

El MSX Red Book (la versión revisada 1997/08/06)

Las notas del editor:

- El libro fue escudriñado y se convirtió (por O.C.R.) Por una persona y editado por otro (usando una PC de la IBM compatible), independientemente.
- Todas las páginas tienen un tamaño de apuro de 64 rayado. Se justificó que la anchura simplifica modificaciones futuras, sin embargo ninguna línea es más larga que 80 columnas.
- Este libro sólo encubre a MSX estándar. La entrada BIOS apunta de 0000H para 01B5H debería ser usado en lugar de las llamadas entradas descritas en el libro, porque otras máquinas (MSX2, MSX2+, interrogación del turbo MSX y hechos a la medida) tienen colocaciones diferentes para las rutinas. El uso de rutina interna BIOS dirige la palabra a es responsable de muchos programas sólo corrida en MSX.
- Algún presente de errores en el libro original fuera fijo, aunque fue probado para conservarlo tan inalterado como posible. Todos los números de la página hacen juego con los originales, exceptúan errores no descubiertos ya presente en el original.
- La mayoría de figuras fueron modificadas debido a la naturaleza única en texto de este archivo. El conjunto de caracteres usado durante edición fue la PC Internacional de la IBM un. Los siguientes personajes especiales fueron usados y deberían variarse para los correspondientes de otros conjuntos de caracteres:

+ --- + La Libra Del Marco: £
Los personajes: | | | Micro: μ
+- + -
| | |
+

Convertido para el ASCII formateo por MSXHans 2001

Los CONTENIDOS

La introducción.....	1
1. La Interfaz Periférica Programable.....	3
2. Videograbe Procesador De Despliegue.....	8
3. El Generador En Buen Estado Programable.....	21
4. ROM BIOS.....	26
5. El BASIC ROM Interpreter.....	89
6. La memoria Map.....	208
7. Los Programas De Código De La Máquina.....	240

El Derecho De Autor De Contenidos 1985 El Software Avalon
Iver Lane, Cowley, Middx, UB8 2JD

MSX es una marca registrada de Microsoft Corp.
Z80 es una marca registrada de Zilog Corp.
La ACADEMIA es marca registrada de Alfred.

La INTRODUCCIÓN

Las metas

Este libro se trata de computadoras MSX y cómo operan. Para razones técnico y comercial MSX los fabricantes de la computadora sólo hacen disponible una cantidad limitada de información para el usuario final acerca del diseño de sus máquinas. Usualmente ésta será una descripción medianamente detallada de Microsoft MSX BASIC conjuntamente con un contorno ancho del hardware de sistema. Mientras este nivel de documentación es adecuado para el usuario informal inevitablemente probará limitar para cualquiera comprometido en más programación sofisticada.

La meta de este libro debe proveer una descripción del hardware estándar MSX y el software en un nivel de detalle lo suficiente como para satisfacer tan más demandando de usuarios, el programador de código de la máquina. No es un curso introductorio en programar y está necesariamente de una naturaleza más bien técnica. Está asumido que usted ya posee, o tiene la intención de adquirir por otra manera, una comprensión del Z80 Microprocessor en el nivel de código de la máquina. Como hay tantos libros generales de propósito ya en la existencia acerca del Z80 que cualquier descripción de sus características simplemente duplicaría ampliamente información disponible.

La organización

El MSX Standard especifica lo siguiente como los componentes funcionales principales en cualquier computadora MSX:

- (1) Zilog Z80 Microprocessor
- (2) Intel 8255 Interface Periférico Programable
- (3) Tejas 9929 El De Vídeo Display Processor
- (4) General Instrumente 8910 Generator Atinado Programable
- (5) 32 KB MSX BASIC ROM
- (6) 8 el KB GOLPEAN DURAMENTE mínimo

Aunque hay obviamente unos un montón de componentes adicionales involucrados en el diseño de una computadora MSX son todos poco programables en pequeña escala, y por consiguiente "invisible" para el usuario. Los fabricantes generalmente tienen libertad considerable en la selección de estos componentes en pequeña escala. Los componentes programables no pueden ser variados y por consiguiente todas las máquinas MSX son idénticas hasta donde el programador está afectado.

Divide en capítulos 1, 2 y 3 describen la operación del Interface Periférico Programable, el De

Vídeo Display Processor y Generator Atinado Programable respectivamente. Estos tres dispositivos

- 1 -

La INTRODUCCIÓN

Provea la interfaz entre el Z80 y el hardware periférico en una máquina estándar MSX. Todo ocupan colocaciones en el Z80 yo / O (la entrada /salida) Bus.

Divida en capítulos 4 cubiertas el software contenido en la primera parte divida del MSX ROM. Este capítulo del ROM está preocupado con controlar el hardware de la máquina en el nivel fino de detalle y es conocido como los ROM BIOS (el Sistema Básico de Salida Input). Está estructurado en tal muy ese la mayor parte de las funciones que un programador de código de la máquina precisa, como teclado y conductores de vídeo, está fácilmente disponible.

Capítulo 5 describe el software contenido en el resto del ROM, el Microsoft MSX BASIC Interpreter. Aunque éste es mayormente un programa conducido en texto, y consecuentemente de menos uso para el programador, un examen cercano revele muchas proposiciones no documentadas por fabricantes.

Capítulo 6 está afectado con la organización de memoria de sistema. La atención particular es pagada al Workspace Area, eso secciona de RAM de F380H para FFFFH, como esto es utilizado como un bloc para apuntes por los BIOS y el BASIC Interpreter y contiene muy información de uso para cualquier programa de aplicación.

Capítulo 7 da algunos ejemplos de programas de código de la máquina que hacen uso de características ROM para minimizar diseñá esfuerzo.

Se cree que este libro contiene defectos de cero, si usted sabe que de otra manera el autor tendría mucho gusto en escuchar de usted.
Este libro está dedicado al Nightmare Andante.

1. LA INTERFAZ PERIFÉRICA PROGRAMABLE

El 8255 PPI es un propósito general paralelamente la interfaz que el dispositivo configuró como tres ocho puertos de datos mordidos, llamó A, B y C, y un puerto de modo. Se aparece ante el Z80 como cuatro que yo / la O exporto a través de cuál el teclado, la memoria cambiando hardware, el motor del casete, la salida del casete, el Caps Lock DIRIGIDO y la salida audia Click Crucial puede controlarse. Una vez que el PPI ha sido inicializado gane acceso a para un pedazo particular de hardware justamente involucra escribir para o leer lo pertinente yo el puerto de la / O.

PPI Exporte UNO (yo / la O Exporto a A8H)

```
7 6 5 4 3 2 1 0
+
| Pagine 3 | Página 2 | Página 1 | Página 0
| PSLOT# |PSLOT# |PSLOT# |PSLOT
| C000-FFFF 8000-BFFF 4000-7FFF 0000-3FFF
+
```

Figura 1: El Registro Primario De La Ranura

Esta terminal de salida, conocido como el Slot Primario Register en la terminología MSX, se usa para controlar la memoria cambiando hardware.

El Z80 Microprocessor sólo puede ganar acceso a 64 el KB de memoria directamente. Esta limitación es actualmente supuesto tan también restrictiva y varias de los métodos personales más nuevos de la planilla de computadoras para vencerla.

Las máquinas MSX pueden tener dispositivos múltiples de memoria en la dirección del mismo y el Z80 puede cambiar de decisión en cualesquier de ellos según se requiera.

El espacio de la dirección del procesador es apreciado como ser duplicado "lateralmente" en cuatro áreas 64 separadas de KB, llamado Slots Primario 0 para 3, cada uno del cual recibe su propia ranura señal selecta a lo largo de las señales normales del autobús Z80. Los contenidos del Primary Slot Register determinan cuál ranura señal selecta es activa y por consiguiente cuál el Primario Slot es seleccionado.

Para aumentar flexibilidad cada uno "que página" de 16 KBES del espacio de la dirección Z80 puede ser seleccionado de un Slot Primario diferente. Como se muestra en Figure 1 dos añicos del Primary Slot Register estén obligados a definir el número Primario Slot para cada página.

La primera operación realizada por el MSX ROM en. El Slot Primario Register está luego listo a fin de que las ranuras pertinentes son seleccionadas así haciendo la RAM permanentemente disponible. La configuración de memoria de cualquier máquina MSX puede ser

1. LA INTERFAZ PERIFÉRICA PROGRAMABLE

Decidido exhibiendo al Primario Slot Register sedimentándose con la declaración de BASIC:

ESCRIBA EN LETRAS DE IMPRENTA a RIGHT\$ ("0000000" N\$ (INP (y HA8)), 8)

Por poner un ejemplo "10100000" se producirían en una Toshiba HX10 donde las páginas 3 y 2 (la RAM) que ambos vengan de Primario Slot 2 y página 1 y 0 (el MSX ROM) del Primario Slot 0. El MSX ROM siempre debe ser colocado en el Primario Slot 0 por un fabricante como ésta sea la ranura seleccionada por el hardware en energizar. Otros dispositivos de memoria, otra RAM y cualquier ROM adicional, pueden ser colocados en cualquier ranura por un fabricante.

Una máquina típica UK tendrá a un Slot Primario conteniendo al MSX ROM, un conteniendo 64 que el KB de RAM y dos ranuras sacó para los conectores externos. La mayoría de máquinas japonesas llevan puestas un conector de tipo del cartucho cada uno de estas ranuras externas pero las máquinas UK usualmente tienen un conector del cartucho y un conector IDC.

Los extensores

La memoria de sistema puede ser incrementada hasta un máximo teórico de dieciséis áreas de 64 KBES por interfaces utilizadoras del extensor. Unos tapones del extensor en cualquier Primary Hacen Una Ranura En proveer cuatro Ranuras de 64 KBES de Secundario, numeraron 0 para 3, en lugar de una primaria. Cada extensor tiene su propio hardware local, llamado un Secundario Slot Register, para seleccionar cuál del Slots Secundario debería aparecer en el Slot Primario. Como antes de que páginas puede ser seleccionado de Slots Secundario diferente.

```
7 6 5 4 3 2 1 0
+
| Pagine 3 | Página 2 | Página 1 | Página 0
| SSLOT# |SSLOT# |SSLOT# |SSLOT
+
```

Figura 2: El Registro Secundario De La Ranura

Cada Slot Secundario Register, al realmente ser una lectura de ocho bits / escriba picaporte, está hecho en aparecer como posición de memoria FFFFH de su Slot Primario por el hardware del extensor. Para ganar acceso a esta posición en un extensor particular usualmente hay que primero cambiar página 3 (C000H para FFFFH) de ese Slot Primario en el espacio de la dirección del procesador. El Slot Secundario Register luego puede ser modificado y, si la página necesaria, 3 restituyó para su Slot Primario original sedimentándose. Ganar acceso a la memoria en extensores puede convertirse más bien un proceso con muchas curvas.

Es aparente que allí debe ser alguna forma de determinar si un Slot Primario contiene RAM ordinaria o un extensor adentro

1. LA INTERFAZ PERIFÉRICA PROGRAMABLE

Haga el pedido ganar acceso a eso correctamente. Para lograr esto son los Registros Secundarios Slot diseñados para invertir sus contenidos cuándo la espalda leída. Durante el posición memoria búsqueda RAM que se energiza arriba FFFFH de cada Slot Primario es examinado para determinar si se comporta normalmente o si la ranura contiene un extensor. Los resultados de estas pruebas son almacenados en el mapa del recurso de sistema Workspace Area EXPTBL para el posterior uso. Esto está hecho en que se energiza arriba por la dificultad en funcionar prueba cuando el Secondary Slot Registers realmente contenga viva colocaciones.

El cambiar memorias es obviamente un área demandando de más cautela, particularmente con los mecanismos jerárquicos necesitados para monitorear extensores. El cuidado debe ser tomado para evitar alternación fuera de la página en la cual un programa está marchando o, si estuviera siendo usado, la página conteniendo la pila. Hay un número de rutinas estándar disponibles para el programador de código de la máquina en el capítulo BIOS del MSX ROM para simplificar el proceso.

El BASIC Interpreter mismo tiene cuatro métodos de ROMs de la extensión que gana acceso a. La primera parte tres de estos es pues el uso con ROMs de código de la máquina colocó en página 1 (4000H para 7FFFH), son:

- (1) Se Abrocha (El Capítulo 6).
- (2) La "llamada" declaración (el Capítulo 5).

Los nombres (el Capítulo 5) (3) Adicionales del dispositivo.

El BASIC Interpreter también puede ejecutar un programa de BASIC ROM detectado en página 2 (8000H para BFFFH) durante la búsqueda que se impulsa arriba ROM. Lo que no puede hacer el BASIC Interpreter es usar cualquier RAM escondida detrás de otros dispositivos de memoria. Esta limitación es una reflexión de la dificultad en convertir un programa establecido a tomar ventaja de más nuevo, más máquinas complicadas. Una situación similar existe con la versión de BASIC Microsoft disponible en la PC de la IBM. Fuera de un espacio de memoria de 1 MEGABYTES sólo 64 para los que el KB puede servir programan almacenamiento.

PPI Exporte a B (yo / la O Exporto a A9H)

```

7 6 5 4 3 2 1 0
+
| Las Entradas De La Columna Del Teclado
+

```

Figura 3

Esta terminal de entrada se usa para leer los ocho añicos de datos de la columna de la fila actualmente seleccionada del teclado. El teclado MSX es un software escudriñado once la fila por ocho la matriz de la columna de interruptores normalmente abiertos. La corriente que las máquinas usualmente sólo tienen tecléea cero de filas para ocho. La conversión de depresiones cruciales

En los códigos de caracteres es realizado por el MSX ROM interrumpa al manipulador, este proceso está descrito en Chapter 4.

PPI Exporte a C (yo / la O Exporto a AAH)

```

    7 6 5 4 3 2 1 0
    +
    | Afine Fila Del | Teclado De La | Gorra Cas Selecta
    | El chasquido | LLEVADO | Fuera De | Motor
    +
  
```

Figura 4

Esta terminal de salida controla una variedad de funciones. Los añicos Select Row cuatro Keyboard seleccionan cuál de las once filas del teclado, numeradas de 0 para 10, debe ser leído adentro por B PPI Port.

El bit Cas Motor determina la condición del relevador del motor del casete: 0 = Adelante, 1 = Off.

El bit Cas Out es filtrado y estilizado antes de ser llevado al conector de ESTRÉPITO del casete como la señal MIC. Toda generación de tono del casete es realizada por software.

El Cap que LED mordió determina que la condición del Caps Lock LLEVÓ LA DELANTERA: 0 = Adelante, 1 = Off.

La salida Crucial Click está estilizada y mixta con la salida audia del Generator Atinado Programable. Para realmente generar un estrecho que este bit debería ser lanzado de vez en cuando.

Noto que hay rutinas estándar en los ROM BIOS para ganar acceso a todas las funciones disponibles con este puerto. Estos deberían ser usados preferentemente a la manipulación directa del hardware si es posible.

PPI Mode Port (yo / la O Exporto a ABH)

```

    7 6 5 4 3 2 1 0
    +
    | 1 | A Y C|A|C|B Y C|B|C
    |   |   | El Modo De Modo Dir Dir
    +
  
```

Figura 5: PPI Mode Selection

Este puerto se usa para colocar el modo operativo del PPI. Como el hardware MSX es diseñado para operar en un detalle

1. LA INTERFAZ PERIFÉRICA PROGRAMABLE

La configuración sólo este puerto no debería ser modificada bajo cualquier condiciones. Los detalles son dados para la plenitud sólo.

El bit 7 debe tener 1 años de edad para alterar el modo PPI, cuando es 0 el PPI realiza la sola función mordida del set /reanudación exteriorizada en Figure 6.

La A y los añicos C Mode determinan el modo operativo de Babor UNO y el cuatro bits superior sólo de C De Babor: 00 = el Normal Mode (MSX), 01 = Strobed Mode, 10 = el Bidireccional Mode

El modo de la A Dir determina la dirección de Babor UNO: 0 = Devuelven (MSX), 1 = Input.

La C Dir mordió determina la dirección del cuatro bits superior sólo de C De Babor: 0 = Devuelven (MSX), 1 = Input.

La B y los añicos C Mode determinan el modo operativo de B De Babor y el cuatro bits inferior sólo de C De Babor: 0 = el Normal Mode (MSX), 1 = Strobed Mode.

La B Dir mordió determina la dirección de B De Babor:0 = devuelven, 1 = Input (MSX).

La C Dir mordió determina la dirección del cuatro bits inferior sólo de C De Babor: 0 = Devuelven (MSX), 1 = Input

```
7 6 5 4 3 2 1 0
+
| 0 | No usado | Mordido Number | Set
+
```

Figura 6: PPI Mordió a SetReset

El PPI Mode Port puede estar acostumbrado directamente colocar o volver a arrancar cualquier bit de C De Babor cuándo bit 7 es 0. El Number Mordido, de 0 para 7, determina cuál bit debe ser afectado. Su valor nuevo está resuelto por el bit SetReset: 0 = Reanudán, 1 = Set. La ventaja de este modo es que una sola salida puede estar con holgura modificada. Por poner un ejemplo el Caps Lock DIRIGIDO puede ser animado con la declaración de BASIC OUT y HAB, 12 y completamente con la declaración OUT y HAB, 13.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

Lo 9929 VDP contiene todo el sistema de circuitos necesario para generar el despliegue de vídeo. Se aparece ante el Z80 como dos que yo los puertos de la / O llámé la Data Exporta y el Command Port. Aunque el VDP tiene su propio 16 KB de VRAM (la RAM De Vídeo), los contenidos de los cuales define la imagen de la pantalla, ésta no puede ser en seguida a la que se ganó acceso por el Z80. En lugar de eso debe usar lo dos yo puertos de la / O para modificar al VRAM y colocar al VDP diversos manejando condiciones.

Datos Port (yo / la O Port 98H)

La Data Port está acostumbrada a leer o escribir solos bytes para el VRAM. El VDP posee un registro interno de la dirección apuntando hacia una posición en el VRAM. Leer a la Data Port introducirá en la computadora el byte de esta posición VRAM mientras escribir a la Data Port almacenará un byte allí. Después de una lectura o escriba que el registro de la dirección es automáticamente incrementado para apuntar para la siguiente posición VRAM. Los bytes secuenciales pueden ser a los que se ganó acceso simplemente por lecturas continuas o escribe a la Data Port.

La orden Port (yo / la O Port 99H)

El Command Port sirve para tres propósitos:

- (1) To establezca el registro de la dirección Data Port.
- (2) To lea al VDP Status Register.
- (3) To escriba a uno del VDP Mode Registers.

La dirección Register

El registro de la dirección Data Port debe ser establecido de diferentes maneras según que la subsiguiente vía de entrada debe ser una lectura o uno escriba. El registro de la dirección puede estar colocado para cualquier valor de 0000H para 3FFFH por primera escritura el LSB (Byte Significativo Mínimo) y luego el MSB (La Mayoría De Byte Significativo) para el Command Port. Los añicos 6 y 7 del MSB son usados por el VDP a determinar si el registro de la dirección está siendo colocado arriba para subsiguientes lecturas o escribe como sigue:

```
+
| Xxxxxxxx 00xxxxxx leído
+
+
| Escriba a xxxxxxxx 01xxxxxx
+
```

Figura 7: VDP Dirija La Palabra A Setup

Es importante que ningún otras vías de entrada hagan al VDP en medio de escribir al LSB y el MSB como ésta contrariará su

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

La sincronización. El MSX ROM interrumpe manipulador continuamente lee al VDP Status Register como una tarea de fondo tan interrumpe debería estar discapacitado tan necesario.

VDP Status Register

Leer al Command Port introducirá en la computadora los contenidos del VDP Status Register. Esto contiene banderas diversas como debajo:

```
7 6 5 4 3 2 1 0
+
| F | 5S | C | Fifth Sprite Number
|| La Bandera De La | Bandera De La Bandera
+-----+ + Figura 8: VDP Status Register
```

Los Quintos añicos Sprite Number contienen el número (0 para 31) del bloque de objetos movibles incentivando al Quinto Sprite Flag.

El Coincidence Flag es normalmente 0 sino que está listo para 1 si cualquier bloques de objetos movibles tienen a uno o más pixels imbricados. Leer al Status Register volverá a arrancar esta bandera para uno 0. Noto que la coincidencia es sólo comprobada como cada pixel sea generado durante un marco de vídeo, en una máquina UK ésta es cada 20 señoras Si rápido el traslado cada uno el que los bloques de objetos movibles pasan encima en medio no inspecciona luego coincidencia estará enlosada.

El Quinto Sprite Flag es normalmente 0 sino que está listo para 1 cuando hay más que cuatro bloques de objetos movibles en cualquier línea del pixel. Leer al Status Register volverá a arrancar esta bandera para uno 0.

El Frame Flag es normalmente 0 sino que está listo para uno 1 al final de la última línea activa del marco de vídeo. Para máquinas UK con una tasa del marco 50 Hz esto ocurrirá cada 20 Señoras Reading que el registro Status volverá a arrancar esta bandera para uno 0. Hay una señal asociada de salida del VDP que genera Z80 interrumpe en la misma tasa, esto conduce al MSX ROM interrumpe manipulador.

VDP Mode Se Registra

El VDP tiene ocho registros que se escriben sólo, numerado 0 para 7, cuál controla su operación general. Un registro particular es determinado por primero escribir un byte de datos luego un byte de selección de registro para el Command Port. El byte de selección de registro contiene el número de registro en los añicos tres inferiores: El 1000RRR. Como el Mode Se Registra es que se escribe sólo, y no puede ser leído, el MSX ROM mantiene una copia conforme de los ocho registros en el Workspace Area de RAM (el Capítulo 6). Usando las rutinas del estándar MSX ROM para VDP las funciones aseguran que esta imagen de registro están correctamente actualizadas.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

El Registro De Modo 0

```
7 6 5 4 3 2 1 0
+
| 0 | 0 | 0 | 0 | 0 | 0 | M3 | EV
+----- + Figura 9
```

El VDP Externo que el bit determina ya sea VDP externo introducen en la computadora debe ser facultado o discapacitado: 0 = Lisieron, 1 = Enabled.

El bit M3 es uno de los tres añicos de selección de modo VDP, sede Mode Register 1.

El Registro De Modo 1

```
7 6 5 4 3 2 1 0
+
| 4/16K | Blank | IE | M1 | M2 | 0 | Size | Mag
+----- + Figura 10
```

La Magnificación que el bit determina ya sea duendes serán normales o doblados en el tamaño: 0 = Normal, 1 = Doubled.

El Tamaño que el bit determina ya sea cada patrón de bloque de objetos movibles serán añicos del 8x8 o añicos 16x16: 0=8x8, 1=16x16.

El M1 y añicos M2 determinan al VDP manejando modo en conjunción con el bit M3 de Mode Register 0:

```
M1 M2 M3
0 0 0 32x24 Text Mode
0 0 Mode 1 Graphics
0 1 Mode 0 Multicolour
1 0 0 40x24 Text Mode
```

El Interrupt que Enable mordió posibilita o lisia lo interrumpe señal de salida del VDP: 0 = Lisian, 1 = Enable.

El bit En Blanco se usa para posibilitar o lisiar el despliegue de vídeo entero: 0 = Lisian, 1 = Enable. Cuando el despliegue es al que se ganó limpiamente será el mismo color como el borde.

El bit del 4/16K altera al VDP VRAM dirigiendo características a satisfacer ya sea 4 KB o 16 patatas fritas de KB: 0=4 KB, 1=16 KB.

El Registro De Modo 2

```
7 6 5 4 3 2 1 0
+
| 0 | 0 | 0 | 0 | Name Posponen Base
+
```

Figura 11

Modo Register 2 define el discurso de puesta en marcha del Name Table en el VDP VRAM. Los cuatro añicos disponibles sólo le especifican colocación 00BB BB00 0000 0000 de la dirección completa así es que registran contenidos de 0FH resultarían en un discurso bajo de 3C00H.

El Registro De Modo 3

```
7 6 5 4 3 2 1 0
+
| La Base De La Mesa De Color
+
```

Figura 12

Modo Register 3 define el discurso de puesta en marcha del Colour Table en el VDP VRAM. Los ocho añicos disponibles sólo le especifican colocación 00BB BBBB BB00 0000 de la dirección completa así es que registran contenidos de FFH resultarían en un discurso bajo de 3FC0H. En bit único Mode Graphics 7 es efectivo así ofreciendo una base de 0000H o 2000H. Los añicos 0 para 6 deben tener 1 años de edad.

El Registro De Modo 4

```
7 6 5 4 3 2 1 0
+
| 0 | 0 | 0 | 0 | 0 | Character Pattern
+
```

Figura 13

Modo Register 4 define el discurso de puesta en marcha del Character Pattern Table en el VDP VRAM. Los tres añicos disponibles sólo le especifican colocación 00BB B000 0000 0000 de la dirección completa así es que registran contenidos de 07H resultarían en un discurso bajo de 3800H. En bit único Mode Graphics 2 es efectivo así ofreciendo una base de 0000H o 2000H. Los añicos 0 y 1 deben tener 1 años de edad.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

El Registro De Modo 5

7 6 5 4 3 2 1 0
+
| 0 | Sprite Atribuyen base
+

Figura 14

Modo Register 5 define el discurso de puesta en marcha del Sprite Attribute Table en el VDP VRAM. Los siete bits disponibles sólo le especifican colocación 00BB BBBB B000 0000 de la dirección completa así es que registran contenidos de 7FH resultarían en un discurso bajo de 3F80H.

El Registro De Modo 6

7 6 5 4 3 2 1 0
+
| 0 | 0 | 0 | 0 | 0 | 0 | Sprite Pattern
+

Figura 15

Modo Register 6 define el discurso de puesta en marcha del Sprite Pattern Table en el VDP VRAM. Los tres bits disponibles sólo le especifican colocación 00BB B000 0000 0000 de la dirección completa así es que registran contenidos de 07H resultarían en un discurso bajo de 3800H.

El Registro De Modo 7

7 6 5 4 3 2 1 0
+
| El Color Del Texto 1 El Color | Fronterizo
+

Figura 16

Los bits Fronterizos Colour determinan el color de la región rodeando el contorno de vídeo activo en todos los cuatro modos VDP. También determinan el color de todos los 0 pixels en la pantalla en 40x24 Text Mode. Noto que el borde que la región realmente extiende a través de la pantalla entera sino sólo se volverá visible en el área activo si el pixel que yace encima de es transparente.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

El Text Colour 1 a 15 determina el color de todos los 1 pixels en 40x24 Text Mode. No tienen efecto en los otros tres modos donde la flexibilidad mayor es provista a través del uso del Colour Table. Los códigos de color VDP son:

0 Transparente 4 Dark Azulan 8 Rojos 12 Verdeoscuros
1 Purple 13 Rojo Fuerte 9 Azul Claro 5 Negro
2 Verdean 6 El Moreno Red 10 Amarillos 14 Gris
3 El Ligero Green 7 Sky Blue 11 White 15 Amarillo Claro

La pantalla Modes

El VDP tiene cuatro modos operativos, cada uno ofrece un set ligeramente diferente de capacidades. Generalmente hablando, como la resolución se aumenta el precio a estar pagado en el tamaño VRAM y la complejidad modernizante también aumentan. En una aplicación dedicada estos asociaron hardware y las expensas del software son consideraciones importantes. Para una máquina MSX son irrelevantes, por consiguiente parece una piedad que un intento mayor no estaba hecho para estandarizar en un modo particular. El Graphics Mode es capaz adecuadamente realizando todas las funciones de los otros modos con sólo reservaciones menores.

Una dificultad agregada en usar al VDP se levanta porque la concesión insuficiente estaba hecha en su diseño para la sobre-tomografía usada por la mayoría de televisiones. La pérdida resultante de personajes en los bordes de la pantalla ha forzado todo el software relacionado en vídeo MSX en basarse en tamaños peculiares de la pantalla. Las máquinas UK normalmente usan sólo a los central treinta y siete personajes disponibles en 40x24 Text Mode. El nipón labra a máquina, con salidas del vídeo NTSC (el Nacional Television Standards Committee), use los símbolos treinta y nueve centrales.

El elemento central en el VDP, de punto de vista del programador, está el Name Table. Ésta es una lista sencilla de solo byte que los códigos de caracteres sostuvieron en VRAM. Tiene 960 bytes de largo en 40x24 Text, Mode 768 bytes de largo en 32x24 Text Mode, Graphics Mode y Multicolour Mode. Cada colocación en el Name Table es propia de una posición particular en la pantalla.

Durante un marco de vídeo el VDP secuencialmente leerá cada código de caracteres del Name Table, comenzando en la base. Como a cada código de caracteres le es leído el patrón correspondiente del 8x8 de pixels es mirado arriba en el Character Pattern Table y exhibido en la pantalla. La apariencia de la pantalla así puede ser modificada por tampoco cambiar los códigos de caracteres en el Name Table o los patrones del pixel en el Character Pattern Table.

Noto que el VDP no tiene facilidad del cursor del hardware, si uno es requerido que debe ser software generado.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

40x24 Text Mode

El Name Table ocupa 960 bytes de VRAM de 0000H para 03BFH:

```

0123456789012345678901234567890123456789
0000H + ----- + 0
0028H +-----+ 1
0050H +-----+ 2
0078H +-----+ 3
00A0H +-----+ 4
00C8H +-----+ 5
00F0H +-----+ 6
0118H +-----+ 7
0140H +-----+ 8
0168H +-----+ 9
0190H +-----+ 10
01B8H +-----+ 11
01E0H +-----+ 12
0208H +-----+ 13
0230H +-----+ 14
0258H +-----+ 15
0280H +-----+ 16
02A8H +-----+ 17
02D0H +-----+ 18
02F8H +-----+ 19
0320H +-----+ 20
0348H +-----+ 21
0370H +-----+ 22
0398H +-----+ 23
+
012345678901234567890123456789

```

Figura 17: 40x24 Name Table

El patrón Table ocupa 2 KB de VRAM de 0800H para 0FFFH. Cada uno que bloque de ocho bytes contiene el pixel modela para un código de caracteres:

```

+
|0 0 1 0 0 0 0 0| Byte 0
|0 1 0 1 0 0 0 0| Byte 1
|1 0 0 0 1 0 0 0| Byte 2
|1 0 0 0 1 0 0 0| Byte 3
|1 1 1 1 1 0 0 0| Byte 4
|1 0 0 0 1 0 0 0| Byte 5
|1 0 0 0 1 0 0 0| Byte 6
|0 0 0 0 0 0 0 0| Byte 7
+-----+ + Figura 18: El carácter Pattern Block (No. 65 mostrado ' UNO ')

```

El primer bloque contiene el patrón para el código de caracteres 0, el segundo el patrón para el código de caracteres 1 etcétera para el código de caracteres 255. Note eso sólo el leftmost seis pixels es de hecho exhibido en este modo. Los colores de los 0 y 1 pixels en este modo están definidos por VDP Mode Register 7, inicialmente son azules y blancos.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

32x24 Text Mode

El Name Table ocupa 768 bytes de VRAM de 1800H para 1AFFH. Como en 40x24 Text la operación normal Mode involucra a colocar códigos de caracteres en la colocación requerida en la mesa. El "VPOKE" declaración puede estar acostumbrado a lograr familiaridad con el plan de la pantalla:

```

01234567890123456789012345678901
1800H + ----- + 0
1820H ++++++ | 1
1840H ++++++ | 2
1860H ++++++ | 3
1880H ++++++ | 4
18A0H ++++++ | 5
18C0H ++++++ | 6
18E0H ++++++ | 7
1900H ++++++ | 8
1920H ++++++ | 9
1940H ++++++ | 10
1960H ++++++ | 11
1980H ++++++ | 12
19A0H ++++++ | 13
19C0H ++++++ | 14
19E0H ++++++ | 15
1A00H ++++++ | 16
1A20H ++++++ | 17
1A40H ++++++ | 18
1A60H ++++++ | 19
1A80H ++++++ | 20
1AA0H ++++++ | 21
1AC0H ++++++ | 22
1AE0H ++++++ | 23
+
01234567890123456789012345678901

```

Figura 19: 32x24 Name Table

El Character Pattern Table ocupa 2 KB de VRAM de 0000H para 07FFH. Su estructura es lo mismo como en 40x24 Text Mode, todos los ocho pixels de un 8x8 el patrón son ahora exhibido.

El color fronterizo está definido por VDP Mode Register 7 y es inicialmente azul. Una mesa adicional, el Colour Table, determina el color de los 0 y 1 pixels. Esto ocupa treinta y dos bytes de VRAM de 2000H para 201FH. Cada entrada en el Colour Table define los 0 y 1 colores pixel para un grupo de ocho códigos de caracteres, mientras inferior cuatro bits definiendo el color 0 pixel, cuatro bits superior el color 1 pixel. La primera entrada en la mesa define los colores para los códigos de caracteres 0 para 7, el segundo para los códigos de caracteres 8 para 15 etcétera para treinta y dos entradas. El MSX ROM inicializa todas las entradas para el mismo valor, azul y el blanco, y no provee instalaciones para individuales cambiantes.

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

Gráfico Mode

El Name Table ocupa 768 bytes de VRAM de 1800H para 1AFFH, así como en 32x24 Text Mode. La mesa es en el que se puso las iniciales con la secuencia de código de caracteres 0 para 255 repetido tres veces y se queda luego sin tocar, en este modo que es el Character Pattern Table que es modificado durante la operación normal.

El Character Pattern Table ocupa 6 KB de VRAM de 0000H para 17FFH. Mientras su estructura es lo mismo tan en los modos del texto que no contiene un conjunto de caracteres sino es inicializado para todos los 0 pixels. La primera parte 2 el KB del Character Pattern Table es tratada según los códigos de caracteres de la primera parte en tercer lugar del Name Table, el segundo 2 KB por la tercera parte central del Name Table y el último 2 KB por la tercera parte final del Name Table.

Por el patrón secuencial en el Name Table el Character entero Pattern Table es leído fuera de linearily durante un marco de vídeo. Colocando un punto en que la pantalla implica resolución donde el bit correspondiente está en el Character Pattern Table y encendiéndolos. Para que un programa de BASIC convierta a X, las coordenadas Y para una dirección ven la rutina del estándar MAPXYC en Chapter 4.

01234567890123456789012345678901	
0000H + ----- + 0	
0100H ++++++	1
0200H ++++++	2
0300H ++++++	3
0400H ++++++	4
0500H ++++++	5
0600H ++++++	6
0700H ++++++	7
0800H ++++++	8
0900H ++++++	9
0A00H ++++++	10
0B00H ++++++	11
0C00H ++++++	12
0D00H ++++++	13
0E00H ++++++	14
0F00H ++++++	15
1000H ++++++	16
1100H ++++++	17
1200H ++++++	18
1300H ++++++	19
1400H ++++++	20
1500H ++++++	21
1600H ++++++	22
1700H ++++++	23
+	
01234567890123456789012345678901	

Figura 20: La Mesa Del Patrón De Carácter De Gráficos

El color fronterizo está definido por VDP Mode Register 7 y es inicialmente azul. El Colour Table ocupa 6 KB de VRAM de

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

El 2000H para 37FFH. Hay un byte exacto para el byte haciendo mapas del Character Pattern Table para el Colour Table pero, porque lleva un byte entero a definir a lo 0 colores pixel y 1

pixel, hay una resolución inferior para colores que para pixels. El cuatro bits inferior de una entrada Colour Table define el color de todos los 0 pixels en la correspondiente línea de ocho pixeles. El cuatro bits superior define el color de lo 1 pixels. El Colour Table es inicializado a fin de que el color 0 pixel y el color 1 pixel son azules para la mesa entera. Porque ambos colores son lo mismo hay que alterar un color cuándo un poco es el set en el Character Pattern Table.

Mode multicolor

El Name Table ocupa 768 bytes de VRAM de 0800H para 0AFFH, la pantalla haciendo mapas es lo mismo como en 32x24 Text Mode.

La mesa es en la que se puso las iniciales con el siguiente patrón de código de caracteres:

- El 00H para 1FH (Repetidas cuatro veces)
- El 20H para 3FH (Repetidas cuatro veces)
- El 40H para 5FH (Repetidas cuatro veces)
- El 60H para 7FH (Repetidas cuatro veces)
- El 80H para 9FH (Repetidas cuatro veces)
- A0H para BFH (Repetidas cuatro veces)

Al igual que con Graphics Mode éste está simplemente un patrón del "conductor" de código de caracteres, es el Character Pattern Table que es modificado durante la operación normal.

La mesa Character Pattern ocupa 1536 bytes de VRAM de 0000H para 05FFH. Como en los otros modos cada carácter codifican mapas encima de un ocho bloque de byte en el Character Pattern Table.

Por la resolución inferior en este modo sólo dos bytes del bloque del patrón son realmente necesarios a definir un patrón del 8x8:

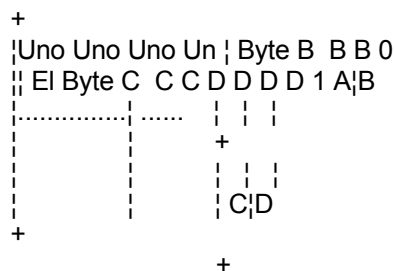


Figura 21: El Bloque Multicolor Del Patrón

Como puede ser visto de Figure 21 que cada capítulo de cuatro bits del bloque de dos bytes contiene un código de color y así define el COLOR de un cuadrante del patrón 8x8 pixel. A fin de que los bytes ocho enteros del patrón se bloquean puede ser utilizado un código de caracteres dado usará un el capítulo dos diferente de byte dependiendo

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

En la posición de la pantalla del código de caracteres (o sea su colocación en el Name Table):

- Videograbre fila 0, 4, 8, 12, 16, 20 bytes Uses 0 y 1
- Videograbre fila 1, 5, 9, 13, 17, 21 bytes Uses 2 y 3
- Videograbre fila 2, 6, 10, 14, 18, 22 bytes Uses 4 y 5

Videograbado fila 3, 7, 11, 15, 19, 23 bytes Usos 6 y 7

Cuando el Name Table se llena del conductor especial secuencia de códigos de caracteres exteriorizado por encima del Character Pattern Table será leído fuera de linealmente durante un marco de vídeo:

01234567890123456789012345678901	
0000H + ----- + 0	
0002H ++++++	1
0004H ++++++	2
0006H ++++++	3
0100H ++++++	4
0102H ++++++	5
0104H ++++++	6
0106H ++++++	7
0200H ++++++	8
0202H ++++++	9
0204H ++++++	10
0206H ++++++	11
0300H ++++++	12
0302H ++++++	13
0304H ++++++	14
0306H ++++++	15
0400H ++++++	16
0402H ++++++	17
0404H ++++++	18
0406H ++++++	19
0500H ++++++	20
0502H ++++++	21
0504H ++++++	22
0506H ++++++	23
+ 01234567890123456789012345678901	

Figura 22: La Mesa Multicolora Del Patrón De Carácter

El color fronterizo está definido por VDP Mode Register 7 y es inicialmente azul. No hay el separado Colour Table como los colores están definidos directamente por los contenidos del Character Pattern Table, éste se llena inicialmente de azul.

Los bloques de objetos móviles

El VDP puede controlar treinta y dos bloques de objetos móviles en todos los modos excepto 40X24 Text Mode. Su tratamiento es idéntico en todos los modos e independiente de cualquier actividad orientada en carácter.

El Sprite Attribute Table ocupa 128 bytes de VRAM de

2. VIDEOGRABADO PROCESADOR DE DESPLIEGUE

El 1B00H para 1B7FH. La mesa contiene treinta y dos bloques de byte, uno para cada bloque de objetos móviles. El primer bloque controla bloque de objetos móviles 0 (el bloque de

objetos movibles "máximo"), el segundo controla bloque de objetos movibles 1 etcétera para el bloque de objetos movibles 31. El formato de cada bloque está como debajo:

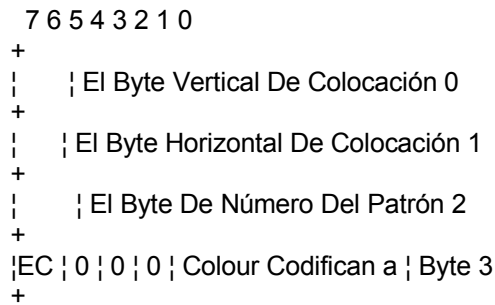


Figura 23: El Bloque De Atributo De Bloque De Objetos Movibles

El byte 0 especifica la coordenada vertical (Y) del pixel dejado en parte superior del bloque de objetos movibles. El sistema de coordenadas marcha de - 1 (FFH) para la línea máxima del pixel en la pantalla hasta 190 (BEH) para el meollo del asunto. Aprecia menos que - 1 puede estar acostumbrado a deslizar el bloque de objetos movibles en desde lo alto de la pantalla. Los valores exactos necesitados obviamente dependerán del tamaño del bloque de objetos movibles. Curiosamente no ha habido intento en MSX BASIC para reconciliar este sistema de coordenadas con el rango normal de gráficos de Y = 0 para 191.

Como una consecuencia un bloque de objetos movibles siempre será un pixel inferior en la pantalla que el punto gráfico equivalente. Noto que el valor coordinado vertical especial de 208 (D0H) colocó en un bloque de atributo de bloque de objetos movibles causará el VDP para ignora todos subsiguientes bloques en el Sprite Attribute Table. De manera impresionante esto quiere decir que cualquier bloques de objetos movibles inferiores desaparecerán de la pantalla.

El byte 1 especifica la coordenada horizontal (X) del pixel dejado en parte superior del bloque de objetos movibles. El sistema de coordenadas marcha de 0 para el leftmost pixel para 255 (FFH) para lo de la extrema derecha. Como este sistema de coordenadas no provee mecanismo para deslizar un bloque de objetos movibles en de la izquierda un bit especial en el byte 3 está usado con este propósito, vea debajo.

El byte 2 selecciona uno de los doscientos patrones del pedacito de cincuenta y seis 8x8 disponible en el Sprite Pattern Table. Si el bit Size es incrustado en VDP Mode Register 1, entonces resultando en patrones del bit del 16x16 ocupando treinta y dos bytes cada uno, los dos añicos menos significativos del número del patrón están ignorados. Así el patrón números 0, 1, 2 y 3 todos seleccionarían patrón número 0.

En Byte 3 los añicos Code cuatro Colour definen el color de lo 1 pixels en el bloque de objetos movibles los patrones, 0 pixels están siempre

2. VIDEOGRABE PROCESADOR DE DESPLIEGUE

Transparente. El bit Tempranero Clock es normalmente 0 sino que intercambiarán el bloque de objetos movibles treinta y dos pixels para la izquierda cuándo set para 1. Esto está a fin de que los bloques de objetos movibles no pueden ir en disminución adentro de la izquierda de la

pantalla, allí ser coordenadas de repuesto en la dirección horizontal.

El Sprite Pattern Table ocupa 2 KB de VRAM de 3800H para 3FFFH. Contiene doscientos patrones cincuenta y seis 8x8 pixel, numerados de 0 para 255. Si el Size mordió en VDP Mode Register 1 es 0, resultante en bloques de objetos movibles del 8x8, entonces luego cada uno que bloque del patrón de bloque de objetos movibles de ocho bytes está estructurado al modo del patrón de carácter se bloquea mostrado en Figure 18. Si el Size mordido tiene 1 años de edad, resultando en bloques de objetos movibles del 16x16, entonces luego cuatro ocho bloques de byte son necesarios para definir el patrón como debajo:

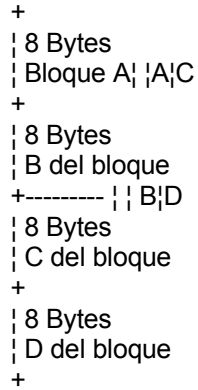


Figura 24: 16x16 Sprite Pattern Block

Así como canales en buen estado tres controlantes lo 8910 PSG contienen dos ocho puertos mordidos de datos, llamado UNO y B, a través del cual interconecta las palancas de control y la entrada del casete. El PSG se aparece ante el Z80 como tres que yo los puertos de la / O le llamé el Address Port, la Data Write Port y la Data Port Leído.

La dirección Port (yo / la O exporto a A0H)

El PSG contiene dieciséis registros internos que completamente definen su operación. Un registro específico es seleccionado escribiendo su número, de 0 para 15, para este puerto. Una vez que se selecciona, repitieron que las vías de entrada para ese registro pueden estar hechas por los dos puertos de datos.

Los datos Escriben A Port (yo / la O exporto a A1H)

Este puerto se usa para escribir a cualquier registrador una vez que ha sido seleccionado por el Address Port.

Los datos Leen a Port (yo / la O exporto a A2H)

Este puerto se usa para leer cualquier registro una vez que ha sido seleccionado por el Address Port.

Registra 0 Y 1

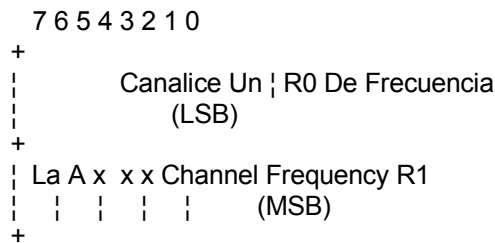


Figura 25

Estos dos registros se usan para definir la frecuencia del Tone Electrónico para A Channel. Las frecuencias variables se producen dividiendo una frecuencia maestra fija con el número mantenido en Registers 0 y 1, este número puede estar del rango 1 para 4095. Regístrele 0 agarres los añicos menos ocho significativos y Register 1 lo más significativo cuatro. El PSG divide una la frecuencia 1.7897725 externa MHz a las dieciséis para producir una frecuencia maestra Electrónica Tone de 111,861 Hz. La salida del Tone Generator

Por consiguiente puede extenderse de 111,861 Hz (divida a las 1) hasta 27.3 Hz (divida a las 4095). Por poner un ejemplo para producir una "A" intermedia (440 Hz) el valor divisionista en Registers 0 y 1 serían 254.

Registra 2 Y 3

Estos dos registros controlan la B Channel Tone Electrónico por lo que respecta a Channel.

Registra 4 Y 5

Estos dos registros controlan la C Channel Tone Electrónico por lo que respecta a Channel.

Registro 6

```

7 6 5 4 3 2 1 0
+
| x x Noise Frequency
+----- + Figura 26

```

Además de ola Tone de tres cuadrados Generators el PSG contiene un solo Generador Noise. La frecuencia fundamental de la fuente de ruido puede controlarse en una moda similar para el Tone Generators. Los cinco ańicos menos significativos de Register 6 sostienen un valor divisionista de 1 para 31. La frecuencia maestra Electrónica Noise es 111,861 Hz como antes.

Registro 7

```

7 6 5 4 3 2 1 0
+
| Exporte A | El De Babor C|B|A|C|B|A
|| El Tono De | Tono De | Tono De | Ruido De | Ruido De | Ruido Del Dir De La | A B Dir
+----- + Figura 27

```

Este registro faculta o lisa al Tone Generator y Noise Electrónico para cada uno de los tres canales: 0 = Facultan A 1 = Disable. También controla la dirección de A de puertos de la interfaz y B, para el cual las palancas de control y el casete están adjuntos: 0 = Introducen En La Computadora, 1 = Output. Registro 7 siempre debe contener 10xxxxxx o el daño posible podría resultar para el PSG, hay dispositivos activos a los que se conectó su yo los alfileres de la / O. La declaración de "sonido" de BASIC forzará estos ańicos para el valor correcto para Register 7 pero no hay protección en el nivel de código de la máquina.

3. EL GENERADOR EN BUEN ESTADO PROGRAMABLE

Registro 8

```

7 6 5 4 3 2 1 0

```



```

+
| La A x x Mode Channel Amplitude
+

```

Figura 28

Los bits cuatro Amplitude determinan la amplitud de A Channel de un mínimo de 0 para un máximo de 15. El bit Mode selecciona a uno u otro centró o moduló amplitud: 0 = Se Concentraron, 1 = Modulated. Cuando la amplitud modulada es seleccionada el valor fijo de amplitud es ignorado y el canal está modulado por la salida del Envelope Generator.

Registro 9

Este registro controla la amplitud de B Channel por lo que respecta a Channel.

Registro 10

Este registro controla la amplitud de C Channel por lo que respecta a Channel.

Registra 11 Y 12

```

7 6 5 4 3 2 1 0
+
|El sobre Frequency (LSB) | R11
+
|El sobre Frequency (MSB) | R12
+

```

Figura 29

Estos dos registros controlan que la frecuencia del solo Envelope Generator usó para la modulación de amplitud. Por lo que respecta al Tone Generators esta frecuencia está resuelta colocando una cuenta divisionista en los registros. El valor divisionista puede extenderse desde 1 para 65535 con Register 11 soporte los bits menos ocho significativos y Register 12 lo más significativo. La frecuencia maestra para el Envelope Generator es 6991 Hz así es que el sobre frecuencia puede extenderse de 6991 Hz (divida a las 1) para 0.11 Hz (divida a las 65535).

- 23 -

3. EL GENERADOR EN BUEN ESTADO PROGRAMABLE

Registro 13

```

7 6 5 4 3 2 1 0
+
| x x x Envelope Shape
+

```


al único le pueden ser leídos a la vez. El que debe ser seleccionado para leer está resuelto por el bit Selecto Joystick en PSG Register 15.

