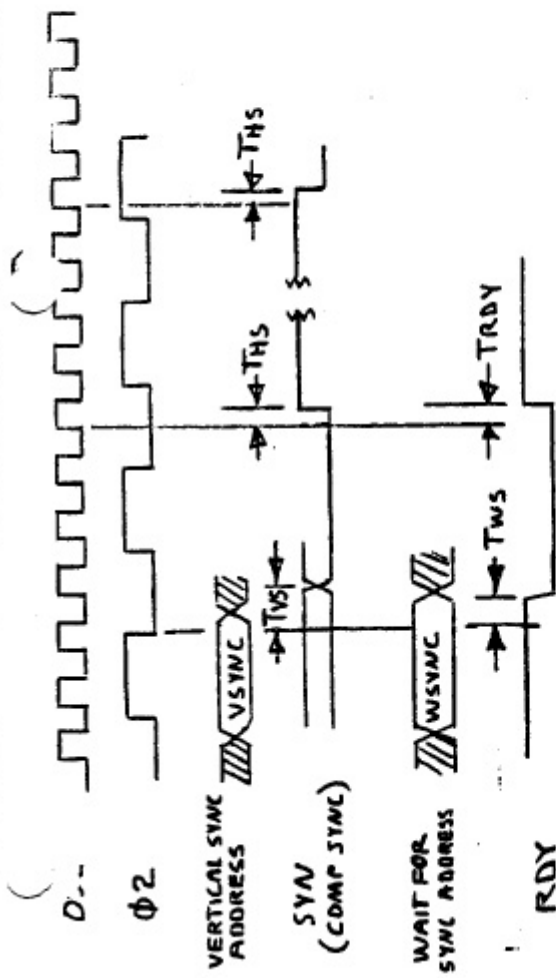
TIA $\Phi 0 - \Phi 2$ AND LUM TIMING



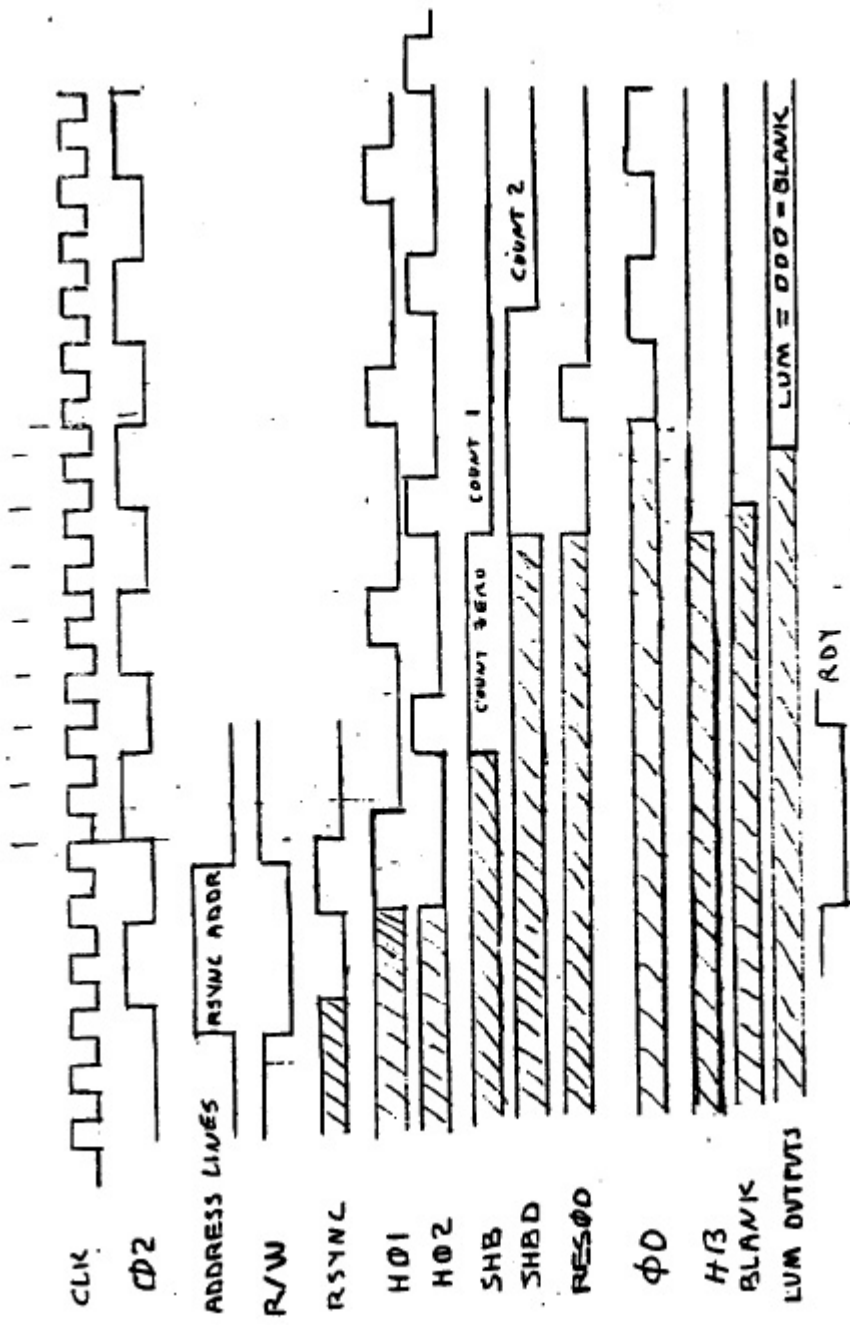
TIA WRITE TIMING CHARACTERISTICS