El Keyboard Mode mordió es no acostumbrado en UK labra a máquina. En máquinas japonesas está atado a un enlace del vestido sin mangas para determinar el conjunto de caracteres del teclado.

El Cassette Input está acostumbrado a leer la señal de la salida de la OREJA del casete. Esto es hecho pasar por un comparador para limpiar los bordes y para mutar para niveles digitales pero está otrora natural.

Registro 15

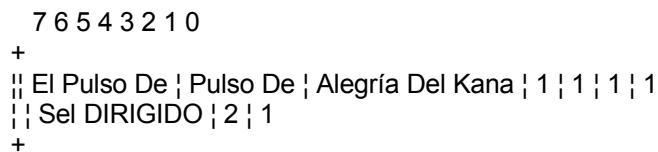


Figura 33

Este registro se usa para devolver para B PSG Port. Los cuatro añicos menos significativos están conectados por TTL amortiguadores colectores en claro para alfileres 6 y 7 de cada conector de la palanca de control. Están normalmente listos a uno 1, cuando un remo o la palanca de control está conectada, a fin de que los alfileres pueden funcionar como las entradas. Cuando un touchpad está conectado son utilizados como salidas de apretón de manos.

Los añicos dos Pulse se usan para generar un pulso que sale de positivo pequeño para cualquier remos adjunto a la presente para conectores de la palanca de control 1 o 2. Cada remo contiene a un cronometrador monoestable con un reostato variable controlando su largo de pulso. Una vez que el cronometrador se desencadena la colocación del reostato variable puede estar resuelta contando hasta las interrupciones monoestables.

El bit Selecto Joystick determina cuál conector de la palanca de control está conectado para la A PSG Port para la entrada: 0 Connector 1, 1 = Connector 2.

La salida Kana LED está sin uso en UK labra a máquina. En máquinas japonesas se usa para conducir un señalizador de modo del teclado.

4. ROM BIOS

El propósito del MSX ROM tiene importancia si los programas de código de la máquina deben ser desarrollados eficazmente y Operate de fuente fidedigna. Casi cada programa, incluir el BASIC Interpreter mismo, tomará un cierto set de funciones del hombre primitivo a manejar.

Estos incluyen conductores de la pantalla y de la impresora, un decodificador del teclado y otro hardware relataron funciones. Separando estas rutinas del BASIC Interpreter puede estar a los que se hizo disponible a cualquier programa de aplicación. El capítulo de ROM de 0000H para 268BH es mayormente devoto para tales rutinas y se llama los ROM BIOS (el Sistema Básico de Salida Input).

Este capítulo da una descripción funcional de cada rutina de la separata del reconocibly en los ROM BIOS. La atención especial es dada a las rutinas "estándar". Estos están documentados por Microsoft y garantizados guardar consecuencia cambios del hardware directo posible y del software. La primera parte pocos cien bytes del ROM consta de instrucciones Z80 JP que proveen arregladas sitúan puntos de entrada para estas rutinas. Para la compatibilidad máxima con software futuro un programa de aplicación debería restringir su adicción en el ROM para estas posiciones sólo. La descripción del ROM empieza con esta lista de proposiciones de entrada para las rutinas estándar. Un comentario conciso está posado con cada punto de entrada, la descripción llena recibe con la rutina misma.

Datos Areas

Se espera que la mayoría de usuarios tendrán el deseo de desensamblar al ROM hasta cierto punto (el listado lleno va en carrera casi cuatrocientas páginas). Para aliviar este proceso las áreas de datos, cuál no contiene código ejecutable Z80, es exteriorizado debajo:

73E4H-73E4H de 4B3AH-4B4CH de 185DH-1863H de 0004H-0007H
752EH-7585H de 4C2FH-4C3FH de 1B97H-1BAAH de 002BH-002FH
7754H-7757H de 555AH-5569H de 1BBFH-23BEH de 0508H-050DH
7BA3H-7BCAH de 5D83H-5DB0H de 2439H-2459H de 092FH-097FH
7ED8H-7F26H de 6F76H-6F8EH de 2CF1H-2E70H de 0DA5H-0EC4H
7F41H-7FB6H de 70FFH-710CH de 3030H-3039H de 1033H-105AH
7FBEH-7FFFH de 7182H-7195H de 3710H-3719H de 1061H-10C1H
71A2H-71B5H de 392EH-3FE1H de 1233H-1252H
71C7H-71DAH de 43B5H-43C3H de 13A9H-1448H
72A6H-72B9H de 46E6H-46E7H de 160BH-1612H

Noto que estas áreas de datos son para el UK ROM, hay diferencias leves en el ROM japonés referente al decodificador del teclado y el conjunto de caracteres de vídeo. Las disparidades entre los ROMs están restringidas para estas regiones con la masa del código siendo idénticas en ambos casos.

- 26 -

4. ROM BIOS

La terminología

La referencia está frecuentemente hecha en este capítulo para las rutinas estándar y para las variables Workspace Area. Cuandoquiera esto está hecho el nombre recomendable a Microsoft es usado en cartas de caja alta, para el ejemplo "la rutina del estándar FILVRM" y "SCRMOD está listo.

Subrutines que no es nombrado es referido por una dirección del parenthesized, "la pantalla es

despejada (0777H)" por ejemplo. Cuándo la referencia es hecha a las buenas costumbres de lenguaje de ensambla de banderas de estado Z80 son usadas, pues el ejemplo "bandera C" significaría que lo lleve la bandera es set mientras "la Bandera NZ" quiere decir que la bandera de cero es vuelta a arrancar. Las condiciones "EI" y "DI" término medio posibilitado interrumpen y deshabilitado interrumpe respectivamente.

ADDR. NOMBRE FUNCIONAR

----- 0000H CHKRAM 02D7H 0004H que se energiza arriba de la RAM del cheque..... Dos bytes, la dirección de 0006H de conjunto de caracteres ROM..... Un byte, el 0007H de número VDP Data Port..... Un byte, el 000BH de carácter de programa de BASIC de número VDP Data Port 0008H SYNCHR 2683H Check..... NOP 000CH RDSL 01B6H Leyó la RAM en cualquier 000FH de la ranura..... NOP 0010H CHRGR 2686H Get después 0013H de carácter de programa de BASIC..... NOP 0014H WRSL 01D1H Write para METER A LA FUERZA cualquier 0017H de la ranura..... El 1B45H NOP 0018H OUTDO Output para 001BH del dispositivo en uso en estos momentos..... La rutina NOP 001CH CALSL 0217H Call en cualquier 001FH de la ranura..... El registro NOP 0020H DCOMPR 146AH Compare para al 0023H HL y Delaware..... NOP 0024H ENASLT 025EH Enable cualquier ranura permanentemente 0027H..... El 002BH de tipo del operand de BASIC NOP 0028H GETYPR 2689H Get..... Rutina Call 0205H CALLF Version Number 0030H de cinco bytes en cualquier 0033H de la ranura..... Manipulador Interrupt 0C3CH KEYINT cinco NOPs 0038H, teclado escudriña que 003BH INITIO 049DH Initialize que yo la llave de función de dispositivos de la / O 003EH INIFNK 139DH Initialize ato con una cuerda pantalla de la pantalla 0041H DISSCR 0577H Disable 0044H ENASCR 0570H Enable 0047H WRTVDP 057FH Write para cualquier registro VDP 004AH RDVRM 07D7H Leyó el byte de byte VRAM 004DH WRTVRM 07CDH Write para VRAM 0050H SETRD 07ECH Set arriba de VDP para leído 0053H SETWRT 07DFH Set arriba de VDP para escribe el bloque 0056H FILVRM 0815H Fill de VRAM con bloque de byte de datos 0059H LDIRMV 070FH Copy para la memoria de bloque VRAM 005CH LDIRVM 0744H Copy para VRAM, de modo de memoria 005FH CHGMOD 084FH Change VDP

- 27 -

4. ROM BIOS

El 0065H de colores 0062H CHGCLR 07F7H Change VDP..... El manipulador NOP 0066H NMI 1398H Non Maskable Interrupt 0069H CLRSR 06A8H Clear todo bloque de objetos movibles 006CH INITXT 050EH Initialize VDP para 40x24 Text Mode 006FH INIT32 0538H Initialize VDP para 32x24 Text Mode 0072H INIGRP 05D2H Initialize VDP para Graphics Mode 0075H INIMLT 061FH Initialize VDP para el Multicolor Mode 0078H SETTXT 0594H Colocó VDP para 40x24 Text Mode 007BH SETT32 05B4H Colocó a VDP para 32x24 Text Mode 007EH SETGRP 0602H Set VDP para Graphics Mode 0081H SETMLT 0659H Set VDP para dirección Multicolora Mode 0084H CALPAT 06E4H Calculate de dirección del patrón de bloque de objetos movibles 0087H CALATR 06F9H Calculate de carácter de tamaño de bloque de objetos movibles de atributo de bloque de objetos movibles 008AH GSPSIZ 0704H Get 008DH GRPPRT 1510H Print en gráficoLa pantalla 0090H GICINI 04BDH Initialize PSG 0093H WRTPSG 1102H (GI Chip) Write para cualquier registro PSG 0096H RDPSG 110EH Leyó de cualquier amortiguador del teclado de música de registro PSG 0099H STRTMS 11C4H Start dequeuing 009CH CHSNS 0D6AH Sense para el carácter de carácter 009FH CHGET 10CBH Get de carácter de prueba de estado de la impresora de salida de carácter de la impresora de salida de carácter del amortiguador del teclado 00A2H CHPUT 08BCH Screen 00A5H LPTOUT 085DH Line 00A8H LPTSTT 0884H Line 00ABH CNVCHR 089DH Convert (la espera) con línea gráfica del encabezado 00AEH PINLIN 23BFH Get de línea de la consola 00B1H INLIN 23D5H Get (el editor) de consola 00B4H QINLIN 23CCH (el editor) Display "?", Traiga línea de consola 00B7H BREAKX 046FH Check (el editor) CTRL-STOP llave directamente 0848H de Lingüísticas Computacionales del 00C3H del pip del 1113H del PIP del 00C0H de la llave de la llave 00BAH

ISCNTC 03FBH Check CRTL-STOP 00BDH CKCNTC 10F9H Check CTRL-STOP Go colocación del cursor del 088EH del 00C6H de la pantalla Clear POSIT Set 00C9H FNKSB 0B26H Check si funcione despliegue crucial en llaves función despliegue llave función 00CCH ERAFNK 0B15H Erase 00CFH DSPFNK 0B2BH Display 00D2H TOTEXT 083BH Return VDP para la entrada de estado del remo de estado de la almohadilla de toque de estado del gatillo de estado de la palanca de control de modo del texto 00D5H GTSTCK 11EEH Get 00D8H GTTRIG 1253H Get 00DBH GTPAD 12ACH Get 00DEH GTPDL 1273H Get 00E1H TAPION 1A63H Tape EN 00E4H TAPIN1ABCH Tape la entrada 00E7H TAPIOF 19E9H Tape la entrada FUERA DE la salida 00EAH TAPOON 19F1H Tape EN la salida de salida 00EDH TAPOUT 1A19H Tape 00F0H TAPOOF 19DDH Tape FUERA DE motor 00F3H STMOTR 1384H Turn ON/OFF 00F6H LFTQ 14EBH Space salió en el byte de la cola de música 00F9H PUTQ 1492H Put en dirección física del pixel de corriente de derecho de la dirección física del pixel de corriente de la cola de música 00FCH RIGHTC 16C5H Move 00FFH LEFTC 16EEH Move izquierda

- 28 -

4. ROM BIOS

La dirección física del pixel de corriente 0102H UPC 175DH Move arriba de 0105H TUPC 173CH Test luego UPC si la dirección legal de reconocimiento médico del pixel de corriente 0108H DOWNC 172AH Move abajo de 010BH TDOWNC 170AH Test luego DOWNC si gráfico legal de coordenadas de gráficos 010EH SCALXY 1599H Scale 0111H MAPXYC 15DFH Map coordina para el byte de atributo de la dirección física del pixel de corriente de la dirección física del pixel de corriente de la dirección física 0114H FETCHC 1639H Fetch 0117H STOREC 1640H Store 011AH SETATR 1676H Set 011DH READC 1647H Leyese el atributo de atributo coetáneo pixel 0120H SETC 167EH Set de atributo coetáneo pixel 0123H NSETCX 1809H Set de número de proporción dimensional pixels 0126H GTASPC 18C7H Get 0129H PNTINI 18CFH Paint inicializa a 012CH SCANR 18E4H ScanLos pixels para enderezar a 012FH los pixels SCANL 197AH Scan para izquierdo 0132H CHGCAP 0F3DH Change Caps salida sonido Lock LED 0135H CHGSND 0F7AH Change Key Click 0138H RSLREG 144CH Read Primary Slot Register 013BH WSLREG 144FH Write para fila Primaria Slot Register 013EH RDVDP 1449H Read VDP Status Register 0141H SNSMAT 1452H Read de matriz del teclado 0144H PHYDIO 148AH Disk, ningún 148EH de FORMATO de 0147H de acción Disk, ningún acción 014AH ISFLIO 145FH Check para archivo yo la salida / O 014DH OUTDLP 1B63H Formatted para aplicar delineador a amortiguador de la teclado de la puntero de la voz de la música de la puntero de la voz de la música de la impresora 0150H GETVCP 1470H Get 0153H GETVC2 1474H Get 0156H KILBUF 0468H Clear 0159H CALBAS 01FFH Call paraEl BASIC de cualquier 015CH de la ranura..... Los NOPs para 01B5H para la expansión

La dirección ... 01B6H
 El nombre..... RDSL
 La entrada..... Un = Slot Idaho, HL = Address
 La salida..... Un = Byte leyó
 Modifica.. AF, BC, DE, DI

La rutina estándar para leer un solo byte de memoria en cualquier ranura. El Slot Identifier está compuesto de un número Primario Slot un número Secundario Slot y una bandera:

7 6 5 4 3 2 1 0
 +
 | Bandera | 0 | 0 | 0 La Ranura | Secundaria La Ranura # | Primaria
 +-----+ + Figura 34: La ranura Idaho

La bandera es normalmente 0 sino que deben tener

er 1 años de edad si un número Secondary Slot es incluido en el Slot Idaho. La dirección de memoria y Slot Idaho son primero tratado (027EH) para ceder un set de máscaras mordidas para aplicarse al registro pertinente de la ranura. Si un número Secundario Slot es

- 29 -

4. ROM BIOS

Especifiqué que luego el Slot Secundario Register es primero modificado para seleccionar la página pertinente de ese Slot Secundario (02A3H). El Slot Primario es entonces cambiado en para el espacio de la dirección Z80, el byte lee y el Primario Slot recuperado para su original incrustando a por la rutina RDPRIM en el Workspace Area. Finalmente, si un número Secundario Slot es incluido en el Slot Idaho, entonces la Ranura Secundaria original Registro sedimentándose está recuperado (01ECH).

Note eso, a menos que es la ranura conteniendo al Workspace Area, cualquier intento para ganar acceso a página 3 (C000H para FFFFH) causará el sistema para chocar como RDPRIM se cambiará a sí mismo fuera. Noto también eso interrumpe se queda deshabilitada por toda la memoria cambiando rutinas.

La dirección ... 01D1H
El nombre..... WRSLT
La entrada..... Una = Ranura CÉDULA DE IDENTIDAD, HL = Discurso, E = Byte a escribir
La salida..... Ninguno
Modifica.. AF, BC, D, DI

La rutina estándar para escribir un solo byte para la memoria en cualquier ranura. Su operación es fundamentalmente así como eso de la rutina del estándar RDSLTL excepto que la rutina Workspace Area WRPRIM es usada en vez de RDPRIM.

La dirección ... 01FFH
El nombre..... CALBAS
La entrada..... IX = Address
La salida..... Ninguno
Modifica.. AF ', BC ', DE ', HL ', IY, DI

La rutina estándar para llamar una dirección en el BASIC Interpreter de cualquier ranura. Usualmente esto será de un programa de código de la máquina marchando en una extensión ROM en página 1 (4000H para 7FFFH). El byte alto de par de registro IY es cargado con el MSX ROM Slot Idaho (00H) y control transfiere para la rutina del estándar CALSLT.

La dirección ... 0205H
El nombre..... CALLF
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF ', BC ', DE ', HL ', IX, IY, DI

La rutina estándar para llamar una dirección en cualquier ranura. El Slot Idaho y dirección son suministrados como parámetros del inline en vez de en registros para acomodar dentro de un gancho (el Capítulo 6), por ejemplo:

RST 30H
DEFB Haga Una Ranura En Idaho

DEFW Address
ENRIE

El Slot Idaho es primero cobrado y acomodado en el byte alto de par de registro IY. La dirección es luego colocada en par de registro

- 30 -

4. ROM BIOS

IX y caídas de control en la rutina del estándar CALSLT.

La dirección ... 0217H
El nombre..... CALSLT
La entrada..... IY (el byte Alto) = Haga Una Ranura En = Address Idaho, IX
La salida..... Ninguno
Modifica.. AF ', BC ', DE ', HL ', DI

La rutina estándar para llamar una dirección en cualquier ranura. Su operación es fundamentalmente así como eso de la rutina del estándar RDSLTL excepto que la rutina Workspace Area CLPRIM es usada en vez de RDPRIM. Noto que CALBAS y CALLF son proposiciones de entrada simplemente especializadas para esta rutina estándar que ofrecen una reducción en la cantidad de código tomó.

La dirección ... 025EH
El nombre..... ENASLT
La entrada..... Un = Slot Idaho, HL = Address
La salida..... Ninguno
Modifica.. AF, BC, DE, DI

La rutina estándar a cambiar de decisión en una página permanentemente de cualquier ranura. A diferencia del RDSLTL, WRSLTL y rutinas del estándar CALSLTL el Primary Slot cambiando de decisión son realizados directamente y no por una rutina Workspace Area. Consecuentemente las direcciones de adentro mandan a llamar 0 (0000H para 3FFFH) causarán un choque inmediato de sistema.

La dirección ... 027EH

Esta rutina es usada por la memoria cambiando rutinas estándar para revolver una dirección, en par de registro HL, y un Slot Idaho, adentro registra UNO, en un set de máscaras mordidas. Como un ejemplo un Slot Idaho de FxxxSSPP y una dirección en Page 1 (4000H para 7FFFH) devolvería lo siguiente:

Registro B = 00 00 PP 00 (la máscara de Oregon)
Registro C = 11 11 00 11 (Y la máscara)
El registro D=PP PP PP (Replegado)
Registro E = 00 00 11 00 (la máscara de la Página)

La B de registros y C están derivativas del número Primario Slot y la máscara de la página. Están más tarde acostumbrados a mezclar el número Primario nuevo Slot en los contenidos existentes del Slot Primario Register.
La D de registro contiene el número de la Ranura Primario replegado cuatro veces y la E de registro la máscara de la página. Esto se produce examinando los dos añicos más significativos de la dirección, para determinar el número de la página, y luego intercambiando la máscara

adelante para la colocación pertinente. Estos registros son más tarde usados durante el Secundario Slot cambiando de decisión.

Como la rutina termina bit 7 del Slot Idaho es probado, determinar si un Slot Secundario ha estado especificado, y la Bandera la M regresó si esto está así.

- 31 -

4. ROM BIOS

La dirección ... 02A3H

Esta rutina es usada por la memoria cambiando rutinas estándar para modificar a un Slot Secundario Register. El Slot a Idaho está suministrado en A de registro mientras D de registros y E contienen las máscaras mordidas mostradas en la rutina previa.

Los bits 6 y 7 de D de registro son primero copiado en el registro Primario Slot. Esto hace cambiar en página 3 del Slot Primario especificado por el Slot Idaho y le hace disponible al Slot Secundario requerido Register. Esto es entonces leído de posición de memoria FFFFH y la máscara de la página, invertido, solido descongestionarse los bits dos requeridos. El número Secundario Slot es intercambiado para la colocación pertinente y mezclado adentro. Finalmente el trasfondo nuevo es colocado en el Slot Secundario Register y el Slot Primario Register restituyó a su original sedimentándose.

La dirección ... 02D7H
El nombre..... CHKRAM
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, SP

La rutina estándar para realizar inicialización de memoria en energizar. Poco destructivamente experimenta para RAM en páginas 2 y 3 en todas las dieciséis ranuras posibles luego coloca que el Slot Primario y Secundario se registra para cambiar de decisión en el área más grande encontrado. De el Workspace entero Area (F380H para FFC9H) es puesto en el cero y EXPTBL y SLTTBL trazar un mapa cualquier interfaces de expansión en existencia Interrupt Mode 1 está colocado y el control se traslada para el resto de la rutina que se impulsa arriba (7C76H) de inicialización.

La dirección ... 03FBH
El nombre..... ISCNTC
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, EI

La rutina estándar a comprobar si el CTRL-STOP o DETENER llaves se han apretado. Es usado por el BASIC Interpreter al final de cada declaración para revisar en busca de terminación de programa. BASROM es primero examinado para ver si contiene un valor de poco cero, si así es que la rutina termina inmediatamente. Esto debe impedir a los usuarios rompiendo relaciones en cualquier extensión ROM conteniendo un programa de BASIC.

INTFLG es entonces a cuadros para determinar si lo interrumpe manipulador ha colocado al CTRL-STOP o códigos de la llave de ALTO (03H o 04H) allí. Si el ALTO ha sido detectado luego el cursor se enciende (09DAH) e INTFLG continuamente inspeccionó hasta uno de los dos códigos cruciales reaparece. El cursor es entonces revuelto apagado (0A27H) y, si la llave fuera SE DETIENE, la rutina termina.

Si CTRL-STOP ha sido detectado luego el amortiguador del teclado es primero aclarado por la rutina del estándar KILBUF y TRPTBL es

- 32 -

4. ROM BIOS

A cuadros para ver si uno "en STOP GOSUB" declaración es activo. Si así es que la entrada pertinente en TRPTBL es actualizada (0EF1H) y la rutina termina como el acontecimiento será manipulado por el Interpreter Runloop. De otra manera la rutina del estándar ENASLT se usa para cambiar en página 1 del MSX ROM, en el caso una extensión ROM usa la rutina, y las transferencias de control para el "alto" manipulador de declaración (63E6H).

La dirección ... 0468H
El nombre..... KILBUF
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. HL

El estándar Routine para descongestionar el cuarenta carácter amortiguador que se escribe a máquina delante del teclado KEYBUF. Hay dos punteros en este amortiguador, PUTPNT donde lo interrumpe personajes de lugares del manipulador, y GETPNT donde los programas de aplicación los van a traer de. Como el número de personajes en el amortiguador está indicado por la diferencia entre estos dos punteros KEYBUF es vaciado simplemente haciéndolos ambos iguales.

La dirección ... 046FH
El nombre..... BREAKX
La entrada..... Ninguno
La salida..... La C de la bandera si la llave CTRL-STOP presionó
Modifica.. AF

La rutina estándar que directamente experimenta hace avanzar con el remo 6 y 7 del teclado a determinar si el CTRL y las llaves de ALTO son ambos presionados. Si son luego KEYBUF es descongestionado y fila 7 de OLDKEY modificado para impedir lo interrumpe manipulador recogiendo las llaves igualmente. Esta rutina a menudo puede ser más adecuada para el uso por un programa de aplicación, preferentemente a ISCNTC, como surtirá efecto cuando interrumpe es deshabilitado, durante casete yo / la O por ejemplo, y no egresa para el Interpreter.

La dirección ... 049DH
El nombre..... INITIO
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, E, EI

La rutina estándar para inicializar al PSG y el Centronics Status Port. PSG Registre 7 es primero determinado para 80H haciendo B PSG Port = Output y A PSG Port = Input. PSG Registre 15 está listo para CFH para poner las iniciales en el hardware de control del conector Joystick. PSG Registre 14 son entonces leídos y el Keyboard Mode mordido colocado en KANAMD, éste no tiene relevancia para máquinas UK.

Finalmente un valor de FFH es salida para el Centronics Status Port (yo / la O exporto a 90H) para ajustar la señal de la LUZ ESTROBOSCÓPICA a gran altura. El control luego cae en la

rutina del estándar GICINI para completar inicialización.

- 33 -

4. ROM BIOS

La dirección ... 04BDH
El nombre..... GICINI
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. El

La rutina estándar para inicializar al PSG y las variables Workspace Area se asoció con el "juego" declaración. QUETAB, VCBA, VCB B y VCBC son primero inicializados con los valores mostrado en Chapter 6. PSG Registers 8, 9 y 10 están luego dispuestos a poner en el cero amplitud y PSG Register 7 para B8H. Esto faculta al Tone Generator y lisa al Noise Electrónico en cada canal.

La dirección ... 0508H

Este seis byte la mesa contiene el "juego" parámetros de declaración inicialmente colocados en VCBA, VCB B y VCBC por la rutina del estándar GICINI: La octava = 4, Length = 4, Tempo = 120, Volume 88H, = el 00FFH Envelope.

La dirección ... 050EH
El nombre..... INITXT
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para inicializar al VDP para el 40x24 Text Mode. La pantalla es por ahora deshabilitada por la rutina del estándar DISSCR y SCRMOD y OLDSCR determinado para 00H. Los parámetros requeridos por la rutina del estándar CHPUT son establecidos por LINI.40 copiador para LINLEN, TXTNAM para NAMBAS y TXTCGP para CGPBAS. Los colores VDP están luego colocados por la rutina del estándar CHGCLR y la pantalla es despejada (077EH). El conjunto de caracteres coetáneo es emulado en el VRAM Character Pattern Table (071EH). Finalmente el modo VDP y direcciones bajo están colocados por la rutina del estándar SETTXT y la pantalla es habilitada.

La dirección ... 0538H
El nombre..... EI INIT32
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para inicializar al VDP para el 32x24 Text Mode. La pantalla es por ahora deshabilitada por la rutina del estándar DISSCR y SCRMOD y OLDSCR determinado para 01H. Los parámetros requeridos por la rutina del estándar CHPUT son establecidos por LINL32 copiador para LINLEN, T32NAM para NAMBAS, T32CGP para CGPBAS, T32PAT para PATBAS y T32ATR para ATRBAS. Los colores VDP están luego colocados por la rutina del estándar CHGCLR y la pantalla es despejada (077EH). El conjunto de caracteres coetáneo es emulado en el VRAM Character Pattern Table (071EH) y todos los bloques de objetos movibles aclarados (06BBH). Finalmente el modo VDP y direcciones bajo están colocados por la rutina del estándar del SETT32 y la pantalla es habilitada.

4. ROM BIOS

La dirección ... 0570H
El nombre..... ENASCR
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, EI

La rutina estándar para habilitar la pantalla. Esto simplemente implica bit de trasfondo 6 de VDP Mode Register 1.

La dirección ... 0577H
El nombre..... DISSCR
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, EI

La rutina estándar para desactivar la pantalla. Esto simplemente implica reanudar bit 6 de VDP Mode Register 1.

La dirección ... 057FH
El nombre..... WRTVDP
La entrada..... El byte de la B = Data, el número C=VDP Mode Register
La salida..... Ninguno
Modifica.. AF, B, EI

La rutina estándar para escribir un byte de datos para cualquier VDP Mode Register. El byte de selección de a registro le es primero escrito para el VDP Command Port, seguido por el byte de datos. Esto es entonces copiado para la imagen pertinente de registro, RGOSAV para RG7SAV, en el Workspace Area

La dirección ... 0594H
El nombre..... SETTXT
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para parcialmente colocar al VDP para el 40x24 Text Mode. El añicos de modo M1, M2 y M3 son primero incrustados en VDP Mode Registers 0 y 1. La cinco base de la mesa VRAM las direcciones, el comienzo con TXTNAM, es entonces copiado del Workspace Area en VDP Mode Registers 2, 3, 4, 5 y 6 (0677H).

La dirección ... 05B4H
El nombre..... El SETT32
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para parcialmente colocar al VDP para el 32x24 Text Mode. El añicos de modo M1, M2 y M3 son primero incrustados en VDP Mode Registers 0 y 1. La cinco base de la mesa VRAM las direcciones, el comienzo con T32NAM, es entonces copiado del Workspace Area en VDP Mode Registers 2, 3, 4, 5 y 6 (0677H).

4. ROM BIOS

La dirección ... 05D2H
El nombre..... INIGRP
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para inicializar al VDP para Graphics Mode. La pantalla es por ahora deshabilitada por la rutina del estándar DISSCR y SCRMOD determinado para 02H. Los parámetros requeridos por la rutina del estándar GRPPRT son establecidos por GRPPAT copiadore para PATBAS y GRPATR para ATRBAS. El patrón del conductor de código de caracteres es entonces copiada en el VDP Name Table, la pantalla despejado (07A1H) y todos los bloques de objetos movibles aclarados (06BBH). Finalmente el modo VDP y direcciones bajo están colocados por la rutina del estándar SETGRP y la pantalla es habilitada.

La dirección ... 0602H
El nombre..... SETGRP
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para parcialmente colocar al VDP para Graphics Mode. El añicos de modo M1, M2 y M3 son primero incrustados en VDP Mode Registers 0 y 1. La cinco base de la mesa VRAM las direcciones, el comienzo con GRPNAM, es entonces copiado del Workspace Area en VDP Mode Registers 2, 3, 4, 5 y 6 (0677H).

La dirección ... 061FH
El nombre..... INIMLT
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para inicializar al VDP para el Multicolor Mode. La pantalla es por ahora deshabilitada por la rutina del estándar DISSCR y SCRMOD determinado para 03H. Los parámetros requeridos por la rutina del estándar GRPPRT son establecidos por MLTPAT copiadore para PATBAS y MLTATR para ATRBAS. El patrón del conductor de código de caracteres es entonces copiada en el VDP Name Table, la pantalla despejado (07B9H) y todos los bloques de objetos movibles aclarados (06BBH). Finalmente el modo VDP y direcciones bajo están colocados por la rutina del estándar SETMLT y la pantalla es habilitada.

La dirección ... 0659H
El nombre..... SETMLT
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para parcialmente colocar al VDP para el Multicolor Mode. El añicos de modo M1, M2 y M3 son primero incrustados en VDP Mode Registers 0 y 1. La cinco base de la mesa VRAM las direcciones, el comienzo con MLTNAM, es entonces copiado del Workspace

Area

- 36 -

4. ROM BIOS

Para VDP Mode Registers 2, 3, 4, 5 y 6.

La dirección ... 0677H

Esta rutina es usada por el SETTXT, SETT32, SETGRP y las rutinas del estándar SETMLT para copiar un bloque de direcciones de cinco mesas de bajo del Workspace Area en VDP Mode Registers 2, 3, 4, 5 y 6. En las proposiciones del par de registro de entrada HL para el grupo pertinente de direcciones. Cada dirección bajo es coleccionada a su vez intercambiado el número requerido de lugares y luego escrito para el Mode pertinente Register por la rutina del estándar WRTVDP.

La dirección ... 06A8H
El nombre..... CLRSPR
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para aclarar todos los bloques de objetos movibles. El entero Sprite de 2 KBES Pattern Table se llena primero de ceros por la rutina del estándar FILVRM. La coordenada vertical de cada uno de los treinta y dos bloques de atributo de bloque de objetos movibles es luego determinado - 47 (D1H) para colocar el bloque de objetos movibles por encima de la parte superior de la pantalla, la coordenada horizontal es a la que se dejó tal cual.

El patrón numera en el Sprite Attribute Table es en lo que se puso las iniciales con la serie 0, 1, 2, 3, 4, ... 31 para bloques de objetos movibles del 8x8 o la serie 0, 4, 8, 12, 16, ... 124 para bloques de objetos movibles del 16x16. La serie a ser generada está resuelta por el Size mordió en VDP Mode Register 1. Finalmente el byte de color de cada bloque de objetos movibles que el bloque de atributo está suplido con el código de color contuvo en FORCLR, éste es inicialmente blanco.

Noto que los añicos Size y Mag en VDP Mode Register 1 no son afectados por esta rutina. Note también que el INIT32, INIGRP y rutinas del estándar INIMLT usan esta rutina con un punto de entrada en 06BBH, dejando al Sprite Pattern Table ecuánime.

La dirección ... 06E4H
El nombre..... CALPAT
La entrada..... Un número del patrón = Sprite
La salida..... La dirección del patrón HL = Sprite
Modifica.. AF, DE, HL

La rutina estándar para calcular el discurso de un patrón de bloque de objetos movibles. El número del patrón es primero multiplicado por ocho luego, si los bloques de objetos movibles del 16x16 son seleccionados, multiplicado por un más factor de cuatro. Esto se agrega luego a la dirección bajo Table Pattern Sprite, tomado de PATBAS, para producir la dirección de cierre.

Este sistema de numeración está formados en línea con uso de BASIC Interpreter de números del patrón en vez de los VDP's cuando los bloques de objetos movibles del 16x16 son seleccionados. Como un ejemplo mientras el Interpreter llama el segundo patrón número uno, es realmente

4. ROM BIOS

VDP mode número cuatro. Este uso significa que el número máximo del patrón que esta rutina debería permitir, cuando los bloques de objetos móviles del 16x16 son seleccionados, es sesenta y tres. No hay cheque real en este acabado así es que el patrón grande que los números producirán dirige la palabra más grande que 3FFFH. Tales direcciones, estando aprobado para el otro VDP las rutinas, se enrollarán aproximadamente después del cero y le corromperán al Character Pattern Table en VRAM.

La dirección ... 06F9H
El nombre..... CALATR
La entrada..... Un número = Sprite
La salida..... La dirección de atributo HL = Sprite
Modifica.. AF, DE, HL

La rutina estándar para calcular el discurso de un bloque de atributo de bloque de objetos móviles. El número de bloque de objetos móviles, de cero para treinta y uno, es multiplicado a las cuatro y se suma a la dirección bajo Table Attribute Sprite tomado de ATRBAS.

La dirección ... 0704H
El nombre..... GSPSIZ
La entrada..... Ninguno
La salida..... Un = Bytes en patrón de bloque de objetos móviles (8 o 32)
Modifica.. AF

La rutina estándar para devolver el número de bytes ocupados por cada patrón de bloque de objetos móviles en el Sprite Pattern Table. El resultado es determinado simplemente examinando al Size bit en VDP Mode Register 1.

La dirección ... 070FH
El nombre..... LDIRMV
La entrada..... AC la dirección = Length, DE=RAM, la dirección HL=VRAM
La salida..... Ninguno
Modifica.. AF, BC, DE, EI

La rutina estándar para copiar un bloque en la memoria principal del VDP VRAM. El VRAM echando a andar dirección está listo por la rutina del estándar SETRD y luego los bytes secuenciales leyeron del VDP Data Port y colocado en la memoria principal.

La dirección ... 071EH

Esta rutina se usa para emular un conjunto de caracteres de 2 KBES en el VDP Character Pattern Table en cualquier modo. La dirección bajo del Character Pattern Table en VRAM es tomada de CGPBAS. La dirección de puesta en marcha del conjunto de caracteres es tomada de CGPNT. La rutina del estándar RDSLTL se usa para leer los datos de carácter tan este puede estar situada en una extensión ROM.

En CGPNT que se energiza arriba es inicializado con la dirección contenida en posición ROM 0004H, lo cual es 1BBFH. CGPNT está con holgura alterado para producir algunos resultados interesantes, PINCHAZO y HF920, y HC7:PANTALLA 0 provee un ejemplo a fondo confuso.

4. ROM BIOS

La dirección ... 0744H
El nombre..... LDIRVM
La entrada..... AC la dirección = Length, DE=VRAM, la dirección HL=RAM
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para copiar un bloque para VRAM de memoria principal. El VRAM echando a andar dirección está listo por la rutina del estándar SETWRT y los bytes luego secuenciales tomados de memoria principal y escrito para el VDP Data Port.

La dirección ... 0777H

Esta rutina despejará la pantalla en cualquier modo VDP. En 40x24 Text Mode y 32x24 Text Mode el Name Table, quién es aquel cuya dirección bajo es tomada de NAMBAS, se llena primero de espacios de ASCII. El cursor es luego determinado para la colocación de la casa (0A7FH) y LINTTB, la mesa de terminación de la línea, en la que se repuso las iniciales. Finalmente el despliegue de la llave de función está recuperado, si es posibilitado, por la rutina del estándar FNKSB.

En Graphics Mode el color fronterizo está primero determinado por VDP Mode Register 7 (0832H). El Colour Table se llena luego del código de color de fondo, tomado de BAKCLR, para ambos 0 y 1 pixels. Finalmente el Character del que Pattern Table es llenado pone en el cero.

En el Multicolor Mode el color fronterizo es primero determinado por VDP Mode Register 7 (0832H). El Character Pattern Table se llena luego del color de fondo tomado de BAKCLR.

La dirección ... 07CDH
El nombre..... WRTVRM
La entrada..... Una dirección = Data de byte, HL=VRAM
La salida..... Ninguno
Modifica.. EI

La rutina estándar para escribir un solo byte para el VDP VRAM. La dirección a VRAM le es primero establecido por la rutina del estándar SETWRT y luego el byte de datos para el VDP Data Port. Noto que el dos aparentemente instrucciones EX espurio (SP), HL en esta rutina, y varios otros, estoy obligado a encontrar las restricciones de oportunidad del momento de VDP's.

La dirección ... 07D7H
El nombre..... RDVRM
La entrada..... La dirección HL=VRAM
La salida..... Un = Byte leyó
Modifica.. AF, EI

La rutina estándar para leer un solo byte del VDP VRAM. La dirección VRAM es primero establecida por la rutina del estándar SETRD y luego el byte leyó del VDP Data Port.

4. ROM BIOS

La dirección ... 07DFH
El nombre..... SETWRT
La entrada..... La dirección HL=VRAM
La salida..... Ninguno
Modifica.. AF, EI

La rutina estándar para hacer caer en una trampa al VDP para subsiguiente escribe a VRAM por la Data Port. La dirección contuvo en par de registro HL está escrito para el VDP Command Port LSB primero, el segundo MSB como se muestra en Figure 7. Dirige la palabra más gran que el 3FFFH se enrollará aproximadamente después del cero como los dos años más significativos de la dirección estén ignorados.

La dirección ... 07ECH
El nombre..... SETRD
La entrada..... La dirección HL=VRAM
La salida..... Ninguno
Modifica.. AF, EI

La rutina estándar para hacer caer en una trampa al VDP para subsiguientes lecturas de VRAM por la Data Port. La dirección contuvo en par de registro HL está escrito para el VDP Command Port LSB primero, el segundo MSB como se muestra en Figure 7. Dirige la palabra más gran que el 3FFFH se enrollará aproximadamente después del cero como los dos años más significativos de la dirección estén ignorados.

La dirección ... 07F7H
El nombre..... CHGCLR
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, HL, EI

La rutina estándar para colocar los colores VDP. SCRMOD es primero examinado para determinar el curso de la acción apropiado. En 40x24 Text Mode los contenidos de BAKCLR y a FORCLR les es para VDP Mode Register 7 para colocar el color de los 0 y 1 pixels, éstos son inicialmente azules y blancos. Note eso en este modo no hay muy de especificar el color fronterizo, esto será lo mismo como el color 0 pixel. En 32x24 Text Mode, Graphics Mode o Mode Multicolor los contenidos de a BDRCLR le es para VDP Mode Register 7 para colocar el color del borde, éste es inicialmente azul. También en 32x24 Text Mode que los contenidos de BAKCLR y FORCLR son emulados para todo el Colour Table para determinar los 0 y 1 colores pixel.

La dirección ... 0815H
El nombre..... FILVRM
La entrada..... Una dirección = Data de byte, BC = Length, HL=VRAM
La salida..... Ninguno
Modifica.. AF, BC, EI

La rutina estándar de la que llenar un bloque del VDP VRAM un byte de datos del soltero. El VRAM echando a andar dirección, contenido en par de registro HL, es primero al que se hizo caer en una trampa por el estándar SETWRT

La rutina. El byte de a datos le es luego repetidamente para el VDP Data Port para llenar posiciones sucesivas VRAM.

La dirección ... 083BH
El nombre..... TOTEXT
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para devolver al VDP para ya sea 40x24 Text Mode o 32x24 Text Mode si está actualmente de Modo Graphics o el Multicolor Mode. Es usado por el BASIC Interpreter Mainloop y por la "entrada" manipulador de declaración. Cuandoquiera que el INITXT o rutinas estándar INIT32 está usado el byte de modo, 00H o 01H, es emulado en OLDSCR. Si el modo se varía subsiguientemente a Graphics Mode o el Multicolor Mode, y luego tiene que ser devuelto a uno de los dos modos del texto pues la entrada del teclado, esta rutina asegura que regresa al mismo.

SCRMOD es primero examinado y, si la pantalla está ya en ya sea el modo del texto, la rutina simplemente termina sin acción.
De otra manera el modo previo del texto es tomado de OLDSCR y pasado para la rutina del estándar CHGMOD.

La dirección ... 0848H
El nombre..... Las Lingüísticas Computacionales
La entrada..... Z de la bandera
La salida..... Ninguno
Modifica.. AF, BC, DE, EI

La rutina estándar para despejar la pantalla en cualquier modo, eso nada sino llama la rutina en 0777H. Éste son de hecho los "CLS" manipulador de declaración y, porque esto señala que hay texto ilegal después de la declaración, simplemente regresará si es entrado con Flag NZ.

La dirección ... 084FH
El nombre..... CHGMOD
La entrada..... Un modo = Screen requerido (0, 1, 2, 3)
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para ajustar un modo nuevo de la pantalla. La A de registro, conteniendo el modo requerido de la pantalla, es probado y control transferido para INITXT, INIT32, INIGRP o INIMLT.

La dirección ... 085DH
El nombre..... LPTOUT
La entrada..... Un = Character a escribir en letras de imprenta
La salida..... La C de la bandera si la terminación CTRL-STOP
Modifica.. AF

La rutina estándar para devolver un carácter para la impresora de la línea por el Centronics Port. El estado de la impresora es continuamente

Probado, por el estándar LPTSTT la rutina, hasta que el impresor se libre. Al carácter le es luego escado para el Centronics Data Port (yo / la O exporto a 91H) y la señal de la LUZ ESTROBOSCÓPICA del Centronics Status Port (yo / la O exporto a 90H) en breve el punto bajo pulsado. Noto que la rutina del estándar BREAKX se usa para experimentar para la llave de ALTO de CTRL-- si el impresor está ocupado. Si CTRL-STOP es detectado, entonces un código de RETORNO DE CARRO está escrito para el Centronics Data Port, para sonrojar el amortiguador de la línea de la impresora, y la rutina termina con ustedstedsuador .

La dirección ... 0884H
El nombre..... LPTSTT
La entrada..... Ninguno
La salida..... Una = 0 Z y Flag si la impresora ocupado
Modifica.. AF

La rutina estándar para probar la señal de ocupado Centronics Status Port. Esto justamente implica lectura yo el 90H del puerto de la / O y examinando la condición de bit 1: 0 = Alistan, 1 = Busy.

La dirección ... 088EH
El nombre..... COLOQUE
La entrada..... = La Columna De La H, = La Fila L
La salida..... Ninguno
Modifica.. AF, EI

La rutina estándar para ajustar el cursor coordina. La fila y coordenadas de la columna son enviadas a la rutina del estándar OUTDO como los parámetros en un ESC, "Y", Row + 1FH, secuencia del + 1FH Column. Noto que la casa BIOS que la colocación tiene se coordina de 1,1 en vez del 0,0 usado por el BASIC Interpreter.

La dirección ... 089DH
El nombre..... CNVCHR
La entrada..... Un = Carácter
La salida..... La bandera Z, Carolina Del Norte = Header; La bandera NZ, C = Graphic; La Bandera Z, = Normalidad C
Modifica.. AF

La rutina estándar para la que probar, y convertir si es necesario, personajes con encabezados gráficos. Los personajes menos de 20H son normalmente interpretados por los conductores del dispositivo de salida como personajes de control. Un código de caracteres en este rango puede ser tratado como un carácter explayable precediendo él con un código gráfico (01H) de control del encabezado y añadiendo 40H para su valor. Para que el ejemplo directamente exhibe código de caracteres 0DH, en vez de lo ha interpretado como un retorno de carro, hay que devolver el 01H de dos bytes, 4DH. Esta rutina es usada por los conductores del dispositivo de salida, algo semejante como la rutina del estándar CHPUT, a comprobar para tales secuencias.

Si el carácter es un encabezado gráfico GRPHED está colocado para 01H y la rutina ultima, entonces GRPHED diferente es puesto en el cero. Si el carácter está fuera del 40H de rango para 5FH, entonces es al que se dejó tal cual. Si está dentro este rango, y GRPHED contiene 01H indicando una caída de cabeza gráfica previa, entonces se convierte sustrayendo 40H.

La dirección ... 08BCH
El nombre..... CHPUT
La entrada..... Un = Carácter
La salida..... Ninguno
Modifica.. El

La rutina estándar para devolver un carácter para la pantalla en 40x24 Text Mode o 32x24 Text Mode. SCRMOD es primero a cuadros y, si el VDP estuviera ya sea el Modo Graphics o Mode Multicolor, la rutina termina sin acción. De otra manera el cursor está removido (0A2EH), el carácter descifrado (08DFH) y luego el cursor repuesto (09E1H). Finalmente la colocación de la columna del cursor es colocada en TTYPOS, pues el uso por la declaración "estampada", y la rutina terminan.

La dirección ... 08DFH

Esta rutina es usada por la rutina del estándar CHPUT para descifrar un carácter y tomar la acción apropiada. La rutina del estándar CNVCHR se usa primero para inspeccionar para un carácter gráfico, si el carácter es un código del encabezado (01H) que luego la rutina termina sin acción. Si el carácter es uno gráfico convertido luego el capítulo de desciframiento de código de control es saltado. ESCCNT diferente es al que se dio jaque para ver si un carácter previo (1BH) ESC ha sido recibido, si así es que control transfiere para el procesador de secuencia ESC (098FH). De otra manera el carácter es comprobado para ver si es más pequeño que 20H, si así es que las transferencias de control para el control codifican procesador (0914H). El carácter es entonces a cuadros para ver si es DEL (7FH), si así es que el control se traslada para lo suprime rutina (0AE3H).

Asumir el carácter es explayable que las coordenadas del cursor son tomadas de CSRY y CSRX y colocaron en = Row de registro del par HL, H = Column, L. Estos son luego convertidos a una dirección física en el VDP Name Table y el carácter acomodado allí (0BE6H). La colocación de la columna del cursor es entonces incrementada (0A44H) y, asumiendo la columna de la extrema derecha no ha sido excedido, la rutina termina. De otra manera la entrada de la fila en LINTTB, la mesa de terminación de la línea, es puesta en el cero para indicar un verso lógico extendido, el número de la columna está colocado para 01H y un LF es realizado.

La dirección ... 0908H

Esta rutina realiza la operación LF para el procesador de código de control de rutina del estándar CHPUT. La pelea del cursor es incrementada (0A61H) y, asumiendo que la fila mínima no ha sido excedida, la rutina termina. De otra manera la pantalla es enrollada arriba y la fila mínima borrada (0A88H).

La dirección ... 0914H

Éste es el procesador de código de control para la rutina del estándar CHPUT. La mesa en 092FH es registrada para un encuentro con lo

- 43 -

4. ROM BIOS

El código y el control transferido para la dirección asociada.

La dirección ... 092FH

Esta mesa contiene los códigos de control, cada uno con una dirección asociada, reconocida por la rutina del estándar CHPUT:

CODIFIQUE PARA FUNCIONAR

07H 1113H TOME LA FORMA DE UNA CAMPANA, salga de a emitir un pip
08H 0A4CH BS, cursor a la izquierda
La ETIQUETA 09H 0A71H, el cursor para después tabular colocación
0AH 0908H LF, cursor abajo de una fila
0BH 0A7FH DIRÍJASE HACIA EL BLANCO, cursor para la casa
0CH 077EH FORMFEED, casa y pantalla clara
EI RETORNO DE CARRO 0DH 0A81H, el cursor para columna del leftmost
1BH 0989H ESC, introduce secuencia de escapada
EI DERECHO 1CH 0A5BH, el derecho del cursor
1DH 0A4CH SALIÓ, el cursor a la izquierda
1EH 0A57H UP, cursor arriba
La PELUSA 1FH 0A61H, la pelusa del cursor.

La dirección ... 0953H

Esta mesa contiene los códigos de control ESC, cada uno con una dirección asociada, reconocida por la rutina del estándar CHPUT:

CODIFIQUE PARA FUNCIONAR

6AH 077EH ESC, "j", casa y pantalla clara
45H 077EH ESC, "E", casa y pantalla clara
4BH 0AEEH ESC, "K", claro a acabar de línea
4AH 0B05H ESC, "J", claro a acabar de pantalla
6CH 0AECH ESC, "I", línea clara
4CH 0AB4H ESC, "L", línea del inserto
4DH 0A85H ESC, "M", suprime línea
59H 0986H ESC, "Y", cursor determinado se coordina
41H 0A57H ESC, "un", cursor arriba
42H 0A61H ESC, "B", pelusa del cursor
43H 0A44H ESC, "C", derecho del cursor
44H 0A55H ESC, "D", cursor a la izquierda
48H 0A7FH ESC, "H", casa del cursor
78H 0980H ESC, "x", cursor de cambio
79H 0983H ESC, "y", cursor de cambio

La dirección ... 0980H

Esta rutina realiza al ESC, "x" operación para el procesador de código de control de rutina del estándar CHPUT. ESCCNT está listo para 01H para señalar que el siguiente carácter recibido es un parámetro.

La dirección ... 0983H

Esta rutina realiza al ESC, "y" operación para el CHPUT

El decodificador estándar de código de control de rutina. ESCCNT está listo para 02H para señalar que el siguiente carácter recibido es un parámetro.

La dirección ... 0986H

Esta rutina realiza al ESC "Y" operación para el procesador de código de control de rutina del estándar CHPUT. ESCCNT está listo para 04H para señalar que el siguiente carácter recibido es un parámetro.

La dirección ... 0989H

Esta rutina realiza la operación ESC para el procesador de código de control de rutina del estándar CHPUT. ESCCNT está listo para FFH para señalar que el siguiente carácter recibido es el segundo carácter de control.

La dirección ... 098FH

Esto es el procesador de secuencia de rutina del estándar CHPUT ESC. Si ESCCNT contiene a FFH luego el carácter es el segundo carácter de control y las transferencias de control para el control codifican procesador (0919H) para registrar la mesa de código ESC en 0953H.

Si ESCCNT contiene 01H luego el carácter es el solo parámetro del ESC, "x" secuencia. Si el parámetro es "4" (34H) luego CSTYLE está listo para 00H resultando en un cursor del bloque. Si el parámetro es "5" (35H) luego CSRSW es determinado para 00H haciendo el cursor normalmente deshabilitado.

Si ESCCNT contiene 02H luego el carácter es el solo parámetro en el ESC, "y" secuencia. Si el parámetro es "4" (34H) luego CSTYLE está listo para 01H resultando en un cursor del subrayado. Si el parámetro es "5" (35H) luego CSRSW es determinado para 01H haciendo el cursor normalmente posibilitado.

Si ESCCNT contiene 04H luego el carácter es el primer parámetro del ESC, "Y" secuencia y es la coordenada de la fila.

El parámetro tiene 1FH sustraído y es colocado en CSRY, ESCCNT es luego decremented para 03H.

Si ESCCNT contiene 03H luego el carácter es el segundo parámetro del ESC, "Y" secuencia y es la coordenada de la columna.

El parámetro tiene 1FH sustraído y es colocado en CSRX.

La dirección ... 09DAH

Esta rutina es usada, por la rutina del estándar CHGET por ejemplo, para exteriorizar el carácter del cursor cuando es normalmente deshabilitada. Si CSRSW es poco ponga en el cero, entonces la rutina simplemente termina sin acción, de otra manera el cursor es exhibido (09E6H).

La dirección ... 09E1H

Esta rutina es usada, por la rutina del estándar CHPUT para

El ejemplo, a exteriorizar el carácter del cursor cuando es normalmente posibilitado. Si CSRSW es ponga en el cero, entonces la rutina simplemente termina sin acción. SCRMOD es al que se dio jaque y, si la pantalla estuviera de Modo Graphics o Mode Multicolor, la rutina termina sin acción. De otra manera las coordenadas del cursor son convertidas a una dirección física en el VDP Name Table y el personaje leyó de esa posición (0BD8H) e invirtió a CURSAV.

Los 's de carácter ocho el pixel de byte modela es leída del VDP Character Pattern Table en el amortiguador LINWRK (0BA5H). El patrón del pixel está entonces invertido, todos los ocho bytes si CSTYLE indica un cursor del bloque, sólo la raíz tres si CSTYLE indica un cursor del subrayado. El patrón del pixel es copiado de regreso a la colocación para el código de caracteres 255 en el VDP Character Pattern Table (0BBEH). El código de carácter 255 es luego colocado en la posición del cursor en uso en estos momentos en el VDP Name Table (0BE6H) y la rutina termina.

Este método de generar el carácter del cursor, por ahí usar código de carácter 255, puede producir efectos curiosos bajo ciertas condiciones. Estos pueden ser demostrados ajustando la declaración de BASIC FOR N = 1 TO 100: ESCRIBA EN LETRAS DE IMPRENTA a CHR\$(255);:DESPUÉS y luego presionando el cursor arriba de llave.

La dirección ... 0A27H

Esta rutina es usada, por la rutina del estándar CHGET por ejemplo, para remover el carácter del cursor cuando es normalmente deshabilitada. Si CSRSW es poco ponga en el cero, entonces la rutina simplemente termina sin acción, de otra manera el cursor está removido (0A33H).

La dirección ... 0A2EH

Esta rutina está usada, por la rutina del estándar CHPUT por ejemplo, .to remueve el carácter del cursor cuando es normalmente posibilitado. Si CSRSW es ponga en el cero, entonces la rutina simplemente termina sin acción. SCRMOD es al que se dio jaque y, si la pantalla estuviera de Modo Graphics o Mode Multicolor, la rutina termina sin acción. De otra manera las coordenadas del cursor son convertidas a una dirección física en el VDP Name Table y el carácter hizo a en CURSAV cumplir escrito que la posición (0BE6H).

La dirección ... 0A44H

Esta rutina realiza al ESC, "C" operación para el procesador de código de control de rutina del estándar CHPUT. Si la coordenada de la columna del cursor está ya en la columna de la extrema derecha, determinó por LINLEN, entonces luego la rutina termina sin acción. De otra manera la coordenada de la columna es incrementada y CSRX actualizado. .

La dirección ... 0A4CH

Esta rutina realiza la operación BS/LEAVE para el procesador de código de control de rutina del estándar CHPUT. La columna del cursor

- 46 -

4. ROM BIOS

La coordenada es decremented y CSRX actualizado. Si la coordenada de la columna se ha mudado más allá del leftmost colocación, entonces es determinada para la colocación de la extrema derecha, de LINLEN, y una operación LEVANTADA realizada.

La dirección ... 0A55H

Esta rutina realiza al ESC, "D" operación para el procesador de código de control de rutina del estándar CHPUT. Si la coordenada de la columna del cursor está ya en la colocación del leftmost, entonces luego la rutina termina sin acción. De otra manera la coordenada de la columna es decremented y CSRX actualizado.

La dirección ... 0A57H

Esta rutina realiza al ESC, "una" (ARRIBA) operación para el procesador de código de control de rutina del estándar CHPUT. Si la coordenada de la fila del cursor está ya en la colocación más alta, entonces la rutina termina sin acción. De otra manera la coordenada de la fila es decremented y CSRY actualizado.

La dirección ... 0A5BH

Esta rutina realiza la operación CORRECTA para el procesador de código de control de rutina del estándar CHPUT. La coordenada de la columna del cursor es incrementada y CSRX actualizado. Si la coordenada de la columna se ha mudado más allá de la colocación de la extrema derecha, determinada por LINLEN, es determinada para el leftmost, entonces la colocación (01H) y una operación DETENIDA realizada.

La dirección ... 0A61H

Esta rutina realiza al ESC, "B" (ABAJO) operación para el procesador de código de control de rutina del estándar CHPUT. Si la coordenada de la fila del cursor está ya en lo más bajo la colocación, determinada por CRTCNT y CNSDFG (0C32H), entonces luego la rutina termina sin acción. De otra manera la coordenada de la fila es incrementada y CSRY actualizado.

La dirección ... 0A71H

Esta rutina realiza la operación de la ETIQUETA para el procesador de código de control de rutina del estándar CHPUT. Los espacios de ASCII son salida (08DFH) hasta que CSRX sea un múltiplo de ocho y uno (las columnas BIOS 1, 9, 17, 25, 33).

La dirección ... 0A7FH

Esta rutina realiza al ESC, "H" (la CASA) operación pues el procesador de código de control de rutina del estándar CHPUT, CSRX y CSRY es simplemente se sedimentó para 1,1. El sistema de coordenadas del cursor ROM BIOS, mientras funcionalmente idéntico para tan usado por el BASIC Interpreter, numera las filas de la pantalla de 1 para 24 y las columnas de 1 para 32/40.

- 47 -

4. ROM BIOS

La dirección ... 0A81H

Esta rutina realiza la operación de RETORNO DE CARRO para el procesador de código de control de rutina del estándar CHPUT, CSRX está simplemente listo a 01H.

La dirección ... 0A85H

Esta rutina realiza al ESC, "M" función para el procesador de código de control de rutina del

estándar CHPUT. Una operación de RETORNO DE CARRO es primero realizado para ajustar la columna del cursor coordinada para la colocación del leftmost. El número de filas de la fila en uso en estos momentos para la raíz de la pantalla es luego determinado, si éste es cero la fila en uso en estos momentos es simplemente borrado (0AECH). La cuenta de la fila se usa primero para desplazar hacia arriba de la pantalla el capítulo pertinente de LINTTB, la mesa de terminación de la línea, por un byte. Se usa luego para desplazar hacia arriba de la pantalla el capítulo pertinente de la pantalla una fila a la vez. Comenzando en la fila debajo de la fila en uso en estos momentos, cada línea es copiada del VDP Name Table en el amortiguador LINWRK (0BAAH) entonces copiado de regreso a la fila una Table Name más alto (0BC3H). Finalmente la fila mínima en la pantalla es borrada (0AECH).

La dirección ... 0AB4H

Esta rutina realiza al ESC, "L" operación para el procesador de código de control de rutina del estándar CHPUT. Una operación de RETORNO DE CARRO es primero realizado para ajustar la columna del cursor coordinada para la colocación del leftmost. El número de filas de la fila en uso en estos momentos para la raíz de la pantalla es luego determinado, si éste es cero la fila en uso en estos momentos es simplemente borrado (0AECH). La cuenta de la fila se usa primero para desplazar hacia abajo de la pantalla el capítulo pertinente de LINTTB, la mesa de terminación de la línea, por un byte. Se usa luego para desplazar hacia abajo de la pantalla el capítulo pertinente de la pantalla una fila a la vez.

Comenzando en la fila penúltima de la pantalla, cada línea es copiada del VDP Name Table en el amortiguador LINWRK (0BAAH), luego copiada de regreso a la fila una Table Name más bajo (0BC3H).

Finalmente la fila en uso en estos momentos es borrada (0AECH).

La dirección ... 0AE3H

Esta rutina se usa para realizar la operación DEL para el procesador de código de control de rutina del estándar CHPUT. Una operación IZQUIERDA es primero realizado. Si esto no puede ser completado, porque el cursor está ya en la colocación de la casa, entonces luego la rutina termina sin acción. De otra manera a un espacio le es escrito para el VDP Name Table en la posición física del cursor (0BE6H).

La dirección ... 0AECH

Esta rutina realiza al ESC, "I" operación para el procesador de código de control de rutina del estándar CHPUT. La coordenada de la columna del cursor es determinada para 01H y control cae en el ESC, "K" rutina.

- 48 -

4. ROM BIOS

La dirección ... 0AEEH

Esta rutina realiza al ESC, "K" operación para el procesador de código de control de rutina del estándar CHPUT. La entrada de la fila en LINTTB, la mesa de terminación de la línea, es primer poco cero hecho para señalar que la línea lógica no está extendida (0C29H). Las coordenadas del cursor son convertidas a una dirección física (0BF2H) en el VDP Name Table y el VDP para escribe por la rutina del estándar SETWRT. A los espacios les es luego directamente para el VDP Data Port hasta la columna de la extrema derecha, está resuelto por LINLEN, es alcanzado.

La dirección ... 0B05H

Esta rutina realiza al ESC, "J" operación para el procesador de código de control de rutina del estándar CHPUT. Una operación ESC, "K" es realizada en filas sucesivas, comenzando con lo actual, hasta que la raíz de la pantalla es alcanzada.

La dirección ... 0B15H
El nombre..... ERAFNK
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, DE, EI

La rutina estándar para revolver el mostrador de la llave de función completamente. CNSDFG es primero puesto en el cero y, si el VDP estuviera de Modo Graphics o Mode Multicolor, la rutina termina sin más acción. Si el VDP está de Modo del 40x24 Text o 32x24 Text Mode la última fila en la pantalla es entonces borrado (0AECH).

La dirección ... 0B26H
El nombre..... FNKSB
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, EI

La rutina estándar para exteriorizar el despliegue de la llave de función si es posibilitado. Si CNSDFG es ponga en el cero, entonces la rutina termina sin acción, el control diferente cae en la rutina del estándar DSPFNK..

La dirección ... 0B2BH
El nombre..... DSPFNK
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, BC, DE, EI

La rutina estándar para revolver el mostrador de la llave de función adelante. CNSDFG está listo para FFH y, si el VDP estuviera de Modo Graphics o Mode Multicolor, la rutina termina sin más acción. De otra manera la coordenada de la fila del cursor es comprobada y, si el cursor estuviera en la última fila de la pantalla, un LF que el código (0AH) emitió para la rutina del estándar OUTDO para enrollar la pantalla arriba.

- 49 -

4. ROM BIOS

El par de registro HL está luego colocado a apuntar ya sea la función no desviada o desviada forma una serie en la Workspace Area dependiendo en ya sea la tecla de mayúsculas se aprieta. LINLEN tiene cuatro sustraídos, para permitir un mínimo de espacio del mismo entre campos, y está en desacuerdo a las cinco para determinar el tamaño del campo para cada cuerda. Los personajes sucesivos son luego tomados de cada uno cuerda de función, comprobadas para encabezados gráficos por la rutina del estándar CNVCHR y están posado en el amortiguador LINWRK hasta que la cuerda está agotada o la zona está llena. Cuando todos los cinco instrumentos de cuerda son completados el amortiguador LINWRK es escrito para la última fila en el VDP Name Table (0BC3H).

La dirección ... 0B9CH

Esta rutina está usada por el despliegue de la llave de función rutinas estándar relacionadas. Los

contenidos de A de registro son colocados en CNSDFG que luego SCRMOD probó y que Flag Carolina Del Norte regresó si la pantalla está de Modo Graphics o el Multicolor Mode.

La dirección ... 0BA5H

Esta rutina emula ocho bytes del VDP VRAM en el amortiguador LINWRK, el VRAM a la dirección física está suministrado en par de registro HL.

La dirección ... 0BAAH

Esta rutina copia una fila completa de personajes, con el largo determinado por LINLEN, del VDP VRAM en el amortiguador LINWRK. La fila del cursor a la coordenada está suministrada en L de registro.

La dirección ... 0BBEH

Esta rutina emula ocho bytes del amortiguador LINWRK en el VDP VRAM, el VRAM a la dirección física está suministrado en par de registro HL.

La dirección ... 0BC3H

Esta rutina copia una fila completa de personajes, con el largo determinado por LINLEN, del amortiguador LINWRK en el VDP VRAM. La fila del cursor a la coordenada está suministrada en L de registro.

La dirección ... 0BD8H

Esta rutina lee un solo byte del VDP VRAM en C de registro. La columna a la coordenada está suministrada en registro H, la fila coordinada en L de registro.

La dirección ... 0BE6H

Esta rutina convierte un par de coordenadas de la pantalla, la columna en H de registro y la fila en registro L, en una dirección física en el VDP Name Table. Esta dirección es devuelta adentro

- 50 -

4. ROM BIOS

El par de registro HL.

La coordenada de la fila es primera multiplicado por treinta y dos o cuarenta, dependiendo del modo de la pantalla, y añadido para la coordenada de la columna. Esto se agrega luego a la dirección bajo Table Name, tomado de NAMBAS, para producir una dirección inicial.

Por la anchura variable de la pantalla, tan contenido en LINLEN, un offset adicional tiene que agregarse para la dirección inicial para conservar la región activa apenas puesta en el centro dentro de la pantalla. La diferencia entre el número "verdadero" de personajes por fila, treinta y dos o cuarenta, y la anchura coetánea son disminuidas a la mitad y luego redondeadas arriba para producir el offset de la izquierda. Para una máquina UK, con una treinta y siete anchura de carácter en 40x24 Text Mode, esto resultará en dos caracteres sin uso en el lado de la izquierda y uno a la derecha. La IMPRESIÓN de declaración (41-WID) \2, dónde WID es cualquier anchura de la pantalla, exteriorizará el offset de la columna de la izquierda en 40x24 Text Mode.

Un programa completo de BASIC que emula esta rutina es dado debajo:

```

10 CPR = 40:NAM=BASE (0):WID=PEEK (y HF3AE)
20 SCRMD=PEEK (Y HFCAF):IF SCRMD = 0 LUEGO 40
30 CPR = 32:NAM=BASE (5):WID=PEEK (y HF3AF)
40 LH = (CPR 1-WID) \2
50 ADDR=NAM * CPR + (COL-1) + (ROW-1) + LH

```

Este programa es diseñado pues la FILA y el sistema de coordenadas COL usado por los ROM BIOS donde la casa sea 1,1. Línea 50 puede ser simplificada, quitando lo "- 1" factores, si BASIC Interpreter coordínese, entonces el sistema debe ser usado.

La dirección ... 0C1DH

Esta rutina calcula el discurso de entrada de una fila en LINTTB, la mesa de terminación de la línea. La fila a la coordenada está suministrada en L de registro y que la dirección devolvió en par de registro Delaware.

La dirección ... 0C29H

Esta rutina hace la entrada de una fila en el poco cero LINTTB cuándo introducida en 0C29H y el cero estando entrado en 0C2AH. La fila a la coordenada está suministrada en L de registro.

La dirección ... 0C32H

Esta rutina devuelve el número de filas en la pantalla en A de registro. Normalmente devolverá veinticuatro si el despliegue de la llave de función es deshabilitado y veintitres si es posibilitado. Noto que el tamaño de la pantalla es determinado por CRTCNT y puede ser modificado con una declaración de BASIC, PINCHAZO y HF3B1H, 14:PANTALLA 0 por ejemplo.

- 51 -

4. ROM BIOS

```

La dirección ... 0C3CH
El nombre..... KEYINT
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. El

```

La rutina estándar para procesar a Z80 interrumpe, estos son generados por el VDP una vez cada 20 ms en un UK labran a máquina. El VDP Status Register es primera lectura y bit 7 a cuadros para asegurar que éste es un marco tasa interrumpe, en caso de que no la rutina termine sin acción. Los contenidos del Status Register son ahorrados en STATFL y mordieron 5 comprobado para la coincidencia de bloque de objetos movibles. Si el Coincidence Flag es activo luego la entrada pertinente en TRPTBL está actualizada (0EF1H).

INTCNT, el "intervalo" mueble mostrador, es luego decremented. Si esto ha cumplido ponga en el cero, entonces la entrada pertinente en TRPTBL es actualizada (0EF1H) y el mueble mostrador vuelto a arrancar con los contenidos de INTVAL.

El MOMENTITO, el "tiempo" el mueble mostrador, es entonces incrementado. Este mueble mostrador justamente se enrolla aproximadamente cero cuando se derrama.

MUSICF es examinado para determinar si cualquier de lo tres música hace cola generado por el "juego" declaración es activo. Para cada cola activa la rutina del dequeueing (113BH) se siente llamada a ir a traer el siguiente paquete de música y escribir ella para el PSG.

SCNCNT es luego decremented para determinar si una palanca de control y un teclado escudriñan es requerido, en caso de que no lo interrumpe manipulador termina sin más acción. Así de en dirección opuesta está acostumbrado a aumentar rendimiento específico y minimizar problemas del keybounce asegurando eso uno escudriñan es sólo efectuado cada tres interrumpen. Asumiendo uno escudriñe es conector requerido de la palanca de control 1 es seleccionado y los añicos dos Trigger leyeron (120CH), seguidos por los añicos dos Trigger de conector de la palanca de control 2 (120CH) y la llave de ESPACIO de fila 8 del teclado (1226H). Estas cinco entradas, que es todo relacionado con el "STRIG" declaración, está combinado en un solo byte donde 0 = Presionado, 1 Not presionado:

```

  7 6 5 4 3 2 1 0
+
|La Alegría 2 | La Alegría 2 | La Alegría 1 | Alegría 1 | 0 | 0 | 0 | El Espacio
|Trg.B Trg.A Trg.B Trg.A
+

```

Figura 35: "STRIG" Entradas

Esta lectura es comparada con la previa, está sujeto en TRGFLG, para producir un byte activo de transición y TRGFLG está actualizado con la lectura nueva. El byte activo de transición es normalmente cero pero contiene uno 1 en cada colocación donde una transición de no en apuros para en apuros ha ocurrido. Este byte activo de transición

- 52 -

4. ROM BIOS

Es intercambiado fuera poco a poco y la entrada pertinente en TRPTBL actualizado (0EF1H) para cada dispositivo activo.

Uno completo escudriña de la matriz del teclado es entonces realizado para identificar depresiones cruciales y nuevas, más encontradas es traducido a códigos cruciales y está posado en KEYBUF (0D12H). Se encuentra que si KEYBUF está vacío al final de este proceso REPCNT es decremented para ver si lo autorepito que el retraso ha caducado, en caso de que no la rutina termina. Si el período de retraso ha caducado, entonces REPCNT es vuelto a arrancar con el ayuno repite valor (60 ms), el mapa del teclado OLDKEY es en el que se repuso las iniciales y el teclado escudriñado otra vez (0D4EH). Cualquier claves que se aprietan continuamente se notarán como transiciones nuevas durante esto escudriñe. Noto que las llaves sólo autorepetirán mientras un programa de aplicación entretiene a KEYBUF vacío leyendo a personajes. Lo interrumpe a que el manipulador entonces termina.

La dirección ... 0D12H

Esta rutina funciona uno completo escudriñe de todas las once filas de la matriz del teclado para lo interrumpe manipulador. Cada uno de las once filas son leídas adentro por el PPI y están posado en ascender ' haga el pedido en NEWKEY. ENSTOP es entonces a cuadros para ver si principios calientes es facultado. Si sus contenidos son poco cero y el CÓDIGO de llaves, entonces GRAPH, CTRL y CAMBIO son transferencias de control en apuros para el BASIC Interpreter (409BH) por la rutina del estándar CALBAS. Esta facilidad es útil como aun un programa de código de la máquina puede ser terminado tan largo como lo interrumpe manipulador corre.

Los contenidos de NEWKEY son con los que se comparó los previos escudriñe contenido en OLDKEY. Si cualquier cambio en todo ha ocurrido, entonces REPCNT está cargado con la inicial autorepite retraso (780 ms). Cada uno hace avanzar con el remo 1, la lectura de NEWKEY es entonces comparada con la previa, está sujeto en OLDKEY, para producir un byte activo de transición y OLDKEY está actualizado con la lectura nueva. El byte activo de transición es normalmente cero pero contiene uno 1 en cada colocación donde una transición de no en apuros para en apuros ha ocurrido. Si la fila contiene cualquier transiciones éstas son descifradas y colocadas en KEYBUF como códigos cruciales (0D89H). Cuando todas las once filas han sido completadas la rutina inspecciona si hay cualquier personajes en KEYBUF, sustrayendo a GETPNT de PUTPNT, y termina.

La dirección ... 0D6AH
 El nombre..... CHSNS
 La entrada..... Ninguno
 La salida..... La bandera NZ si los personajes en KEYBUF
 Modifica.. AF, EI

La rutina estándar a comprobar si cualquier personajes del teclado está lista. Si la pantalla está de Modo Graphics o el Multicolor Mode luego GETPNT es sustraído de PUTPNT (0D62H) y la rutina termina. Si la pantalla está de Modo del 40x24 Text o 32x24 Text Mode la condición de la tecla de mayúsculas es también examinado y la función que la llave el despliegue actualizó, entonces por la rutina del estándar DSPFNK, si se ha alterado.

- 53 -

4. ROM BIOS

La dirección ... 0D89H

Esta rutina se reduce a cada bit activo en un byte de transición de la fila del teclado un código crucial. Un bit se convierte primero en un número crucial determinado por su colocación en la matriz del teclado:

```

+
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Row 0
| (07H) | (06H) | (05H) | (04H) | (03H) | (02H) | (01H) | (00H)
+
| ; | ] | [ | \ | = | - | 9 | 8 | Row 1
| (0FH) | (0EH) | (0DH) | (0CH) | (0BH) | (0AH) | (09H) | (08H)
+
| B | A . | , | ' | ` | ' | ' | Fila 2
| (17H) | (16H) | (15H) | (14H) | (13H) | (12H) | (11H) | (10H)
+
| La J | Yo | La Fila H | G | F | E | D | C 3
| (1FH) | (1EH) | (1DH) | (1CH) | (1BH) | (1AH) | (19H) | (18H)
+
| ; | La Fila De Interrogación Q | P | O | N | M | L | K 4
| (27H) | (26H) | (25H) | (24H) | (23H) | (22H) | (21H) | (20H)
+
| Z | Y | X | W | V | U | T | S | Haga Avanzar Con El Remo 5
| (2FH) | (2EH) | (2DH) | (2CH) | (2BH) | (2AH) | (29H) | (28H)

```

```

+
| F3 F2 F1 CODE|CAP|GRAPH|CTRL|SHIFT | Row 6
|(37H) |(36H) |(35H) |(34H) |(33H) |(32H) |(31H) |(30H)
+
| CR|SEL|BS|STOP|TAB|ESC | F5 | F4 | Row 7
|(3FH) |(3EH) |(3DH) |(3CH) |(3BH) |(3AH) |(39H) |(38H)
+
|RIGHT|DOWN|UP|LEAVE|DEL|IN|HOME|SPACE | Haga Avanzar Con El Remo 8
|(47H) |(46H) |(45H) |(44H) |(43H) |(42H) |(41H) |(40H)
+
| 4 | 3 | 2 | 1 | 0 | | | Row 9
|(4FH) |(4EH) |(4DH) |(4CH) |(4BH) |(4AH) |(49H) |(48H)
+
| . | , | ; | - | 9 | 8 | 7 | 6 | 5 | Row 10
|(57H) |(56H) |(55H) |(54H) |(53H) |(52H) |(51H) |(50H)
+
7 6 5 4 3 2 1 0 Column

```

Figura 36: Teclee Números

- 54 -

4. ROM BIOS

El número crucial se convierte luego en un código crucial y colocado en KEYBUF (1021H). Cuando todos los ocho añicos posibles han sido procesados la rutina termina.

La dirección ... 0DA5H

Esta mesa contiene los códigos cruciales de 00H crucial de números para 2FH para combinaciones diversas de las llaves de control. Una entrada de cero en la mesa quiere decir que ningún código crucial se producirá cuando esa llave se aprieta:

```

37H 36H 35H 34H 33H 32H 31H 30H Row 0
3BH 5DH 5BH 5CH 3DH 2DH 39H 38H Row 1
62H 61H 9CH 2FH 2EH 2CH 60H 27H Row NORMAL 2
6AH 69H 68H 67H 66H 65H 64H 63H Row 3
72H 71H 70H 6FH 6EH 6DH 6CH 6BH Row 4
7AH 79H 78H 77H 76H 75H 74H 73H Row 5

26H 5EH 25H 24H 23H 40H 21H 29H Row 0
3AH 7DH 7BH 7CH 2BH 5FH 28H 2AH Row 1
CAMBIO 42H 41H 9CH 3FH 3EH 3CH 7EH 22H Row 2
4AH 49H 48H 47H 46H 45H 44H 43H Row 3
52H 51H 50H 4FH 4EH 4DH 4CH 4BH Row 4
5AH 59H 58H 57H 56H 55H 54H 53H Row 5

```

FBH F4H BDH EFH BAH ABH ACH 09H Row 0
06H 0DH 01H 1EH F1H 17H 07H ECH Row 1
GRÁFICA 11H C4H 9CH 1DH F2H F3H BBH 05H Row 2
C6H DCH 13H 15H 14H CDH C7H BCH Row 3
18H CCH DBH C2H 1BH 0BH C8H DDH Row 4
0FH 19H 1CH CFH 1AH C0H 12H D2H Row 5

00H F5H 00H 00H FCH FDH 00H 0AH Row 0
04H 0EH 02H 16H F0H 1FH 08H 00H Row 1
CAMBIO 00H FEH 9CH F6H AFH AEH F7H 03H Row 2
GRÁFICA CAH DFH D6H 10H D4H CEH C1H FAH Row 3
A9H CBH D7H C3H D3H 0CH C9H DEH Row 4
F8H AAH F9H D0H D5H C5H 00H D1H Row 5

E1H E0H 98H 9BH BFH D9H 9FH EBH Row 0
B7H DAH EDH 9CH E9H EEH 87H E7H Row 1
CÓDIGO 97H 84H 9CH A7H A6H 86H E5H B9H Row 2
91H A1H B1H 81H 94H 8CH 8BH 8DH Row 3
93H 83H A3H A2H A4H E6H B5H B3H Row 4
85H A0H 8AH 88H 95H 82H 96H 89H Row 5

00H 9DH 9CH BEH 9EH ADH D8H Row 0
B6H EAH E8H 00H 00H 00H 80H E2H Row 1
CAMBIO 00H 8EH 9CH A8H 00H 8FH E4H B8H Row 2
CÓDIGO 92H 00H B0H 9AH 99H 00H 00H 00H Row 3
00H E3H 00H A5H 00H B4H B2H Row 4
00H 00H 00H 00H 90H 00H 00H Row 5

7 6 5 4 3 2 1 0 Column

- 55 -

4. ROM BIOS

La dirección ... 0EC5H

Controle transferencias para esta rutina, de 0FC3H, para completar desciframiento de las cinco llaves de función. La entrada pertinente en FNKFLG es primero a cuadros para determinar si la clave es con la que se asoció uno "en KEY GOSUB" declaración. Si es así, y con tal que CURLIN muestra el BASIC Interpreter estar de modo de programa, la entrada pertinente en TRPTBL es actualizada (0EF1H) y la rutina termina. Si se amarró a la llave uno "en KEY GOSUB" declaración, o si el Interpreter está directo modo, entonces la cadena de caracteres asociada con la llave de función es devuelta en lugar de eso. El número crucial está mu

ltiplicado por dieciséis, como cada cuerda es dieciséis personajes largos, y añadidos para la dirección de puesta en marcha de los instrumentos de cuerda de la llave de función en el Workspace Area. Los personajes secuenciales son luego tomados de la cuerda y están posado en KEYBUF (0F55H) hasta que el exterminador de byte de cero sea alcanzado.

La dirección ... 0EF1H

Esta rutina se usa para actualizar la entrada de un dispositivo en TRPTBL cuando ha producido un programa de BASIC interrumpa. En las proposiciones del par de registro de entrada HL para el byte de estado del dispositivo en la mesa. El bit 0 del byte de estado es primer a lo que se dio jaque, si el dispositivo no está encendido luego la rutina termina sin acción. El bit 2, la

bandera de acontecimiento, es entonces a cuadros. Si esto está ya listo luego la rutina termina, de otra manera es determinada para señalar que un acontecimiento ha ocurrido. El bit 1, el "alto" bandera, es entonces a cuadros. Si el dispositivo está detenido luego la rutina termina sin más acción. ONGSBF diferente es incrementado para hacer señales a para el Interpreter Runloop que el acontecimiento ahora debería ser tramitado.

La dirección ... 0F06H

Este capítulo del decodificador crucial procesa la llave de la CASA sólo. La condición de la tecla de mayúsculas es determinada por fila 6 de NEWKEY y el código crucial para A CASA (0BH) o CLS (0CH) colocado en KEYBUF (0F55H) consecuentemente.

La dirección ... 0F10H

Este capítulo de los procesos del decodificador del teclado afina 30H de números para 57H con la excepción de la GORRA, F1 para las llaves F5, de ALTO y de la CASA. El número crucial se usa simplemente para buscar el código crucial en la mesa en 1033H y éste son luego colocados en KEYBUF (0F55H).

La dirección ... 0F1FH

Este capítulo del decodificador del teclado procesa la llave MUERTA encontrada en máquinas europeas MSX. En UK labra a máquina la llave en fila 2, columna 5 siempre genera el código de la llave de libra (9CH) exteriorizado en la mesa en 0DA5H. En máquinas del europeo esta mesa tendrá el código de la llave FFH en las mismas posiciones. Este código crucial sólo sirve de una bandera para señalar que la siguiente llave en apuros, si eso

- 56 -

4. ROM BIOS

Es un vocal, debería ser modificado para producir un carácter acentuado de gráficos.

La condición del CAMBIO y el CÓDIGO las llaves es determinada por fila 6 de NEWKEY y una de lo siguiente colocado en KANAST: El 01H = DEAD, 02H DEAD+SHIFT, 03H DEAD+CODE, 04H DEAD+SHIFT+CODE.

La dirección ... 0F36H

Este capítulo del decodificador del teclado procesa la llave de la GORRA. La condición coetánea de CAPST es invertida y el control cae en la rutina del estándar CHGCAP.

La dirección ... 0F3DH

El nombre..... CHGCAP

La entrada..... Uno = En / El Anticuado Switch

La salida..... Ninguno

Modifica.. AF

La rutina estándar para dar vuelta al Caps Lock LLEVÓ LA DELANTERA o completamente como determinó por los contenidos de A de registro: El 00H = On, NZ = Off. El LED está modificado usando la facilidad mordida del set /reanudación del PPI Mode Port. Como CAPST no se varía esta rutina no afecta a los personajes producidos por el teclado.

La dirección ... 0F46H

Este capítulo del decodificador del teclado procesa la llave de ALTO. La condición de la llave CTRL es determinada por fila 6 de NEWKEY y el código crucial para el ALTO (04H) o CTRL/STOP (03H) producido según el caso. Si el código CTRL/STOP se produce es emulado para INTFLG, entonces para el uso por la rutina del estándar ISCNTC, y luego colocado en KEYBUF (0F55H). Si el código de ALTO se produce es también copiado para INTFLG pero no es colocado en KEYBUF, entonces en lugar de eso sólo un chasquido es generado (0F64H). Esto quiere decir que un programa de aplicación no puede leer el código de la llave de ALTO por las rutinas del estándar ROM BIOS.

La dirección ... 0F55H

Este capítulo del decodificador del teclado coloca un código crucial en KEYBUF y genera un chasquido audible. La dirección correcta en el amortiguador del teclado es primero tomada de PUTPNT y el código acomodado allí. El discurso es entonces incrementada (105BH). Si se ha enrollado alrededor y ha alcanzado a GETPNT, entonces luego la rutina termina sin más acción como el amortiguador del teclado esté lleno. PUTPNT diferente está actualizado con la dirección nueva.

CLIKSW y CLIKFL son luego ambos dado jaque para determinar si un chasquido es precisado. CLIKSW es un general posibilite / lisa interruptor mientras CLIKFL está acostumbrado a impedir chasquidos múltiples cuando las llaves de función se aprietan. Asumir un chasquido es precisado que la salida Crucial Click es determinada por el PPI Mode Port y, después de un retraso de 50 µs, el control cae en la rutina del estándar CHGSND.

- 57 -

4. ROM BIOS

La dirección ... 0F7AH
El nombre..... CHGSND
La entrada..... Uno = En / El Anticuado Switch
La salida..... Ninguno
Modifica.. AF

La rutina estándar a ajustar o poner a cero la salida Crucial Click por el PPI Mode Port: El 00H = Reanudó, NZ = Set. Esta salida audia es CORRIENTE ALTERNA enganchada así es que las polaridades absolutas no deberían ser tomadas también seriamente.

La dirección ... 0F83H

Este capítulo de los procesos del decodificador del teclado afina 00H de números para 2FH. La condición del CAMBIO, GRÁFICA y CÓDIGO las llaves es determinada por fila 6 de NEWKEY y combinada con el número crucial formar una dirección de búsqueda en la mesa en 0DA5H. El código crucial es luego tomado de la mesa. Si es ponga en el cero, entonces la rutina termina sin más acción, si es control FFH transfiere para el procesador (0F1FH) TOTALMENTE crucial. Si el código está en el 40H de rango para 5FH o 60H para 7FH y la llave CTRL es presionado luego el código correspondiente de control es colocado en KEYBUF (0F55H). Si el código está en el 01H de rango para 1FH luego un código gráfico (01H) del encabezado es primero colocado en KEYBUF (0F55H) seguido por el código con 40H añadido. Si el código está en el 61H de rango para 7BH y CAPST señala que el cerrojo de gorras está encendido luego es convertido a caja alta sustrayendo 20H. Dado que KANAST contiene cero, como siempre hará en UK labra a máquina, luego el código crucial es colocado en KEYBUF (0F55H) y la rutina termina. En MSX europeo las máquinas, con una llave MUERTA en lugar de una llave de libra, luego la llave codifica correspondiente a los vocales uno, e, yo, o, u puede estar adicionalmente modificada en códigos de gráficos.

La dirección ... 0FC3H

Este capítulo del decodificador del teclado procesa las cinco llaves de función. La condición de la tecla de mayúsculas es examinada por fila 6 de NEWKEY y cinco se sumaron al número crucial si se aprieta.

El control luego se traslada para 0EC5H para completar procesamiento.

La dirección ... 1021H

Esta rutina registra que la mesa en 1B97H para determinar cuál grupo de llaves el número crucial suministró en C de registro pertenece para.

La dirección asociada es luego tomada de la mesa y control transferida para ese capítulo del decodificador del teclado. Noto que la mesa misma es de hecho moteada en el centro de la rutina del estándar OUTDO como resultado de las modificaciones hechas al ROM japonés.

La dirección ... 1033H

Esta mesa contiene los códigos cruciales de 30H crucial de números para 57H aparte de la GORRA especial de llaves, F1 para F5, el ALTO y la CASA. uno

- 58 -

4. ROM BIOS

La entrada de cero en la mesa quiere decir que ningún código crucial se producirá cuando esa llave se aprieta:

```
00H 00H 00H 00H 00H 00H 00H Row 6
0DH 18H 08H 00H 09H 1BH 00H 00H Row 7
1CH 1FH 1EH 1DH 7FH 12H 0CH 20H Row 8
34H 33H 32H 31H 30H 00H 00H 00H Row 9
2EH 2CH 2DH 39H 38H 37H 36H 35H Row 10
```

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

La dirección ... 105BH

Esta rutina simplemente pone en el cero a KANAST y luego transfiere control para 10C2H.

La dirección ... 1061H

Esta mesa contiene los caracteres de gráficos que reemplazan los vocales una, e, yo, o, u en máquinas europeas.

La dirección ... 10C2H

Esta rutina incrementa el consejo del amortiguador del teclado, ya sea PUTPNT o GETPNT, abastecida en par de registro HL. Si el puntero luego excede el fin del teclado amortiguador, entonces es envuelto de regreso al comienzo.

La dirección ... 10CBH

El nombre..... CHGET

La entrada..... Ninguno

La salida..... Un = Character de teclado
Modifica.. AF, EI

La rutina estándar para ir a traer a un personaje del amortiguador del teclado. El amortiguador es primero a cuadros para ver si ya contiene un carácter (0D6AH). En caso de que no el cursor se enciende (09DAH), el amortiguador comprobado repetidamente hasta que un carácter aparece (0D6AH) y luego el cursor se desactivó (0A27H). El carácter es tomado del amortiguador usando a GETPNT que es entonces incrementado (10C2H).

La dirección ... 10F9H
El nombre..... CKCNTC
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, EI

La rutina estándar a comprobar si el CTRL-STOP o DETENER llaves se han apretado. Es usado por el BASIC Interpreter dentro de declaraciones intensivas en procesador, algo semejante como "la ESPERA" y "el CÍRCULO", a comprobar para la terminación de programa. El par de registro HL es primero control puestos en el cero y luego transferidos para la rutina del estándar ISCNTC.

Cuando el Interpreter corre el par de registro HL normalmente contiene la dirección del carácter coetáneo en el BASIC

- 59 -

4. ROM BIOS

El texto de programa. Si ISCNTC es CTRL-STOP terminado esta dirección será colocado en OLDTXT por el "alto" manipulador de declaración (63E6H) para el uso por un posterior "CONT" declaración. Poner en el cero par de registro HL de antemano da señas de para el "CONT" manipulador que la terminación ocurrió dentro de una declaración y eso surgirá uno "no puede mantener" error si la continuación es intentada.

La dirección ... 1102H
El nombre..... WRTPSG
La entrada..... Un byte = Register de número, E = Data
La salida..... Ninguno
Modifica.. EI

La rutina estándar para escribir un byte de datos para cualquier de los registros dieciséis PSG. El número de selección de a registro le es escrito para el PSG Address Port y el byte de datos escrito para el PSG Data Write Port.

La dirección ... 110EH
El nombre..... RDPSG
La entrada..... Un número = Register
La salida..... Un byte = Data leyó de PSG
Modifica.. uno

La rutina estándar para leer un byte de datos de cualquier de los registros dieciséis PSG. El número de selección de a registro le es escrito para el PSG Address Port y el byte de datos leído del PSG Data Leyó a Port.

La dirección ... 1113H
El nombre..... EMITA UN PIP
La entrada..... Ninguno

La salida..... Ninguno
Modifica.. AF, BC, E, EI

La rutina estándar para producir un pip por el PSG. La A del canal está colocada para producir un tono de 1316Hz luego posibilitado con una amplitud de siete. Después de un retraso de 40 ms las transferencias de control para la rutina del estándar GICINI para reinicializar el PSG.

La dirección ... 113BH

Esta rutina es usada por ahí lo interrumpe manipulador para reparar una cola de música. Como hay tres de estos, cada alimentación un canal PSG, la cola a ser reparados es especificada suministrando su número en A de registro: 0 = VOICAQ, 1 = VOICBQ Y 2 = VOICCCQ.

Cada cuerda en un "juego" declaración es traducida a una serie de paquetes de datos por el BASIC Interpreter. Estos son colocados en la cola apropiada seguidos por un fin de byte de datos (FFH). La tarea de dequeuing los paquetes, descifrándolos y colocar al PSG queda para lo interrumpe manipulador. El Interpreter está así en libertad para proceder inmediatamente hacia la siguiente declaración sin tener que esperar notas para terminar.

- 60 -

4. ROM BIOS

Los primeros dos bytes de cualquier paquete especifican su cuenta de byte y duración. Los tres bits más significativos del primer byte especifican el número de bytes siguiendo el encabezado en el paquete. El resto del encabezado especifica la duración de acontecimiento en 20 unidades ms. Esta cuenta de duración determina cuánto tiempo será antes de que el siguiente paquete sea leído de la cola.

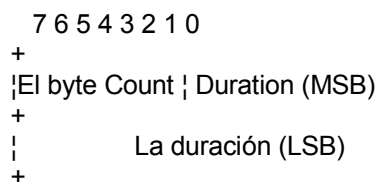
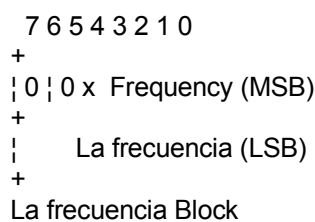


Figura 37: El paquete Header

El encabezado del paquete puede ser seguido por el cero o más bloques, en cualquier orden, conteniendo información de frecuencia o de amplitud:



```

7 6 5 4 3 2 1 0
+
| X | 1 x x x x
+
| El sobre Frequency (MSB)
+
| El sobre Frequency (LSB)
+
El sobre Block

```

```

7 6 5 4 3 2 1 0
+
| 1 x Mode AmplitudeShape
+
La amplitud Block

```

Figura 38: El Bloque Del Paquete Escribe A Máquina

La primera parte de rutina halla el mueble mostrador de duración en uso en estos momentos en el amortiguador pertinente (VCBA, VCBB o VCBC) de voz por el GETVCP

- 61 -

4. ROM BIOS

Los decrementos y rutina estándar eso. Si el mueble mostrador ha cumplido cero luego el siguiente paquete debe ser leído de la cola, de otra manera la rutina termina.

El número de la cola es colocado en QUEUEN y un byte leído de la cola (11E2H). Esto es entonces a cuadros para ver si es el fin de marca de datos (FFH), si así es que la cola termina (11B0H).

De otra manera la cuenta de byte es colocada en C de registro y la duración MSB en el amortiguador pertinente de voz. El segundo byte es lectura (11E2H) y la duración que LSB colocó en el amortiguador pertinente de voz. La cuenta de byte es entonces examinada, si no hay bytes para seguir el encabezado del paquete que la rutina termina. Los bytes sucesivos diferentes son leída de la cola, y la acción apropiada tomado, hasta que la cuenta de byte está agotada.

Si un bloque de frecuencia es encontrado luego a un segundo byte le es leído y ambos bytes para Registros PSG 0 y 1, 2 y 3 o 4 y 5 a merced del número de la cola.

Si un bloque de amplitud es encontrado, entonces el Amplitude y ańicos a Mode les es para PSG Registers 8, 9 o 10 a merced del número de la cola. Si el Mode mordido tiene 1 años de edad, entonces hacer una selección moduló en vez de la amplitud fija, luego al byte le es también para PSG Register 13 para colocar la forma del sobre.

Si un bloque del sobre es encontrado, o si mordió 6 de un bloque de amplitud son determinados, entonces luego unos adicionalmente dos bytes son leídos de la cola y escrito para Registros PSG 11 y 12.

La dirección ... 11B0H

Esta rutina es usada cuando un fin de marca de datos (FFH) es encontrado en una de las tres colas de música. Un valor de amplitud de a cero le es para PSG Register 8 9 o 10, dependiendo del número de la cola, para cerrar el canal. El bit del canal en MUSICF es entonces vuelto a arrancar y el control cae en la rutina del estándar STRTMS.

La dirección ... 11C4H
 El nombre..... STRTMS
 La entrada..... Ninguno
 La salida..... Ninguno
 Modifica.. AF, HL

La rutina estándar usada por el "juego" manipulador de declaración para iniciar dequeueing de música por ahí lo interrumpe manipulador. MUSICF es primero examinado, si cualquier canales ya ejecutan la rutina termina sin acción. PLYCNT es luego decremented, si hay ningún más "juego" forma una serie al que se hizo cola arriba de la rutina termina. De otra manera los tres muebles mostradores de duración, en VCBA, VCBB y VCBC, estén listos para 0001H, a fin de que el primer paquete del grupo nuevo será sacado de la cola lo siguiente interrumpa, y MUSICF está listo para 07H para habilitar todos los tres canales.

- 62 -

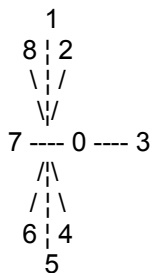
4. ROM BIOS

La dirección ... 11E2H

Este registro de cargas de rutina UNO con el número actual de la cola, de QUEUEN, y luego lecturas un byte de eso hace cola (14ADH).

La dirección ... 11EEH
 El nombre..... GTSTCK
 La entrada..... Un = Joystick Idaho (0, 1 o 2)
 La salida..... Un código de colocación = Joystick
 Modifica.. AF, B, DE, HL, EI

La rutina estándar para leer la colocación de una palanca de control o las cuatro llaves del cursor. Si el Idaho abastecido es ponga en el cero, entonces la condición de las llaves del cursor es leída por PPI Port B (1226H) y se convierte para un código de colocación usando la mesa de búsqueda en 1243H. El conector diferente de la palanca de control 1 o 2 es leído (120CH) y los cuatro añicos de dirección mutaron para un código de colocación usando la mesa de búsqueda en 1233H. Los códigos devueltos de colocación son:



La dirección ... 120CH

Esta rutina lee el conector de la palanca de control especificado por los contenidos de A de registro: 0 Connector 1, 1 = Connector 2. La corriente contenida de PSG Register 15 es leída en luego espaldas con el bit Selecto Joystick apropiadamente se sedimenta. PSG Register 14 son entonces leídos en A de registro (110CH) y la rutina termina.

La dirección ... 1226H

Esta fila de lecturas de rutina 8 de la matriz del teclado. Los contenidos coetáneos de C PPI Port son leídos en luego espaldas con el cuatro set de añicos Keyboard Row Select para fila 8. Las entradas de la columna son entonces leídas en A de registro de B PPI Port.

La dirección ... 1253H

El nombre..... GTTRIG

La entrada..... Un = Trigger Idaho (0, 1, 2, 3 o 4)

La salida..... Un código = Status

Modifica.. AF, BC, EI

La rutina estándar para comprobar el gatillo de la palanca de control o espaciar estado crucial. Si el Idaho abastecido es fila de cero 8 de la matriz del teclado es leída (1226H) y convertida para un estado el código. De otra manera

- 63 -

4. ROM BIOS

El conector de la palanca de control 1 o 2 es leído (120CH) y convertido para un código de estado. Los IDs de selección son:

0 = ESPACIAN LLAVE

1 = ALEGRÍA 1, GATILLO UNO

2 = SE REGOCIJAN 2, SE ACCIONAN UNO

3 = SE REGOCIJAN 1, TRIGGER B

4 = SE REGOCIJAN 2, TRIGGER B

El valor devuelto es FFH si el gatillo pertinente se aprieta y cero otrora.

La dirección ... 1273H

El nombre..... GTPDL

La entrada..... Un = Paddle Idaho (1 para 12)

La salida..... Un valor = Paddle (0 para 255)

Modifica.. AF, BC, DE, EI

La rutina estándar para leer el valor de cualquier remo adjunto para un conector de la palanca de control. Cada uno del seis rayado de entrada (cuatro la dirección y dos gatillos) por conector puede soportar un remo tan doce es posible enteramente. Los remos pegaron para conector de la palanca de control 1 tienen identificadores de entrada 1, 3, 5, 7, 9 y 11.

Esos pegaron para conector de la palanca de control 2 tienen identificadores de entrada 2, 4, 6, 8, 10 y 12. Cada remo es básicamente un generador de pulso de un disparos, el largo del pulso controlándose por un reostato variable. Un pulso de principio es expedido para el conector especificado de la palanca de control por PSG Register 15. Una cuenta es entonces mantenida cuántas veces PSG Register 14 tiene que ser leída hasta la entrada pertinente cronometra fuera. Cada incremento de la unidad representa un período aproximado de 12 μ s en una máquina MSX con una condición de espera.