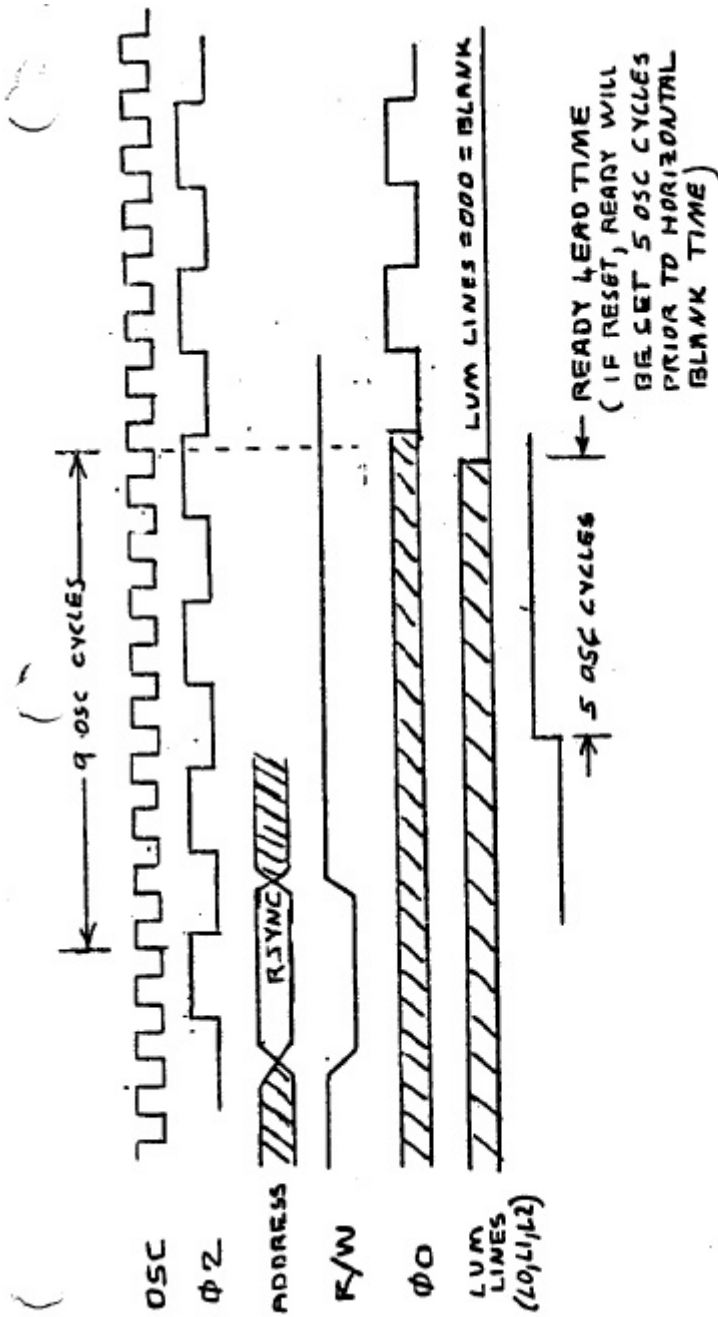
TIA READ TIMING CHARACTERISTICS



TIA COMP-SYN AND READY TIMING



RSYNC - RES00 - H01 - H02 - SHB - D2 - 00



TIA RSYNC AND BLANK AND READY TIMING