La dirección ... 12ACH

El nombre..... GTPAD

La entrada..... Un código = Function (0 para 7)
La salida..... Un = Status o valor
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para ganar acceso a un touchpad adjunto para cualquier de los conectores de la palanca de control. Los códigos disponibles de funciones para conector de la palanca de control 1 son:

0 = Devuelven A Activity Status
"X" 1 = Return coordenada
2 = Devuelven a "Y" coordenada
3 = Devuelven A Switch Status

Los códigos de función 4 para 7 tienen el efecto del mismo con relación a conector de la palanca de control 2. La función Activity Status devuelve a FFH si el Touchpad está siendo emocionado y ponga en el cero de otra manera. La función Switch Status devuelve a FFH si el interruptor está siendo presionado y ponga en el cero de otra manera. Las dos funciones coordinadas de petición devuelven las coordenadas de la última posición emocionada. Estas coordenadas son

- 64 -

4. ROM BIOS

Realmente almacenado en el variables Workspace Area PADX y PADY cuando una llamada con código de función 0 o 4 detecten actividad. Noto que estas variables son compartidas por ambos conectores de la palanca de control.

La dirección ... 1384H
El nombre..... STMOTR
La entrada..... Un ON = Motor / FUERA DEL código
La salida..... Ninguno
Modifica.. AF

La rutina estándar para revolver el relevador del motor del casete adelante o completamente por C PPI Port: El 00H = Off, 01H On, condición de corriente FFH = Reverse.

La dirección ... 1398H
El nombre..... NMI
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. Ninguno

La rutina estándar para procesar a un Z80 Non Maskable Interrupt, simplemente devuelve en una máquina estándar MSX.

La dirección ... 139DH
El nombre..... INIFNK
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. BC, DE, HL

La rutina estándar para poner las iniciales en el de diez funciones de crucial coloca en serie para sus valores que se impulsan arriba. Lo ciento sesenta bytes de datos comenzando en 13A9H es emulado para el amortiguador FNKSTR en el Workspace Area.

La dirección ... 13A9H

Este área contiene los instrumentos de cuerda que se energizan arriba para las diez llaves de función. Cada cuerda es dieciséis personajes bastante, colocaciones sin uso contenga ponen en el cero:

F1 Para F5 F6 Para F10
El color de color 15,4,4 el RETORNO DE CARRO
El automóvil cload "
El RETORNO DE CARRO goto cont
La lista de la lista. CR ARRIBA DE ARRIBA
El RETORNO DE CARRO andado de Lingüísticas Computacionales de carrera de
RETORNO DE CARRO

La dirección ... 1449H
El nombre..... RDVDP
La entrada..... Ninguno
La salida..... Un = VDP Status Register contenta
Modifica.. uno

La rutina estándar para introducir en la computadora los contenidos del VDP Status Register por ahí leer al Command Port. Note esa lectura el VDP

- 65 -

4. ROM BIOS

El estado Register despejará las banderas asociadas y puede hacer mella lo interrumpe al manipulador.

La dirección ... 144CH
El nombre..... RSLREG
La entrada..... Ninguno
La salida..... Un Slot = Primario Register contenta
Modifica.. uno

La rutina estándar para introducir en la computadora los contenidos de la ranura Primaria Register por ahí leer A PPI Port.

La dirección ... 144FH
El nombre..... WSLREG
La entrada..... Un = Value al que escribir
La salida..... Ninguno
Modifica.. Ninguno

La rutina estándar para colocar al Slot Primario Register por ahí escribir para la A PPI Port.

La dirección ... 1452H
El nombre..... SNSMAT
La entrada..... Un número de la fila = Keyboard
La salida..... Unos datos = Column de teclado reman
Modifica.. AF, C, EI

La rutina estándar para leer una fila completa de la matriz del teclado. La C PPI Port es leída

en luego espalda escrita con el número de la fila ocupando los ańicos Select Row cuatro Keyboard. La B PPI Port es entonces leída en A de registro para devolver las ocho entradas de la columna.

Las cuatro salidas misceláneas de control de C PPI Port están no afectadas por esta rutina.

La dirección ... 145FH
El nombre..... ISFLIO
La entrada..... Ninguno
La salida..... La bandera NZ si el archivo yo / la O activo
Modifica.. AF

La rutina estándar a comprobar si el BASIC Interpreter actualmente dirige su entrada o su salida por uno y o el amortiguador de la / O. Esto está resuelto examinando a PTRFIL. Es normalmente cero pero contendrá una dirección del amortiguador FCB (el Archivo Control Block) mientras las declaraciones como "impresión # 1", "entrada # 1", etc está siendo ejecutado por el Interpreter.

La dirección ... 146AH
El nombre..... DCOMPR
La entrada..... HL, DE
La salida..... La bandera Carolina Del Norte si la Z HL>DE, Flag si la C HL=DE, Flag si HL<DE
Modifica.. AF

La rutina estándar usada por el BASIC Interpreter a comprobar lo

- 66 -

4. ROM BIOS

Los valores relativos de pares de registro HL y Delaware.

La dirección ... 1470H
El nombre..... GETVCP
La entrada..... Un número = Voice (0, 1, 2)
La salida..... HL = Address en amortiguador de voz
Modifica.. AF, HL

La rutina estándar para devolver la dirección de byte 2 en el amortiguador especificado (VCBA, VCBB o VCBC) de voz.

La dirección ... 1474H
El nombre..... El GETVC2
La entrada..... El número de la L = Byte (0 para 36)
La salida..... HL = Address en amortiguador de voz
Modifica.. AF, HL

La rutina estándar para devolver la dirección de cualquier byte en el amortiguador de voz (VCBA, VCBB o VCBC) especificó por el número de voz en VOICEN.

La dirección ... 148AH
El nombre..... PHYDIO
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. Ninguno

La rutina estándar para el uso por el BASIC Disk, simplemente regresa en máquinas estándar MSX.

La dirección ... 148EH
El nombre..... FORMATEE
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. Ninguno

La rutina estándar para el uso por el BASIC Disk, simplemente regresa en máquinas estándar MSX.

La dirección ... 1492H
El nombre..... PUTQ
La entrada..... Un byte = Queue de número, E = Data
La salida..... La Z de la bandera si la cola lleno
Modifica.. AF, BC, HL

La rutina estándar en la que colocar un byte de datos una de las tres colas de música. La cola obtenga y las colocaciones puestas son primero tomadas de QUETAB (14FAH). La colocación puesta es por ahora incrementada y comparada lo obtiene colocación, si son iguales la rutina termina como la cola esté llena. De otra manera la dirección de la cola es tomada de QUETAB y la colocación puesta se sumó a ella. El byte de datos es colocado en esta posición en la cola, la colocación puesta es incrementada y la rutina termina. Noto que las colas de música son circulares, si lo llega

- 67 -

4. ROM BIOS

O los punteros puestos cumplen la última colocación en la cola alrededor de la que ellos se enrollan de regreso al principio.

La dirección ... 14ADH

Esta rutina es usada por ahí lo interrumpe manipulador para leer un byte de una de las tres colas de música. La cola al número está suministrada en A de registro, el byte de datos es devuelta en A de registro y los ingresos de rutina Señala Con Una Bandera Z si la cola está vacía. La cola obtenga y las colocaciones puestas son primero tomadas de QUETAB (14FAH). Si la bandera del putback es activa luego el byte de datos es tomado de QUEBAK y la rutina termina (14D1H), esta facilidad está sin uso en las versiones coetáneas del MSX ROM. La colocación puesta es entonces comparada lo obtiene colocación, si son iguales la rutina termina como la cola esté vacía.

De otra manera la dirección de la cola es tomada de QUETAB y lo se agrega colocación para ella. El byte de datos es leído de esta posición en la cola, lo obtiene colocación es incrementada y la rutina termina.

La dirección ... 14DAH

Esta rutina es usada por la rutina del estándar GICINI para poner las iniciales en el bloque de control de una cola en QUETAB. El bloque de control está primero ubicado en QUETAB (1504H) y lo puesto, llegue y bytes putback puestos en el cero. El byte de tamaño es determinado de B de registro y la dirección de la cola de par de registro Delaware.

La dirección ... 14EBH

El nombre..... LFTQ
La entrada..... Un número = Queue
La salida..... HL = Desbloquee espacio izquierdo en cola
Modifica.. AF, BC, HL

La rutina estándar para devolver el número de bytes libres dejó en una cola de música. La cola obtenga y las colocaciones puestas son tomadas de QUETAB (14FAH) y el espacio libre determinó sustrayendo puesto de llegue.

La dirección ... 14FAH

Esta rutina devuelve los parámetros de control de una cola de QUETAB, la cola al número está suministrada en A de registro. El bloque de control está primero ubicado en QUETAB (1504H), la colocación puesta es luego colocada en registro B, lo mete en colocación C de registro y la bandera del putback en A de registro.

La dirección ... 1504H

Esta rutina localiza el bloque de control de una cola en QUETAB. La cola al número está suministrada en A de registro y que la dirección del bloque de control devolvió en par de registro HL. El número de la cola está simplemente multiplicado por seis, ya que hay seis bytes por bloque, y añadido para la dirección de QUETAB tan sostenido en QUEUES.

- 68 -

4. ROM BIOS

La dirección ... 1510H
El nombre..... GRPPRT
La entrada..... Un = Carácter
La salida..... Ninguno
Modifica.. EI

La rutina estándar para exteriorizar un carácter en la pantalla en ya sea Graphics Mode o el Multicolor Mode, es funcionalmente equivalentes a la rutina del estándar CHPUT.

La rutina del estándar CNVCHR se usa primero para inspeccionar para un carácter gráfico, si el carácter es un código del encabezado (01H) que luego la rutina termina sin acción. Si el carácter es uno gráfico convertido luego el capítulo de desciframiento de código de control es saltado. De otra manera el carácter es comprobado para ver si es un código de control. Sólo el código de RETORNO DE CARRO (0DH) es reconocido (157EH), todos otros personajes más pequeños que 20H están ignorados.

Asumir el carácter es explayable su patrón del pixel de ocho bytes es copiado del carácter ROM determinado en el amortiguador PATWRK (0752H) y FORCLR del que se recibió bien el mensaje de radio para ATRBYT para colocar su color. Las coordenadas coetáneas de gráficos son luego tomadas de GRPACX y GRPACY y están acostumbrado a coloque la dirección física del pixel en uso en estos momentos por el SCALXY y rutinas del estándar MAPXYC.

El patrón de ocho bytes en PATWRK es procesado un byte a la vez. Al principio de cada byte la dirección física del pixel en uso en estos momentos sea obtenida por la rutina del estándar FETCHC y se salve. Los ocho añicos son entonces examinados a su vez. Si el bit es uno 1 el pixel asociado está colocado por la rutina del estándar SETC, entonces si es una 0 ninguna acción es tomada.

Después de que cada uno mordió la dirección física del pixel en uso en estos momentos es movida bien (16ACH). Cuando el byte está acabado, o el borde de la derecha de la pantalla es alcanzado, la dirección física del pixel en uso en estos momentos inicial está recuperada y pasea a lo largo de una posición por la rutina del estándar TDOWNC.

Cuando el patrón es completo, o la raíz de la pantalla ha sido alcanzada, GRPACX está actualizado. En Graphics Mode su valor es aumentado a las ocho, en el Multicolor Mode a las treinta y dos. Si GRPACX luego excede 255, el borde de la derecha de la pantalla, una operación de RETORNO DE CARRO es realizada (157EH).

La dirección ... 157EH

Esta rutina realiza la operación de RETORNO DE CARRO para la rutina del estándar GRPPRT, estas funciones de código como un RETORNO DE CARRO combinado, LF. GRPACX es puesto en el cero y ocho o treinta y dos, dependiendo del modo de la pantalla, añadido para GRPACY. Si GRPACY luego excede 191, entonces la raíz de la pantalla, está colocado para poner en el cero.

GRPACX y GRPACY pueden ser manipulados directamente por un programa de aplicación para compensar el número limitado de funciones de control disponibles.

- 69 -

4. ROM BIOS

La dirección ... 1599B
El nombre..... SCALXY
La entrada..... BC=X coordínese, coordenada DE=Y
La salida..... La bandera Carolina Del Norte si recortado
Modifica.. AF

La rutina estándar para recortar un par de gráficos se coordina si es necesario. El BASIC que Interpreter puede producir se coordina en el rango - 32768 para + 32767 si bien éste lejos exceda el tamaño real de la pantalla. Esta rutina modifica valores coordinados excesivos para acomodar dentro del rango físicamente realizable. Si la coordenada de la X es mayor que 255 que es determinada para 255, entonces si la coordenada de la Y es mayor que 191 que es determinada para 191. Si o se coordina es negativa (más gran que 7FFFH) que está colocada para poner en el cero.

Finalmente si la pantalla está de, entonces el Modo Multicolor ambas coordenadas están divididos a las cuatro según lo solicitado por la rutina del estándar MAPXYC.

La dirección ... 15D9H

Esta rutina se usa para comprobar el modo coetáneo de la pantalla, devuelve Z Flag si la pantalla está de Modo Graphics.

La dirección ... 15DFH
El nombre..... MAPXYC
La entrada..... BC=X coordínese, coordenada DE=Y
La salida..... Ninguno
Modifica.. AF, D, HL

La rutina estándar para convertir un par de coordenada de gráficos en la dirección física del pixel en uso en estos momentos. La posición en el Character Pattern Table del byte conteniendo

el pixel es colocada en CLOC. La máscara mordida identificando el pixel dentro de ese byte es colocada en CMASK. Los métodos de conversión ligeramente diferentes sirven para Graphics Mode y el Multicolor Mode, programas equivalentes en el BASIC son:

Gráfico Mode

```
10 INPUT"X, Y Coordinates ";X, Y
20 UNO = (Y\8) * 256 + (Y AND 7) + (X AND 7) HF8)
30 PRINT"ADDR = ";+ la A HEX$ (la Base (12)) ""H ";
40 RESTITUYEN 100
50 FOR N = 0 PARA (X AND 7):M$ LEÍDO: SIGUIENTE N
60 PRINT"MASK = ";M$
70 GOTO 10
100 DATOS 10000000
110 DATOS 01000000
120 DATOS 00100000
130 DATOS 00010000
140 DATOS 00001000
150 DATOS 00000100
160 DATOS 00000010
170 DATOS 00000001
```

- 70 -

4. ROM BIOS

Mode multicolor

```
10 INPUT"X, Y Coordinates ";X, Y
20 X = X\4:Y-Y\4
30 UNO = (Y\8) * 256 + (Y AND 7) + (X * 4 Y y HF8)
40 PRINT"ADDR = ";+ la A HEX$ (la BASE (17)) ""H ";
50 SI 2=0 X MOD THEN MS "11110000" ELSE MS "00001111"
60 PRINT"MASK = ";M$
70 GOTO 10
```

El rango admisible de entrada para ambos programas es X = 0 para 255 y Y = 0 para 191. Las declaraciones de datos en el programa Graphics Mode corresponden para la mesa de la máscara de ocho bytes comenzando en 160BH en el MSX ROM. Línea 20 en el programa Multicolor Mode realmente es propia de la división a las cuatro en la rutina del estándar SCALXY. Es incluido para hacer el sistema de coordenadas consistente para ambos programas.

La dirección ... 1639H
El nombre..... FETCHC
La entrada..... Ninguno
La salida..... Un = CMASK, HL=CLOC
Modifica.. Un, HL

La rutina estándar para devolver la dirección física del pixel en uso en estos momentos, el par de registro HL es cargado de A CLOC y de registro de CMASK.

La dirección ... 1640H
El nombre..... STOREC
La entrada..... Un = CMASK, HL=CLOC

La salida..... Ninguno
Modifica.. Ninguno

La rutina estándar para colocar la dirección física del pixel en uso en estos momentos, el par de registro HL es copiado para CLOC y la A de registro es copiada para CMASK.

La dirección ... 1647H
El nombre..... READC
La entrada..... Ninguno
La salida..... Un código = Colour de pixel en uso en estos momentos
Modifica.. AF, EI

La rutina estándar para devolver el color del pixel en uso en estos momentos. La dirección física VRAM es primero obtenido por la rutina del estándar FETCHC. Si la pantalla está de, entonces el Modo Graphics hacia el que el byte apuntó por CLOC es leído del Character Pattern Table por la rutina del estándar RDVRM. El bit requerido está entonces aislado por CMASK y está acostumbrado a seleccionar ya sea el cuatro bits superior o inferior de la entrada correspondiente en el Colour Table.

Si la pantalla está de, entonces el Modo Multicolor hacia que el byte apuntó por CLOC es leído del Character Pattern Table por el RDVRM

- 71 -

4. ROM BIOS

La rutina estándar. CMASK está luego acostumbrado a seleccionar ya sea el cuatro bits superior o inferior de este byte. El valor devuelto en cualquier caso será un código normal de color VDP de cero para quince.

La dirección ... 1676H
El nombre..... SETATR
La entrada..... Un código = Colour
La salida..... La C de la bandera si el código ilegal
Modifica.. Las banderas

La rutina estándar para colocar el color de tinta de gráficos usado por el SETC y rutinas del estándar NSETCX. El código de color, de cero para quince, es simplemente colocado en ATRBYT.

La dirección ... 167EH
El nombre..... SETC
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF, EI

La rutina estándar para colocar el pixel en uso en estos momentos para cualquier color, el código de color es tomado de ATRBYT. La dirección física VRAM de pixel es primero obtenido por la rutina del estándar FETCHC. En Graphics Mode ambos el Character Pattern Table y Colour Table están entonces modificados (186CH).

En el Multicolor Mode el byte apuntado para por CLOC es leído del Character Pattern Table por la rutina del estándar RDVRM. Los contenidos de ATRBYT son luego colocados en el cuatro bits superiores o inferior, tan determinados por CMASK, y el byte espalda escrita por la rutina del estándar WRTVRM

La dirección ... 16ACH

Esta rutina conmueve la colocación coetánea del mismo de la dirección física del pixel bien. Si el borde de la derecha de la pantalla es excedido, entonces regresa con ustedstedst Flag y la dirección física es igual. En Graphics Mode CMASK está primer intercambiado bit bien, si el pixel todavía permanece dentro del byte que la rutina termina. Si CLOC está en casa de la célula de carácter (LSB = F8H para FFH) de la extrema derecha, entonces luego la rutina termina con Flag C (175AH).

CMASK diferente está listo para 80H, el leftmost pixel, y 0008H añadido para CLOC.

En el Multicolor Mode las transferencias de control para una rutina separada (1779H).

La dirección ... 16C5H
El nombre..... RIGHTC
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF

La rutina estándar para mover la dirección física del pixel en uso en estos momentos

- 72 -

4. ROM BIOS

Un derecho de colocación. En Graphics Mode CMASK está primer intercambiado bit bien, si el pixel todavía permanece dentro del byte que la rutina termina. CMASK diferente está listo para 80H, el leftmost pixel, y 0008H añadido para CLOC. Noto que las direcciones incorrectas serán producidas si el borde de la derecha de la pantalla es excedido.

En el Multicolor Mode las transferencias de control para una rutina separada (178BH).

La dirección ... 16D8H

Esta rutina mueve la izquierda de colocación del mismo de la dirección física del pixel en uso en estos momentos. Si el borde de la izquierda de la pantalla es excedido, entonces devuelve C Flag y la dirección física es igual. En Graphics Mode que CMASK está que lo primer intercambiado mordió izquierda, si el pixel todavía permanece dentro del byte, entonces la rutina termina. Si CLOC está en casa de la célula de carácter del leftmost (LSB = 00H para 07H), entonces luego la rutina termina con Flag C (175AH). CMASK diferente está listo para 01H, el pixel de la extrema derecha, y 0008H sustraído de CLOC.

En el Multicolor Mode las transferencias de control para una rutina separada (179CH).

La dirección ... 16EEH
El nombre..... LEFTC
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF

La rutina estándar para mover la colocación coetánea del mismo de la dirección física del pixel dejó. En Graphics Mode que CMASK está que lo primer intercambiado mordió izquierda, si el pixel todavía permanece dentro del byte, entonces la rutina termina. CMASK diferente está listo para 01H, el leftmost pixel, y 0008H sustraído de CLOC. Noto que las direcciones incorrectas serán producidas si el borde de la izquierda de la pantalla es excedido.

En el Multicolor Mode las transferencias de control para una rutina separada (17ACH).

La dirección ... 170AH
El nombre..... TDOWNC
La entrada..... Ninguno
La salida..... La C de la bandera si fuera de pantalla
Modifica.. AF

La rutina estándar para mover la colocación coetánea del mismo de la dirección física del pixel abajo. Si el borde más bajo de la pantalla es excedido, entonces devuelve C Flag y la dirección física es igual. En Graphics Mode CLOC es primero incrementado, si todavía permanece dentro de una ocho demarcación de byte que la rutina termina. Si CLOC estaba de la pelea más bajo (CLOC > = 1700H) de carácter, entonces luego la rutina termina con Flag C (1759H). Se el se sumó a 00F8H diferente

- 73 -

4. ROM BIOS

CLOC.

En el Multicolor Mode las transferencias de control para una rutina separada (17C6H).

La dirección ... 172AH
El nombre..... DOWNC
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. AF

La rutina estándar para mover la colocación coetánea del mismo de la dirección física del pixel abajo. En Graphics Mode CLOC es primero incrementado, si todavía permanece dentro de una ocho demarcación de byte que la rutina termina. El 00F8H diferente se agrega para CLOC. Noto que las direcciones incorrectas serán producidas si el borde más bajo de la pantalla es excedido.

En el Multicolor Mode las transferencias de control para una rutina separada (17DCH).

La dirección ... 173CH
El nombre..... TUPC
La entrada..... Ninguno
La salida..... La C de la bandera si fuera de pantalla
Modifica.. AF

La rutina estándar para conmovier la colocación coetánea del mismo de la dirección física del pixel arriba. Si el carel de la pantalla es excedido, entonces regresa con ustedstedst Flag y la dirección física es igual. En Graphics Mode CLOC está primer decremented, si todavía permanece dentro de una ocho demarcación de byte que la rutina termina. Si CLOC estaba de la pelea máxima (CLOC < 0100H) de carácter, entonces luego la rutina termina con ustedstedst Flag. El 00F8H diferente es sustraído de CLOC.

En el Multicolor Mode las transferencias de control para una rutina separada (17E3H).

La dirección ... 175DH
El nombre..... UPC
La entrada..... Ninguno

La salida..... Ninguno
Modifica.. AF

La rutina estándar para conmovir la colocación coetánea del mismo de la dirección física del pixel arriba. En Graphics Mode CLOC está primer decremented, si todavía permanece dentro de una ocho demarcación de byte que la rutina termina. El 00F8H diferente es sustraído de CLOC. Noto que las direcciones incorrectas serán producidas si el carel de la pantalla es excedido.

En el Multicolor Mode las transferencias de control para una rutina separada (17F8H).

- 74 -

4. ROM BIOS

La dirección ... 1779H

Ésta es la versión Multicolora Mode de la rutina en 16ACH. Es idéntico para la versión Graphics Mode excepto que CMASK es cambiado de posición que cuatro mordieron derecha de colocaciones y se convierte en F0H si un límite de la célula es cruzado.

La dirección ... 178BH

Ésta es la versión Multicolora Mode de la rutina del estándar RIGHTC. Es idéntico para la versión Graphics Mode excepto que CMASK es cambiado de posición que cuatro mordieron derecha de colocaciones y se convierte en F0H si un límite de la célula es cruzado.

La dirección ... 179CH

Ésta es la versión Multicolora Mode de la rutina en 16D8H. Es idéntico para la versión Graphics Mode excepto que CMASK es cambiadas de posición que colocaciones de cuatro bits dejaron y se convierte en 0FH si un límite de la célula es cruzado.

La dirección ... 17ACH

Ésta es la versión Multicolora Mode de la rutina del estándar LEFTC. Es idéntico para la versión Graphics Mode excepto que CMASK es cambiadas de posición que colocaciones de cuatro bits dejaron y se convierte en 0FH si un límite de la célula es cruzado.

La dirección ... 17C6H

Ésta es la versión Multicolora Mode de la rutina del estándar TDOWNC. Es idéntico para la versión Graphics Mode excepto que la dirección más bajo de demarcación es 0500H en lugar de 1700H.

Hay un problema no resuelto en el software en esta rutina que lo causará para comportarse imprevisiblemente si MLTCGP, la dirección bajo Table Pattern Character, se varía de su valor normal de cero. Debería haber una instrucción Delaware EX, HL inserto en 17CEH de la dirección.

Si la base Character Pattern Table es aumentada, entonces la rutina pensará que ha alcanzado la raíz de la pantalla cuando realmente no tiene. Esta rutina está usada por la "pintura" declaración así es que lo siguiente demuestra el defecto:

```
10 BASAN (17) = y H1000
20 OCULTAN 3
30 PSET (200,0)
```

40 DRAW"D180L100U180R100 "
50 PINTAN (150,90)
60 GOTO 60

La dirección ... 17DCH

Ésta es la versión Multicolora Mode de la rutina del estándar DOWNC, es idéntica para la versión Graphics Mode.

- 75 -

4. ROM BIOS

La dirección ... 17E3H

Ésta es la versión Multicolora Mode de la rutina del estándar TUPC. Es idéntico para la versión Graphics Mode excepto que es tiene un problema no resuelto en el software tan citado anteriormente, esta vez debería haber una instrucción Delaware EX, HL en 17EBH de la dirección.

Si el discurso bajo Table Pattern Character es aumentada, entonces la rutina pensará que está dentro de la mesa cuando realmente ha excedido el carel de la pantalla. Esto puede ser demostrado quitándole al "R100" parte de Line 40 en el programa previo.

La dirección ... 17F8H

Ésta es la versión Multicolora Mode de la rutina del estándar UPC, es idéntica para la versión Graphics Mode.

La dirección ... 1809H
El nombre..... NSETCX
La entrada..... HL = Pixel llena cuenta
La salida..... Ninguno
Modifica.. AF, BC, DE, HL, EI

La rutina estándar para colocar el color de pixels múltiples horizontalmente rightwards de la dirección física del pixel en uso en estos momentos. Aunque su función puede ser multiplicada por dos por el SETC y rutinas del estándar RIGHTC ésta resultaría significativamente operación más lenta. La cuenta suministrada del pixel debería estar seleccionada a fin de que el borde de derecha de la pantalla no es hecho pasar por éste producirá comportamiento anómalo. La dirección física del pixel en uso en estos momentos es igual por esta rutina.

En Graphics Mode CMASK es primero examinado para determinar el número de pixels a la derecha dentro de la célula de carácter en uso en estos momentos. Asumiendo lo llena cuenta es bastante grande que éstas están luego listas (186CH). El faltante llena cuenta está dividida a las ocho para determinar el número de células enteras de carácter. Los bytes sucesivos en el Character Pattern Table son entonces puestos en el cero y los bytes correspondientes en el Colour Ponon Sobre Una Mesa set de ATRBYT para llenar estas células enteras. El faltante llena cuenta es luego convertida a una máscara mordida, usando la mesa de siete bytes en 185DH, y estos pixels están colocados (186CH).

En el Multicolor Mode las transferencias de control para una rutina separada (18BBH).

La dirección ... 186CH

Esta rutina incrusta para ocho pixels dentro de una célula para un color especificado en Graphics Mode. ATRBYT contiene el código de color, par de registro HL la dirección del byte pertinente en la A Character Pattern Table y de registro una máscara mordida, 11100000 por ejemplo, dónde cada 1 especifican un poco ser determinados.

- 76 -

4. ROM BIOS

Si ATRBYT hace juego con el color 1 existente pixel en el byte correspondiente Colour Table luego cada bit especificado está colocado para 1 en el byte Character Pattern Table. Si ATRBYT hace buena pareja el existente color 0 pixel en el byte correspondiente Colour Table luego cada uno el bit especificado está colocado para 0 en el byte Character Pattern Table.

Si ATRBYT no hace juego con cualquier de los colores existentes en el Colour Table Byte luego normalmente cada bit especificado está colocado para 1 en el byte Character Pattern Table y lo 1 color pixel cambiado en el byte Colour Table. De cualquier forma que si esto resultaría en todos los ratos siendo determinado para 1 en el byte Character Pattern Table luego cada bit especificado está colocado para 0 y lo 0 color pixel cambiado en el byte Colour Table.

La dirección ... 18BBH

Ésta es la versión Multicolora Mode de la rutina del estándar NSETCX. El SETC y rutinas del estándar RIGHTC son llamados lo llena cuenta está agotada. La velocidad de operación no es tan importante en el Modo Multicolor por la resolución inferior de la pantalla y la reducción consiguiente en el número de operaciones requeridas.

La dirección ... 18C7H
El nombre..... GTASPC
La entrada..... Ninguno
La salida..... Delaware = ASPCT1, HL = ASPCT2
Modifica.. DE, HL

La rutina estándar para devolver el "círculo" proporciones dimensionales de incumplimiento de declaración.

La dirección ... 18CFH
El nombre..... PNTINI
La entrada..... Un color = Boundary (0 para 15)
La salida..... La C de la bandera si el color ilegal
Modifica.. AF

La rutina estándar para colocar el color de demarcación para la "pintura" declaración. En el Multicolor Mode el código suministrado de color es colocado en BDRATR. En Graphics Mode BDRATR es emulado de ATRBYT en su estado actual posible para tener colores de la separata de pintura y de demarcación.

La dirección ... 18E4H
El nombre..... SCANR
La entrada..... El interruptor de la B = Fill, la cuenta de Delaware = Skip
La salida..... La cuenta Delaware = Skip de remanente, HL = Pixel
Modifica.. AF, BC, DE, HL, EI

La rutina estándar usada por la "pintura" manipulador de declaración para registrar rightwards

de la dirección física del pixel en uso en estos momentos hasta un código de color igual para BDRATR es encontrada o el borde de lo

- 77 -

4. ROM BIOS

La pantalla es alcanzada. La colocación que termina se convierte en la dirección física del pixel en uso en estos momentos y la colocación inicial es devuelta en CSAVEA y CSAVEM. El tamaño de la región atravesada es devuelto en par de registro HL y FILNAM + 1. La región atravesada está normalmente llena en pero esto puede estar inhibido, en Graphics Mode sólo, por ahí utilizador un parámetro de entrada de cero en B de registro. El escrutinio de salto en par de registro Delaware determina el número máximo de pixels del color requerido que se ignoró de la colocación inicial de puesta en marcha. Esta facilidad es usada por la "pintura" manipulador de declaración para ir en busca de aberturas en una demarcación horizontal bloqueando su progreso ascendente.

La dirección ... 197AH
El nombre..... SCANL
La entrada..... Ninguno
La salida..... La cuenta HL = Pixel
Modifica.. AF, BC, DE, HL, EI

La rutina estándar a investigar hacia la izquierda de la dirección física del pixel en uso en estos momentos hasta un código de color igual para BDRATR es encontrada o el borde de la pantalla es alcanzado. La colocación que termina se convierte en la dirección física del pixel en uso en estos momentos y el tamaño de la región atravesada es devuelto en par de registro HL. La región atravesada está siempre llena adentro.

La dirección ... 19C7H

Esta rutina es usada por el SCANL y rutinas del estándar SCANR para cotejar a actual el color de pixel contra el color de demarcación en BDRATR.

La dirección ... 19DDH
El nombre..... TAPOOF
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. EI

La rutina estándar para detener el casete motor después de datos ha estado escrita para el casete. Después de un retraso de 550 ms, en MSX las máquinas con una condición de espera, el control cae en la rutina del estándar TAPIOF.

La dirección ... 19E9H
El nombre..... TAPIOF
La entrada..... Ninguno
La salida..... Ninguno
Modifica.. EI

La rutina estándar para detener el casete motor después de datos ha sido leída del casete. El relevador motor es abierto por el PPI Mode Port. La nota que interrumpe, cuál debe ser deshabilitado durante transferencias de datos del casete para cronometrar razones, es posibilitado como esta rutina termine.

4. ROM BIOS

La dirección ... 19F1H
El nombre..... TAPOON
La entrada..... Un interruptor de largo = Header
La salida..... La C de la bandera si la terminación CTRL-STOP
Modifica.. AF, BC, HL, DI

La rutina estándar para revolver el casete motor de adelante, espere a 550 ms para la cinta venidera levantado para acelerar y luego escribir un encabezado para el casete. Un encabezado es una ráfaga de Hawaii se cicla escrita delante de cada bloque de datos así es que la tasa de baudio puede ser determinada cuando los datos es vuelta a leer.

El largo del encabezado es determinado por los contenidos de A de registro: El encabezado del 00H = Short, el encabezado NZ = Long. Las declaraciones del casete de BASIC "salvo", "CSAVE" y "BSAVE" todo generan una caída de cabeza larga al principio del archivo, en parte delantera del bloque de la identificación, y después uso pone en cortocircuito encabezados entre bloques de datos. El número de ciclos en el encabezado está también modificado por la tasa coetánea de baudio para conservar su constante de duración:

1200 CORTOCIRCUITO Baud ... 3840 Ciclos ... 1.5 Segundos
1200 Baud LONG ... 15360 Ciclos ... 6.1 Segundos
2400 CORTOCIRCUITO Baud ... 7936 Ciclos ... 1.6 Segundos
2400 Baud LONG ... 31744 Ciclos ... 6.3 Segundos

Después de que el motor se haya encendido y el retraso ha caducado los contenidos de ENCABEZADO son multiplicados a las doscientos cincuenta y seis y, si la A de registro fuera poco pone en el cero, por un más factor de cuatro para producir la cuenta ciclista. Los ciclos de Hawaii son entonces generados (1A4DH) hasta que la cuenta está agotada después de lo cual el control se traslada para la rutina del estándar BREAKX. Porque la llave CTRL-STOP es sólo examinada en la terminación es imposible romper fuera en parte muy a través de esta rutina.

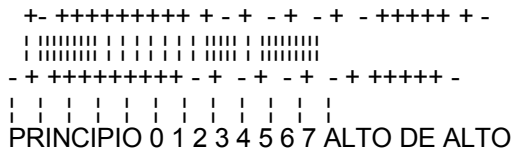
La dirección ... 1A19H
El nombre..... TAPOUT
La entrada..... Un byte = Data
La salida..... La C de la bandera si la terminación CTRL-STOP
Modifica.. AF, B, HL

La rutina estándar para escribir un solo byte de datos para el casete. El MSX ROM usa un software método propulsado FSK (la Frecuencia Shift Tecleó) para almacenar información en el casete. En la tasa de 1200 baudios esto es idéntico para el Kansas City Standard usado por el BBC para la distribución de BASICODE programa.

En 1200 el baudio a cada 0 bit le es escritos como un ciclo 1200 completo Hz LO y cada 1 bit como dos ciclos 2400 completos Hz Hawaii. La tasa de datos es así de constante como 0 y 1 añicos tienen la misma duración. Cuando lo 2400 baudio evalúe es seleccionado que las dos frecuencias se convierten en 2400 Hz y 4800 Hz pero el formato es otrora igual.

4. ROM BIOS

Un byte de datos está escrito con un 0 bit de principio (1A50H), ocho añicos de datos con el bit menos significativo primero, y dos 1 añicos de alto (1A40H). En la tasa de 1200 baudios un solo byte tendrá una duración nominal de $11 \times 833 \mu s = 9.2$ señora Después De Que los añicos de a alto les hayan sido que las transferencias de control para la rutina del estándar BREAKX para comprobar al CTRL-STOP teclean. El byte debajo del que el 43H es mostrado como él estaría escrito para casete:



1 transiciones de = dos (como los AÑICOS de ALTO) "cortocircuitos"
0 = una transición "larga" (como el PRINCIPIO MORDIDO)

Figura 39: El Byte De Datos Del Casete

Es importante no dejar demasiado tiempo un intervalo entre los bytes cuando los datos de la escritura como éste aumentarán la tasa de error. Una abertura de interbyte de $80 \mu s$, por ejemplo, produce una tasa leída de fracaso de aproximadamente doce por ciento. Si una cantidad sustancial de procesamiento es precisada entre cada byte luego moderador debería usarse para aglomerar datos en bloques del headered. El BASIC "salvo" formato es de este tipo.

La dirección ... 1A39H

Esta rutina escribe un solo ciclo LO con un largo de aproximadamente $816 \mu s$ para el casete. El largo de cada mitad del ciclo es tomado de transferencias de PUNTO BAJO y de control para el generador ciclista general (1A50H).

La dirección ... 1A40H

Esta rutina escribe dos ciclos de Hawaii para el casete. El primer ciclo es generado (1A4DH) seguido por un 17 retraso μs y luego el segundo ciclo (1A4DH).

La dirección ... 1A4DH

Esta rutina escribe un solo ciclo de Hawaii con un largo de aproximadamente $396 \mu s$ para el casete. El largo de cada mitad del ciclo es tomado de ALTO y el control cae en el generador ciclista general.

La dirección ... 1A50H

Esta rutina escribe un solo ciclo para el casete. El largo de primer tiempo del ciclo es suministrado en L de registro y su segundo medio en H de registro. El primer largo es pelusa contada y luego el Cas Out mordió set por el PPI Mode Port. El segundo largo es pelusa contada y la reanudación del bit Cas Out.

4. ROM BIOS

En todas las máquinas MSX el Z80 corre en una frecuencia del reloj de 3.579545 MHz (280 ns) con una condición de espera durante el ciclo M1. Como esta rutina cuenta cada 16T declara cada incremento de la unidad en la cuenta de largo representa un período de 4.47 μ s. Hay también unos gastos fijos fijos de 20.7 μ s asociados con la rutina cualquier cosa la cuenta de largo.

La dirección ... 1A63H
El nombre..... TAPION
La entrada..... Ninguno
La salida..... La C de la bandera si la terminación CTRL-STOP
Modifica.. AF, BC, DE, HL, DI

La rutina estándar para revolver el casete motor de adelante, lea el casete hasta que un encabezado sea encontrado y luego determina la tasa de baudio. Los ciclos sucesivos son leídos del casete y el largo de cada uno medido (1B34H). Cuando 1,111 ciclos han sido encontrados menos de variación 35 μ s en sus largos un encabezado ha estado ubicado.

Los siguientes 256 ciclos son entonces leídos (1B34H) y promediados para determinar el largo de ciclo del casete Hawaii. Esta figura está multiplicado por 1.5 y colocado en LOWLIM donde define el largo lo suficientemente bueno mínimo de un 0 bit de principio. El largo de ciclo de Hawaii es colocado en WINWID y se usará para discriminar entre ciclos LO y Hawaii.

La dirección ... 1ABCH
El nombre..... TAPIN
La entrada..... Ninguno
La salida..... Un = Byte le leyó, la C Flag si CTRL-STOP o yo el error de la / O
Modifica.. AF, BC, DE, L

La rutina estándar para leer un byte de datos del casete. El casete es primera lectura continuamente hasta que un bit de principio es encontrado. Esto está hecho localizando una transición negativa, midiendo el siguiente pedazo ciclista (1B1FH) y comparando así de para ver si es mayor que LOWLIM.

Cada uno de los ocho añicos de datos son entonces leídos contando el número de transiciones dentro de un período de tiempo fijo (1B03H). Si el cero o uno las transiciones es encontrado que es un 0 bit, entonces si dos o tres son encontrados que es un 1 bit. Si más que tres transiciones fueran encontradas la rutina terminan con ustedstedss Flag como éste - se supone - es un error del hardware de algún tipo. Después del valor de cada bit ha estado resuelto uno que adicionalmente a uno o dos transiciones les es leídos (1B23H) para retener sincronización. Con un mismo impar de la cuenta de transición a más les serán leídos, con una cuenta pareja de transición dos más.

La dirección ... 1B03H

Esta rutina es usada por la rutina del estándar TAPIN para contar el número de casete efectúa una transición dentro de un período fijo de

- 81 -

4. ROM BIOS

El tiempo. La duración de la ventana es contenida en WINWID y es aproximadamente 1.5 veces

el largo de un ciclo de Hawaii como se muestra debajo:

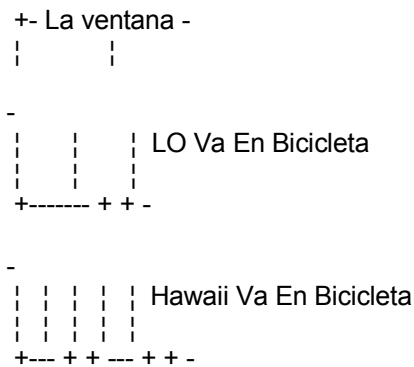


Figura 40: El casete Window

El bit Cas Input es continuamente probado por PSG Register 14 y comparado con la lectura previa mantenido en E de registro. Cada vez que un cambio de condición es encontrado la C de registro es incrementada. La tasa de muestreo es una vez cada 17.3 μ s así el valor en WINWID, lo cual estaba resuelta por la rutina del estándar TAPION con una tasa de la cuenta de 11.45 μ s, lo está de manera impresionante multiplicarse un y que una mitad cronometra.

La dirección ... 1B1FH

Esta rutina mide el tiempo para la siguiente transición de entrada del casete. El Cassette Input mordió es continuamente probado por PSG Register 14 hasta que se altere de la condición abastecida en E de registro. La bandera estatal es entonces invertida y la cuenta de duración regresó en registro C, cada incremento de la unidad representa un período de 11.45 μ s.

La dirección ... 1B34H

Esta rutina mide el pedazo de un ciclo completo del casete de transición negativa para negar transición. El Cassette Input mordió es probado por PSG Register 14 hasta que vaya al cero. La bandera de transición en E de registro está colocada para poner en el cero y el tiempo para la transición positiva medida (1B23H). El tiempo para la transición negativa es entonces medido (1B25H) y el total regresó en C de registro.

La dirección ... 1B45H

El nombre..... SUPERE

La entrada..... Un = Carácter a la salida

La salida..... Ninguno

Modifica.. El

La rutina estándar usada por el BASIC Interpreter para devolver un carácter para el dispositivo en uso en estos momentos. La rutina del estándar ISFLIO es

4. ROM BIOS

La primera parte usada para cheque ya sea la salida es actualmente direccionada uno yo el amortiguador de la / O, si así es que el control se traslada para el conductor secuencial (6C48H) de salida por la rutina del estándar CALBAS. Si PRTFLG es, entonces ponga en el cero transferencias de control para la rutina del estándar CHPUT para devolver el carácter para la

pantalla. Asumir que la impresora es RAWPRT activo es comprobado. Si esto es poco ponga en el cero, entonces el carácter está aprobado directamente para la impresora (1BABH), el control diferente cae en la rutina del estándar OUTDLP.

La dirección ... 1B63H
El nombre..... OUTDLP
La entrada..... Un = Carácter a la salida
La salida..... Ninguno
Modifica.. El

La rutina estándar para devolver un carácter para la impresora. Si el carácter es, entonces un código de la ETIQUETA (0, 8, 16 etcétera (09H) espacios es expedido para la rutina del estándar OUTDLP hasta que LPTPOS sea un múltiplo de ocho.). Si el carácter es, entonces un código de RETORNO DE CARRO (0DH) LPTPOS es puesto en el cero si es cualquier otro código de control LPTPOS está no afectado, si es un carácter explayable LPTPOS es incrementado.

Si NTMSXP es cero, entonces significar una impresora específica a MSX está conectado, el carácter está aprobado directamente para la impresora (1BABH). Asumir una impresora normal está conectado que la rutina del estándar CNVCHR se usa para revisar en busca de personajes gráficos. Si el carácter es un encabezado codifique (01H), entonces la rutina termina sin acción. Si es un carácter gráfico convertido, entonces es reemplazado por un espacio, todo otros personajes son pasados a la impresora (1BACH).

La dirección ... 1B97H

Este veinte byte la mesa es usado por el decodificador del teclado para encontrar la rutina correcta para un número crucial dado:

TECLEE NÚMERO PARA FUNCIONAR

El 00H para 2FH 0F83H Rows 0 para 5
El 30H para el CAMBIO 32H 0F10H, CTRL, GRÁFICA
33H 0F36H LLEGUE AL CLÍMAX
El CÓDIGO 34H 0F10H
El 35H para 39H 0FC3H F1 para F5
El 3AH para 3BH 0

F10H ESC, ETIQUETA

3CH 0F46H DETÉNGASE
El 3DH para 40H 0F10H BS, el RETORNO DE CARRO, SEL, ESPACIO
41H 0F06H DIRÍJASE HACIA EL BLANCO
El 42H para 57H 0F10H Indianas, DEL, CURSOR

La dirección ... 1BABH

Esta rutina es usada por la rutina del estándar OUTDLP para pasar un carácter a la impresora. Es enviado por la rutina del estándar LPTOUT, si esto devuelve C Flag que el control se traslada para lo

- 83 -

4. ROM BIOS

"El dispositivo yo" el generador de "error de la / O" (73B2H) por la rutina del estándar CALBAS.

La dirección ... 1BBFH

El siguiente 2 KB contiene el conjunto de caracteres que se impulsa arriba. Los primeros ocho bytes contienen el patrón para el código de caracteres 00H, los segundos ocho bytes el patrón para 01H de código de caracteres etcétera para el código de caracteres FFH.

La dirección ... 23BFH

El nombre..... PINLIN

La entrada..... Ninguno

La salida..... HL = Eche A Andar de C del texto, Flag si la terminación CTRL-STOP

Modifica.. AF, BC, DE, HL, EI

La rutina estándar usada por el BASIC Interpreter Mainloop para coleccionar una línea lógica de texto de la consola. Controle transferencias para la rutina del estándar INLIN poco después del punto donde la línea previa ha estado cortada (23E0H).

La dirección ... 23CCH

El nombre..... QINLIN

La entrada..... Ninguno

La salida..... HL = Eche A Andar de C del texto, Flag si la terminación CTRL-STOP

Modifica.. AF, BC, DE, HL, EI

La rutina estándar usada por la "entrada" manipulador de declaración para coleccionar una línea lógica de texto de la consola. ¿Los personajes "? " Es exhibido por la rutina del estándar OUTDO y el control cae en la rutina del estándar INLIN.

La dirección ... 23D5H

El nombre..... INLIN

La entrada..... Ninguno

La salida..... HL = Eche A Andar de C del texto, Flag si la terminación CTRL-STOP

Modifica.. AF, BC, DE, HL, EI

La rutina estándar usada por la "entrada de la LÍNEA" manipulador de declaración para coleccionar una línea lógica de texto de la consola. Los personajes son leídos del teclado hasta ya sea el RETORNO DE CARRO o las llaves CTRL-STOP se aprietan. La línea lógica es entonces leída del carácter de la pantalla por el carácter y está posado en el amortiguador del texto Workspace Area BUF.

Las coordenadas coetáneas de la pantalla son primero tomadas de CSRX y CSRY y están posado en FSTPOS. La fila de la pantalla inmediatamente por encima de la actual luego tiene su entrada en LINTTB hecho poco cero (0C29H) para detenerla extendiéndose lógicamente en la fila de corriente.

Cada teclado que el carácter leyó por la rutina del estándar CHGET es cotejado contra (0919H) la mesa crucial editora en 2439H.

El control luego se traslada para una de las rutinas editoras o para el manipulador crucial predeterminado (23FFH) según el caso. Este proceso

- 84 -

4. ROM BIOS

Continúa hasta que la C Flag sea devuelta por el CTRL-STOP o rutinas de RETORNO DE CARRO. El par de registro HL está luego colocado a apuntar hacia el principio de BUF y la rutina termina. Note que lo lleve, la bandera es despejada cuando Flag NZ es también devuelto para

distinguir entre un RETORNO DE CARRO o una terminación protegida CTRL-STOP y una terminación normal CTRL-STOP.

La dirección ... 23FFH

Esta rutina procesa que todos los personajes para la rutina del estándar INLIN exceptúan la edición especial las llaves. Si el carácter es, entonces un código de la ETIQUETA (09H) espacios es expedido (23FFH) hasta que CSRX sea un múltiplo de ocho y uno (las columnas 1, 9, 17, 25, 33). Si el carácter es un encabezado gráfico código (01H), entonces es simplemente del que se hizo eco para la rutina del estándar OUTDO. Todo otro control codifica más pequeño que 20H es de lo que se hizo eco para la rutina del estándar OUTDO después de la cual INSFLG y CSTYLE son puestos en el cero. Pues el personajes explayable INSFLG es primero a cuadros y un espacio inserto (24F2H) si aplicable antes del carácter es de lo que se hizo eco para la rutina del estándar OUTDO.

La dirección ... 2439H

Esta mesa contiene la edición especial que las llaves reconocieron por la rutina del estándar INLIN conjuntamente con las direcciones pertinentes:

CODIFIQUE PARA FUNCIONAR

08H 2561H BS, backspace
12H 24E5H Indianas, palanca de apoyo intercala modo
1BH 23FEH ESC, ninguna acción
02H 260EH CTRL-B, palabra previa
06H 25F8H CTRL-F, después palabra
0EH 25D7H CTRL-N, fin de línea lógica
05H 25B9H CTRL-E, claro a acabar de línea
03H 24C5H CTRL-STOP, termine
EI RETORNO DE CARRO 0DH 245AH, termine
15H 25AEH CTRL-U, línea clara
7FH 2550H DEL, suprime carácter

La dirección ... 245AH

Esta rutina realiza la operación de RETORNO DE CARRO para la rutina del estándar INLIN. Las coordenadas de puesta en marcha de la línea lógica son encontradas (266CH) y el cursor cambió de dirección de la pantalla (0A2EH). Hasta 254 personajes es entonces leído del VDP VRAM (0BD8H) y está posado en BUF. Cualquier códigos nulos (00H) están ignorados, cualquier personajes más pequeños que 20H son reemplazado por un código gráfico (01H) del encabezado y el carácter mismo con 40H añadido. Como el fin de cada fila física es alcanzado LINTTB es al que se dio jaque (0C1DH) para ver si la línea lógica se extiende hasta la siguiente fila física. Arrastrando espacios está entonces pelado de BUF y un byte de cero añadido como un fin de rotulador del texto. El cursor es

- 85 -

4. ROM BIOS

Recuperado para la pantalla (09E1H) y sus coordenadas se sedimenta para la última fila física de la línea lógica por la rutina del estándar POSIT. Un código LF es expedido para la rutina del estándar OUTDO, INSFLG es puesto en el cero y la rutina termina con un código de RETORNO DE CARRO (0DH) en NZ de registro de la A y Flag, C. Este código de RETORNO DE CARRO

será del que se hizo eco para la pantalla por el mainloop de rutina del estándar INLIN poco antes de que termine.

La dirección ... 24C5H

Esta rutina realiza la operación CTRL-STOP para la rutina del estándar INLIN. La última fila física de la línea lógica es encontrada por LINTTB examinante (0C1DH), CSTYLE es puesto en el cero, un byte de cero es colocado al principio de BUF y todo las variables de música son aclaradas por la rutina del estándar GICINI. TRPTBL es entonces examinado (0454H) para ver si uno "en el ALTO" la declaración es activo, si así es que el cursor es vuelto a arrancar (24AFH) y la rutina ultima con Flag NZ, C. BASROM es entonces a cuadros para ver si un ROM protegido corre, si así es que el cursor es vuelto a arrancar (24AFH) y la rutina ultima con Flag NZ, C. De otra manera el cursor es vuelto a arrancar (24B2H) y la rutina ultima con Flag Z, C.

La dirección ... 24E5H

Esta rutina realiza la operación de Indianas para la rutina del estándar INLIN. La condición coetánea de INSFLG es invertida y el control termina por el CSTYLE ajustando rutina (242CH).

La dirección ... 24F2H

Esta rutina intercala un carácter de espacio para el capítulo crucial predeterminado de la rutina del estándar INLIN. El cursor es removido (0A2EH) y el cursor en uso en estos momentos se coordina tomado de CSRX y CSRY. El carácter en esta colocación es leído del VDP VRAM (0BD8H) y reemplazado con un espacio (0BE6H). Los personajes sucesivos son luego copiado colocación de la columna a la derecha hasta que el fin de la fila física sea cumplido.

En este punto LINTTB es examinado (0C1DH) para determinar si la línea lógica está extendida, si así es que el proceso continúa en la siguiente fila física. De otra manera el carácter tomado de la última colocación de la columna es examinado, si éste es un espacio que la rutina termina reemplazando el cursor (09E1H).

De otra manera la entrada de la fila física en LINTTB es puesta en el cero para indicar un verso lógico extendido. El número de la siguiente fila física es comparado con el número de filas en la pantalla (0C32H). Si la siguiente fila es lo último la pantalla es enrollada arriba (0A88H), entonces de otra manera una fila en blanco es inserta (0AB7H) y el proceso copiador continúa.

La dirección ... 2550H

Esta rutina realiza la operación DEL para la rutina del estándar INLIN. Si la colocación coetánea del cursor está en lo

- 86 -

4. ROM BIOS

La columna de la extrema derecha y la línea lógica no está extendida ninguna acción está ocupada aparte de para escribir un espacio para el VDP VRAM (2595H). De otra manera un código CORRECTO (1CH) es expedido para la rutina del estándar OUTDO y el control cae en la rutina BS.

La dirección ... 2561H

Esta rutina realiza la operación BS para la rutina del estándar INLIN. El cursor es primero removido (0A2EH) y el decremented de coordenada de la columna del cursor a menos que queda a la colocación del leftmost y la fila previa no está extendida.

Los personajes son entonces leídos del VDP VRAM (0BD8H) y el de atrás escrito colocación a la izquierda (0BE6H) hasta que el fin de la línea lógica es cumplido. En este punto un espacio es escrito para el VDP VRAM (0BE6H) y el carácter del cursor está recuperado (09E1H).

La dirección ... 25AEH

Esta rutina realiza la operación CTRL-U para la rutina del estándar INLIN. El cursor está removido (0A2EH) y el principio de la línea lógica localizada (266CH) y está posado en CSRX y CSRY. El verso lógico entero es entonces aclarada (25BEH).

La dirección ... 25B9H

Esta rutina realiza la operación CTRL-E para la rutina del estándar INLIN. El cursor está removida (0A2EH) y el resto de la fila física aclarado (0AEEH). Este proceso es repetido para filas físicas sucesivas hasta que el fin de la línea lógica sea encontrado en LINTBB (0C1DH). El cursor está entonces recuperado (09E1H), INSFLG puesto en el cero y CSTLYE reanudación para un cursor del bloque (242DH).

La dirección ... 25D7H

Esta rutina realiza la operación CTRL-N para la rutina del estándar INLIN. El cursor está removido (0A2EH) y la última fila física de la línea lógica encontrada por examen de LINTTB (0C1DH). Comenzando en la columna de la extrema derecha de estos personajes físicos de la fila es que la lectura del VDP VRAM (0BD8H) hasta un carácter de poco espacio es encontrada. Las coordenadas del cursor son luego colocan una columna a la derecha de esta colocación (0A5BH) y la rutina termina restituyendo el cursor (25CDH).

La dirección ... 25F8H

Esta rutina realiza la operación CTRL-F para la rutina del estándar INLIN. El cursor está removido (0A2EH) y emocionado sucesivamente bien (2624H) hasta que un carácter poco alfanumérico sea encontrado. El cursor está luego emocionado sucesivamente bien (2624H) hasta que un carácter alfanumérico sea encontrado. La rutina termina restituyendo el cursor (25CDH).

- 87 -

4. ROM BIOS

La dirección ... 260EH

Esta rutina realiza la operación CTRL-B para la rutina del estándar INLIN. El cursor está removido (0A2EH) y emocionado sucesivamente izquierda (2634H) hasta que un carácter alfanumérico sea encontrado. El cursor está luego emocionado sucesivamente izquierda (2634H) hasta que un carácter poco alfanumérico sea encontrado y luego emocionado colocación bien (0A5BH). La rutina termina restituyendo el cursor (25CDH).

La dirección ... 2624H

Esta rutina conmueve la colocación del mismo del cursor bien (0A5BH), las cargas registran D con el número de la columna de la extrema derecha, registran E con el número más bajo de la fila y entonces las pruebas para un carácter alfanumérico en la colocación del cursor (263DH).

La dirección ... 2634H

Esta rutina mueve la izquierda de colocación del mismo del cursor (0A4CH), las cargas registran D con el número de la columna del leftmost y registran E con el número máximo de la fila. Las coordenadas coetáneas del cursor son comparadas con estos valores y la rutina termina Z Flag si el cursor queda a esta colocación. De otra manera el carácter en esta colocación es leído del VDP VRAM (0BD8H) e inspeccionado para la sede si es alfanumérica. Si así es que la rutina ultima a Flag NZ, entonces C otrora ultima Flag NZ, Carolina Del Norte.

Los personajes alfanuméricos son los dígitos "0" para "9" y las cartas "A" para "Z" y "uno" para "z". También incluido es el 86H de personajes de gráficos para 9FH y A6H para FFH, estos fuera cartas originalmente japonesas y debería haber quedado excluido durante la conversión para el UK ROM.

La dirección ... 266CH

Esta rutina encuentra el principio de una línea lógica y los ingresos en los que su pantalla coordina registran par HL. Cada fila física por encima de la actual es comprobada por la mesa LINTTB (0C1DH) hasta que una fila poco extendida sea encontrada. La fila inmediatamente debajo de éste en la pantalla es el principio de la línea lógica y su número de la fila es colocado en L de registro. Esto es entonces comparado con FSTPOS, lo cual contiene el número de la fila cuando la rutina del estándar INLIN fue primero introducido, para ver si el cursor está todavía en la misma línea. Si así es que la columna coordinada en H de registro está colocada para su colocación inicial de FSTPOS. La H diferente de registro está colocada para la colocación del leftmost para devolver la línea entera.

La dirección ... 2680H, JP para.

La dirección ... 2683H, JP para la rutina del estándar SYNCHR (558CH).

La dirección ... 2686H, JP para la rutina del estándar CHRGR (4666H).

La dirección ... 2689H, JP para la rutina del estándar GETYPR (5597H).

- 88 -

5. EL INTÉRPRETE DE BASIC ROM

El BASIC Microsoft ha evolucionado sobre los años para su colocación presente como el estándar de la industria. Estaba originalmente escrito para lo 8080 Microprocessor y aun la versión MSX se celebra en 8080 forma Assembly Language. Este proceso de desarrollo continuo quiere decir que hay menos instrucciones específicas a Z80 que se esperarían en un más programa moderno. También quiere decir que los cambios numerosos han sido hechos y el resultado es un programa más bien enrollado. La estructura del Interpreter la hace improbable que un programa de aplicación podrá usar su muchas instalaciones. No importa cuán la mayoría de programas necesitarán cooperarle hasta cierto punto tan este capítulo da una descripción detallada de su operación.

Hay cuatro áreas fácilmente identificables de importancia dentro del Interpreter, el que más familiar para cualquier usuario es el Mainloop (4134H). Esto colecciona rayado numerado de texto de la consola y los pone en su lugar en el Program Text Area de memoria hasta que una declaración directa sea recibida.

El Runloop (4601H) es responsable de la ejecución de un programa. Examina la primera señal de cada línea de programa y llama la rutina apropiada para procesar el resto de la declaración. Esto continúa hasta que ninguno de los más restos del texto de programa, el control luego

ingresos para el Mainloop.

El análisis de operands numéricos o de la cuerda dentro de una declaración es realizado por el Evaluador Expression (4C64H).

Cada expresión está compuesta de factores, a su vez analizadas por el Evaluador Factor (4DC7H), cuáles son acoplados por operadores diádicos del infix. Como hay varios tipos de operand, notablemente revisten números, lo cual no puede formar en parte de una expresión en el BASIC Microsoft que el término "evaluó" está sólo acostumbrado a aplicarse a esos que pueden. De otra manera un término tan "computado" será usado.

El un punto a notar al examinar al Interpreter en detalle es que contiene una parte de truco el código. Los escritores parecen particularmente cariñosos metiéndose en el centro de instrucciones para proveer proposiciones múltiples de entrada para una rutina. Como un ejemplo tome la instrucción:

3E D1 Normal: LD A, 0D1H

Estando encontrado en la forma usual esto por supuesto cargará el acumulador con el valor D1H. De cualquier forma que si es introducido en "Normal" luego será ejecutado como una instrucción POP de Delaware.

El Interpreter tiene muchos capítulos de modo semejante oscuros.

- 89 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 268CH

Esta rutina es usada por el Evaluador Expression para sustraer dos operands dobles de precisión. El primer operand es contenido en DAC y el segundo en ARG, el resultado es devuelto en DAC.

Segundo el signo mantissa de operand es invertido y el control cae en la rutina de adición.

La dirección ... 269AH

Esta rutina es usada por el Evaluador Expression para añadir dos operands dobles de precisión. El primer operand es contenido en DAC y el segundo en ARG, el resultado es devuelto en DAC. Si el segundo operand es ponga en el cero, entonces la rutina termina sin acción, si el primer operand es cero el segundo operand es copiado para DAC (2F05H) y la rutina termina. Los dos exponentes son comparados, si difieren por más que 10^{15} que la rutina termina con el operand mayor como el resultado.

De otra manera la diferencia entre los dos exponentes se usa para poner en línea el mantissae intercambiando a lo más pequeño rightwards (27A3H), por ejemplo:

```
19.2100 .1921*10^2 .192100
+ .7436 .7436*10^0 .007436
```

Si los dos signos del mantissa son iguales el mantissae se agrega luego (2759H), entonces si son diferentes el mantissae es sustraído (276BH). El exponente del resultado es simplemente el

más grande de los dos exponentes originales. Si un excedente fue producido por adición el mantissa de resultado es desviado enmiende un dígito (27DBH) y el exponente incrementado. Si la conducción pone en el cero fue producido por sustracción, entonces el mantissa de resultado es renormalizado intercambiando izquierda (2797H). El byte del guarda es entonces examinado y el resultado redondeado de arriba de si el decimoquinto dígito es iguales o más gran que cinco.

La dirección ... 2759H

Esta rutina se añade que el mantissae de precisión de dos dobles contuvo en DAC y ARG y devuelve el resultado en DAC.

La adición empieza por lo menos colocaciones significativas, DAC + 7 y ARG + 7, e ingresos dos dígitos a la vez para los siete bytes.

La dirección ... 276BH

Esta rutina sustrae el mantissae de precisión de dos dobles contenido en DAC y ARG y devuelve el resultado en DAC.

La sustracción empieza en los bytes del guarda, DAC + 8 y ARG + 8, e ingresos dos dígitos a la vez para los ocho bytes. Si el resultado poco fluye se corrige sustrayéndolo de cero e invirtiendo el mantissa signo, entonces por ejemplo:

$$0.17-0.85 = 0.32 = - 0.68$$

- 90 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 2797H

Esta rutina intercambia el mantissa de precisión doble contenido en izquierda de dígito del mismo DAC.

La dirección ... 27A3H

Esta rutina desvía un mantissa doble de precisión bien. El número de dígitos a desviar es suministrado en A de registro, la dirección de byte más significativo de mantissa es suministrada en par de registro HL. La cuenta de dígito es primero dividido a las dos para poner aparte el byte y cuentas de dígito. El número requerido de bytes enteros son luego derecho desviado y los bytes más significativos puestos en el cero. Si algunos dígitos fueron especificados que el mantissa es entonces intercambiado uno más derecho de dígito.

La dirección ... 27E6H

Esta rutina es usada por el Evaluatodor Expression para multiplicar dos operands dobles de precisión. El primer operand es contenido en DAC y el segundo en ARG, el resultado es devuelto en DAC. Si cualquier operand es ponga en el cero, entonces la rutina termina con un resultado de cero (2E7DH). De otra manera los dos exponentes se agregan para producir el exponente de resultado. Si esto es más pequeño que 10^{63} que la rutina termina con un resultado de cero, si eso son mayores que 10^{63} un "error de excedente" es generado (4067H). Los dos signos del mantissa son entonces tratados para producir el signo del resultado, si son lo mismo el resultado es positivo, si difieren que es negativo.

Aun sin embargo el mantissae está en formato BCD que son multiplicados usando la normalidad binaria añadida y desvíe método. Para lograr esto es el primer operand sucesivamente multiplicado por dos (288AH) para producir la constante $X * 80$, $X * 40$, $X * 20$, $X * 10$, $X * 8$, $X * 4$, $X * 2$, y X en el amortiguador del HOLD8. El segundo operand se queda en ARG y DAC es

puesto en el cero para funcionar como el acumulador del producto. La multiplicación procede tomando pares sucesivos de dígitos del segundo operand echando a andar con el par menos significativo. Para cada 1 bit en el par de dígito el múltiplo apropiado del primer operand se agrega para el producto. Como un ejemplo el solo 1823×96 de multiplicación produciría:

$$1823 \times 10010110 = (1823 \times 80) + (1823 \times 10) + (1823 \times 4) + (1823 \times 2)$$

Como cada par de dígito es completado el producto es intercambiado que dos dígitos enmiendan. Cuando todo siete pares de dígito han sido procesados la rutina termina renormalizando y redondeando arriba del producto (26FAH).

El tiempo requerido pues una multiplicación depende mayormente del número de 1 añicos en el segundo operand. El peor caso, cuándo todos los dígitos es sevens, pueden hacerse cargo para 11 ms se comparó al promedio de aproximadamente 7 señora

- 91 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 288AH

Esta rutina duplica un mantissa doble de precisión tres veces sucesivas para producir la productos $X \times 2$, $X \times 4$ y $X \times 8$. La dirección de byte menos significativo de mantissa es suministrada en par de registro Delaware. Los productos se guardan en direcciones sucesivamente inferiores comenzando inmediatamente debajo del operand.

La dirección ... 289FH

Esta rutina es usada por el Evaluatodor Expression para dividir dos operands dobles de precisión. El primer operand es contenido en DAC y el segundo en ARG, el resultado es devuelto en DAC. Si el primer operand es ponga en el cero, entonces la rutina termina con un resultado de cero si el segundo operand es cero una "división por el cero" el error es generado (4058H). De otra manera los dos exponentes son sustraídos para producir el exponente de resultado y los dos signos del mantissa procesados para producir el signo del resultado. Si son lo mismo el resultado es positivo, entonces si difieren es negativo.

El mantissae está dividido usando el método largo normal de división. El segundo operand es repetidamente sustraído desde el comienzo hasta el poco flujo para producir un solo dígito del resultado.

El segundo operand se agrega luego de regreso a restituir el resto (2761H), el dígito es almacenado en el AGARRE y el primer operand es intercambiado izquierda de dígito. Cuando el primer operand ha estado completamente intercambiado apagado el resultado es emulado de AGARRE para DAC entonces renormalizado y redondeado levantado (2883H). El tiempo requerido pues una división se maximiza de aproximadamente 25 ms cuando el primer operand es compuesto mayormente de nines y el segundo operand de mismos. Esto requerirá el número máximo de sustracciones.

La dirección ... 2993H

Esta rutina es usada por el Evaluatodor Factor para ejercer los "Colorados" función para un operand de precisión doble contenido en DAC.

El operand es primero multiplicado (2C3BH) a las $1 / (2 \times \text{la PI})$ a fin de que la unidad es propia de un completo ciclo de 360 grados. El operand luego tiene 0.25 (90 grados) sustraídos (2C32H), su signo del mantissa está invertido (2E8DH) y control cae en el "pecado" rutina.

La dirección ... 29ACH

Esta rutina es usada por el Evaluatodor Factor para ejercer el "pecado" función para un operand de precisión doble contenido en DAC.
El operand es primero multiplicado (2C3BH) a las $1 / (2 * \text{la PI})$ a fin de que la unidad es propia de un completo ciclo de 360 grados. Como la función es periódica sólo la parte fraccionada del operand es ahora requerido. Esto es extraído empujando al operand (2CCCH) obteniendo la parte de entero (30CFH) y copiándola para ARG (2C4DH), haciendo estallar el operand entero para DAC (2CE1H) y luego sustrayendo la parte de entero (268CH).

- 92 -

5. EL INTÉRPRETE DE BASIC ROM

El primer dígito del mantissa es entonces examinado para determinar el cuadrante de operand. Si está en el primer cuadrante, entonces es igual. Si está en el segundo cuadrante, entonces es sustraído de 0.5 (180 grados) para reflejarlo acerca del Eje Vertical. Si está en el tercer cuadrante, entonces es sustraído de 0.5 (180 grados) para reflejarlo acerca del axis de la X. Si está en el cuarto cuadrante 1.0 (360 grados) es sustraído para reflejarlo acerca de ambos ejes. La función es entonces computada por la aproximación polinómica (2C88H) usando la lista de coeficientes en 2DEFH.
¡Éstas son las primeras ocho condiciones en la X-(de serie Taylor $X^{3/3} + (X^{5/5}) - (X^{7/7} !)$) ... con los coeficientes multiplicado por los factores sucesivos de $2 * \text{PI}$ para compensar el sarro inicial.

La dirección ... 29FBH

Esta rutina es usada por el Evaluatodor Factor para ejercer la función "color café claro" para un operand de precisión doble contenido en DAC.
La función es computada usando el = PECADO trigonométrico (X) de BRONCEADO de identidad (X) / Colorados (X).

La dirección ... 2A14H

Esta rutina es usada por el Evaluatodor Factor para ejercer al "ATN" función para un operand de precisión doble contenido en DAC.
La función es computada por la aproximación polinómica (2C88H) usando la lista de coeficientes en 2E30H. Éstas son las primeras ocho condiciones en la X-(de serie Taylor $X^{5/5}$) - ... con los coeficientes modificados ligeramente para resumir la serie $(x^{3/3}) (X^{7/7})$.

La dirección ... 2A72H

Esta rutina es usada por el Evaluatodor Factor para ejercer el "leño" función para un operand de precisión doble contenido en DAC.
La función es computada por la aproximación polinómica usando la lista de coeficientes en 2DA5H.

La dirección ... 2AFFH

Esta rutina es usada por el Evaluatodor Factor para ejercer al "SQR" función para un operand de precisión doble contenido en DAC.
La función es computada usando el proceso de Newton-Raphson, un programa equivalente de BASIC es:

```

10 INPUT"NUMBER ";X
20 ADIVINAN = 10
30 FOR N = 1 TO 7
40 ADIVINAN A = (GUESS+X/GUESS) /2
50 SIGUIENTE N
60 ESCRIBEN EN LETRAS DE IMPRENTA ADIVINANZA
70 PRINT SQR (X)

```

El citado anteriormente programa usa una adivinanza inicial fija. Mientras esto es

- 93 -

5. EL INTÉRPRETE DE BASIC ROM

Precisa sobre un máximo limitado de rango la exactitud sólo será lograda si la adivinanza inicial está junta a la raíz. El método usado por el ROM es disminuir a la mitad el exponente, con redondeo arriba, y luego dividir los primeros dos dígitos del operand a las cuatro e incrementar el primer dígito.

La dirección ... 2B4AH

Esta rutina es usada por el Evaluador Factor para ejercer al "EXP" función para un operand de precisión doble contenido en DAC.

El operand es primero multiplicado por 0.4342944819, lo cual es LEÑO (e) para Base 10, a fin de que el problema se convierte en computar 10^X en vez del e^X . Esto que los resultados en la simplificación considerable como el entero la parte pueden ser con los que se trató fácilmente.

La función es entonces computada por la aproximación polinómica usando la lista de coeficientes en 2D6BH.

La dirección ... 2BDFH

Esta rutina es usada por el Evaluador Factor para ejercer al "RND" función para un operand de precisión doble contenido en DAC.

Si el operand es ponga en el cero el número aleatorio actual es copiado para DAC de RNDX y la rutina termina. Si el operand es negativo, entonces es copiado para RNDX para colocar el número aleatorio actual.

El número aleatorio nuevo se produce copiando a RNDX a HACER, la constante en 2CF9H cumplir ARG, la constante en 2CF1H para DAC y luego multiplicando (282EH). Los catorce dígitos menos significativos del producto doble de largo son emulados para RNDX para formar el mantissa del número aleatorio nuevo. El byte del exponente en DAC es determinado para 10^0 para devolver un valor en el rango 0 a 1.

La dirección ... 2C24H

Esta rutina es usada por los manipuladores "nuevos", "cristalinos" y "andados" de declaración para inicializar a RNDX con la constante en 2D01H.

La dirección ... 2C2CH

Esta rutina añade la constante a cuya dirección está suministrada en par de registro HL para el operand doble de precisión contuvo en DAC.

La dirección ... 2C32H

Esta rutina sustrae la constante a cuya dirección está suministrada en par de registro HL del

operand doble de precisión contuvo en DAC.

La dirección ... 2C3BH

Esta rutina multiplica el operand de precisión doble contenido en DAC por la constante a cuya dirección está suministrado en par de registro HL.

- 94 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 2C41H

Esta rutina divide el operand de precisión doble contenido en DAC por la constante a cuya dirección está suministrado en par de registro HL.

La dirección ... 2C47H

Esta rutina realiza la operación de relación en el operand doble de precisión contenida en DAC y la constante a cuya dirección está suministrada en par de registro HL.

La dirección ... 2C4DH

Esta rutina copia un operand de precisión de ocho bytes de doble de DAC para ARG.

La dirección ... 2C59H

Esta rutina copia un operand de precisión de ocho bytes de doble de ARG para DAC.

La dirección ... 2C6FH

Esta rutina intercambia los ocho bytes en DAC con los ocho bytes actualmente en la raíz de la pila Z80.

La dirección ... 2C80H

Esta rutina invierte el signo del mantissa del operand contenido en DAC (2E8DH). La misma dirección es entonces empujada encima de la pila para restituir el signo cuando la persona que llama termina.

La dirección ... 2C88H

Esta rutina genera una serie extraña basada en el operand de precisión doble contenido en DAC. La serie es de la formalidad:

$$X^{1 * (Kn)} + X^{3 * (Kn-1)} + X^{5 * (Kn-2)} + X^{5 * (Kn-3)}$$

La dirección de la lista de coeficiente es suministrada en par de registro HL. El primer byte de la lista contiene la cuenta de coeficiente, los coeficientes dobles de precisión siguen con K1 primero y Kn de último. La serie pareja es generada (2C9AH) y multiplicada (27E6H) por el operand original.

La dirección ... 2C9AH

Esta rutina genera una serie pareja basada en el operand de precisión doble contenido en

DAC. La serie es de la formalidad:

$$X^0 * (Kn) x^2 * (Kn-1) x^4 * (Kn-2) x^6 * (Kn-3)$$

La dirección de la lista de coeficiente es suministrada en registro

- 95 -

5. EL INTÉRPRETE DE BASIC ROM

El par HL. El primer byte de la lista contiene la cuenta de coeficiente, los coeficientes dobles de precisión siguen con K1 primero y Kn de último. El método usado para computar el polinomio es conocido como el método de Horner. Sólo precisa una multiplicación y una adición por el término, el equivalente de BASIC es:

```
10 X=X*X
20 EL PRODUCTO = 0
30 RESTITUYEN 100
40 LEEN LA CUENTA
50 FOR N = 1 PARA CONTAR
60 LEEN LA K
70 EL PRODUCTO = (PRODUCT*X) + K
80 SIGUIENTE N
90 ACABAN
100 DATOS 8
Kn-7 de 110 DATOS
Kn-6 de 120 DATOS
Kn-5 de 130 DATOS
Kn-4 de 140 DATOS
Kn-3 de 150 DATOS
Kn-2 de 160 DATOS
Kn-1 de 170 DATOS
Kn de 180 DATOS
```

El polinomio es tramitado del coeficiente final directo para el primer coeficiente a fin de que el producto parcial puede usarse para ahorrar operaciones innecesarias.

La dirección ... 2CC7H

Estos empujones de rutina un operand de precisión de ocho bytes de doble de ARG encima del Z80 se apilan bien.

La dirección ... 2CCCH

Estos empujones de rutina un operand de precisión de ocho bytes de doble de DAC encima del Z80 se apilan bien.

La dirección ... 2CDCH

Estas gaseosas de rutina un operand de precisión de ocho bytes de doble del Z80 se apilan bien en ARG.

La dirección ... 2CE1H

Estas gaseosas de rutina un operand de precisión de ocho bytes de doble del Z80 se apilan bien en DAC.

La dirección ... 2CF1H

Esta mesa contiene la precisión doble que las constantes usaron por las rutinas de matemáticas. Las primeras tres constantes tienen cero en la colocación del exponente en su estado actual en un especial forma intermedia

- 96 -

5. EL INTÉRPRETE DE BASIC ROM

Usado por el generador aleatorio de número.

ADDR. ADDR PERSEVERANTE. La CONSTANTE

2CF1H .14389820420821 RND 2DAEH 6.2503651127908
2CF9H .21132486540519 RND 2DB6H - 13.682370241503
2D01H .40649651372358 2DBEH 8.5167319872389
2D09H .43429448190324 PONGA EN BITÁCORA a (e) 2DC6H 5 LEÑO
2D11H .50000000000000 2DC7H 1.00000000000000
2D13H .00000000000000 2DCFH - 13.210478350156
2D1BH 1.00000000000000 2DD7H 47.925256043873
2D23H .25000000000000 2DDFH - 64.906682740943
2D2BH 3.1622776601684 SQR (10) 2DE7H 29.415750172323
2D33H .86858896380650 de 8 PECADO de 2 ^ LOGARITMOS (e) de 2DEFH
2D3BH 2.3025850929940 2DF0H de 1 / LEÑOS (e) - .69215692291809
2D43H 1.5707963267949 PI/2 2DF8H 3.8172886385771
2D4BH .26794919243112 BRONCEE a (PI/12) 2E00H - 15.094499474801
2D53H que 1.7320508075689 BRONCEAN (PI/3) 2E08H 42.058689667355
2D5BH .52359877559830 PI/6 2E10H - 76.705859683291
2D63H .15915494309190 1 / (2 ^ la PI) 2E18H 81.605249275513
2D6BH 4 EXP 2E20H - 41.341702240398
2D6CH 1.00000000000000 2E28H 6.2831853071796
2D74H 159.37415236031 2E30H 8 ATN
2D7CH 2709.3169408516 2E31H - .05208693904000
2D84H 4497.6335574058 2E39H .07530714913480
2D8CH 3 EXP 2E41H - .09081343224705
2D8DH 18.312360159275 2E49H .11110794184029
2D95H 831.40672129371 2E51H - .14285708554884
2D9DH 5178.0919915162 2E59H .19999999948967
2DA5H 4 el LEÑO 2E61H - .3333333333160
2DA6H - .71433382153226 2E69H 1.00000000000000

La dirección ... 2E71H

Esta rutina devuelve el signo del mantissa de un operand Point Flotante contenido en DAC. El byte del exponente es probado y el resultado regresó en A de registro y las banderas:

El cero..... Un = 00H, Flag Z, Carolina Del Norte Positive ... UN = 01H, Flag NZ, Carolina Del Norte Negative ... A=FFH, Flag NZ, C

La dirección ... 2E7DH

Esta rutina simplemente pone en el cero el byte del exponente en DAC.

La dirección ... 2E82H

Esta rutina es usada por el Evaluador Factor para ejercer el "sistema de frenos ABS" función para un operand contenido en DAC. El signo de operand's es primero a cuadros (2EA1H), si es positivo que la rutina simplemente termine. El tipo de operand es entonces a cuadros por la rutina del estándar GETYPR. Si es una cuerda un "bodorrio de tipo" error es generada (406DH). Si es un entero, entonces es negado

- 97 -

5. EL INTÉRPRETE DE BASIC ROM

(322BH). Si es una precisión doble o singularice, entonces el operand de precisión que el signo del mantissa mordió en DAC está invertido.

La dirección ... 2E97H

Esta rutina es usada por el Evaluador Factor para ejercer al "SGN" función para un operand contenido en DAC. Los operand's en los que el signo es comprobado (2EA1H), extendido registran par HL y luego colocados en DAC como un entero:

El cero..... El 0000H
El positivo ... 0001H
La negativa ... FFFFH

La dirección ... 2EA1H

Esta rutina devuelve el signo de un operand contenido en DAC. El tipo de operands es primero a cuadros por la rutina del estándar GETYPR. Si es una cuerda un "bodorrio de tipo" error es generada (406DH). Si es una sola precisión o dóblese, entonces el operand de precisión el signo del mantissa es examinado (2E71H). Si es un entero su valor es tomado de DAC + 2 y traducidos en las banderas mostrados en 2E71H.

La dirección ... 2EB1H

Estos empujones de rutina un operand de precisión de cuatro bytes de solo de DAC encima del Z80 se apilan bien.

La dirección ... 2EC1H

Esta rutina emula los contenidos de registro C, B, E y D para DAC.

La dirección ... 2ECCH

Esta rutina emula los contenidos de DAC para registro C, B, E y D.

La dirección ... 2ED6H

Estos registros de cargas de rutina C, B, E y D de posiciones ascendentemente secuenciales comenzando en la dirección abastecida en registro porean a HL.

La dirección ... 2EDFH

Estos registros de cargas de rutina E, D, C y B de posiciones ascendentemente secuenciales comenzando en la dirección abastecida en registroorean a HL.

La dirección ... 2EE8H

Esta rutina copia un solo operand de precisión de DAC para

- 98 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección abastecida en par de registro HL.

La dirección ... 2EEFH

Esta rutina copia cualquier operand de la dirección abastecida en par de registro HL para ARG. El largo del operand es contenido en VALTYP: 2 = Integer, 3 = String, 4 = El Solo Precision, 8 = El Doble Precision.

La dirección ... 2F05H

Esta rutina copia cualquier operand de ARG para DAC. El largo del operand es contenido en VALTYP: 2 = Integer, 3 = String, 4 = El Solo Precision, 8 = El Doble Precision.

La dirección ... 2F0DH

Esta rutina copia cualquier operand de DAC para ARG. El largo del operand es contenido en VALTYP: 2 = Integer, 3 = String, 4 = El Solo Precision, 8 = El Doble Precision.

La dirección ... 2F21H

Esta rutina es usada por el Evaluador Expression para encontrar la relación ($<$ $>$ $=$) entre dos solos operands de precisión. La primera parte en la que el operand es contenido registra a C, B, E y D y el segundo en DAC. El resultado es devuelto en A de registro y las banderas:

Operand 1 = Operand 2 ... UN = 00H, Flag Z, Carolina Del Norte
Operand 1 < Operand 2 ... UN = 01H, Flag NZ, Carolina Del Norte
Operand 1 > Operand 2 ... A=FFH, Flag NZ, C

Debería ser notable que para operadores del relational que la negativa del maximally de mis saludos afectuosos del Evaluador Expression numera como números pequeños y positivos maximally tan grandes.

La dirección ... 2F4DH

Esta rutina es usada por el Evaluador Expression para encontrar la relación ($<$ $>$ $=$) entre dos operands de entero. La primera parte que el operand es contenido en par de registro Delaware y el segundo en registroorea a HL. Los resultados están por lo que respecta a la sola versión de precisión (2F21H).

La dirección ... 2F83H

Esta rutina es usada por el Evaluador Expression para encontrar la relación ($<$ $>$ $=$) entre dos operands dobles de precisión. El primer operand es contenido en DAC y el segundo en ARG. Los resultados están por lo que respecta a la sola versión de precisión (2F21H).

La dirección ... 2F8AH

Esta rutina es usada por el Evaluador Factor a aplicar lo

- 99 -

5. EL INTÉRPRETE DE BASIC ROM

"CINT" funcione para un operand contenido en DAC. El tipo del operand es primero a cuadros por la rutina del estándar GETYPR, si es ya entero que la rutina simplemente termina. Si es una cuerda un "bodorrio de tipo" error es generada (406DH). Si es una sola precisión u operand doble de precisión ella es convertido a un entero binario indicado a través de señas en par de registro Delaware (305DH) y luego colocado en DAC como un entero. Fuera de alcance aprecia el resultado en un "Overflow" error (4067H).

La dirección ... 2FA2H

Esta rutina inspecciona si DAC contiene la sola precisión operand - 32768, si así es que ella la reemplaza con el 8000H de equivalente de entero. Este paso es precisado durante la conversión numérica (3299H) de entrada por el rango asimétrico de número entero.

La dirección ... 2FB2H

Esta rutina es usada por el Evaluador Factor para ejercer al "CSNG" función para un operand contenido en DAC. El tipo de operand's es primero a cuadros por la rutina del estándar GETYPR, si es ya sola precisión que la rutina simplemente termina. Si es una cuerda un "bodorrio de tipo" error es generada (406DH). Si es doble, entonces la precisión VALTYP se varíe (3053H) y el mantissa redondeado arriba del séptimo dígito (2741H). Si el operand es un entero, entonces se convierte de binario para un máximo de cinco dígitos BCD por divisiones sucesivas usando las constantes 10000, 1000, 100, 10, 1. Estos son colocados en DAC para formar el solo mantissa de precisión. El exponente es igual al número de dígitos significativos en el mantissa. Por ejemplo si hay cinco, entonces el exponente sería 10^5 .

La dirección ... 3030H

Esta mesa contiene las cinco constantes usadas por el "CSNG" rutina: -- 1000, -- 10, - 10000, 100, 1

La dirección ... 303AH

Esta rutina es usada por el Evaluador Factor para ejercer al "CDBL" función para un operand contenido en DAC. El tipo de operand's es primero a cuadros por la rutina del estándar GETYPR, si es ya precisión doble que la rutina simplemente termina. Si es una cuerda un "bodorrio de tipo" error es generada (406DH). Si es un entero él se convierte primero a singularizar precisión (2FC8H), los ocho dígitos menos significativos son entonces puestos en el cero y VALTYP se sedimentó para 8.

La dirección ... 3058H

Esta rutina inspecciona que el operand en uso en estos momentos es un tipo de la cuerda, en caso de que no un "bodorrio de tipo" error es generado (406DH).

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 305DH

Esta rutina es usada por el "CINT" rutina (2F8AH) para convertir una precisión del soltero BCD o el operand doble de precisión en un entero binario indicado a través de señas en par de registro Delaware, devuelve C Flag si un excedente ha ocurrido. Los dígitos sucesivos son tomados del mantissa y se suman al producto echando a andar con lo más significativo. Después de cada adición que el producto está multiplicado por diez. El número de dígitos a proceso es determinado por el exponente, por ejemplo cinco dígitos estarían prendados de un exponente de 10^5 . Finalmente el signo del mantissa es comprobado y el producto negado (3221H) si es necesario.

La dirección ... 30BEH

Esta rutina es usada por el Evaluatodor Factor para ejercer el "apuro" función para un operand contenido en DAC. El tipo de operand's es primero a cuadros por la rutina del estándar GETYPR, si es un entero que la rutina simplemente termina. El signo del mantissa es entonces a cuadros (2E71H), si es positivo que el control transfiere para el "INT" rutina (30CFH). De otra manera el signo está invertido para positivo, el "INT" función es realizado (30CFH) y el signo recuperado para negar.

La dirección ... 30CFH

Esta rutina es usada por el Evaluatodor Factor para ejercer al "INT" función para un operand contenido en DAC. El tipo de operand's es primero a cuadros por la rutina del estándar GETYPR, si es un entero que la rutina simplemente termina. El número de dígitos fraccionados está resuelto sustrayéndole al exponente de la cuenta de dígito del tipo, 6 para la sola precisión, 14 para la precisión doble.

Si el signo del mantissa es positivo, entonces estos dígitos fraccionados son simplemente puestos en el cero. Si el signo del mantissa es negativo, entonces cada dígito fraccionado es examinado antes de que sea puesto en el cero. Si todos los dígitos fueron previamente ponga en el cero, entonces la rutina simplemente termina. De otra manera - 1.0 se agregan para el operand por la sola rutina de adición de precisión (324EH) o la rutina doble (269AH) de adición de precisión. Debería ser notable que el tipo de operand no se varía normalmente por el "CINT" función.

La dirección ... 314AH

Esta rutina multiplica los enteros binarios sin firmar en pares de registro BC y Delaware, el resultado es devuelto en par de registro Delaware. El cambio estándar y se añade que el método es usado, el producto lo hace sucesivamente multiplicado a las dos y el registro el par AC se sumó a él para cada 1 bit en par de registro Delaware. La rutina es usada por la rutina Variable (5EA4H) de búsqueda para computar una colocación de element's dentro de un Array, un error "fuera del alcance escrito debajo de una letra" es generado (601DH) si un excedente ocurre.

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3167H

Esta rutina es usada por el Evaluador Expression para sustraer dos operands de entero. La primera parte que el operand es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. El segundo operand es negado (3221H) y el control cae en la rutina de adición.

La dirección ... 3172H

Esta rutina es usada por el Evaluador Expression para añadir dos operands de entero. La primera parte que el operand es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. Los operands binarios firmados están normalmente justamente añadidos y colocados en DAC. Sin embargo, si un excedente ha ocurrido, entonces ambos operands se convierten para singularizar transferencias de precisión (2FCBH) y de control para el solo sumador de precisión (324EH). Un excedente ha ocurrido cuando ambos operands son del mismo signo y el resultado lo es del signo opuesto, por ejemplo:

$$30000+15000 = - 20536$$

La dirección ... 3193H

Esta rutina es usada por el Evaluador Expression para multiplicar dos operands de entero. La primera parte que el operand es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. Los dos signos del operand son salvados por ahora y ambos operands hicieron positivo (3215H). Los ingresos de multiplicación usando el cambio binario estándar y añade método con par de registro HL como el acumulador del producto, par de registro BC conteniendo el primer operand y el par de registro Delaware el segundo. Si el producto excede 7FFFH en cualquier momento durante la multiplicación ambos operands se convierte para singularizar precisión (2FCBH) y controle transferencias para el solo multiplicador de precisión (325CH). De otra manera los signos iniciales están recuperados y, si difirieran, el producto negado antes de que siendo colocado en DAC como un entero (321DH).

La dirección ... 31E6H

Esta rutina es usada por el Evaluador Expression para el entero divida (\) dos operands de entero. La primera parte que el operand es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. Si el segundo operand es cero que una "división por" el error de "cero" es generado (4058H), entonces de otra manera los dos signos del operand son salvados y ambos operands hicieron positivo (3215H). Los ingresos de división usando el cambio binario estándar y sustrae método con par de registro HL conteniendo el resto, par de registro BC que el segundo operand y registro arregla en pares Delaware el primer operand y el producto. Cuando la división es completa los signos iniciales están recuperados y, si difirieran, el producto es negado antes de que siendo colocado en DAC como un entero (321DH).

- 102 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3215H

Esta rutina se usa para hacer dos enteros binarios indicados a través de señas, en pares de

registro HL y Delaware, positivo. Ambos los signos iniciales del operand son devueltos como una bandera adentro bit 7 de B de registro: 0 = De La Misma Forma, 1 = Different. Cada operand es entonces examinado y, si es negativo, positivo hecho sustrayéndolo de cero.

La dirección ... 322BH

Esta rutina es usada por el "sistema de frenos ABS" función para hacer un entero negativo contenida en positivo DAC. El operand es tomado de DAC, negado y luego recoloca en DAC (3221H). Si el valor de operand es 8000H, entonces se convierte para singularizar precisión (2FCCH) como no hay entero de valor + 32768.

La dirección ... 323AH

Esta rutina es usada por el Evaluador Expression para operands dos de entero "a la última". La primera parte que el operand es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. El signo del primer operand es salvado y los dos operands divididos (31E6H). Como el resto es devuelto doblado por el par de registro de proceso de división Delaware es intercambió un derecho del lugar para restituirlo. El signo del primer operand está entonces recuperado y, si es negativo, que lo demás sea negado antes de que siendo colocado en DAC como un entero (321DH).

La dirección ... 324EH

Esta rutina es usada por el Evaluador Expression para añadir dos solos operands de precisión. La primera parte en la que el operand es contenido registra a C, B, E, D y el segundo en DAC, el resultado es devuelto en DAC. El primer operand es copiado para ARG (3280H), el segundo operand se convierte para duplicar transferencias de precisión (3042H) y de control para el sumador doble (269AH) de precisión.

La dirección ... 3257H

Esta rutina es usada por el Evaluador Expression para sustraer dos solos operands de precisión. La primera parte en la que el operand es contenido registra a C, B, E, D y el segundo en DAC, el resultado es devuelto en DAC. El segundo operand es negado (2E8DH) y el control se traslada para el solo sumador de precisión (324EH).

La dirección ... 325CH

Esta rutina es usada por el Evaluador Expression para multiplicar dos solos operands de precisión. La primera parte en la que el operand es contenido registra a C, B, E, D y el segundo en DAC, el resultado es devuelto en DAC. El primer operand es copiado para ARG (3280H), el segundo operand se convierte para duplicar transferencias de precisión (3042H) y de control para el multiplicador doble (27E6H) de precisión.

- 103 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3265H

Esta rutina es usada por el Evaluador Expression para dividir dos solos operands de precisión. La primera parte en la que el operand es contenido registra a C, B, E, D y el segundo en DAC, el resultado es devuelto en DAC. La primera parte y segundos operands son intercambiados a fin de que la primera parte está en DAC y el segundo en los registros. El segundo operand es entonces copiado para ARG (3280H), el primer operand se convierte para

duplicar transferencias de precisión (3042H) y de control para el divisor doble (289FH) de precisión.

La dirección ... 3280H

Esta rutina copia el solo operand de precisión contenido en registro C, B, E y D para ARG y luego pone en el cero los cuatro bytes menos significativos.

La dirección ... 3299H

Esta rutina convierte un número en forma textual para uno de los tipos numéricos internos estándar, está usado durante la tokenización y por el "VAL", "la ENTRADA" y los manipuladores "leídos" de Declaración. En las proposiciones del par de registro de entrada HL para el primer carácter de la cuerda del texto a ser convertido. En par de la registro de la salida HL apunta hacia el carácter siguiendo la cuerda, el operand numérico está en DAC y el código de tipo en VALTYP. Los ejemplos de los tres tipos son:

```
+
| | | FFH | 7FH
+
El Entero + 32767

+
| 42H 17H 39H 04H
+
Singularice .173904*10^2 De Precisión

+
| C2H 17H 39H 04H 62H 70H 93H 13H
+
Duplica Precisión - .17390462709313*10^2
```

Figura 41: El numérico Types en DAC

Un entero es un número de dieciséis ratos de binario en forma de complemento de two's, es LSB almacenado primero, segundo MSB en DAC + 2.

Un entero puede extenderse de 8000H (- 32768) para 7FFFH (+ 32767).

Un punto flotante número consta de un byte del exponente y un mantissa tres o de siete bytes. El exponente es guardado en forma binaria firmada y puede alinear en orden a de 01H (- 63) a través de 40H (0) a 7FH (+ 63), el valor especial de 00H sirve para el cero de número. Estos valores del exponente son para un mantissa normalizado. El Interpreter presenta números de la forma de exponente para el usuario con uno

- 104 -

5. EL INTÉRPRETE DE BASIC ROM

El dígito principal, esto resulta en un rango asimétrico del exponente de E-64 para E + 62. El bit 7 del byte del exponente mantiene el signo del mantissa, 0 para positivo y 1 para la negativa, el mantissa mismo es guardado en forma apiñada BCD con dos dígitos por el byte. Debería ser notable que el Interpreter usa los contenidos de VALTYP para determinar el tipo de un número, no el formato del número mismo.

La conversión comienza examinando el primer símbolo del texto. Si éste es uno "y" control se

traslada para la rutina especial (4EB8H) de conversión del radix, entonces si es un carácter principal del signo es por ahora ahorrado. Los símbolos numéricos sucesivos son luego tomados y añadidos para el producto de entero con multiplicaciones apropiadas a las diez como cada dígito nuevo es encontrado. Si el valor del producto excede 32767, o un punto decimal es encontrado, entonces el producto se convierte para singularizar precisión y cualquier más personajes acomodadas directamente en DAC. Si un séptimo dígito es encontrado, entonces el producto se varía para duplicar precisión, si más que catorce dígitos son encontrados que los dígitos excedentes son leídos pero ignorados.

La conversión cesa cuando un carácter poco numérico es encontrado. Si esto un carácter de definición de tipo ("el %", o "!") la rutina apropiada de conversión se llama y que transferencias de control para el punto de la salida (331EH). Si es un prefijo del exponente ("E", "e", "D" o "d") una de las rutinas de conversión también será usada y luego los siguientes dígitos mutaron para un exponente binario en E de registro. En el punto de la salida (331EH) el tipo de product's es comprobado por la rutina del estándar GETYPR. Si está soltera la precisión o duplique precisión, entonces el exponente se calcula por primero sustraer la cuenta fraccionada de dígito, en registro B, de la cuenta total de dígito, adentro se registra D, para producir la cuenta delantera de dígito. Esto se agrega luego a cualquier exponente explícitamente declarado, adentro registra E, y colocado en DAC + 0 como el exponente.

El carácter principal del signo está recuperado y el producto negado si es requerido (2E86H), si el producto es entero que la rutina luego termina. Si el producto está soltera el control de precisión termina revisando en busca del valor especial de - 32768 (2FA2H). Si el producto es al doble, entonces el control de precisión termina redondeando arriba del decimoquinto dígito (273CH).

La dirección ... 340AH

Esta rutina es usada por el manipulador de error para exteriorizar el mensaje "de adentro" (6678H) seguido por el número de la línea abastecido en par de registro HL (3412H).

La dirección ... 3412H

Estos despliegues de rutina en los que el entero binario sin firmar suministró registran par HL. El operand es colocado en DAC como un entero (2F99H), convertido para texto (3441H) y luego exhibido (6677H).

- 105 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3425H

Esta rutina convierte al operand numérico contenido en DAC a la forma textual en FBUFFR. La dirección del primer carácter del texto resultante es devuelta en par de registro HL, el texto está terminado por un byte de cero. El operand se convierte primero a duplicar precisión (375FH). Los dígitos BCD del mantissa son entonces desempacados, convertidos para el ASCII y están posado en FBUFFR (36B3H).

La colocación del punto decimal es determinada por el exponente, por ejemplo:

$$\begin{aligned} .999 * 10 ^ + 2 &= 99.9 \\ .999 * 10 ^ + 1 &= 9.99 \\ .999 * 10 ^ + 0 &= .999 \\ .999 * 10 ^ - 1 &= .0999 \end{aligned}$$

Si el exponente está fuera del rango 10^{-1} para 10^{14} el número se replantea en forma exponencial. En este caso el punto decimal está posado después del primer dígito y el exponente se convierte de binario y sigue el mantissa.

Un punto alternativo de entrada para la rutina existe en 3426H para la "impresión USANDO" manipulador de declaración. Con este punto de entrada el número de personajes para prefijar al punto decimal le es suministrado en registro B, el número de personajes para apuntar apuro él en C de registro y un byte del formato en A de registro:

```

    7 6 5 4 3 2 1 0
    +
    | 1 |, $ Sign | 0
    +

```

Figura 42: El formato Byte

La operación en este modo es medianamente similar al modo normal pero con la adición de instalaciones adicionales. Una vez que el operand se ha convertido para duplicar precisión la forma exponencial estará asumido si fue mordido 0 del byte del formato está listo. El mantissa es intercambiado a la derecha en DAC y redondeado arriba para perder dígitos no deseados (377BH) de postfijo. Como el mantissa es convertido a las comas de ASCII (36B3H) estarán insertas en las proposiciones apropiadas si son mordidas 6 del byte del formato está listo. Durante la postconversión formateando (351CH) colocaciones de prefijo sin uso se llenará de asteriscos si es mordido 5 está listo, un prefijo de libra puede agregarse colocando bit 4. El bit 3 posibilita el signo para números positivos si es colocado, de otra manera un espacio es usado. Mordido de 2 posiciones sociales de cualquier firman en el frente si reanudan y al fondo si se sedimentan.

El punto de entrada para la rutina en 3441H se usa para convertir enteros sin firmar, notablemente revestir números, para su forma textual. Para ejemplo 9000H, estando tratado como un entero normal, se convirtió a - 28672. Usando este punto de entrada 36864 sería producido en lugar de eso. El operand se convierte por ahí sucesivo

- 106 -

5. EL INTÉRPRETE DE BASIC ROM

La división con los factores 10000, 1000, 100, 10 y 1 y los dígitos resultantes colocaron en FBUFFR (36DBH).

La dirección ... 3710H

Esta mesa contiene las cinco constantes usadas por la rutina numérica de salida: 10000, 1000, 100, 10, 1.

La dirección ... 371AH

Esta rutina es usada por el "N\$" función para convertir a un operand numérico contenido en DAC a la forma textual. La B de registro está cargada de las transferencias de tamaño en coro (1) y de control para la rutina general (3724H) de conversión.

La dirección ... 371EH

Esta rutina es usada por el "OCT\$" función para convertir a un operand numérico contenido en DAC a la forma textual. La B de registro está cargada de las transferencias de tamaño en coro (3) y de control para la rutina general (3724H) de conversión.

La dirección ... 3722H

Esta rutina es usada por el "HEX\$" función para convertir a un operand numérico contenido en DAC a la forma textual. La B de registro está cargada del tamaño en coro (4) y el operand mutó para un entero binario en par de registro HL (5439H). Los grupos sucesivos de 1, 3 o 4 añicos son rightwards intercambiados fuera del operand, convertidos para los dígitos de ASCII y están posado en FBUFFR. Cuándo el operand es todo pone en el cero la rutina termina con la dirección del primer carácter del texto en par de registro HL, la cuerda está terminada con un byte de cero.

La dirección ... 3752H

Esta rutina es usada durante la salida numérica para devolver la cuenta de dígito de operand en B de registro y la dirección de su byte menos significativo en par de registro HL. Para sola precisión B = 6 y HL=DAC + 3, para precisión doble B = 14 y HL=DAC + 7.

La dirección ... 375FH

Esta rutina es usada durante la salida numérica para convertir al operand numérico en DAC a duplicar precisión (303AH).

La dirección ... 377BH

Esta rutina es usada durante la salida numérica para intercambiar el mantissa en DAC rightwards (27DBH), lo inverso de la cuenta de dígito es abastecido en A de registro. El resultado está entonces redondeado levantado del decimoquinto dígito (2741H).

- 107 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 37A2H

Esta rutina es usada durante la salida numérica para devolver lo inverso de la cuenta fraccionada de dígito en un operand flotante del punto. Esto es computado sustrayendo el exponente de la cuenta de dígito de operand (6 o 14).

La dirección ... 37B4H

Esta rutina es usada durante la salida numérica para hallar el último dígito de poco cero del mantissa contenido en DAC. Su dirección es devuelta en par de registro HL.

La dirección ... 37C8H

Esta rutina es usada por el Evaluatodor Expression para exponentiate (^) dos operands de precisión del soltero. La primera parte en la que el operand es contenido registra a C, B, E, D y el segundo en DAC, el resultado es devuelto en DAC. El primer operand es copiado para ARG (3280H), empujado encima de la pila (2CC7H) e intercambiado con DAC (2C6FH). El segundo operand es entonces al que se hizo estallar en ARG y el control cae en la rutina doble de exponentiación de precisión.

La dirección ... 37D7H

Esta rutina es usada por el Evaluador Expression para exponentiate (^) dos operandos de precisión del doble. El primer operando es contenido en DAC y el segundo en ARG, el resultado es devuelto en DAC. El resultado es usualmente computado utilizando:

$$X^P = \text{EXP}(P * \text{LOG}(X))$$

Una alternativa, mucho más rápida, método es posible si el operando de poder es un entero. Esto es probado para extrayendo la parte de entero del operando y asemejándose para la igualdad con el valor original (391AH). Un resultado positivo para esta prueba quiere decir que el método más rápido puede ser usado, éste es descrito más abajo.

La dirección ... 383FH

Esta rutina es usada por el Evaluador Expression para exponentiate (^) dos operandos de entero. La primera parte que el operando es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. La rutina funciona estudiando en sus partes el problema en multiplicaciones simples:

$$6^{13} = 6^{1101} = (6^8) * (6^4) * (6^1)$$

Como el operando de poder está de formalidad binaria un cambio correcto simple es determinar si un producto intermedio particular necesita ser incluido en el resultado. Los productos intermedios mismos son obtenidos por la multiplicación acumulativa del operando cada vez que el lazo de computación es atravesado. Si

- 108 -

5. EL INTÉRPRETE DE BASIC ROM

El producto se derrama en a cualquier hora que se convierte para singularizar precisión. En la terminación el operando de poder es comprobado, si es negativo que el producto sea recíproco como $X^{-P} = 1/X^P$.

La dirección ... 390DH

Esta rutina es usada durante la exponentiación para multiplicar dos enteros (3193H), devuelve a Flag NZ si el resultado se ha derramado singularizar precisión.

La dirección ... 391AH

Esta rutina es usada durante la exponentiación para comprobar si un operando doble de poder de precisión consta sólo de una parte de entero, si así es que devuelve Flag Carolina Del Norte.

La dirección ... 392EH

Esta mesa de direcciones es usada por el Interpreter Runloop para encontrar al manipulador para una señal de declaración. Aunque no divide de la mesa las palabras claves asociadas son incluidas debajo:

PARA STMT PARA STMT PARA STMT

EL CÍRCULO del 5B11H de Lingüísticas Computacionales del 00C3H de FIN del 63EAH

EI 4524H PARA COLOR del 7980H de ANCHURA del 51C9H
EI 485DH del 6527H NEXT SI NO el EMPATE del 5D6EH
La PINTURA de DATOS del 485BH 6438H TRON 59C5H
La ENTRADA del 4B6CH el PIP 6439H TROFF 00C0H
EI 5E9FH el JUEGO del 73E5H de TRUEQUE del 643EH DIM
EI 4B9FH LEYÓ 6477H ERASE 57EAH PSET
EI 4880H DEJÓ el 57E5H de ERROR del 49AAH PRESET
47E8H SONIDO del 73CAH del CURRICULUM VITAE GOTO 495DH
EI 53E2H de CARRERA del 479EH la PANTALLA del 79CCH DELETE
EI AUTOMÓVIL del 49B5H de DECLARACIÓN IF del 49E5H 7BE2H VPOKE
EI 63C9H RESTAURA 5468H BLOQUE DE OBJETOS MOVIBLES RENUM 7A48H
47B2H GOSUB 4718H DEFSTR 7B37H VDP
EI REGRESO del 4821H la BASE 471BH DEFINT 7B5AH
485DH LLAMADA REM 471EH DEFSNG 55A8H
EI ALTO del 63E3H el TIEMPO 4721H DEFDBL 7911H
La LLAVE del 786CH de la LÍNEA del 4B0EH de la IMPRESIÓN del 4A24H
EI 7E4BH de CLARO del 6AB7H de CLARO del 64AFH MAX
EI MOTOR del 73B7H del CAMPO del 7C52H de la LISTA del 522EH
EI 775BH del 6286H NEW GET 6EC6H BLOAD
EI 48E4H EN 7758H PUSO a 6E92H BSAVE
EI ESPACIO CERRADO del 6C14H de ESPERA del 401CH 7C16H DSKO\$
EI 7C1BH de CARGA 501DH DEF 6B5DH SE SEDIMENTÓ
EI 6B5EH del PINCHAZO del 5423H el NOMBRE del 7C20H MERGE
6424H CONT 6C2FH ARCHIVA PRESA del 7C25H
6FB7H CSAVE 7C48H LSET 7C2AH IPL
703FH COPIA CLOAD 7C4DH RSET 7C2FH
EI 4016H FUERA DE 6BA3H SALVO 7C34H CMD

4A1DH LPRINT 6C2AH LFILES 7766H LOCATE
5229H LLIST

- 109 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 39DEH

Esta mesa de direcciones es usada por el Evaluador Factor para encontrar al manipulador para una señal de función. Aunque no divida de la mesa las palabras claves asociadas son incluidos con las direcciones exteriorizado debajo:

PARA FUNCIONAR FUNCIONAR FUNCIONAR

EI APURO del 30BEH DE PUNTO DE VENTA 6861H LEFT\$ 4FCCH
6891H RIGHT\$ 67FFH VARA LEN 7940H
La TRIGONOMETRÍA 689AH MID\$ 6604H TR\$ 794CH
2E97H SGN 68BBH VAL 795AH PDL
30CFH ALMOHADILLA INT 680BH ASC 7969H
EI SISTEMA DE FRENOS ABS del 2E82H 681BH CHR\$ 7C39H DSKF
EI ATISBO 2AFFH SQR 541CH 6D39H FPOS
2BDFH RND 7BF5H VPEEK 7C66H CVI
EI PECADO del 29ACH 6848H SPACE\$ 7C6BH CVS
EI LEÑO del 2A72H 7C70H OCT\$ 7C70H CVD
2B4AH EXP 65FAH HEX\$ 6D25H EOF
2993H Colorado 4FC7H LPOS 6D03H LOC

EI BRONCEADO del 29FBH 6FFFH N\$ 6D14H LOF
2A14H ATN 2F8AH CINT 7C57H MKI\$
69F2H FRE 2FB2H CSNG 7C5CH MKS\$
4001H INP 303AH CDBL 7C61H MKD\$

La dirección ... 3A3EH

Esta mesa de direcciones es usada durante la tokenización de programa como un índice en la mesa de palabra clave de BASIC (3A72H). Cada uno de las veintiseis entradas definen la dirección de puesta en marcha de uno de los subbloques de palabra clave. La primera entrada apunta hacia las palabras claves a partir de la carta "A", el segundo para esos a partir de la carta "B" etcétera.

3A72H ... UN 3B9FH ... J 3C8EH ... S
3A88H ... B 3BA0H ... K 3CDBH ... T
3A9FH ... C 3BA8H ... L 3CF6H ... U
3AF3H ... D 3BE8H ... M 3CFFH ... V
3B2EH ... E 3C09H ... N 3D16H ... W
3B4FH ... F 3C18H ... O 3D20H ... X
3B69H ... G 3C2BH ... P 3D24H ... Y
3B7BH ... H 3C5DH ... Q 3D25H ... Z
3B80H ... yo 3C5EH ... interrogación

La dirección ... 3A72H

Esta mesa contiene las palabras claves de BASIC y señales. Cada uno de los veintiseis bloques dentro de la mesa contienen todas las palabras claves a partir de una carta particular, es terminado con un byte de cero. Cada palabra clave es almacenada en el texto puro con el set 7 mordido para demarcar el último carácter, ésta es seguida inmediatamente por la señal asociada. El primer carácter de la palabra clave no necesita ser almacenado como esto es insinuado por su colocación en lo

- 110 -

5. EL INTÉRPRETE DE BASIC ROM

Posponga ' Las palabras claves y las señales están listadas debajo de en su totalidad, notan que la " J ", " Q ", " Y " y " Z " los bloques están vacíos:

La LISTA del AUTOMÓVIL A9H DSKF 26H 93H REM 8FH
Y el CURRÍCULUM VITAE de EMPATE F6H BEH LFILES BBH A7H
EI 06H del SISTEMA DE FRENOS ABS SI NO LEÑO A1H 0AH RSET B9H
EI FIN ATN 0EH 81H LOC 2CH RIGHT\$ 02H
ASC 15H ERASE A5H LEN 12H RND 08H
EI ERROR ATTR\$ E9H A6H LEFT\$ 01H RENUM AAH
BASE PANTALLA C9H ERL E1H LOF 2DH C5H
EI MOTOR BSAVE D0H ERR E2H CEH SPRITE C7H
EI 90H de ALTO BLOAD CFH EXP 0BH MERGE B6H
PIP C0H EOF 2BH FBH SWAP A LA ÚLTIMA A4H
N\$ 1DH EQV F9H MKI\$ 2EH COLOCÓ a D2H
La LLAMADA CAH FOR 82H MKS\$ 2FH SALVO
EI CAMPO CERCANO B4H B1H MKD\$ 30H SPC (DFH
COPIA D6H ARCHIVA a B7H MID\$ 03H STEP DCH
CONT 99H FN DEH TOPA CDH SGN 04H

DESCONGESTIONE a 92H FRE 0FH NEXT 83H SQR 07H
 EI 09H de PECADO de NOMBRE del 21H de APURO CLOAD 9BH D3H
 CSAVE 9AH FPOS 27H NEW 94H STR\$ 13H
 CSRLIN E8H GOTO 89H NO E0H STRING\$ E3H
 EI EMPUJE CINT 1EH PARA CLARO del 89H B0H SPACE\$ 19H
 CSNG 1FH GOSUB 8DH FUERA DEI SONIDO del 9CH C4H
 CDBL 20H GET B2H EN 22H de la VARA del 95H
 CVI 28H HEX\$ 1BH Oregon F7H STRIG 23H
 La ENTRADA CVS 29H 85H OCT\$ 1AH THEN DAH
 EI 8BH de DECLARACIÓN IF CVD 2AH FUERA DE EBH TRON A2H
 La IMPRESIÓN de Colorados 0CH INSTR E5H 91H TROFF A3H
 CHR\$ 16H INT 05H PUSO ETIQUETA B3H (DBH
 EI 98H del PINCHAZO del CÍRCULO BCH INP 10H PARA D9H
 EI 11H del DIABLILLO de COLOR BDH FAH POS TIME CBH
 EI 0DH de BRONCEADO del 17H de Lingüísticas Computacionales 9FH INKEY\$ ECH
 PEEK
 CMD D7H IPL D5H PSET C2H USANDO a E4H
 SUPRIMA PRESA A8H D4H PRESET C3H USR DDH
 EI PUNTO de la LLAVE del 84H de DATOS CCH EDH VAL 14H
 OSCUREZCA PINTURA 86H LPRINT 9DH BFH VARPTR E7H
 DEFSTR ABH LLIST 9EH PDL 24H VDP C8H
 La ALMOHADILLA DEFINT ACH LPOS 1CH 25H VPOKE C6H
 EI JUEGO del 88H DEFSNG ADH LET C1H VPEEK 18H
 La ANCHURA del 8EH de REGRESO DEFDBL AEH LOCATE D8H A0H
 DSKO\$ D1H REVISTA 96H de ESPERA del 87H AFH READ
 La CARRERA de CARGA DEF 97H B5H 8AH XOR F8H
 DSKI\$ EAH LSET B8H RESTITUYE 8CH

La dirección ... 3D26H

Este veintiuno byte la mesa es usado por el Interpreter durante la tokenización de programa. Contiene las diez solas palabras claves de carácter y sus señales:

+ ... F1H * ... F3H ^ ... F5H ' ... E6H = ... EFH - ... F2H / ... F4H \ ... FCH > ... EEH < ... F0H

- 111 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3D3BH

Esta mesa es usada por el Evaluador Expression para determinar el nivel de precedencia para un operador dado del infix, mientras más alto el valor de la mesa mayor la precedencia del operador. No incluido es las precedencias para los operadores del relational (64H), el "no" operador (5AH) y el operador de negación (7DH), estos están definidos directamente por el Expression y Evaluadores Factor.

79H ... + 46H ... Oregon
 79H ... - 3CH ... XOR
 7CH ... * 32H ... EQV
 7CH ... / 28H ... DIABLILLO
 7FH ... ^ 7AH ... MOD
 50H ... Y 7BH \

La dirección ... 3D47H

Esta mesa se usa para convertir al resultado de un usuario la función definida al mismo tipo como el Variable usado en la definición de función. Contiene las direcciones de las rutinas de conversión de tipo:

303AH ... CDBL
0000H ... No usado
2F8AH ... CINT
3058H ... Check coloquen en serie tipo
2FB2H ... CSNG

La dirección ... 3D51H

Esta mesa de direcciones es usada por el Evaluador Expression para encontrar al manipulador para un operador particular de matemáticas del infix cuando ambos operands son precisión del doble:

269AH
268CH ... -
27E6H
289FH ... /
37D7H
2F83H ... Relation

La dirección ... 3D5DH

Esta mesa de direcciones es usada por el Evaluador Expression para encontrar al manipulador para un operador particular de matemáticas del infix cuando ambos operands son precisión del soltero:

324EH
3257H ... -
325CH
3267H ... /
37C8H
2F21H ... Relation

- 112 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3D69H

Esta mesa de direcciones es usada por el Evaluador Expression para encontrar al manipulador para un operador particular de matemáticas del infix cuando ambos operands son entero:

3172H
3167H ... -
3193H
4DB8H ... /
383FH
2F4DH ... Relation

La dirección ... 3D75H

Esta mesa contiene los mensajes de error Interpreter, cada uno se guarda en el texto puro con un exterminador de byte de cero. Los códigos asociados de error son exteriorizados debajo para la referencia sólo, no forman en parte de la mesa:

01 DESPUÉS fuera PARA 19 Device yo el error de la / O
02 el error Syntax 20 el error Verify
03 DEVUELVEN SIN GOSUB 21 CURRICULUM VITAE NINGÚN
04 Fuera De los DATOS 22 REANUDAN sin error
05 la llamada Ilegal de función 23 el error Impublicable
06 Se Derraman 24 Perdiendo operand
07 Fuera De la memoria 25 Line amortiguador se derraman
08 la línea Indefinida número 50 el excedente del CAMPO
09 Escrito Debajo De Una Letra fuera de alcance 51 el error Interno
10 el conjunto imponente Redimensioned 52 el Mal número del archivo
11 Division de por ahí ponen en el cero 53 Archivo no encontrado
12 Ilegal dirigen 54 File ya claro
13 Escriben A Máquina bodorrio 55 fin pasado Input
14 Fuera De espacio de la cuerda 56 Bad archivan nombre
15 Ensartan demasiado tiempo 57 Directo declaración en archivo
16 Colocan En Serie fórmula Sequential demasiado 58 complicado yo / la O sólo
17 No Pueden Mantener 59 File cerrado
18 el usuario Indefinido funcionan

La dirección ... 3FD2H

Éste es el mensaje de texto puro "adentro" terminado por un byte de cero.

La dirección ... 3FD7H

Éste es el mensaje de texto puro "Oklahoma", RETORNO DE CARRO, LF ultimado por un byte de cero.

La dirección ... 3FDCH

Éste es el mensaje de texto puro "que Break" terminó por un byte de cero.

- 113 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 3FE2H

Esta rutina registra la pila Z80 para el "FOR" bloque de parámetro del lazo cuyo lazo la dirección Variable es suministrada en par de registro Delaware. La búsqueda es iniciada cuatro bytes por encima del Z80 actual SP para tener en cuenta la dirección del remitente de la persona que llama y la dirección del remitente Runloop. Si ningún "FOR" señal (82H) existe, entonces la rutina ultima a Flag NZ, si uno es encontrado que la dirección del lazo Variable es tomada del bloque de parámetro e inspeccionada. La rutina termina Z Flag en un encuentro exitoso con par de registro HL apuntando hacia el byte de tipo del bloque de parámetro. De otra manera los movimientos de búsqueda arriba de veintidos bytes para el principio del siguiente parámetro se bloquean.

La dirección ... 4001H

Esta rutina es usada por el Evaluador Factor para ejercer al "INP" función para un operand contenido en DAC. El número de babor es comprobado (5439H), el puerto leyó y el resultado colocó en DAC como un entero (4FCFH).

La dirección ... 400BH

Esta primera parte de rutina evalúa un operand en el rango - 32768 para + 65535 (542FH) y los lugares él en par de registro BC. Después de revisar en busca de una coma, por la rutina del estándar SYNCHR, eso evalúa un segundo operand en el rango 0 para 255 (521CH) y coloca esto en A de registro.

La dirección ... 4016H

Éste es el manipulador "apagado" de declaración. El número de babor y byte de a datos les es evaluado (400BH) y el byte de datos para el puerto pertinente Z80.

La dirección ... 401CH

Éste es la "espera" manipulador de declaración. El número de babor y "AND" operands son primero evaluado (400BH) seguido por el "XOR" optativo operand (521CH). El puerto es entonces repetidamente leídos y los operands aplicados, XOR luego Y, hasta que un resultado de poco cero es obtenido. En contra de la información dado en algunos manuales MSX el lazo puede estar quebrado por la llave CTRL-STOP como la rutina del estándar CKCNTC sea llamada dentro de él.

La dirección ... 4039H

Esta rutina es usada por el Runloop cuando encuentra el fin del texto de programa mientras en el modo de programa. ONEFLAG es al que se dio jaque para ver si eso todavía contiene un código activo de error.

Si así es que un "ningún CURRICULUM VITAE" error es generado, entonces el programa diferente la terminación continúa normalmente (6401H). La idea detrás de esta rutina es atrapar cualquier "en el ERROR" los manipuladores sin un "curriculum vitae" declaración al final.

- 114 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 404FH

Esta rutina es usada por el manipulador "leído" de declaración cuando un error es encontrado en una "DATA" declaración. La línea que el número contuvo en DATLIN es copiada para CURLIN así es que el manipulador de error señalará con una bandera los "datos" línea como la declaración ilegal en vez de la línea de programa. El control luego cae en el "error de sintaxis" generador.

La dirección ... 4055H

Éste es un grupo de nueve generadores de error, la E de registro está cargada del código pertinente de error y el control cae en el manipulador de error:

ADDR. EI ERROR

El error del 4055H Syntax
El 4058H Division por el cero
El 405BH DESPUÉS sin PARA
El conjunto imponente 405EH Redimensioned
La función del usuario del 4061H Undefined
El CURRICULUM VITAE del 4064H sin error
El error del 4067H Overflow
El 406AH Perdiendo operand
El bodorrio del 406DH Type

La dirección ... 406F

Éste es el manipulador de error Interpreter, todos los generadores de error transferencia para aquí con un código de error en E de registro. VLZADR es primero a cuadros para ver si el "VAL" manipulador de declaración ha cambiado el texto de programa, si así es que el carácter original está recuperado de VLZDAT. El número actual de la línea es entonces copiado de CURLIN para ERRLIN y PUNTEE y la pila Z80 está recuperada de SAVSTK (62F0H). El código de error es colocado en ERRFLG, para el uso por el "ERR" función, y el programa actual que la colocación del texto copió de SAVTXT a ERRTXT para el uso por el "curriculum vitae" manipulador de declaración. El número de la línea de error y colocación del texto de programa son también copiados para OLDLIN y OLDTXT para el uso por el "CONT" manipulador de declaración. ONELIN es entonces a cuadros para ver si uno previo "en el ERROR" la declaración ha sido ajustado. Si es así, y con tal que ningún código de error es ya activo, las transferencias de control para el Runloop (4620H) para ejecutar las declaraciones de recuperación de ERROR BÁSICO.

De otra manera el código de error se usa para contar a través de la mesa de mensaje de error en 3D75H hasta que lo requerido es alcanzado. Un RETORNO DE CARRO, LF es expedido (7323H) y la pantalla contenida para el modo del texto por la rutina del estándar TOTEXT. Un código de la CAMPANA es entonces expedido y el error que el mensaje exteriorizó (6678H). Asumir al Interpreter está adentro modo de programa, más bien que el modo directo, esto sea seguido por la línea numere (340AH) y controle caídas en el punto "muy bueno".

- 115 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 411FH

Éste es el punto de reentrada para el Interpreter Mainloop para un programa que termina. La pantalla se ve forzada al modo del texto por la rutina del estándar TOTEXT, la impresora es despejado (7304H) y yo amortiguador de la / O 0 cerrado (6D7BH). Un RETORNO DE CARRO, LF es entonces expedido para la pantalla (7323H), el mensaje "Oklahoma" es exhibido (6678H) y control cae en el Mainloop.

La dirección ... 4134H

Éste es el Interpreter Mainloop. CURLIN está primero listo a FFFFH para indicar modo directo y AUTFLG al que se dio jaque ver si "el AUTOMÓVIL" el modo está encendido. Si así es que el siguiente número de la línea es tomado de AUTLIN y exhibido (3412H). El Program Text Area es entonces registrado para ver si esta línea ya existe (4295H) y ya sea un asterisco o un espacio exhibido consecuentemente.

La rutina del estándar ISFLIO se usa luego para determinar si una "carga" declaración es

activa. Si así es que la línea de programa es coleccionada del casete (7374H), entonces de otra manera es tomada de la consola por la rutina del estándar PINLIN. Si la línea es vacía o la llave CTRL-STOP ha sido en apuros control se traslada de regreso al principio del Mainloop (4134H) sin más acción. Si la línea empieza con una línea numere esto es convertido a un entero sin firmar en par de registro Delaware (4769H).

La línea es luego convertida a forma del tokenized y está posado en KBUF (42B2H). Si ningún número de la línea fue encontrado al principio del control de la línea entonces transferencias para el Runloop (6D48H) para ejecutar la declaración.

Asumir la línea comienza con un número de la línea que es probada para ver si es otrora vacía y el resultado por ahora salvado. El número de la línea es copiado para PUNTEAR y AUTLIN aumentado por los contenidos de AUTINC, si AUTLIN ahora excede 65530 que el modo del "automóvil" está apagado. El Program Text Area es entonces registrado (4295H) para encontrar un número que hace juego de la línea o, a falta de esto, la colocación del siguiente número más alto de la línea. Si ningún número que hace juego de la línea es encontrado y la línea es vacía y "el AUTOMÓVIL" el modo está apagado un "número indefinido de la línea" error es generado (481CH). Si un número que hace juego de la línea es encontrado y la línea es vacía y "el AUTOMÓVIL" el modo está en el Mainloop simplemente saltos para la siguiente declaración (4237H).

Otrora cualquier punteros en el Program Text Area se convierten de regreso a revestir números (54EAH) y cualquier línea de programa existentes suprimidas (5405H). Asumir la línea nueva de programa está poco vacío que el Program Text Area es abierto arriba por la cantidad requerida (6250H) y la línea de programa del tokenized emulado de KBUF.

Los vínculos Program Text Area son entonces calculados de nuevo (4257H), el Storage Variable Areas son descongestionadas (629AH) y controlan transferencias de regreso al principio del Mainloop.

- 116 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 4253H

Esta rutina calcula de nuevo los vínculos Program Text Area después de una modificación de programa. Los primeros dos bytes de cada línea de programa contienen la dirección de puesta en marcha de la siguiente línea, éste es llamado el enlace. Aunque los incrementos del enlace por los que la cantidad de almacenamiento tomó programan línea grandemente se achica el tiempo requerido por el Interpreter para localizar una línea dada.

Un ejemplo de una línea típica de programa es exteriorizado debajo, en este caso la línea "10 ESTAMPADO 9" situados al principio del Program Text Area (8001H):

```
+  
| 09H 80H 0AH 00H 91H 20H 1AH 00H  
+
```

Figura 43: El programa Line

En el citado anteriormente ejemplo el enlace se guarda en la orden de palabra Z80 (LSB, MSB) y es inmediatamente seguido por el número binario de la línea, también en la orden de palabra. La declaración misma está compuesta de una señal "estampada" (91H), un solo

espacio, el número nueve y el fin de carácter de la línea (00H). Los más detalles del formato de almacenamiento pueden ser encontrados en la rutina del tokenizing (42B2H).

Cada vínculo es calculado de nuevo simplemente escudriñando a través de la línea hasta que el fin de carácter de la línea es encontrado. El proceso es completo cuando el fin del Program Storage Area, demarcada por el enlace especial de 0000H, es cumplido.

La dirección ... 4279H

Esta rutina es usada por la "lista" manipulador de declaración a coleccionar hasta dos operands de número de la línea del texto de programa. Si el primer número de la línea está presente, entonces es convertido a un entero sin firmar en par de registro Delaware (475FH), en caso de que no un valor predeterminado de 0000H es devuelto. Si el segundo número de la línea está presente, entonces debe ser precedido por ahí uno "-" la señal (F2H) y es devuelta en la pila Z80, en caso de que no un valor predeterminado de 65530 es devuelto. El control luego cae en la rutina de búsqueda del texto de programa para encontrar la primera línea a la que se puso notas de programa.

La dirección ... 4295H

Esta rutina le registra al Program Text Area para la línea de programa cuya línea el número es suministrado en par de registro Delaware. Comenzar en la dirección contuvo en TXTTAB cada línea de programa es examinada para un encuentro. Si un número igual de la línea es encontrado, entonces la rutina termina con Flag Z, C y par de registro BC apuntando hacia el principio de la línea de programa. Si un número más alto de la línea es encontrado, entonces la rutina ultima a Flag NZ, Carolina Del Norte y si el enlace de fin es alcanzado que la rutina ultima a Flag Z, Carolina Del Norte.

- 117 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 42B2H

Esta rutina es usada por el Interpreter Mainloop para tokenize una línea de texto. En las proposiciones del par de registro de entrada HL para el primer carácter del texto en BUF. En salida la línea del tokenized está en KBUF, el par de registro AC tiene aplicación que su largo y registro arreglan en pares puntos HL para su principio.

Exceptúe después de que la abertura abra comillas o después de las palabras claves "REM", de "llamada" o de "datos" cualquier cadena de caracteres haciendo juego con una palabra clave son reemplazados por la señal de esa palabra clave. Los alphabeticos de la letra minúscula se varían para la caja alta para la comparación de palabra clave. El carácter "?" Es reemplazado por la señal "estampada" (91H) y el carácter "por ahí" (3AH), "REM" señal (8FH), " la señal (E6H). La señal "distinta" (A1H) es precedida por un separador de declaración (3AH). Cualquier otros personajes misceláneos en el texto son emulados sin alteración excepto que los alphabeticos de la letra minúscula son convertidos a caja alta. Esas señales más pequeño que 80H, las señales de función, no pueden guardarse directamente en KBUF como estarán en conflicto con texto ordinario. En lugar de eso la secuencia FFH, + 80H de la señal es usada.

Las constantes numéricas se convierten primero en uno de los tipos estándar en DAC (3299H). Se guardan luego en una de varias formas dependiendo de su tipo y la magnitud, la idea general siendo minimizar uso de memoria:

0BH LSB MSB..... El número Octal
0CH LSB MSB..... El número Hex

El 11H para 1AH..... El Entero 0 Para 9
0FH LSB..... El Entero 10 Para 255
1CH LSB MSB..... El Entero 256 Para 32767
El INGENIERO ELÉCTRICO del 1DH DD DD..... Singularice a Precision
El INGENIERO ELÉCTRICO del 1FH DD DD DD DD DD DD ... Precision Doble

No hay señal específica para números binarios, estos son dejados como cadenas de caracteres. Éste parecería ser un legado de anteriores versiones de BASIC Microsoft. Cualquier indique a través de señas prefijar un número es considerado como un operador y se guarda como una señal separada, los números negativos no se producen durante la tokenización. Como los números dobles de precisión ocupan tanto espacio una línea conteniendo demasiados, para IMPRESIÓN de ejemplo 1 #, 1 #, 1 # etc . puede causar a KBUF para se llena. Si esto ocurre, entonces un "excedente del amortiguador de la línea" error es generado.

Cualquier numere después de una de las señales de palabra clave en la mesa en 43B5H - se considera - es un operand de número de la línea y se guarda con una señal diferente:

0DH LSB MSB..... El puntero
0EH LSB MSB..... El número de la línea

Durante la tokenización sólo el tipo normal (0EH) es generado, cuando un programa realmente marcha estos operands de número de la línea son convertidos al tipo del puntero de la dirección (0DH).

- 118 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 43B5H

Esta mesa de señales es usada durante la tokenización para revisar en busca de las palabras claves que toman operands de número de la línea. Las palabras claves mismas están listadas debajo:

RESTAURE HUYE
La LISTA del AUTOMÓVIL
RENUM LLIST
SUPRIMA a GOTO
El REGRESO del CURRICULUM VITAE
ERL LUEGO
SI NO GOSUB

La dirección ... 4524H

Éste es el "FOR" manipulador de declaración. El lazo Variable está primero ubicado y asignado su valor inicial por el manipulador "dejado" (4880H), la dirección del lazo Variable es devuelta en par de registro Delaware. El fin de la declaración es encontrado (485BH) y su dirección colocada en ENDFOR. La pila Z80 es entonces registrada (3FE6H) para cualquier lazo de lo mismo que usa bloques de parámetro Variable. Pues cada uno encontró el ENDFOR actual que la dirección es comparada con eso del bloque de parámetro, si hay un encuentro que el capítulo de la pila es descartado. Esto está hecho en el caso que hay cualquier lazos incompletos como resultado de un "GOTO" de regreso al "FOR" declaración de adentro el lazo.

El operand de terminación y "paso" optativo operand son entonces evaluados y convertidos

para el mismo tipo como el lazo Variable.

Después de comprobar ese espacio de la pila está disponible (625EH) que un veinticinco parámetro de byte el bloque es empujado encima del Z80 pila. Esto está hecho de lo siguiente:

- 2 bytes ... la dirección ENDFOR
- 2 bytes ... el número de la línea Current
- 8 bytes ... el valor de terminación Loop
- 8 bytes ... el valor de PASO
- 1 byte ... el tipo Loop
- 1 dirección de byte ... de PASO
- 2 bytes ... Address de lazo Variable
- 1 byte ... PARA señal (82H)

El parámetro que el bloque permanece en la pila para el uso por el "siguiente" manipulador de declaración hasta la terminación es cumplido, está luego descartado. El tamaño del bloque permanece constante aun sin embargo, para el entero y la sola precisión arroja a Variables, los bytes ocho llenos no son precisados para la terminación y los valores de PASO. En estos casos los bytes menos significativos están apiñados fuera con basura.

Debería ser notable que el tipo de operación de aritmética realizada por el "siguiente" manipulador de declaración, y por lo tanto la ejecución del lazo aceleran, depende enteramente del tipo del lazo Variable

- 119 -

5. EL INTÉRPRETE DE BASIC ROM

Y no el operand escribe a máquina. Para el tipo entero ejecución programa más rápido Variables, el % N por ejemplo, debería ser usado.

La dirección ... 4601H

Éste es el Runloop, cada declaración manipulador regresa aquí en la terminación así es que el Interpreter puede proceder hacia la siguiente declaración. El Z80 actual SP es emulado para SAVSTK para el error que la recuperación tiene en mente y que la llave CTRL-STOP comprobó por la rutina del estándar ISCNTC. Cualquier pendiente interrumpe es procesado (6389H) y la colocación coetánea del texto de programa, está sujeto en par de registro HL a todo lo largo del Interpreter, es emulado para SAVTXT.

El carácter coetáneo de programa es entonces examinado, si ésta es una declaración que el control del separador (3AH) transfiere inmediatamente para el punto de ejecución (4640H). Si es cualquier otra cosa sino que un fin de carácter de la línea (00H) un "error de sintaxis" es generado (4055H), entonces como hay texto espurio al final de la declaración. El par de registro HL es adelantado para el primer carácter de la línea nueva de programa y el enlace examinado, si éste es cero el programa es terminado (4039H). De otra manera el número de la línea es tomado de la línea nueva y está posado en CURLIN. Si TRCFLG es poco cero el número de la línea es exhibido (3412H) adjunto a la presente por paréntesis angulares, el control luego cae en la ejecución punto.

La dirección ... 4640H

Éste es el punto de ejecución Runloop. Una declaración de impuestos para el principio del Runloop (4601H) es empujado encima de la pila Z80 y el primer carácter tomado de la declaración nueva por la rutina del estándar CHRGTTR. Si es, entonces un control de carácter del subrayado (5FH) se traslada para la "llamada" manipulador de declaración (55A7H). Si es más pequeño que 81H, entonces la señal menor de declaración, transferencias de control para el

manipulador "dejado" (4880H). Si es mayor que D8H, entonces la señal más grande de declaración, es comprobado para ver si es una de la función que las señales permitieron como una declaración (51ADH).

De otra manera la dirección del manipulador es tomada de la mesa en 392EH y empujada encima de la pila. El control luego las caídas en la rutina del estándar CHRGTTR para ir a traer el siguiente personaje de programa antes del control transfieren para el manipulador de declaración.

La dirección ... 4666H

El nombre..... CHRGTTR

La entrada..... Las proposiciones HL para corriente programan carácter

La salida..... Un = Siguiente carácter de programa

Modifica.. AF, HL

La rutina estándar para ir a traer al siguiente personaje del texto de programa. El par de registro HL es incrementado y el carácter colocado en A de registro. Si es un espacio, entonces el código de la ETIQUETA (09H) o LF pone código a (0AH) que es del que se saltó por encima. Si es un separador de declaración (3AH) o acabe de carácter de la línea (00H), entonces la rutina ultima con Flag Z, Carolina Del Norte. Si es un dígito de "0" para "9" que la rutina termina con Flag NZ, entonces C. Si es cualquier otro carácter aparte

- 120 -

5. EL INTÉRPRETE DE BASIC ROM

De las señales numéricas de prefijo la rutina ultima a Flag NZ, Carolina Del Norte. Si el carácter es una de las señales numéricas de prefijo luego es colocado en CONSAV y el operand copiado para CONLO.

El código de tipo es colocado en CONTYP y la dirección del carácter de programa de arrastre en CONTXT.

La dirección ... 46E8H

Esta rutina es usada por el Evaluatodor Factor y durante la detokenización para recobrar un operand numérico cuándo una de las señales de prefijo son devueltas por la rutina del estándar CHRGTTR. La señal de prefijo es primero tomada de CONSAV, si es todo menos una señal de la línea de número o del puntero el operand es copiado de CONLO para DAC y el tipo que el código emuló de CONTYP para VALTYP. Si es una línea número, entonces se convierte para singularizar precisión y colocado en DAC (3236H). Si es un puntero el número original de la línea es recobrado del programa para el que se estableció referencias línea, convertida para singularizar precisión y colocado en DAC (3236H).

La dirección ... 4718H

Éste es el "DEFSTR" manipulador de declaración. La E de registro está cargada del código de tipo de la cuerda (03H) y las caídas de control en el general escriben a máquina rutina de definición.

La dirección ... 471BH

Éste es el "DEFINT" manipulador de declaración. La E de registro está cargada del código de tipo de entero (02H) y las caídas de control en el general escriben a máquina rutina de definición.

La dirección ... 471EH

Éste es el "DEFSNG" manipulador de declaración. La E de registro está cargada del solo código de tipo de precisión (04H) y las caídas de control en el general escriben a máquina rutina de definición.

La dirección ... 4721H

Éste es el "DEFDBL" manipulador de declaración. La E de registro está cargada del código doble (08H) de tipo de precisión y el primer rango que el carácter de definición comprobó (64A7H). Si ésta no es caja alta alfabética un "error de sintaxis" es generada (4055H). Si uno "-" la señal (F2H) sigue, entonces el segundo carácter de definición de rango es tomado y comprobado (64A7H), la diferencia entre lo dos determina el número de entradas en DEFTBL que se llena del código de tipo.

La dirección ... 4755H

Esta rutina evalúa un operand y convierte ella para un entero en par de registro Delaware (520FH). Si el operand es negativo, entonces una "llamada ilegal de función" error es generada.

- 121 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 475FH

Esta rutina es usada por los manipuladores de declaración mostrados en la mesa en 43B5H para coleccionar un solo operand de número de la línea del texto de programa y convertir ella para un entero sin firmar en par de registro Delaware. Si el primer carácter en el texto es uno "" (2EH) la rutina termina con los contenidos de PUNTO. Si es una de las señales de número de la línea (0DH o 0EH), entonces la rutina termina con los contenidos de CONLO. Los dígitos sucesivos diferentes son tomados y añadidos para el producto, con multiplicaciones apropiadas a las diez, hasta que un carácter poco numérico es encontrado.

La dirección ... 479EH

Éste es el manipulador "andado" de declaración. Si ningún operand de número de la línea está presente en el texto de programa el sistema es aclarado (629AH) y los ingresos de control para el Runloop con registro parece a HL apuntando hacia el principio del Program Storage Area. Si un operand de número de la línea está presente que el sistema es aclarado (62A1H) y controle transferencias para el "GOTO" manipulador de declaración (47E7H). De otra manera un siguiente filename está asumido, para la CARRERA de ejemplo "CAS:ARCHIVE ", y controle transferencias para la "CARGA " el manipulador de declaración (6B5BH);

La dirección ... 47B2H

Éste es el "GOSUB" manipulador de declaración. Después de comprobar ese espacio de la pila está disponible (625EH) el operand de número de la línea es coleccionado y colocado en par de registro Delaware (4769H). El bloque de parámetro de siete bytes es entonces empujado encima de la pila y control transfiere para el "GOTO" manipulador (47EBH). El bloque de parámetro está hecho de lo siguiente:

2 bytes ... End de dirección de declaración
2 bytes ... el número de la línea Current

2 bytes ... 0000H
1 byte ... la señal GOSUB (8DH)

El parámetro que el bloque permanece en la pila hasta un "regreso" declaración es ejecutado. Se usa luego para determinar la colocación original del texto de programa después de la cual está descartada.

La dirección ... 47CFH

Esta rutina es usada por el Runloop interrumpe procesador (6389H) para crear a un "GOSUB" bloque de parámetro de tipo en la pila Z80. Uno interrumpe bloque es idéntico para un bloque normal excepto que los dos bytes de cero mostrado de arriba sean reemplazados por la dirección de entrada del dispositivo en TRPTBL. Esta dirección será usada por el "regreso" manipulador de declaración para actualizar el dispositivo interrumpe estado una vez que una subrutina ha terminado. Después de empujar los boletos de transbordo de control del bloque de parámetro para el Runloop para ejecutar la línea de programa a cuyo discurso están suministrados en par de registro Delaware.

- 122 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 47E8H

Éste es el "GOTO" manipulador de declaración. El operand de número de la línea es coleccionado (4769H) y colocado en par de registro HL. Si es, entonces un control del puntero se traslada inmediatamente para el Runloop para empezar ejecución en la colocación nueva del texto de programa. De otra manera el número de la línea es comparado con el número actual de la línea para determinar la colocación de puesta en marcha para la búsqueda del texto de programa. Si es mayor, entonces los principios de búsqueda del fin de esta línea (4298H), si eso fueran más pequeños, entonces empieza desde el principio del Program Text Area (4295H). Si la línea a la que se puso notas no puede ser encontrada, entonces un "número indefinido de la línea" error es generado (481CH).

De otra manera el operand de número de la línea es reemplazado por la dirección de la línea a la que se puso notas de programa y su señal se convirtió en el tipo del puntero (5583H). El control luego se traslada para el Runloop para ejecutar el verso para el que se estableció referencias de programa.

La dirección ... 481CH

Éste es el "número indefinido de la línea" generador de error.

La dirección ... 4821H

Éste es el "regreso" manipulador de declaración. Una dirección falsa del lazo Variable es colocada en par de registro Delaware y la pila Z80 registrada (3FE2H) para encontrar el primer bloque de parámetro no formar parte de un "FOR" lazo, este capítulo de pila está descartado. Si ningún "GOSUB" señal (8DH) es encontrado en este punto, entonces un "regreso sin GOSUB" error es generado.

Los siguientes dos bytes son luego tomados del bloque, si son poco cero que el bloque fue generado por uno interrumpe y el "alto" temporal condición está removido (633EH). El texto de programa es entonces examinado, si cualquier cosa sigue que el "regreso" señal mismo - se asume - es un operand de número de la línea y control se traslada para el "GOTO" manipulador

(47E8H). De otra manera el viejo número de la línea y dirección del texto de programa son tomados del bloque y controlan ingresos para el Runloop.

La dirección ... 485BH

Éste es la "DATA" manipulador de declaración. El texto de programa es saltado encima hasta que un separador de declaración (3AH) o fin de carácter de la línea (00H) es encontrado. Esta rutina es también el "REM" y "si no" la manipuladora de declaración por el punto de entrada en 485DH, en este caso sólo el fin de actos de carácter de la línea como un exterminador.

La dirección ... 4880H

Éste es el manipulador "dejado" de declaración. El Variable está primero ubicado (5EA4H), su dirección invirtió a TEMP y el operand evaluado (4C64H). Si es necesario el tipo de operand se varía luego a hacer juego con eso del Variable (517AH). Asumir el operand es uno de los tres tipos numéricos que es simplemente copiado

- 123 -

5. EL INTÉRPRETE DE BASIC ROM

De DAC para el Variable en la Variable Storage Area (2EF3H).

Si el operand es una cuerda escriba a máquina, entonces la dirección del cuerpo de la cuerda es tomada del descriptor e inspeccionada. Si está en KBUF, entonces como sería el caso para una cuerda explícita en una declaración directa, el cuerpo es primero copiado para el String Storage Area y un descriptor nuevo creado (6611H). El descriptor está entonces liberado de TEMPST (67EEH) y copiado para el Variable en el Storage Variable Area (2EF3H).

La dirección ... 48E4H

Esto es lo "en el ERROR", "en DEVICE GOSUB" y "en la EXPRESIÓN" el manipulador de declaración. Si el siguiente carácter del texto de programa no es, entonces un "error" control de la señal (A6H) se traslada para lo "en DEVICE GOSUB" y "en la EXPRESIÓN" el manipulador (490DH). El texto de programa es comprobado para asegurar que una señal "GOTO" (89H) sigue y luego el operand de número de la línea coleccionado (4769H). El texto de programa es registrado para obtener la dirección de la línea a la que se puso notas (4293H) y esto es colocado en ONELIN. Si el operand de número de la línea es poco cero la rutina luego termina. Si el operand de número de la línea es, entonces el cero ONEFLG es comprobado para ver si una situación de error ya existe (insinuar que la declaración está dentro de una rutina de recuperación de ERROR BÁSICO). Si así es que el control se traslada para el manipulador de error (4096H) para obtener a la fuerza un error inmediato, entonces de otra manera la rutina termina normalmente.

La dirección ... 490DH

Esto es lo "en DEVICE GOSUB" y "en la EXPRESIÓN" el manipulador de declaración. Si el siguiente carácter del texto de programa no es, entonces un control de la señal del dispositivo (7810H) se traslada para lo "en la EXPRESIÓN" el manipulador (4943H). Después de comprobar el texto de programa para un "GOSUB" señal (8DH) cada uno del número de la línea que los operands requirieron para un dispositivo particular es coleccionado a su vez (4769H). Asumir un operand dado de número de la línea es poco cero el texto de programa es registrado para encontrar la dirección de la línea a la que se puso notas (4293H) y esto es colocado en la entrada del dispositivo en TRPTBL (785CH). La rutina termina cuando ninguno de los más operands de número de la línea son encontrados.

La dirección ... 4943H

Esto es lo "en la EXPRESIÓN" el manipulador de declaración. El operand es evaluado (521CH) y la siguiente señal (89H) "GOSUB" de la señal (8DH) o "GOTO" colocado en A de registro. El operand se usa luego para contar a lo largo del texto de programa hasta que las proposiciones del par de registro HL para la línea requerida numeran operand. El control luego se traslada de regreso al punto de ejecución Runloop (4646H) para descifrar al "GOSUB" o "GOTO" señal.

La dirección ... 495DH

Éste es el "curriculum vitae" manipulador de declaración. ONEFLG es primero a cuadros para hacer seguro que una condición de error ya existe, en caso de que no un "curriculum vitae sin error" es generado (4064H). Si un non

- 124 -

5. EL INTÉRPRETE DE BASIC ROM

El operand de número de la línea de cero sigue transferencias de control para el "GOTO" manipulador (47EBH). Si una "siguiente" señal (83H) sigue, entonces la colocación del error está recuperada de ERRTXT y ERRLIN, el principio de la siguiente declaración es encontrado (485BH) y la rutina termina. Si no hay operand de número de la línea o si es ponga en el cero, entonces la colocación del error es encontrada de ERRTXT y ERRLIN y la rutina termina.

La dirección ... 49AAH

Éste es el "error" manipulador de declaración. El operand es evaluado y colocado en registro E (521CH). Si es ponga en el cero una "llamada ilegal de función" error es generada (475AH), entonces el control diferente se traslada para el manipulador de error (406FH).

La dirección ... 49B5H

Éste es el "automóvil" manipulador de Declaración. El principio optativo y línea de incremento numeran a operands, ambos con un valor predeterminado de diez, son coleccionados (475FH) y colocados en AUTLIN y el poco cero de la hechura AUTINC. After AUTFLG el Runloop regreso se destruye y el control se traslada directamente para el Mainloop (4134H).

La dirección ... 49E5H

Éste es la "declaración IF" manipulador de declaración. El operand es evaluado (4C64H) y, después de inspeccionar para un "GOTO" señal (89H) o "luego" una señal (DAH), su signo es probado (2EA1H). Si el operand es poco ponga en el cero (verdadero) el siguiente texto es ejecutado ya sea por una transferencia inmediata para el Runloop (4646H) o, para un número de la línea operand, el "GOTO" manipulador (47E8H). Si el operand es ponga en el cero (falso), entonces el texto de declaración es escudriñado (485BH) hasta que una señal "distinta" (A1H) no sea encontrada simétrica por una "declaración IF" señal (8BH) y la ejecución recomienza.

La dirección ... 4A1DH

Éste es el "LPRINT" manipulador de declaración. PRTFLG está listo para 01H, a dirigir salida para la impresora, y controlar transfere para el manipulador "estampado" (4A29H).

La dirección ... 4A24H

Éste es el manipulador "estampado" de declaración. El texto de programa es primero a cuadros para un número del amortiguador de arrastre y, si PTRFIL necesario, se dispuso a dirigir salida para lo requerido y / la O el amortiguador (6D57H).

Si ningún más texto de programa existe, entonces un RETORNO DE CARRO, LF es expedido (7328H) y la rutina termina (4AFFH). Los personajes sucesivos diferentes son tomados del texto de programa y analizados. Si una señal "utilizadora" (E4H) es encontrada que control se traslada para el manipulador (60B1H) "que USA IMPRESIÓN". Si Uno "" El carácter es encontrado que control justamente se traslada de regreso al principio para ir a traer el siguiente artículo (4A2EH). Si una coma es encontrada, entonces suficientes espacios son expedidos para traer la colocación estampada coetánea, de TTYPOS, LPTPOS o uno y o el amortiguador de la / O

- 125 -

5. EL INTÉRPRETE DE BASIC ROM

FCB, para un múltiplo integral de catorce. Si la salida es dirigida para la pantalla y la colocación estampada es iguales para o más gran que los contenidos de CLMLST o si la salida es dirigida para la impresora y eso es iguales para o más gran que 238 luego un RETORNO DE CARRO, entonces LF es expedido en lugar de eso (7328H). Si un "SPC (" la señal (DFH) es encontrado, entonces el operand es evaluado (521BH) y el número requerido de espacios es salida. Si una "etiqueta (" la señal (DBH) es encontrada, entonces el operand es evaluado (521BH) y suficientes espacios expedidos para traer la impresión de corriente sitúan, de TTYPOS, LPTPOS o uno y o amortiguador de la / O FCB, para el punto requerido.

Si ninguno de estos personajes son encontrados que el texto de programa contiene unos datos ítem cuál es luego evaluado (4C64H). Si el operand es una cuerda, entonces es simplemente exhibida (667BH). Si es numérico, entonces sea primero convertido a texto en FBUFFR (3425H) y una cuerda que el descriptor creó (6635H). Si la salida es dirigida uno y o el amortiguador de la / O la cuerda resultante es entonces exhibida (667BH). Si la salida es dirigida para la pantalla o la impresora la colocación estampada coetánea, entonces de TTYPOS o LPTPOS, es comparado con el largo de la línea y un RETORNO DE CARRO, LF expedido (7328H) si la salida no se probará la línea. El largo máximo de la línea es 255 para la impresora y es tomado de LINLEN para la pantalla. Una vez que la cuerda ha sido exteriorizado control se traslada de regreso al principio del manipulador.

La dirección ... 4AFFH

Esta rutina pone en el cero a PRTFLG y PTRFIL para devolver a la salida de Interpreter a la pantalla.

La dirección ... 4B0EH

Ésta es la "entrada de la LÍNEA de ENTRADA", de la "línea" y "línea" manipulador de declaración. Si los siguientes programan carácter del texto, entonces es, entonces cualquier cosa aparte de una "entrada" control de la señal (85H) se traslada para la "línea" manipulador de declaración (58A7H). Si los siguientes programan carácter del texto es, entonces unas (23H) transferencias de control para la "línea INTRODUCEN EN LA COMPUTADORA" manipulador de declaración (6D8FH).

Cualquier siguiente cuerda presta es evaluada y exhibida (4B7BH) y el Variable localizado (5EA4H) e inspeccionada para asegurar que es un tipo de la cuerda (3058H). La línea de texto es coleccionada de la consola por la rutina del estándar INLIN, si Flag C (CTRL-STOP) es control devuelto se traslada para el "alto" manipulador de declaración (63FEH). De otra manera la cuerda de entrada es analizada y un descriptor creado (6638H), un control luego se traslada para el manipulador "dejado" de declaración para la asignación (4892H). Debería ser notable que la

pantalla no se ve forzada al modo del texto antes de que la entrada sea coleccionada.

La dirección ... 4B3AH

¿Éste es el mensaje de texto puro "?Rehaga de principio ", el RETORNO DE CARRO, LF ultimado por un byte de cero.

- 126 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 4B4DH

Esta rutina es usada por la "lectura /entrada" declaración manipulador si no ha podido convertir un artículo de datos a forma numérica. Si en el modo "leído" (FLGINP es poco cero) un "error de sintaxis" es generado (404FH). ¿De otra manera el mensaje "?Rehaga de principio " es exhibido (6678H) y el control regresa al manipulador de declaración.

La dirección ... 4B62H

Éste es la "entrada" manipulador de Declaración. El número del amortiguador es evaluado y PTRFIL se dispuso a dirigir entrada de lo requerido yo amortiguador de la / O (6D55H), el control luego se traslada para la "lectura /entrada" combinada manipulador de declaración (4B9BH).

La dirección ... 4B6CH

Éste es la "entrada" manipulador de declaración. Si el siguiente carácter del texto de programa es, entonces un control se traslada para la "entrada" manipulador de declaración (4B62H). De otra manera la pantalla se ve forzada al modo del texto, por la rutina del estándar TOTXT, y cualquier indicador que la cuerda analizó (6636H) y exteriorizó (667BH). Un signo de interrogación es entonces exhibido y una línea de texto cobrado de la consola por la rutina del estándar QINLIN. Si esto devuelve transferencias de control Flag C (CTRL-STOP) para el "alto" manipulador (63FEH). Si el primer carácter en BUF es ponga en el cero (la entrada nula), entonces el manipulador termina saltándose hasta el fin de la declaración (485AH), el control diferente cae en la "lectura /entrada" combinada manipulador.

La dirección ... 4B9FH

Éste es el manipulador "leído" de declaración, un capítulo grande es también usado por la "entrada" y "la ENTRADA" declaraciones así es que la estructura es bastante embarazosa. Cada Variable encontró en el texto de programa está ubicado a su vez (5EA4H), para cada uno el artículo correspondiente de datos es obtenido y asignado para el Variable por el manipulador "dejado" (4893H). Cuando en el modo "leído" los artículos de datos son tomados del texto de programa usando los contenidos iniciales de DATPTR (4C40H). Cuando en "la ENTRADA" o "introduzca en la computadora" modo que los artículos de datos son tomados del amortiguador del texto BUF.

Si los artículos de datos están agotados en el modo "leído" uno "por los DATOS" el error es generado. Si están exhaustos adentro "introduzca en la computadora", entonces el modo dos la pregunta marca es exteriorizado y otra línea ida a traer de la consola por la rutina del estándar QINLIN. Si están exhaustos adentro "introduzca en la computadora" modo que otra línea de texto es copiada para BUF de lo pertinente yo / O amortiguador (6D83H). Si el Variable que la lista es el rato agotado en "la ENTRADA" el modo el mensaje "que la Extra ignoró" es exteriorizado (6678H) y el manipulador termina (4AFFH). En "la ENTRADA" el modo que ningún mensaje está

el rato exteriorizado en el control "leído" de modo lo termina por DATPTR modernizante (63DEH).
¿Si un artículo de datos no puede ser convertido a, entonces la forma numérica (3299H) para hacer juego con un control numérico Variable se traslada para lo "?Rehaga de principio " la rutina (4B4DH).

- 127 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 4C2FH

¿Lo es el mensaje de texto puro "?La extra ignorada ", el RETORNO DE CARRO, LF ultimado por un byte de cero.

La dirección ... 4C40H

Esta rutina es usada por el manipulador "leído" para localizar los siguientes "datos" declaración en el texto de programa, la dirección de la que echar a andar es suministrada en par de registro HL. Cada declaración de programa es examinada hasta que una "DATA" señal (84H) es encontrada después de lo cual la rutina termina (4BD1H). Si el enlace de fin es alcanzado uno "por DATA" error es generado. Como los ingresos de búsqueda el número de la línea de cada línea de programa es colocado en DATLIN para el uso por el manipulador de error.

La dirección ... 4C5FH

Esta rutina inspecciona que el siguiente carácter en el texto de programa es la señal (EFH) y luego cae en el Evaluador Expression. Estando entrado en 4C62H inspecciona para "(".

La dirección ... 4C64H

Éste es el Evaluador Expression. En las proposiciones del par de registro de entrada HL para el primer carácter de la expresión a ser evaluado. En par de la registro de la salida HL apunta hacia el carácter siguiendo la expresión, el resultado está en DAC y el código de tipo en VALTYP. Para un resultado de la cuerda la dirección del descriptor de la cuerda es devuelta en DAC + 2. El descriptor mismo comprendiendo un solo byte para el largo de la cuerda y dos bytes para su dirección, será en TEMPST o dentro de una cuerda Variable.

Una expresión es una lista de factores (4DC7H) acoplados por operadores con niveles discrepantes de precedencia. Para procesar tal expresión correctamente el Evaluador Expression debe poder por ahora apilar un resultado intermedio, si el siguiente operador tiene una precedencia superior que el operador actual, y vuelva a empezar en un cálculo nuevo. Por consiguiente tiene dos operaciones básicas, la PILA y TENGA APLICACIÓN. Por ejemplo:

$3+250\backslash 2^2\cdot 3^3+1$

La PILA: 3 + (\ sigue)

La PILA: 250 \ (sigue)

TENGA APLICACIÓN: $2^2=4$ (* sigue)

La PILA: 4 * (sigue)

TENGA APLICACIÓN: $3^3=27$ (+ sigue)

TENGA APLICACIÓN: $4\cdot 27=108$ (+ sigue)

TENGA APLICACIÓN: $250\backslash 108=2$ (+ sigue)

TENGA APLICACIÓN: $3+2=5$ (+ sigue)

TENGA APLICACIÓN: $5+1=6$ (, sigue)

La evaluación termina cuando el siguiente operador tiene una precedencia

- 128 -

5. EL INTÉRPRETE DE BASIC ROM

El igual para o más abajo de la precedencia inicial y la pila está vacío. El delimitador de expresión, mostrado como una coma en el ejemplo, es considerado como tener una precedencia de cero y así también siempre detendrá evaluación. Normalmente el Expression Evaluador se pone en marcha con una precedencia inicial de cero pero el punto de entrada en 4C67H puede usarse para suministrar un valor alternativo en D de registro. Esta facilidad es usada por el Evaluador Factor para restringir el rango de evaluación al aplicar la negación monádica y "no" los operadores.

La dirección ... 4D22H

Esta rutina es usada por el Evaluador Expression para aplicar un operador de matemáticas del infix (+ - * /) para un par de operands numéricos. Hay rutinas separadas para los operadores del relational (4F57H) y los operadores lógicos (4F78H). El primer operand, su código de tipo, y la señal del a operador les es suministrado en la pila Z80, el segundo operand y su código de tipo son suministrados en DAC y VALTYP. Los tipos de ambos operands son primero comparado, si difieren que el operand mínimo de precisión se convierte para hacer juego con lo más alto. Los operands son luego movidos a las colocaciones requeridas por las rutinas de matemáticas. Para los enteros la primera parte que el operand es colocado en el par de registro Delaware y el segundo en registro parean a HL. Para la sola precisión la primera parte que el operand es colocado en registra a C, B, E, D y el segundo en DAC. Para la precisión doble el primer operand es colocado en DAC y el segundo en ARG. La señal del operador se usa luego para sacar la dirección requerida de la mesa en 3D51H, 3D5DH o 3D69H, dependiendo del tipo del operand, y las transferencias de control para la rutina pertinente de matemáticas.

La dirección ... 4DB8H

Esta rutina es usada por el Evaluador Expression para dividir dos operands de entero. La primera parte que el operand es contenido en par de registro Delaware y el segundo en par de registro HL, el resultado es devuelto en DAC. Ambos operands se convierten para singularizar transferencias de precisión (2FCBH) y de control para la sola rutina de división de precisión (3265H).

La dirección ... 4DC7H

Éste es el Evaluador Factor. En las proposiciones del par de registro de entrada HL para el carácter antes del factor a ser evaluado. En par de la registro de la salida HL apunta hacia el carácter siguiendo el factor, el resultado está en DAC y el código de tipo en VALTYP. Un factor puede tener un año de edad de lo siguiente:

(1) UNA numérica o A de constante de la cuerda (2) la numérica o una A de la cuerda Variable (3) funcionan (4) UN operador monádico (+ - NO) (5) UNA expresión del parenthesized

El primer carácter es tomado del texto de programa por lo

- 129 -

5. EL INTÉRPRETE DE BASIC ROM

La rutina del estándar CHRGR y examinado. Si es un fin de carácter Statement un "operand perdido" error es generado (406AH). Si es un dígito de ASCII, entonces se convierte de forma textual para uno de los tipos numéricos estándar en DAC (3299H).

Si es caja alta alfabética (64A8H) es un Variable y su valor coetáneo es devuelto (4E9BH). Si es una señal numérica el número es copiada de CONLO para DAC (46B8H). Si es, entonces uno del FFH prefijó valores simbólicos de función exteriorizados en la mesa en 39DEH que es descifrado para transferir control para el manipulador pertinente (4ECFH) de función. Si es el operador monádico, entonces es simplemente saltado encima, sólo el monádico "-" el operador (4E8DH) y "NOT" monádico operador (4F63H) precisan cualquier acción.

Si es una cita inaugural la siguiente cuerda explícita es analizada y un descriptor creado (6636H). Si es uno "y" es una constante numérica poco decimal y eso es convertida a uno de los tipos numéricos estándar en DAC (4EB8H). Si no es una de las funciones exteriorizadas debajo de luego debe ser una expresión del parenthesized (4E87H), entonces de otra manera un "error de sintaxis" es generado. Las siguientes señales de función son probadas para directamente y controlan transferidas para la dirección mostrada:

YERRE.... 4DFDH ATTR\$.... EI 7C43H
ERL.... 4E0BH VARPTR ... 4E41H
EI PUNTO.. 5803H USR..... EI 4FD5H
EI TIEMPO ... 7900H INSTR.... EI 68EBH
EI BLOQUE DE OBJETOS MOVIBLES. 7A84H INKEY\$... 7347H
VDP.... 7B47H STRING\$. EI 6829H
La BASE ... 7BCBH INPUT\$... 6C87H
EI JUEGO ... 791BH CSRLIN ... 790AH
EI DSKI\$. 7C3EH FN..... EI 5040H

La dirección ... 4DFDH

Esta rutina es usada por el Evaluador Factor para ejercer al "ERR" función. Los contenidos de ERRFLG son colocados en DAC como un entero (4FCFH).

La dirección ... 4E0BH

Esta rutina es usada por el Evaluador Factor para ejercer al "ERL" función. Los contenidos de ERRLIN son emulados para DAC como un solo número de precisión (3236H).

La dirección ... 4E41H

Esta rutina es usada por el Evaluador Factor para ejercer al "VARPTR" función. Si la señal de función es seguida por ahí, entonces uno el número del amortiguador es evaluado (521BH), lo que yo / la O el amortiguador FCB localicé (6A6DH) y que su dirección colocó en DAC como un entero (2F99H). De otra manera el Variable está ubicado (5F5DH) y su dirección colocó en DAC como un entero (2F99H).

La dirección ... 4E8DH

Esta rutina es usada por el Evaluatodor Factor a aplicar lo monádico "-" el operador. La D de registro está colocada para un valor de precedencia de 7DH, el factor evaluado (4C67H) y luego negado (2E86H).

La dirección ... 4E9BH

Esta rutina es usada por el Evaluatodor Factor para devolver el valor coetáneo de un Variable. El Variable está primero ubicado (5EA4H). Si es una cuerda Variable su dirección es colocada en DAC para apuntar hacia el descriptor. De otra manera los contenidos del Variable son emulados para DAC (2F08).

La dirección ... 4EA9H

Esta rutina devuelve el solo carácter hacia el que se apuntó por par de registro HL en A de registro, si es una letra minúscula alfabética la convierte para la caja alta.

La dirección ... 4EB8H

Esta rutina es usada por el Evaluatodor Factor y la rutina numérica (3299H) de entrada para convertir a un número de ampersand Prefixed ("y") de forma textual a un entero en DAC. Como cada carácter legal es encontrado que el producto está multiplicado por 2, 8 o 16, dependientes en el carácter que inicialmente siguió el ampersand, y el dígito nuevo se sumaron a eso. Si el producto se derrama, entonces un "Overflow" error es generado (4067H). La rutina termina cuando un carácter inaceptable es encontrado.

La dirección ... 4EFCH

Esta rutina es usada por el Evaluatodor Factor para procesarle al FFH las señales colocadas como prefijo de función. Si la señal es cualquier "LEFT\$", "RIGHT\$" o "MID\$" el operand de la cuerda es evaluado (4C62H), la dirección de su descriptor empujado encima de la pila Z80 y el siguiente operand numérica también evaluada (521CH) y se apila. De otra manera parenthesized operand de la función es evaluado (4E87H) y, para "SQR", "para RND", "para PECADO", "para LEÑO", "para EXP", "para LECHUGA ROMANA", "para BRONCEADO" o "para ATN" sólo, convertido para duplicar precisión (303AH). La señal de función se usa luego para sacar la dirección requerida de la mesa en 39DEH y controlar transferencias para el manipulador de función.

La dirección ... 4F47H

Esta rutina es usada por la rutina numérica (3299H) de conversión de entrada a probar para uno o "-" el carácter o la señal. Devuelve registro D = 0 para D=FFH del positivo y de registro para la negativa.

La dirección ... 4F57H

Esta rutina es usada por el Evaluatodor Expression a aplicar uno

- 131 -

5. EL INTÉRPRETE DE BASIC ROM

El operador del relational (< > = o las combinaciones) para un par de operands. Si los operands

son numéricos, entonces la primera parte del Evaluador Expression usa la rutina del operador de matemáticas (4D22H) para ejercer la operación general de relación para los operands. Si los operands son instrumentos de cuerda la rutina de comparación de la cuerda (65C8H) son usados primero.

Cuando el control llega aquí el resultado de relación está en A de registro y los Z80 Flags:

Operand 1 = Operand 2 ... UN = 00H, Flag Z, Carolina Del Norte
Operand 1 < Operand 2 ... UN = 01H, Flag NZ, Carolina Del Norte
Operand 1 > Operand 2 ... A=FFH, Flag NZ, C

El Evaluador Expression también suministra una máscara mordida definiendo a los operadores originales en la pila Z80. Esto tiene uno 1 en cada colocación si la operación asociada es precisada: 00000. La máscara es aplicada para el resultado de relación produciendo cero si ninguna de las condiciones está satisfecha. Esto es luego colocado en DAC como un entero verdadero (2E9AH) (- 1) o falso (0).

La dirección ... 4F63H

Esta rutina es usada por el Evaluador Factor para aplicar al "NOT" monádico operador. La D de registro está colocada para una precedencia inicial ras con ras de 5AH y la expresión evaluada (4C67H) y convertida a un entero (2F8AH). Está entonces invertido y recuperado para DAC.

La dirección ... 4F78H

Esta rutina es usada por el Evaluador Expression para aplicar a un operador lógico ("Oregon", "y", "XOR", "EQV" y "diablillo") o el "MOD" y los operadores para un par de operands numéricos. El primer operand, cuál ya ha sido convertido a un entero, es suministrado en la pila Z80 y el segundo es suministrado en DAC. La operadora a la señal (realmente su nivel de precedencia) está suministrada en B de registro. Después de convertir el segundo operand a un entero (2F8AH) el operador es examinado. Hay rutinas separadas para "MOD" (323AH) y (31E6H) sino los operadores lógicos son procesados localmente usando las instrucciones lógicas Z80 correspondientes en pares de registro Delaware y HL. El resultado es almacenado en DAC como un entero (2F99H).

La dirección ... 4FC7H

Esta rutina es usada por el Evaluador Factor para ejercer a los "LPOS" función para un operand contenido en DAC. Los contenidos de LPTPOS son colocados en DAC como un entero (4FCFH).

La dirección ... 4FCCH

Esta rutina es usada por el Evaluador Factor para ejercer la función "de punto de venta" para un operand contenido en DAC. Los contenidos de TTYPOS son colocados en DAC como un entero (2F99).

- 132 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 4FD5H

Esta rutina es usada por el Evaluador Factor para ejercer al "USR" función. El número del

usuario es coleccionado directamente del texto de programa, no puede ser una expresión, y la dirección asociada tomada de USRTAB (4FF4H). El siguiente parenthesized operand es entonces evaluado (4E87H) e izquierdo en DAC como el parámetro aprobado. Si es una cuerda escriba a máquina, entonces su almacenamiento es liberado (67D3H).

La posición coetánea del texto de programa es empujado encima de la pila Z80 seguido por un regreso para 3297H, la rutina en esta dirección restaurará la colocación del texto de programa después de que la función del usuario haya terminado. El control luego las transferencias para la dirección del usuario con par de registro HL apuntando hacia el primer byte de DAC y el código de tipo, de VALTYP, adentro registra UNO. Adicionalmente, para un parámetro de la cuerda, la dirección de descriptor es tomada de DAC y está posado en par de registro Delaware.

La rutina del usuario puede modificar cualquier registro excepto el Z80 SP y debería terminar con una instrucción RET, interrumpe puede quedarse deshabilitado si es necesario como el Runloop los reposibilitará.

Cualquier parámetro numérico a ser devuelto al Interpreter debería ser colocado en DAC. En rigor éste debería ser el mismo tipo numérico como el parámetro aprobado, de cualquier forma que si VALTYP es modificado que el Interpreter siempre lo aceptará.

Devolver un tipo de la cuerda es más difícil. Usando el mismo método como las funciones de la cuerda del Evaluador Factor, que involucra a copiar la cuerda para el String Storage Area y empujando un descriptor nuevo encima de TEMPST, está complicado y vulnerable para los cambios en el sistema MSX. Un más simple y más método fidedigno es usar el parámetro aprobado para crear el espacio para el resultado. Ésta no debería ser una cuerda explícitamente declarada como el programa que el texto tendrá para ser modificado, en lugar de eso un parámetro implícito debería ser usado. Esto debe haber terminado con cuidado sin embargo, es demasiado fácil ganar la impresión que el Interpreter ha aceptado la cuerda cuando de hecho no tiene. Tome el siguiente ejemplo que las gamas nada excepto devuelven el parámetro aprobado:

```
10 ATIZAN y H9000, y HC9
20 DEFUSR = y H9000
30 A$ =USR (STRING$ (12, "!))
40 ESCRIBEN EN LETRAS DE IMPRENTA a A$
50 B$ STRING$ (9, "X")
60 ESCRIBEN EN LETRAS DE IMPRENTA a A$
```

Al principio parece que la cuerda aprobada haya estado correctamente asignada para A\$. Cuando la línea 60 es alcanzada sin embargo se vuelve aparente que A\$ haya sido corrompido por la subsig

uiente asignación de una cuerda para B\$. Lo que ha ocurrido es que el almacenamiento temporal ubicó para el parámetro aprobado fue rescatado del String Storage Area antes del control transferido para la rutina del usuario. Esta región se usó luego para almacenar la cuerda perteneciendo a B\$ así modificando a A\$.

- 133 -

5. EL INTÉRPRETE DE BASIC ROM

Esta situación puede ser evitada asignando el parámetro a un Variable de antemano y luego pasando al Variable, por ejemplo:

```
10 A$ STRING$ (12, "!)
20 A$ =USR (A$)
```

Revista 10 resultados en doce bytes de la String Storage Area siendo permanentemente ubicado para A\$. Cuando la función del usuario es introducida el descriptor, cuál es hacia el que se apuntó por par de registro Delaware, contendrá la puesta en marcha dirección de la región de doce bytes donde el resultado debería ser colocado. Si la cuerda devuelta es más corta que la aprobada el byte de largo del descriptor puede variarse sin cualquier efectos secundarios. Pues los más detalles en el almacenamiento de la cuerda ven al colector de basuras (66B6H).

Un punto que vale notar es que una operación "cristalina" no es estrictamente menester antes de que un programa de lenguaje de máquina sea cargado. La región entre la parte superior del Array Storage Area y la base de la pila Z80 no es nunca usada por el Interpreter. Un programa puede existir en esta región con tal que las dos áreas de acorralamiento no lo traslapan.

La dirección ... 500EH

Éste es el "DEFUSR" manipulador de declaración. El número del usuario es coleccionado directamente del texto de programa, no puede ser una expresión, y la entrada asociada en USRTAB localizado (4FF4H).

El operand de la dirección es entonces evaluado (542FH) y colocado en USRTAB.

La dirección ... 501DH

Éste es el "DEF FN" y "DEFUSR" manipulador de declaración. Si el siguiente carácter es, entonces un "USR" que el control de la señal (DDH) transfiere para el "DEFUSR" manipulador de declaración (500EH), de otra manera el texto de programa es comprobado para un "FN" de arrastre señal (DEH). El nombre de función Variable es hallado (51A1H) y, después de comprobar que el Interpreter está en el modo de programa (5193H), la colocación coetánea del texto de programa es colocada allí. Cada uno de los Variables en la lista formal de parámetro están luego localizados uno tras otro (5EA4H), éste lo está simplemente para asegurar que ellos son creados. La rutina termina pasando por alto el resto de la declaración (485BH) como el cuerpo de función no es requerido a esta hora.

La dirección ... 5040H

Esta rutina es usada por el Evaluador Factor para ejercer al "FN" función. El nombre de función Variable está primero ubicado (51A1H) a obtener la dirección de la definición de función en el texto de programa. Cada Variable formal de la definición de función está ubicado a su vez (5EA4H) y su dirección empujada encima de la pila Z80. Como cada uno es encontrado el parámetro real correspondiente es evaluado (4C64H) y empujado encima de la pila con eso. Si

- 134 -

5. EL INTÉRPRETE DE BASIC ROM

Necesario el tipo del parámetro real se convierte para hacer juego con eso del parámetro formal (517AH) '

Cuando ambas listas están exhaustos cada dirección Variable formal y parámetro real son a los que se hizo estallar de la pila a su vez. Cada Variable es entonces copiado del Storage Variable Area para PARM2 con su valor repuesto por el parámetro real. Debería ser notable que, porque el PARM2 tiene sólo cien bytes de largo, un máximo de nueve parámetros dobles de precisión está permitido. Cuando todos los parámetros reales han sido emulados para PRM2 los contenidos enteros de PARM1 (el área coetánea de parámetro) son empujados encima del Z80 pila y PARM2 es emulado para PARM1 (518EH). El par de registro HL está luego colocado al

principio del cuerpo de función en el texto de programa y la expresión es evaluada (4C5FH). Los contenidos viejos de PARM1 son a los que se hizo estallar de la pila y restituidos. Finalmente el resultado de la evaluación es tipo converso si es necesario para hacer juego con el tipo de nombre de función (517AH).

Un usuario definió función difiere de una expresión normal en el respeto de único, tiene su propio set del local Variables. Estas Variables son creadas en PARM1 cuando la función es invocada y desaparezca cuando termina. Cuando una búsqueda Variable normal es iniciada por el Evaluatodor Expression la región examinada es el Storage Variable Area. Sin embargo, si NOFUNS es poco cero, entonces indicando al menos una función activa del usuario, PARM1 será registrado en lugar de eso, sólo si esto yerra voluntad el movimiento de búsqueda adelante para el Variables global en el Storage Variable Area. Usar un local área Variable específico para cada invocación de una función quiere decir que los mismos nombres Variables pueden ser usados completamente sin los Variables sobrescribiendo a cada quien o el Variables global.

Vale notar que un usuario definió función es más lento que una expresión del inline o empareje una subrutina. La búsqueda ejecutada para encontrar el nombre de función Variable, y la cantidad grande de hacinamiento y destacking, son gastos generales significativos.

La dirección ... 5189H

Estos movimientos de rutina por los que un bloque de memoria de la dirección apuntó registran par Delaware para tan apuntado para por par de registro HL, el par de registro AC define el largo.

La dirección ... 5193H

Esta rutina genera un error "directo ilegal" si CURLIN muestra al Interpreter estar de modo directo.

La dirección ... 51A1H

Esta rutina comprueba el texto de programa para un "FN" señal (DEH) y luego crea el nombre de función Variable (5EA9H). Estos son distinguidos de Variables ordinarias habiendo mordido 7 estar listo en el primer carácter de nombre de Variable.

- 135 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 51ADH

El control se traslada para esta rutina del punto de ejecución Runloop (4640H) si una señal más gran que D8H es encontrado al principio de una declaración. Si la señal no es, entonces un FFH prefijó valor simbólico de función un "error de sintaxis" es generado (4055H). Si la señal de función es una de esas que se doblan como las transferencias de control de declaraciones para el manipulador pertinente, de otra manera un "error de sintaxis" es generado. Las declaraciones en duda son "MID\$" (696EH), "STRIG" (77BFH) y "intervalo" (77B1H). Hay realmente ninguna señal separada para "el INTERVALO", el "INT" señal (85H) sacia con los personajes restantes siendo al que se dio jaque por el manipulador de declaración.

La dirección ... 51C9H

Éste es la "anchura" manipulador de declaración. El operand es evaluada (521CH) y su magnitud comprobada. Si es cero o más gran que treinta y dos o cuarenta, entonces depender del modo de la pantalla sostuvo en OLDSCR una "llamada ilegal de función" error es generada

(475AH). Si es lo mismo que la corriente contenta de LINLEN, entonces la rutina termina sin fomenta acción. De otra manera la pantalla en uso en estos momentos es despejada con un código de control FORMFEED (0CH) por la rutina del estándar OUTDO en caso la pantalla debe ser a la que se hizo más pequeño. El operand es luego colocado en LINLEN y ya sea LINL32 o LINL40, dependiendo de la pantalla que el modo mantuvo en OLDSCR, y que la pantalla despejó otra vez en caso ha estado hecha más grande. Porque el largo de la línea variable para estar cambiado es seleccionado por OLDSCR, en vez de SCRMOD, la anchura todavía puede variarse aun si la pantalla está actualmente de Modo Graphics o el Multicolor Mode. En este caso el cambio es efectivo cuando un regreso hace al Interpreter Mainloop o una "entrada" declaración es ejecutada.

La dirección ... 520EH

Esta rutina evalúa la siguiente expresión en el texto de programa (4C64H), le convierte a ella para un entero (2F8AH) y coloca el resultado en par de registro Delaware. La magnitud y signo del MSB son entonces probados y la rutina termina.

La dirección ... 521BH

Esta rutina evalúa el siguiente operand en el texto de programa (4C64H) y la convierte para un entero (5212H). Si el operand es mayor que 255 que un error "ilegal de llamada de función" es generado (475AH).

La dirección ... 5229H

Éste es el "LLIST" manipulador de declaración. PRTFLG está listo para 01H, a dirigir salida para la impresora, y controlar cae en la "lista" manipulador de declaración.

- 136 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 522EH

Éste es la "lista" manipulador de declaración. El principio optativo y operands de número de la línea de terminación son coleccionados y la puesta en marcha que la colocación encontró en el texto de programa (4279H). El rayado sucesivo de programa está listado hasta que el enlace de fin es encontrado, la llave CTRL-STOP se aprieta o el número de la línea de terminación es alcanzado, el control luego transfiere directamente para el punto Mainloop "Oklahoma" (411FH). Cada línea de programa está listada exhibiendo su número de la línea (3412H), detokenizing (5284H) y exhibiendo (527BH) la línea misma y luego emitiendo un RETORNO DE CARRO, LF (7328H).

La dirección ... 5284H

Esta rutina es usada por la "lista" manipulador de declaración para convertir una línea de programa del tokenized a la forma textual. En las proposiciones del par de registro de entrada HL para el primer carácter de la línea del tokenized. En salida la línea de texto está en BUF y es terminada por un byte de cero.

Cualquier normalidad o FFH prefijó valor simbólico es convertido a la palabra clave correspondiente por una búsqueda lineal simple de las señales en la mesa en 3A72H. Las excepciones están hechas si ya sea un carácter de cita de la abertura, un "REM" señal, o una "DATA" señal previamente ha sido encontrada. Normalmente estas señales serán seguidas por

el texto puro de cualquier manera, el cheque está hecho para detener personajes de gráficos siendo interpretado como señales. La secuencia de tres bytes "" (3AH), "REM" señal (8FH), señal (E6H) es convertido al solo carácter (27H) y el separador de declaración (3AH) precediendo una señal "distinta" (A1H) es al que se limpió restregando.

Si una de las señales numéricas es encontrada, entonces su valor y su tipo son primero emulado de CONLO y CONTYP para DAC y VALTYP (46E8H). Es luego convertido a forma textual en FBUFFR por lo decimal (3425H), octal (371EH) o rutinas de conversión hex (3722H). Para octal y los tipos hex el número es prefijado por un ampersand y una "O" o "H" carta. Un sufijo de tipo, "o", se agrega para singularizar números de precisión o de precisión doble sólo si no hay parte decimal y ninguna parte del exponente (" E "o" D ").

La dirección ... 53E2H

Éste es el "DELETE" manipulador de declaración. El principio optativo y operands de número de la línea de terminación son coleccionados y la puesta en marcha que la colocación encontró en el texto de programa (4279H). Si cualquier punteros existen en el texto de programa, entonces se convierten de regreso a revestir números (54EAH). La línea de programa que termina es encontrada por una búsqueda del texto de programa (4295H), si esta dirección es más pequeña que eso del programa de puesta en marcha reviste un "llamado de pájaro ilegal de función" error es generado (475AH), de otra manera el mensaje "bien" es exteriorizado (6678H). El bloque de memoria del fin de la línea que termina para el principio del Storage Variable Area es reproducido con mucha exactitud al comienzo de la línea de salida y VARTAB, ARYTAB y STREND son vueltos a arrancar el nuevo (muévase hacia abajo)

- 137 -

5. EL INTÉRPRETE DE BASIC ROM

El fin del texto de programa. El control luego se traslada directamente hasta el fin del Mainloop (4237H) para volver a arrancar los punteros restantes y reconectar el Contorno Program Text. Note eso, porque el control no regrese a la normalidad "apruebe" punto, la pantalla no será devuelto a modo del texto. Si la pantalla está de Modo Graphics o modo Multicolor cuándo un "DELETE" es ajustado, entonces cuál es reconocidamente más bien improbable, el sistema se derrumbará.

La dirección ... 541CH

Esta rutina es usada por el Evaluador Factor para ejercer el "atisbo" función para un operand contenido en DAC. El operand de la dirección es comprobado (5439H) luego la lectura de byte de memoria y colocado en DAC como un entero (4FCFH).

La dirección ... 5423H

Éste es el "pinchazo" manipulador de declaración. El operand de la dirección es evaluado (542FH) luego el operand de datos evaluado (521CH) y está escrito para la memoria.

La dirección ... 542FH

Esta rutina evalúa el siguiente operand en el texto de programa (4C64H) y coloca ella en par de registro Delaware como un entero (5439H).

La dirección ... 5439H

Estos conversos de rutina que el operand numérico contuvo en DAC en un entero en registro

parean a HL. El operand debe estar del rango - 32768 para + 65535 y son normalmente una dirección según lo solicitado por "pinchazo", "atisbo", "BLOAD", etc. El tipo de operand es primero a cuadros por la rutina del estándar GETYPR, si es ya un entero que es simplemente colocado en el par de registro HL (2F8AH).

Asumir el operand es sola precisión o precisión doble su signo es comprobada, si es negativo, entonces sea convertido a entero (2F8AH). De otra manera se convierte para singularizar precisión (2FB2H) y su magnitud comprobada (2F21H). Si es mayor que 32767 y más pequeño que 65536 luego - 65536 se agregan (324EH) antes de que sea convertido a entero (2F8AH).

La dirección ... 5468H

Éste es el "RENUM" manipulador de declaración. Si un operand inicial nuevo de número de la línea existe es coleccionado (475FH), entonces de otra manera un valor predeterminado de diez de tomado. Si un operand inicial viejo de número de la línea existe es coleccionado (475FH), diferente, entonces un valor predeterminado de cero sea tomado. Si un operand de número de la línea de incremento existe es coleccionado (4769H), diferente, entonces un valor predeterminado de diez sea tomado.

El texto de programa es entonces registrado para números de la línea existentes iguales para o más gran que el número inicial nuevo (4295H) de la línea

- 138 -

5. EL INTÉRPRETE DE BASIC ROM

Y el viejo número inicial (4295H) de la línea, una "llamada ilegal de función" error es generado (475AH) si la dirección nueva es más pequeña que la dirección vieja. Esto debe percibir cualquier intento en volver a numerar rayado alto de programa abajo para los bajo existentes.

Una carrera falsa que vuelve a numerar del texto de programa es primero efectuada a inspeccionar que ningún número nuevo de la línea será generado con un valor mayor que 65529. Esto debe estar hecho como un error a medio camino a través de la conversión dejaría el texto de programa en una condición confundida. Asumir todo está sano que cualquier operands de número de la línea en el texto de programa son convertidos a punteros (54F6H). Esto pulcramente soluciona el problema de referencias de número de la línea, GOTO 50 por ejemplo, como el texto de programa no es movido durante volver a numerar. Echando a andar en la colocación inicial vieja del texto de programa cada número existente de la línea de programa es reemplazado con su valor nuevo. Cuando el enlace de fin es alcanzado cualquier punteros del texto de programa se convierten de regreso a revestir número operands (54F1H) y el control transfiere directamente para el punto Mainloop "Oklahoma" (411EH).

La dirección ... 54F6H

Estando entrada en 54F6H esta rutina convierte cada operand de número de la línea en el texto de programa a un puntero. Estando entrado en 54F7H realiza la operación inversa y convierte cada puntero en el texto de programa de regreso a un operand de número de la línea. Empezando desde el principio del Program Text Area cada línea es examinado para una señal del puntero (0DH) o una señal del operand de número de la línea (0EH) dependiendo del modo. En puntero para revestir modo del operand de número el puntero es reemplazado por el número de la línea de la línea a la que se puso notas de programa y que la señal cambió para 0EH. En operand de número de la línea para el modo del puntero el texto de programa es registrado (4295H) para encontrar la línea pertinente, su dirección reemplaza el operand de número de la línea y la señal se varía para 0DH. Si la búsqueda está sin éxito que un mensaje de la forma "línea indefinido NNNN en NNNN" es exhibido (6678H) y el proceso de conversión continúa. Un cheque especial es al que se partió rumbo lo "en para ERROR GOTO 0" para declaración, a

impedir la generación de un mensaje espurio de error, pero ningún cheque está hecho para el "curriculum vitae" similar "0" declaración. En este caso un mensaje de error será exteriorizado, éste debería estar ignorado.

La dirección ... 555AH

Éste es el mensaje de texto puro "línea indefinida" terminada por un byte de cero.

La dirección ... 558CH

El nombre..... SYNCHR

La entrada..... Las proposiciones HL para el carácter a comprobar

La salida..... Un = Siguiente carácter de programa

Modifica.. AF, HL

La rutina estándar para comprobar el carácter coetáneo del texto de programa, a cuya dirección está suministrada en par de registro HL

- 139 -

5. EL INTÉRPRETE DE BASIC ROM

En contra de un carácter remisivo. El carácter remisivo es suministrado como un solo byte inmediatamente después de la LLAMADA o la instrucción RST, por ejemplo:

```
RST 08H
DEFB
```

Si los personajes no hacen juego con un "error de sintaxis" es generado (4055H), entonces las transferencias diferentes de control para la rutina del estándar CHRGR para ir a traer el siguiente personaje de programa (4666H).

La dirección ... 5597H

El nombre..... GETYPR

La entrada..... Ninguno

La salida..... AF = Escriba A Máquina

Modifica.. AF

La rutina estándar para devolver el tipo del operand actual, decidido por VALTYP, como sigue:

```
El entero.....Un = FFH, Flag M, NZ, C
La cuerda.....Un = 00H, Flag P, Z, C
Singularice a Precision ... UN = 01H, Flag P, NZ, C
Duplique Precisión ... UN = 05H, Flag P, NZ, Carolina Del Norte
```

La dirección ... 55A8H

Éste es la "llamada" manipulador de declaración. El nombre extendido de declaración, cuál es una cuerda no citada hasta quince personajes por mucho tiempo ultimados por uno "(", "" O el fin de carácter de la línea (00H), es primero copiado del texto de programa para PROCNM, cualquier bytes sin uso son cero lleno. El bit 5 de cada entrada en SLTATR es entonces examinado para una extensión ROM con un manipulador de declaración. Si un ROM adecuado es encontrado, entonces su colocación en SLTATR es convertida a un Slot Idaho en A de registro y una dirección de la base ROM en registro H (7E2AH). La dirección del manipulador de a declaración le son leídos de posiciones ROM cuatro cinco (7E1AH) y están posado en par de registro IX. El

Slot Idaho es colocado en el byte alto de par de registro IY y el ROM que el manipulador de declaración llamó por la rutina del estándar CALSLT.

El ROM examinará el nombre de declaración y C de regreso Flag si no lo reconoce, de otra manera realiza la operación requerida. Si la llamada ROM no se logra, entonces la búsqueda de SLTATR continúa hasta que la mesa está agotada después de lo cual un "error de sintaxis" es generado (4055H). Si la llamada ROM tiene éxito el manipulador termina.

La dirección ... 55F8H

Esta rutina es usada por el analizador gramatical de nombre del dispositivo (6F15H) cuando no puede reconocer un dispositivo que el nombre encontró en el texto de programa. En registro de entrada las proposiciones del par HL para el primer carácter del nombre y los agarres de la B de registro su largo. El nombre es primero emulado para PROCNM y terminado por un byte de cero. El Bit 6 De Cada Uno

- 140 -

5. EL INTÉRPRETE DE BASIC ROM

La entrada en SLTATR es entonces examinada para una extensión ROM con un manipulador del dispositivo. Si un ROM adecuado es encontrado, entonces su colocación en SLTATR es convertida a un Slot Idaho en A de registro y una dirección de la base ROM en registro H (7E2AH). La dirección del manipulador del a dispositivo le son leídos de posiciones ROM seis siete (7E1AH) y están posado en par de registro IX. El Slot Idaho es colocado en el byte alto de par de registro IY, el código desconocido (FFH) del dispositivo en A de registro y el ROM que el manipulador del dispositivo llamó por la rutina del estándar CALSLT.

El ROM examinará el nombre del dispositivo y C de regreso Flag si no lo reconoce, de otra manera devuelve su propio código interno de cero a tres. Si la llamada ROM no se logra, entonces la búsqueda de SLTATR continúa hasta que la mesa está agotada después de lo cual un "mal nombre del archivo" error es generado (6E6BH). Si la llamada ROM tiene éxito el código interno de ROM está añadido para su SLTATR la colocación, entonces multiplicado por un factor de cuatro, para producir un código global del dispositivo ' El código bajo pues cada entrada en SLTATR es exteriorizado en hexadecimal. Los "SS" y "postdata" rotuladores muestran los correspondientes números Secundarios y Primarios Slot, cada ranura está compuesta de cuatro páginas:

```
SS0 SS1 SS2 SS3
+
| 00 04 08 0C | 10 14 18 1C | 20 24 28 2C | 30 34 38 3C PS0
+
| 40 44 48 4C | 50 54 58 5C | 60 64 68 6C | 70 74 78 7C PS1
+
| 80 84 88 8C | 90 94 98 | el PS2 de CORRIENTE ALTERNA 9C A0 A4 A8 | B0 B4 B8 BC
+
| La CD de la COPIA C0 C4 C8 | D0 D4 D8 | E0 E4 E8 EC | F0 F4 F8 FC | PS3
+
```

Figura 44: El dispositivo Codes

El código global del dispositivo es usado por el Interpreter hasta que el tiempo venga por que el ROM realice una operación real del dispositivo. Se convierte luego de vuelta a Slot Idaho de ROM, dirección bajo y código interno del dispositivo para realizar el acceso ROM. Noto que los códigos de 0 para 8 son reservados para

identificadores de la unidad de disco y esos de FCH para FFH para el dispositivos estándar GRP, CRT, LPT y CAS. Con el hardware MSX en uso en estos momentos estructuran estos códigos sea propio físicamente configuraciones improbables ROM y sea por consiguiente caja fuerte para servir para propósitos específicos por el Interpreter.

La dirección ... 564AH

Esta rutina es usada por el repartidor de función (6F8FH) cuando no encuentra un código del dispositivo formando parte de uno de los dispositivos estándar. El código del dispositivo es primero convertido a una colocación SLTATR y luego para un Slot Idaho en A de registro y la dirección de la base ROM en registro H (7E2DH). El manipulador del dispositivo ROM

- 141 -

5. EL INTÉRPRETE DE BASIC ROM

A la dirección le son leídos de posiciones ROM seis siete (7E1AH) y están posado en par de registro IX. El Slot Idaho es colocado en el byte alto de par de registro IY, el código interno del dispositivo de ROM en DISPOSITIVO y el ROM que el manipulador del dispositivo llamó por la rutina del estándar CALSLT.

La dirección ... 566CH

Este punto de entrada para el analizador gramatical de lenguaje de macro es usado por el "empate" manipulador de declaración, un posterior punto de entrada (56A2H) es usado por el "juego" manipulador de declaración. La cuerda de orden es evaluada (4C64H) y su almacenamiento liberado (67D0H). Después de empujar un bloque de terminación de cero encima de la pila Z80 el pedazo y la dirección del cuerpo de la cuerda es colocado en MCLLEN y MCLPTR y controle caídas en el mainloop de analizador gramatical.

La dirección ... 56A2H

Éste es el analizador gramatical de lenguaje de macro mainloop, se usa para tramitar la cuerda de orden asociada con un "empate" o "juego" declaración ' En la entrada el largo de la cuerda está en MCLLEN, la dirección de la cuerda está en MCLPTR y la dirección de la mesa pertinente de orden está en MCLTAB. Las mesas de orden contienen las cartas legales de orden, conjuntamente con las direcciones asociadas del manipulador de orden, para cada declaración. El "empate" mesa está en 5D83H y el "juego" mesa en 752EH.

La primera parte del mainloop de analizador gramatical va a traer al siguiente personaje de la cuerda de orden (56EEH). Si hay ningún más izquierda de personajes el siguiente descriptor de la cuerda es a la que se hizo estallar de la pila (568CH). Si esto es ponga en el cero, entonces el analizador gramatical termina (5709H) si MCLFLG se ve un "empate" declaración para ser activo, el control diferente se traslada de regreso al "juego" manipulador de declaración (7494H).

Asumir un carácter de orden existe que la mesa de orden en uso en estos momentos es registrada para comprobar su legalidad, si ningún encuentro es encontrado que una "llamada ilegal de función" error es generada (475AH). La entrada de la mesa de orden es entonces examinada, si es mordida 7 es determinado que la orden toma un parámetro numérico optativo. Si esto está presente, entonces es coleccionado y colocado en par de registro Delaware (571CH), de otra manera un valor predeterminado de uno es tomado. Después de empujar una declaración de impuestos para el principio del mainloop de analizador gramatical encima del Z80 que el control de la pila transfiere para el manipulador de orden en la dirección tomada de la mesa de orden.

La dirección ... 56EEH

Esta rutina es usada por el analizador gramatical de lenguaje de macro para ir a traer al siguiente personaje de la cuerda de orden. Si MCLLEN es ponga en el cero, entonces la rutina termina con Flag Z, no hay personajes a la izquierda. De otra manera el siguiente carácter es tomado de la dirección contenida en MCLPTR y regresado en A de registro, si el carácter es letra minúscula que es convertida a la caja alta. MCLPTR es entonces incrementado y decremento MCLLEN Ed.

- 142 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 570BH

Esta rutina es usada por el analizador gramatical de lenguaje de macro para devolver un carácter no deseado a la cuerda de orden. MCLLEN es incrementado y MCLPTR decremented.

La dirección ... 5719H

Esta rutina es usada por el analizador gramatical de lenguaje de macro para coleccionar un parámetro numérico de la cuerda de orden. El resultado es un entero indicado a través de señas y es devuelto en par de registro Delaware, no puede ser una expresión. El primer carácter es tomado y examinado, si es uno que eso está ignorado y el siguiente carácter tomado (5719H). Si es uno "-" un regreso es establecido a la rutina de negación (5795H) y el siguiente carácter tomado (5719H). Si es uno el valor del siguiente Variable es devuelto (577AH). Los personajes sucesivos diferentes son timados y un producto binario acumulado hasta que un carácter poco numérico sea encontrado.

La dirección ... 575AH

Esta rutina es usada por el analizador gramatical de lenguaje de macro y "X" manipuladores. El nombre Variable es emulado para BUF hasta lo "" El delimitador es encontrado, si esto toma más que treinta y nueve personajes para encuentra una "función ilegal llamen" error es generado (475AH). Las transferencias diferentes de control para el manipulador del Evaluador Factor Variable (4E9BH) y los contenidos Variables son devueltos en DAC.

La dirección ... 577AH

Esta rutina es usada por el analizador gramatical de lenguaje de macro para tramitar el carácter en un parámetro de orden. El valor de Variable es obtenido (575AH), convertido para un entero (2F8AH) y está posado en par de registro Delaware.

La dirección ... 5782H

Esta rutina es usada por el analizador gramatical de lenguaje de macro para tramitar a la "X" orden. El Variable es procesado (575AH) y, después de comprobar que el espacio de la pila es disponible (625EH), los contenidos coetáneos de MCLLEN y MCLPTR son apilados. El control luego se traslada para el punto de entrada de analizador gramatical (5679H) para obtener el descriptor de Variable y procesar la cuerda nueva de orden.

La dirección ... 579CH

Esta rutina es usada por declaraciones diversas de gráficos para evaluar un par coordinado en

el texto de programa. Las coordenadas deben ser parenthesized con una coma distanciando los operands componentes. Si el par coordinado es precedido por un "paso" señal (DCH) cada valor componente se agrega para el componente correspondiente de las coordenadas coetáneas de gráficos en GRPACX y GRPACY, entonces de otra manera los valores absolutos son devueltos. La X

- 143 -

5. EL INTÉRPRETE DE BASIC ROM

La coordenada es devuelta en GRPACX, GXPOS y registro parean AC.
La coordenada de la Y es devuelta en GRPACY, GYPOS y registro parean Delaware.

Hay dos proposiciones de entrada para la rutina, el que es usada depende en si la persona que llama está esperando más que un par coordinado. La "línea" declaración, por ejemplo, espera dos pares coordinados la primera parte de la cual sea lo más flexibles. El punto de entrada en 579CH se usa para coleccionar el primer par coordinado y los aceptará a los personajes "-" o "@" "-" tan representantes las coordenadas coetáneas de gráficos. El punto de entrada en 57ABH sirve para el segundo par coordinado y requiere un operand explícito.

La dirección ... 57E5H

Éste es el manipulador "preprogramado" de declaración. El color de fondo actual es tomado de BAKCLR y el control cae en el "PSET" manipulador.

La dirección ... 57EAH

Éste es el "PSET" manipulador de declaración. Después de que el par coordinado haya sido evaluado (57ABH) el color actual del primer plano es tomado de FORCLR y es utilizado como el incumplimiento cuándo colocando el color de tinta (5850H). Las coordenadas coetáneas de gráficos son convertidas a una dirección física, por el SCALXY y rutinas del estándar MAPXYC, y el color del pixel en uso en estos momentos se sedimenta por la rutina del estándar SETC.

La dirección ... 5803H

Esta rutina es usada por el Evaluatodor Factor para ejercer el "punto" función. Los contenidos coetáneos de CLOC, CMASK, GYPOS, GXPOS, GRPACY y GRPACX son apilados y el par de coordenada operand evaluado (57ABH). El color del pixel nuevo es leído por el SCALXY, MAPXYC y rutinas del estándar READC y colocado en DAC como un entero (2F99H), los valores coordinados viejos son entonces a los que se hizo estallar y recuperados. Note ese un valor de -1 es devuelto si las coordenadas del punto están fuera de la pantalla.

La dirección ... 5850H

Esta rutina de gráficos se usa para evaluar un operand optativo de color en el texto de programa y hacerle a ella el color de tinta de corriente. Después de comprobar el modo de la pantalla (59BCH) el operand de color es evaluado (521CH) y colocado en ATRBYT. Si ningún operand existe, entonces el color que el código suministró en A de registro es colocado en ATRBYT en lugar de eso.

La dirección ... 5871H

Estos gráficos que la rutina devuelve la diferencia entre los contenidos de GXPOS y este registro parean AC en par de registro HL. Si

5. EL INTÉRPRETE DE BASIC ROM

El resultado es negativo (GXPOS<BC) que es negado para producir la magnitud absoluta y la C Flag es devuelta.

La dirección ... 5883H

Estos gráficos que la rutina devuelve la diferencia entre los contenidos de GYPOS y este registro parean Delaware en par de registro HL. Si el resultado es negativo (GYPOS<DE), entonces es negado para producir la magnitud absoluta y la C Flag es devuelta.

La dirección ... 588EH

Esta rutina de gráficos intercambia los contenidos de GYPOS y el par de registro Delaware.

La dirección ... 5898H

Esta primera parte de rutina de gráficos intercambia los contenidos de GYPOS y el par de registro Delaware (588EH) luego intercambia los contenidos de GXPOS y el par de registro BC. Estando entrada en 589BH sólo la segunda operación es realizada.

La dirección ... 58A7H

Éste es la "línea" manipulador de declaración. La primera coordenada que el par (X1, Y1) es evaluado (579CH) y acomodado en registro pareo a BC, Delaware. Después de inspeccionar para lo "-" la señal (F2H) el segundo par coordinado (X2, Y2) es evaluada (57ABH) e izquierda en GRPACX, GRPACY y GXPOS, GYPOS. Después de colocar el color de tinta (584DH) el texto de programa es comprobado para una siguiente "B" o "BF" opción y ya sea la caja (5912H), boxfill (58BFH) o la operación linedraw (58FCH) realizada. Ninguno de estos afectos de operaciones los gráficos actuales coordinan en GRPACX y GRPACY, éstos es dejado en X2, Y2.

La dirección ... 58BFH

Esta rutina realiza la operación del boxfill. Como quiera que la coordenada abastecida que los pares definen diagonalmente a los que se opuso enseñar con el dedo de la caja dos cantidades deben estar derivativo de ellas. El tamaño horizontal de la caja es obtenido de la diferencia entre X1 y X2, éste da el número de pixels para colocar por fila. El tamaño vertical es obtenido de la diferencia entre Y1 y Y2 dando el número de filas requeridas. Comenzando en la dirección física de X1, Y1, y mudándose sucesivamente más bajo por la rutina del estándar DOWNC, el número requerido de filas del pixel es rellenado por el repetido uso de la rutina del estándar NSETCX.

La dirección ... 58FCH

Esta rutina realiza la operación del linedraw. Después de sacar el línea (593CH) GXPOS y GYPOS es vuelta a arrancar para X2, Y2 de GRPACX y GRPACY.

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 5912H

Esta rutina realiza la operación de la caja. La caja se produce trazando una línea (58FCH) entre cada uno de las cuatro proposiciones rinconeras. Las coordenadas de cada esquina están derivadas de los operands iniciales intercambiando el componente pertinente del par. La secuencia del dibujo es:

(1) X1, Y2 Para X2, Y2 (2) X1, Y1 Para X2, Y1 (3) X2, Y1 Para X2, Y2 (4) X1, Y1 Para X1, Y2

La dirección ... 593CH

Esta rutina saca una línea entre el proposiciones X1, Y1, abastecida en pares de registro BC y Delaware y X2, Y2, abastecido en GXPOS y GYPOS. La operación del dibujo mainloop (5993H) es más conveniente ilustrada por un ejemplo, diga LÍNEA (0,0) - (10,4). Para alcanzar el punto de fin de la línea de su decena de principio los pasos horizontales (X2-X1) y lo cuatro hacia abajo los pasos (Y2-Y1) deben ser tomados enteramente.

La mejor aproximación para una línea recta por consiguiente requiere dos y unos pasos medio horizontales para cada paso descendente (X2-X1/Y2-Y1). Mientras esto es imposible en la práctica, como los pasos sólo integrales pueden ser tomados, la proporción correcta puede ser lograda por término medio.

El método utilizado es añadir la diferencia de la Y para un mueble mostrador cada vez que un paso del rightward es tomado. Cuando el mueble mostrador excede el valor de la diferencia de la X es puesto a cero y uno descendente que el paso es tomado, éste está de efecto una división de entero de los dos valores de diferencia. Algunas veces los pasos descendentes se producirán cada dos pasos del rightward y algunas veces cada tres pasos del rightward. El promedio, sin embargo, será un paso descendente cada dos y un medio rightward dan un paso. Un programa equivalente de BASIC es mostrado debajo con uno ligeramente línea de BASIC de offset para la comparación:

```
10 OCULTAN 0
20 INPUT"START X, Y ";X1, Y1
30 INPUT"END X, Y ", X2, Y2
40 OCULTAN 2
50 X = X1:Y = Y1:X2-X1 L:Y2-Y1 S:CTR = L/2
60 PSET (X, Y)
70 CTR=CTR+S:SI CTR<L LUEGO 90
80 CTR CTR-L:Y=Y + 1
90 X=X + 1:IF X <= X2 THEN 60
100 REVISTEN (X1, Y1+5) - (X2, Y2+5)
110 GOTO 110
```

El citado anteriormente ejemplo padece de tres limitaciones. La línea debe inclinarse hacia abajo, debe inclinarse a la derecha y la cuesta no puede exceder cuarenta y cinco grados de lo horizontal (un paso descendente para un paso del rightward).

- 146 -

5. EL INTÉRPRETE DE BASIC ROM

La rutina vence la primera limitación examinando al Y1 y Y2 se coordina antes de que el dibujo

comienza. Si Y2 es mayor o corresponda a para Y1, entonces acompañando la línea a cuesta arriba o a ser horizontal, ambos los pares coordinados son intercambiados. La línea ahora se inclina hacia abajo y será sacada del punto de fin para el principio.

La segunda limitación se subsana examinando a X1 y X2 de antemano para determinar por dónde la línea son inclinarse. Si X2 es mayor o igual para X1 la línea se inclina a la derecha y un Z80 JP para la rutina del estándar RIGHTC es colocado en MINUPD/MAXUPD (vea debajo) para el uso por el dibujo mainloop, entonces de otra manera un JP para la rutina del estándar LEFTC es colocado allí.

La tercera limitación se subsana comparando la diferencia de coordenada de la X para la diferencia de coordenada de la Y antes de sacando a determinar el empinamiento de la cuesta. Si $X2-X1$ es más pequeño que $Y2-Y1$, entonces la cuesta de la línea está menos de cuarenta y cinco grados de lo horizontal. El método simple y exteriorizado citado anteriormente para LÍNEA (0,0) - (10,4) no trabajará para cuestas más gran que cuarenta y cinco grados como la tasa máxima de descenso sea lograda cuando un paso descendente es tomado por cada paso horizontal. Surtirá efecto sin embargo si las instrucciones de paso están permutadas. Así la LÍNEA (0,0) - (4,10) precisa un paso del rightward para cada dos y unos pasos medio descendentes. MINUPD cree que un Z80 JP para la rutina "normal" del estándar de dirección de paso para el mainloop del dibujo y MAXUPD le haga a un JP cumplir la "cuesta" rutina del estándar de dirección de paso. Pues ángulos poco hondo MINUPD lega vector para DOWNC y MAXUPD para LEFTC o RIGHTC. Para ángulos de la pendiente MINUPD lega vector para LEFTC o RIGHTC y MAXUPD para DOWNC. Pues la pendiente pesca con caña el mueble mostrador aprecia también debe ser intercambiado, la diferencia de la X ahora debe agregarse para el mueble mostrador y la Y que la diferencia usó como el acabóse del mueble mostrador. El variable MINDEL y MAXDEL son usados por el mainloop del dibujo para mantener estos valores del mueble mostrador, MINDEL los sostiene mientras más pequeño diferencia del punto de fin y MAXDEL mayor.

Un punto interesante es que el mueble mostrador remisivo, mantenido en CTR en el citado anteriormente programa y en par de registro Delaware en el ROM, es precargado medio la diferencia más grande del punto de fin en vez de ser colocado para poner en el cero. Esto tiene como consecuencia dividiendo la primera "escalera" en la línea en dos capítulos, uno al principio de la línea y uno en su fin, y mejorando la apariencia de line's.

La dirección ... 59B4H

Estos gráficos que la rutina intercambia los contenidos de mismo del par de registro Delaware mordieron a la derecha.

La dirección ... 59BCH

Esta rutina genera una "llamada ilegal de función" error (475AH) si la pantalla no está de Modo Graphics o el Multicolor Mode.

- 147 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 59C5H

Éste es la "pintura" manipulador de declaración. El par de coordenada de puesta en marcha es evaluado (579CH), el color de tinta colocado (584DH) y la demarcación optativa que el color el operand evaluó (521CH) y colocó en BDRATR. El par de coordenada de puesta en marcha es comprobado para asegurar que está dentro de la pantalla (5E91H) y está hecho la dirección física del pixel en uso en estos momentos por la rutina del estándar MAPXYC. La distancia para

la demarcación de la derecha está entonces medida y, si fuera pone en el cero, el manipulador termina (5ADCH).

De otra manera la distancia para la demarcación de la izquierda está medida (5AEDH) y la suma de lo dos colocado en par de registro Delaware como la anchura de la zona. La corriente dos veces a la que la colocación es entonces apilada (5ACEH), primero con una bandera de terminación (00H) y luego con una bandera caída (40H) de dirección. El control luego se traslada para la pintura mainloop (5A26H) con una bandera levantada (C0H) de dirección en B de registro.

La dirección ... 5A26H

Éste es el mainloop de pintura. La zona que la anchura es sostenida en par de registro Delaware, la dirección de pintura, arriba o abajo de, en B de registro y la dirección física del pixel en uso en estos momentos está eso del pixel adyacente para el límite de la izquierda. Un paso vertical es llevado a la siguiente línea, por el TUPC o las rutinas del estándar TDOWNC, y la distancia a la derecha dan límite medido (5ADCH). La distancia para la demarcación de la izquierda está entonces medidas y la línea entre los confines llenado adentro (5AEDH). Si ningún cambio es encontrado en calidad de ya sea el control de demarcación transfiere para el principio del mainloop mantener pintura en la misma dirección. Si un cambio es encontrado, entonces una inflexión ha ocurrido y la acción apropiada debe ser tomada.

Hay cuatro tipos de incursive de inflexión, LH o RH, dónde los movimientos pertinentes de demarcación hacia dentro, y LH o RH digresivo, donde se muda hacia afuera. Un ejemplo de cada tipo es exteriorizado debajo con zonas numeradas indicando la orden de pintura durante el movimiento ascendente. Una zona secundaria es mostrada dentro de cada región del inflective para la plenitud:

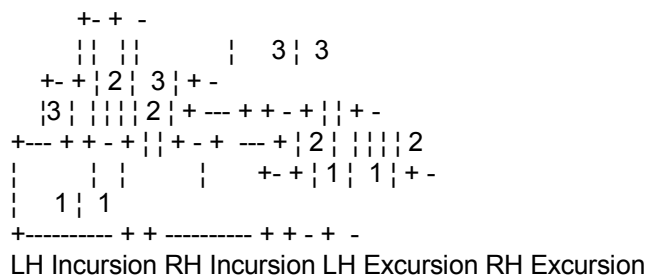


Figura 45: La demarcación Inflections

Una excursión LH ha ocurrido cuando la distancia para la demarcación de la izquierda es poco cero, una excursión RH ha ocurrido cuando lo

5. EL INTÉRPRETE DE BASIC ROM

La anchura coetánea de la zona es mayor que eso de la línea previa. A menos que la excursión esté menos de dos pixels, en cuyo caso será ignorada, que la colocación coetánea (la raíz dejó de zona 3 en figura 45) sea apilada (5AC2H), la pintura que la dirección puso al revés y la pintura se reanuda en la izquierda máxima de la región digresiva.

Una incursión RH ha ocurrido cuando la anchura coetánea de la zona es más pequeña que eso de la línea previa. Si la incursión es total, entonces esto es la anchura coetánea de la zona es cero, una calle sin salida ha sido alcanzada y la última colocación y la dirección son a las que se hizo estallar (5AIFH) y la pintura se reanuda en ese momento. De otra manera la colocación

coetánea y la dirección son apiladas (5AC2H) y la pintura se reanuda en la izquierda más bajo de la región del incursive.

Un LH que la incursión está con la que se trató automáticamente durante la búsqueda para la demarcación de la derecha y no precisa acción explícita por el mainloop de pintura.

La dirección ... 5AC2H

Esta rutina es usada por la "pintura" manipulador de declaración para ahorrar la colocación coetánea de pintura y la dirección en la pila Z80.

El bloque de parámetro de seis bytes está hecho de lo siguiente:

- 2 bytes ... contenidos Coetáneos de CLOC
- 1 byte ... la dirección Coetánea
- 1 byte ... contenidos Coetáneos de CMASK
- 2 bytes ... la anchura de la zona Current

Después de que los parámetros hayan sido apilados un cheque está hecho ese existe suficiente (625EH) de silencio de la espacio pila.

La dirección ... 5ADCH

Esta rutina es usada por la "pintura" manipulador de declaración para hallar la demarcación de la derecha. La anchura de la zona de la línea previa es pasada a la rutina del estándar SCANR en par de registro Delaware, ésta determina el número máximo de pixels de color de demarcación que inicialmente puede ser pasadas por alto. El remanente devuelto de la cuenta de salto es colocado en SKPCNT y el número de poco demarcación que los pixels de color negaron en MOVCNT.

La dirección ... 5AEDH

Esta rutina es usada por la "pintura" manipulador de declaración para hallar la demarcación de la izquierda. El punto de fin de la búsqueda de demarcación de la derecha es por ahora ahorrado y el punto de partida tomado de CSAVEA y CSAVEM y hecho la dirección física del pixel en uso en estos momentos. El límite de la izquierda está luego ubicado por la rutina del estándar SCANL, lo cual también rellena la zona entera, y el cabo de la derecha punto se recuperó y colocó en CSAVEA y CSAVEM.

- 149 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 5B0BH

Esta rutina es usada por el "círculo" manipulador de declaración para negar los contenidos de par de registro Delaware.

La dirección ... 5B11H

Éste es el "círculo" manipulador de declaración. Después de evaluar el par de coordenada del centro (579CH) el radio es evaluado (520FH), multiplicado (325CH) por el PECADO (PI/4) y está posado en CNPNTS. El color de tinta está colocado (584DH), el principio que el ángulo evaluó (5D17H) y colocó en CSTCNT y que el ángulo de fin evaluó (5D17H) y colocó en CENCNT. Si el ángulo de fin es más pequeño que el ángulo de principio los dos valores es intercambiado y

CPL0TF es hecho poco cero. La proporción dimensional es evaluada (4C64H) y, si es mayor que una, su recíproco es tomado (3267H) y CSCLXY es poco cero hecho para indicar un aplastamiento del axis de la X. La proporción dimensional es multiplicada (325CH) a las 256, convertido para un entero (2F8AH) y colocado en el ASPECTO como un solo byte el fragmento binario. Pares de registro HL y Delaware están listos para la colocación de puesta en marcha en el perímetro del círculo (X=RADIUS, Y = 0) y el control cae en el mainloop del círculo.

La dirección ... 5BBDH

Éste es el mainloop del círculo. Por el grado alto de simetría en círculos hay sólo que computar las coordenadas del arco de cero para cuarenta y cinco grados. Los otros siete segmentos son producidos por rotación y reflexión de estas proposiciones. La ecuación del parametric para un círculo de la unidad, con T el ángulo de cero para $\pi/4$, es:

$$\begin{aligned} X &= \cos(T) \\ Y &= \sin(T) \end{aligned}$$

Dirija computación usando esta ecuación, o la forma funcional correspondiente $X = \sqrt{1 - Y^2}$, es demasiado lento, en lugar de eso la primera derivada es usada:

$$\begin{aligned} \frac{dx}{dy} &= -Y/X \end{aligned}$$

Como quiera que la colocación de puesta en marcha es conocida (X=RADIUS, Y = 0), el cambio de coordenada de la X pues cada cambio de coordenada de la Y de la unidad puede ser computado usando la derivada. Además, porque la resolución de gráficos está limitada para un pixel, hay sólo que saber cuándo cumple la suma de los cambios de coordenada de la X unidad y luego para el decremento la coordenada de la X. Por consiguiente:

$$\begin{aligned} \text{La X De Decremento Cuándo } (Y1/X) + (Y2/X) + (Y3/X) + \dots &= > 1 \\ \text{Por consiguiente el decremento cuando } (Y1 Y2 Y3 + \dots) / X &= > 1 \\ \text{Por consiguiente el decremento cuando } Y1 Y2 Y3 + \dots &= > X \end{aligned}$$

Todo lo que está obligado a identificar un cambio de coordenada de la X es totalizar los valores de coordenada de la Y de cada paso hasta la X

- 150 -

5. EL INTÉRPRETE DE BASIC ROM

El valor coordinado es excedido. El mainloop del círculo sostiene la X coordinada en par de registro HL, la Y coordinada en par de registro Delaware y el total corredor en CRCSUM. Un programa equivalente de BASIC para un círculo de radio arbitrario 160 pixels es:

```

10 OCULTAN 2
20 X = 160:Y = 0:CRCSUM = 0
30 PSET (X, 191-Y)
40 CRCSUM=CRCSUM+Y:Y=Y + 1
50 SI CRCSUM<X LUEGO 30
60 CRCSUM CRCSUM-X:X X-1
70 SI X>Y LUEGO 30
80 DAN VUELTAS (0,191), 155
90 GOTO 90

```

Los pares de coordenada generados por el mainloop son esos de un círculo "virtual", tales tareas como la reflexión axial, el tropel elíptico y la traducción del centro son manipulados en un nivel inferior (5C06H).

La dirección ... 5C06H

Esta rutina se usa para por el mainloop del círculo para convertir un par coordinado, en pares de registro HL y Delaware, en ocho proposiciones simétricas en la pantalla. La coordenada de la Y es inicialmente negada (5B0BH), reflectante que ella acerca del axis de la X, y las primeras cuatro proposiciones produjeron por rotaciones en el sentido de las manecillas del reloj sucesivas a través de noventa grados (5C48H). La coordenada de la Y es entonces negada otra vez (5B0BH) y unas adicionalmente cuatro proposiciones producidas (5C48H).

La rotación en el sentido de las manecillas del reloj es realizada intercambiando la X y Y se coordina y negando la coordenada nueva de la Y, así un punto (40,10) vendría bien (10, - 40). Asumiendo una proporción dimensional de 0.5, por ejemplo, la secuencia completa de ocho proposiciones por consiguiente sería:

(1) X, - Y * 0.5 (2) - Y, - X * 0.5 (3) - X, Y * 0.5 (4) Y, X * 0.5 (5) Y, - X * 0.5 (6) - X, - Y * 0.5 (7) - Y, X * 0.5 (8) X, Y * 0.5

Puede verse de lo citado anteriormente ese, ignorando el signo de las coordenadas por el momento, hay sólo cuatro condiciones complejas. Por consiguiente, en vez de realizar la multiplicación de proporción dimensional (5CEBH) relativamente lenta para cada punto, la condiciones X * 0.5 y Y * 0.5 pueden prepararse por adelantado y la secuencia completa generada por ahí intercambiando y negar las cuatro condiciones. Con la proporción dimensional exteriorizada por encima de las condiciones iniciales es establecido a fin de que registra par HL=X, registra par Delaware = - Y * 0.5, CXOFF=Y y CYOFF=X * 0.5 y proposiciones sucesivas se producen por ahí lo

- 151 -

5. EL INTÉRPRETE DE BASIC ROM

Las operaciones:

- (1) Exchange HL y CXOFF, niegan a HL.
- (2) Exchange Delaware y CYOFF, niegan Delaware.

Adentro iguale con la computación de cada coordenada del círculo que el número de proposiciones requirió alcanzar el principio del segmento conteniendo el punto es guardado en CPCNT8. Éste inicialmente será cero y aumentará por 2 RADIUS*SIN (PI/4) como cada uno que rotación de noventa grados está hecha. Como cada uno de las ocho proposiciones se producen su valor de coordenada de la Y se agrega para los contenidos de CPCNT8 y comparado para el principio y el fin ángulos para determinar el curso de la acción apropiado. Si el punto está entre los dos ángulos y CPLOTf es cero, o si está fuera del Anglos y CPLOTf es poco cero, entonces las coordenadas están añadido para las coordenadas del centro del círculo (5CDCH) y el punto determinado por el SCALXY, MAPXYC y rutinas del estándar SETC. Si el punto es igual ya sea de los dos ángulos, y el bit asociado es incrustado en CLINEF, entonces las coordenadas están añadidas para las coordenadas del centro del círculo (5CDCH) y una línea llevada al centro (593CH). Si ninguna de estas condiciones son aplicables ninguna acción está ocupada aparte de para proceder hacia el siguiente punto.

La dirección ... 5CEBH

Esta rutina multiplica el valor coordinado abastecido en par de registro Delaware por la proporción dimensional contenida en el ASPECTO, el resultado es devuelto en par de registro Delaware. El cambio binario estándar y se añade que el método es usado pero la operación es realizada como dos solas multiplicaciones de byte para evitar problemas de excedente.

La dirección ... 5D17H

Esta rutina es usada por el "círculo" manipulador de declaración para convertir un operand del ángulo a que la forma requirió por el círculo mainloop, el resultado es devuelto en par de registro Delaware. Mientras el método usado es básicamente atinado, y elimina una computación trigonométrica por ángulo, los resultados producidos es inexacta. Esto es demostrado por el siguiente ejemplo que saca una línea para el verdadero punto de treinta grados en un perímetro de circle's:

```
10 OCULTAN 2
20 LA PI = 4 * ATN (1)
30 DAN VUELTAS (100,100), 80, PI/6
40 REVISTEN (100,100) - (100+80 * Colorados (PI/6), 100-80 * PEQUE (PI/6))
50 GOTO 50
```

El resultado que la rutina debería producir es el número de proposiciones que deben ser producidas por el mainloop del círculo antes de que el ángulo requerido sea alcanzado. Esto puede ser computado por primero notar que habrá INT (el $\text{ÁNGULO} / (\text{PI}/4)$) segmentos de cuarenta y cinco grados antes del segmento conteniendo el ángulo requerido. Además cada cuarenta y cinco segmento contendrá proposiciones $\text{RADIUS} * \text{SIN} (\text{PI}/4)$ tan

- 152 -

5. EL INTÉRPRETE DE BASIC ROM

Éste es el valor de la coordenada de la Y que termina. Por eso el número de proposiciones requirió cumplir el principio del segmento conteniendo el ángulo es el producto de estos dos números. La cuenta total se produce añadiendo esta figura para el número de proposiciones requeridas a cubrir cualquier ángulo restante dentro del segmento final, que es $\text{RADIUS} * \text{SIN} (\text{PERMANECIENDO} \text{ÁNGULO})$ apunta.

Desafortunadamente la rutina computa el número de proposiciones dentro de un segmento por la aproximación lineal del tamaño total del segmento en la suposición equivocada que las proposiciones sucesivas subtenden ángulos iguales. Así en el citado anteriormente ejemplo el punto que la cuenta computó para el ángulo sea $30/45 * (80 * 0.707107) = 37$ en lugar del valor correcto de cuarenta. El error producido por la rutina está por consiguiente en un máximo en el centro de cada segmento de cuarenta y cinco grados y adelgaza poner en el cero al final apunta.

La dirección ... 5D6EH

Éste es el "empate" manipulador de declaración. El par de registro Delaware está colocado para apuntar hacia la mesa de orden en 5D83H y las transferencias de control para el analizador gramatical de lenguaje de macro (566CH).

La dirección ... 5D83H

Esta mesa contiene las cartas válidas de orden y las direcciones asociadas para el "empate" órdenes de declaración. Esas órdenes que las tomas un parámetro, y consecuentemente tienen, mordieron 7 el círculo en la mesa, son exteriorizados con un asterisco:

CMD PARA

U * 5DB1H
D * 5DB4H
L * 5DB9H
* el 5DBCH de interrogación
M 5DD8H
E * 5DCAH
F * 5DC6H
G * 5DD1H
H * 5DC3H
Un * 5E4EH
B 5E46H
N 5E42H
X 5782H
C * 5E87H
S * 5E59H

La dirección ... 5DB1H

Esto es el "empate" manipulador de orden de declaración "U". La operación de la "D", "L", "interrogación", "E", "F", "G" y "H" órdenes es muy similar así es que ninguna descripción separada de sus manipuladores es dada. El parámetro numérico optativo es suministrado por el analizador gramatical de lenguaje de macro en par de registro Delaware. Este parámetro inicial es

- 153 -

5. EL INTÉRPRETE DE BASIC ROM

Modificado por un manipulador dado en un offset horizontal en registre par BC y un offset vertical en par de registro Delaware. Por ejemplo si hacia la izquierda o el movimiento ascendente es precisado, entonces el parámetro es negado (5B0BH), si el movimiento diagonal es precisado que el parámetro es multiplicado por dos a fin de que igual vertical horizontal y offsets se produce. Una vez que los offsets han sido preparado las transferencias de control para la rutina de dibujo lineal (5DFFH).

La dirección ... 5DD8H

Esto es el "empate" manipulador de orden de declaración "M". El carácter siguiendo la carta de orden es examinado luego los dos parámetros cobrados de la cuerda de orden (5719H). Si el carácter inicial es o "-" los parámetros son considerados como offsets y son modificados a escala (5E66H), alternados a través sucesivos pasos de noventa grados tan determinados por DRWANG y luego sumados a las coordenadas coetáneas (5CDCH) de gráficos para determinar la terminación punto. Si DRWFLG exterioriza a la "B" modo para estar inactivo que una línea es luego sacada de (5CCDH) las coordenadas coetáneas de gráficos para la terminación el punto. Si DRWFLG exterioriza a la "N" modo para estar inactivo, entonces las coordenadas de terminación son colocadas en GRPACX y GRPACY para ir bien con los gráficos actuales nuevos coordina. Finalmente DRWFLG es puesto en el cero, quitar las ganas a la "B" y "N" modos, y el manipulador termina.

La dirección ... 5E42H

Esto es el "empate" manipulador de orden de declaración "N", DRWFLG está simplemente listo a 40H.

La dirección ... 5E46H

Esto es el "empate" manipulador de orden de declaración "B", DRWFLG está simplemente listo a 80H.

La dirección ... 5E4EH

Éste es la "A" de declaración de "empate" manipulador de orden. El parámetro es comprobado para la magnitud y está posado en DRWANG.

La dirección ... 5E59H

Esto es el "empate" manipulador de orden de declaración "S". El parámetro es comprobado para la magnitud y está posado en DRWSCL.

La dirección ... 5E66H

Esta rutina es usada por el "empate" declaración "U", "D", "L", "interrogación", "E", "F", "G", "H" y "M" (en el modo de offset) manipuladores de orden para modificar a escala el offset abastecido en par de registro Delaware por los contenidos de DRWSCL. A menos que DRWSCL es cero, en cuyo caso la rutina simplemente termina, el offset es multiplicado usando repetida adición y luego dividido a las cuatro (59B4H). Para eliminar modificar a escala un "S0" o "S4" orden debería ser usado.

- 154 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 5E87H

Esto es el "empate" manipulador de orden de declaración "C". El parámetro es colocado en ATRBYT por la rutina del estándar SETATR.

No hay cheque en el MSB del parámetro así es que los valores ilegales como "C265" serán aceptados sin un mensaje de error.

La dirección ... 5E91H

Esta rutina es usada por la "pintura" manipulador de declaración a comprobar, por la rutina del estándar SCALXY, que las coordenadas en pares de registro BC y Delaware están dentro de la pantalla. En caso de que no una "llamada ilegal de función" error es generada (475AH).

La dirección ... 5E9FH

Éste es el manipulador "débil" de declaración. Un regreso se establece mecanismo arriba para 5E9AH, tan ese Arrays múltiple puede ser procesado, DIMFLG está hecho que el poco cero y las caídas de control en el Variable investigan rutina.

La dirección ... 5EA4H

Ésta es la rutina Variable de búsqueda. En las proposiciones del par de registro de entrada HL para el primer carácter del nombre Variable en el texto de programa. En registro de la salida las proposiciones del par HL para el carácter siguiendo el nombre y el par de registro Delaware para el primer byte de los contenidos Variables en el Storage Variable Area. El primer carácter del nombre es tomado del texto de programa, comprobado para asegurar que es caja alta alfabética

(64A7H) y colocada en la C de registro. El carácter segundo optativo, con un valor predeterminado de cero, es colocado en registro B, este carácter puede ser alfabético o numérico. Cualquier fomite caracteres alfanuméricos son entonces simplemente saltados encima. Si un carácter de sufijo de tipo ("el %", "\$", "!" o "#") sigue el nombre éste es convertido al tipo correspondiente código (2, 3, 4 o 8) y está posado en VALTYP. De otra manera el tipo predeterminado de Variable es tomado de DEFTBL usando la primera carta del nombre para hallar la entrada apropiada.

SUBFLG es entonces a cuadros para determinar cómo siguiendo cualquier subíndice del parenthesized el nombre debería ser tratado. Esta bandera es normalmente cero pero es modificada por el "ERASE" (01H), "FOR" (64H), "FN" (80H) o "DEF FN" (80H) manipuladores de declaración para obtener a la fuerza un curso de la acción particular. En el "ERASE" caso el control transfere directamente para la rutina de búsqueda Array (5FE8H), ninguna necesidad del subíndice del parenthesized esté presente. En el "FOR", "FN" y "DEF FN" casos monitorean transferencias directamente para la rutina Variable simple (5F08H) de búsqueda, ningún cheque está hecho para un subíndice del parenthesized. Dado que la situación es normal el texto de programa es comprobado para los personajes "(" o "." Si uno u otro es transferencias presentes de control para la rutina de búsqueda Array (5FBAH), las caídas diferentes de control en el Variable simplista investigan rutina.

- 155 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 5F08H

Ésta es la rutina Variable simple de búsqueda. Hay cuatro tipos del simplista Variable que cada uno compuso de un encabezado siguieron por los contenidos Variables. El primer byte del encabezado contiene el código de tipo y los siguientes dos bytes el nombre Variable. Los contenidos del Variable serán una de las formas de tres estándares de numérico o, para el tipo de la cuerda, el largo y dirección de la cuerda. Cada uno de los cuatro tipos son exteriorizados debajo:

```

+
| 02H "A B" LSB|MSB
+
El entero

+
| 03H "A B" LEN|LSB|MSB
+
La cuerda

+
| 04H "A B" EE|DD|DD|DD
+
Singularice a Precision

+
| 08H "A B" EE|DD|DD|DD|DD|DD|DD|DD
+
Duplique Precisión

```

Figura 46: Variables simplista

NOFUNS es primero a cuadros para determinar si un usuario definió función está actualmente

siendo evaluada. Si así es que la búsqueda es efectuada en los contenidos de PARM1 ante todo, sólo si esto yerra voluntad ella múdese encima del Storage Variable principal Area. Un método lineal de búsqueda es usado, los dos personajes de nombre y el byte de tipo de cada Variable en el área de almacenamiento son comparados para los personajes remisivos y el tipo hasta un encuentro es encontrado o el fin del área de almacenamiento es cumplido. Si la búsqueda tiene éxito la rutina termina con la dirección del primer byte de los contenidos Variables en par de registro Delaware. Si la búsqueda está sin éxito, entonces el Array Storage Area es movido arriba y el Variable nuevo se agrega hasta el fin de los existentes e inicializado para poner en el cero.

Hay dos excepciones para esta creación automática de un Variable nuevo. Si la búsqueda está siendo ejecutada por el "VARPTR" función, y esto está resuelto examinando el regreso, entonces la dirección, ningún Variable será creada. En lugar de eso la rutina termina con par de registro Delaware puesto a cero (5F61H) de causar una subsiguiente "llamada ilegal de función" error. La segunda excepción ocurre cuando la búsqueda está siendo ejecutada por el Evaluador Factor, esto es cuando la Variable es recién declarado dentro de una expresión. En este caso DAC es puesto en el cero para tipos numéricos, y está cargado de la dirección de un descriptor falso de largo de cero para un tipo de la cuerda, así devolviendo un resultado de cero (5FA7H). Estos

- 156 -

5. EL INTÉRPRETE DE BASIC ROM

Las acciones son diseñadas para impedir el Evaluador Expression creando a un Variable nuevo ("VARPTR") sea la única función para tomar una discusión Variable directamente en vez de por una expresión y así también precise protección separada). Si ésta no estuviera así bien asignación para un Array, por el manipulador "dejado" de declaración, no se lograría como cualquier simplista Variable creado durante la evaluación de expresión cambiaría la dirección de Array.

La dirección ... 5FBAH

Ésta es la rutina de búsqueda Array. Hay cuatro tipos de Array cada uno serenos de un encabezado y un número de elementos. El primer byte del encabezado contiene el código de tipo, los siguientes dos bytes el nombre Array y lo siguiente dos el offset para el principio del siguiente Array. Esto es seguido por un solo byte conteniendo la dimensionalidad del Array y la lista de la cuenta del elemento. Cada uno que cuenta del elemento de dos bytes contiene el número máximo de elementos por la dimensión. Estos se guardan en orden inverso con lo primer correspondiente al último subíndice.

Los contenidos de cada elemento Array son idénticos para los contenidos del Variable simplista correspondiente. El entero debajo del que el % Array AB (3,4) es mostrado con cada elemento identificó por sus subíndices, la memoria alta está hacia la parte superior de la página:

```

+
|(0,4) (1,4) (2,4) (3,4)
|(0,3) (1,3) (2,3) (3,3)
|(0,2) (1,2) (2,2) (3,2)
|(0,1) (1,1) (2,1) (3,1)
|(0,0) (1,0) (2,0) (3,0)
+

+
| | | | | Imprima En Offset | Cuenta | Oscura De La | Cuenta
| 02H "A B" | 2DH 00H | 02H | 05H 00H | 04H 00H
+

```

Figura 47: El entero Array

Cada subíndice es evaluado, convertido para un entero (4755H) y empujado encima de la pila Z80 hasta que un paréntesis de cierre es encontrado, no necesita hacer juego con lo inaugural. Una búsqueda lineal es luego efectuada en el Array Storage Area para un encuentro con los dos personajes de nombre y el tipo. Si la búsqueda tiene éxito DIMFLG es al que se dio jaque y un "conjunto imponente Redimensioned" error generado (405EH) si muestra una declaración "oscura" para ser activo. A men

os que un "ERASE" declaración es activo, en cuyo caso la rutina termina con par de registro BC apuntando hacia el principio del Array (3297H), la dimensionalidad del Array es entonces a cuadros en contra de la cuenta escrita debaja de una letra y un error "escrito debajo de una letra y fuera del alcance" generado si desencajan. Asumir que estas pruebas son pasó las transferencias de control al punto de computación de la dirección del elemento (607DH).

Si la búsqueda es sin éxito y un "ERASE" declaración es activo, entonces una "llamada ilegal de función" error es generada (475AH)

- 157 -

5. EL INTÉRPRETE DE BASIC ROM

De otra manera el Array nuevo se agrega hasta el fin del Array existente Storage Area. La inicialización del Array nuevo procede almacenando los dos caracteres de nombre, el tipo código y la dimensionalidad (la cuenta escrita debaja de una letra) seguida por la cuenta del elemento para cada dimensión. Si DIMFLG muestra una declaración "oscura" para ser activo, entonces las cuentas del elemento son determinadas por los subíndices.

Si el Array está siendo creado por defecto, entonces con una declaración como "uno (1,2,3) = 5" por ejemplo, un valor predeterminado de once es usado.

Como cada cuenta del elemento es almacenada el tamaño total del Array está acumulado en par de registro Delaware por multiplicaciones sucesivas (314AH) del elemento cuenta y el tamaño del elemento (el tipo Array). Después de que una comprobación que esta cantidad de memoria sea STREND disponible (6267H) es aumentado el área nuevo es puesto en el cero y el tamaño Array es almacenado, adentro ligeramente modificó forma, inmediatamente después de que los dos nombres a los personajes. A menos que el Array esté siendo creado por defecto, en cuyo caso la dirección del elemento debe ser computada, la rutina luego termina.

Éste es el punto de computación de la dirección del elemento de la rutina de búsqueda Array. La posición de un elemento particular dentro de un Array implica la multiplicación (314AH) de subíndices, cuentas del elemento y tamaños del elemento. Como hay una colección variada de formas que esto pudo estar hecho el método real usado es más conveniente ilustrado con un ejemplo. La posición de elemento (1,2,3) en un 4*5*6 Array inicialmente sería computada como $((3*5) + 2) * 4 + 1$.

Esto está luego multiplicado por el tamaño del elemento (el tipo) y añadido para la dirección bajo Array para obtener la dirección del elemento requerido. El método de computación es una forma optimizada que minimiza el número de pasos necesitados, es equivalentes para evaluar $(3 * (4*5)) + (2*4) + (1)$. El elemento que la dirección es devuelta en par de registro Delaware.

La dirección ... 60B1H

Ésta es la "impresión USANDO" manipulador de declaración. Controle transferencias aquí del manipulador "estampado" general de declaración después de que el dispositivo aplicable de salida haya sido establecido. En la terminación el control devuelve para el punto "estampado" general (4AFFH) de la salida de declaración para restaurar la salida de vídeo normal. La cuerda del formato es evaluada (4C65H) y la dirección y largo del cuerpo de la cuerda sacado del descriptor. El perro de muestra del texto de programa se salva luego por ahora. Cada carácter de la cuerda del formato es examinado hasta uno de los personajes posibles de la plantilla es encontrado. Si el carácter no debe estar en una plantilla, entonces es simplemente salida por la

rutina del estándar OUTDO. Una vez que el principio de una plantilla es encontrado esto es escudriñado adelante hasta que un carácter de la poca plantilla es encontrado. El control luego pasa para la rutina numérica (6192H) de salida o la rutina de salida de la cuerda (6211H).

En cualquier caso el puntero del texto de programa está recuperado para registrar par HL y el siguiente operand evaluado (4C64H). Para la salida numérica que la información ganó de la plantilla escudriñe es pasado a la rutina numérica (3426H) de conversión en A de registros, B y C y la cuerda resultante exhibida (6678H). Para

- 158 -

5. EL INTÉRPRETE DE BASIC ROM

La salida de la cuerda la cuenta requerida de carácter es pasada al "LEFT\$" manipulador de declaración (6868H) en C de registro y la cuerda resultante exhibida (667BH). Pues o escribe a máquina de salida que el texto de programa y el formato atan con una cuerda es entonces examinado para determinar si hay cualquier más personajes. Si ninguno de los operands existen, entonces el manipulador termina. Si la cuerda del formato ha estado agotada luego es vuelta a arrancar desde el principio (60BFH), otrora la tomografía continúa de la colocación coetánea para el siguiente operand (60f6H).

La dirección ... 6250H

Esta rutina es usada por el Interpreter Mainloop y el Variable investiga rutina para mover un bloque de memoria arriba. Un cheque está primero hecho para asegurar que suficiente memoria existe (6267H) y luego el bloque de memoria es movido. La fuente máxima a la dirección está suministrada en par de registro BC y el destino máximo dirigen la palabra a en par de registro HL. Copiarse se detiene cuando los contenidos de registro parean AC a igual que esos de registro parean Delaware.

La dirección ... 625EH

Esta rutina se usa para comprobar que suficiente memoria está disponible entre la parte superior del Array Storage Area y la base de la pila Z80. En registro de entrada la C contiene el número de palabras que la persona que llama requiere. Si esto estrecharía la abertura para menos de doscientos bytes uno "por la memoria" el error es generada.

La dirección ... 6286H

Éste es el manipulador "nuevo" de declaración. TRCFLG, AUTFLG y PTRFLG son puestos en el cero y el enlace de fin de cero está posado al principio del Program Text Area. VARTAB está dispuesto a apuntar hacia el byte siguiendo las caídas de fin del enlace y de control en la rutina cristalina en la carrera.

La dirección ... 629AH

Esta rutina es usada por los manipuladores "nuevos", "andados" y "cristalinos" de declaración para inicializar las variables Interpreter. Todo interrumpe es aclarado (636EH) y los tipos predeterminados Variable en DEFTBL se disponen a duplicar precisión. RNDX es vuelto a arrancar (2C24H) y ONEFLG, ONELIN y OLDTXT son puestos en el cero. MEMSIZ es emulado para FRETOP para descongestionar al String Storage Area y DATPTR determinado para el principio del Program Text Area (63C9H). Los contenidos de VARTAB son emulados en ARYTAB y STREND, para descongestionar a cualquier Variables, todo lo yo / que la O modera es cerrado (6C1CH) y NLOONLY es vuelto a arrancar. SAVSTK y el Z80 SP son vueltos a arrancar de STKTOP y TEMPPT es vuelto a arrancar para el principio de TEMPST para aclarar cualquier

descriptores de la cuerda. La impresora es cerrada (7304H) y salida recuperada para la pantalla (4AFFH).

Finalmente PRMLN, NOFUNS, PRMLN2, FUNACT, PRMSTK y SUBFLG son puestos en el cero y la rutina termina.

- 159 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 631BH

Esta rutina es usada por el "dispositivo EN" manipuladores de declaración a posibilitar un interrumpente fuente, la dirección de byte de estado TRPTBI del dispositivo pertinente es suministrada en par de registro HL.

Interrumpente es posibilitado ajustando bit 0 del byte de estado.

Los bits 1 y 2 son entonces examinados y, si el dispositivo ha estado bloqueado y un interrumpente ha ocurrido, ONGSBF es incrementado (634FH) a fin de que el Runloop lo procesará al final de la declaración. Finalmente el bit 1 del byte de estado es vuelto a arrancar para lanzar al mercado cualquier condición existente de alto.

La dirección ... 632EH

Esta rutina está usada por el "dispositivo FUERA DE" manipuladores de declaración para lisar un interrumpente fuente, la dirección de byte de estado TRPTBL del dispositivo pertinente es suministrada en par de registro HL.

Los bits 0 y 2 son examinados para determinar si un interrumpente ha ocurrido desde el fin de la última declaración, si así es que ONGSBF es decremented (6362H) para advertir al Runloop de recogerlo.

El byte de estado es entonces puesto en el cero.

La dirección ... 6331H

Esta rutina es usada por el "alto del DISPOSITIVO" manipuladores de declaración a suspender el procesamiento de interrumpente de un interrumpente fuente, la dirección de byte de estado TRPTBL del dispositivo pertinente es suministrada en par de registro HL. Los bits 0 y 2 son examinados para determinar si un interrumpente ha ocurrido desde el fin de la última declaración, si así es que ONGSBF es decremented (6362H) para advertir al Runloop de recogerlo. El bit 1 del byte de estado está luego colocado.

La dirección ... 633EH

Esta rutina es usada por el "regreso" manipulador de declaración para lanzar al mercado la condición de alto temporal impuesta durante interrumpente subrutinas propulsadas de BASIC, la dirección del byte pertinente de estado device's TRPTBL es suministrada en par de registro HL. Los bits 0, y 2 son examinados para determinar si un bloqueado interrumpente ha ocurrido desde que la subrutina fue primero activado. Si así es que ONGSBF es incrementado (634FH) a fin de que el Runloop lo recogerá al final de la declaración. El bit 1 del byte de estado es entonces vuelto a arrancar. Debería ser notado que cualquier "alto del DISPOSITIVO" Declaración dentro de un interrumpente subrutina propulsada por consiguiente será ineficaz.

La dirección ... 6358H

Esta rutina es usada por el Runloop interrumpente procesador (6389H) para aclarar un interrumpente antes de accionar la subrutina de BASIC, la dirección de byte de estado TRPTBL del

dispositivo pertinente es suministrada en par de registro HL. ONGSBF es decremented y bit 2 del byte de estado es vuelto a arrancar.

- 160 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 636EH

Esta rutina es usada por la rutina cristalina en la carrera (629AH) para aclarar todo interrumpido. Los setenta y ocho bytes de TRPTBL y los diez bytes de FNKFLG son puestos en el cero.

La dirección ... 6389H

Éste es el Runloop interrumpido procesador. ONEFLG es primero examinado para determinar si una condición de error actualmente existe. Si así es que la rutina termina, entonces no interrumpido será procesado hasta los claros de error. CURLIN es entonces examinado y, si el Interpreter está en modo directo, la rutina termina.

Asumir todo es adecuadamente una búsqueda está hecha de los veintiseis bytes de estado en TRPTBL para encontrar la primera parte activa interrumpida. Noto que los dispositivos cerca del principio de la mesa consecuentemente tendrán una prioridad superior que esos más bajo abajo. Cuando el primer byte activo de estado es encontrado, esto es mismo con añicos 0 y 2 se sedimentan, la dirección asociada es tomada de TRPTBL y está posado en par de registro Delaware. Lo interrumpido es entonces aclarado (6358H) y el dispositivo detenido (6331H) antes del control se traslada para el "GOSUB" manipulador (47CFH).

La dirección ... 63C9H

Esto es lo "restaura" manipulador de declaración. Si ningún operand de número de la línea existe, entonces DATPTR está listo para el principio del Program Storage Area. De otra manera el operand es coleccionado (4769H), el texto de programa registrado para encontrar la línea pertinente (4295H) y su dirección colocada en DATPTR.

La dirección ... 63E3H

Éste es el "alto" manipulador de declaración. Si el más texto existe en las transferencias de control de declaración para el "alto ON/OFF/STOP" manipulador de declaración (77A5H). La A diferente de registro está colocada para 01H y el control cae en el "fin" manipulador de declaración.

La dirección ... 63EAH

Éste es el "fin" manipulador de declaración. Es también usado, con proposiciones discrepantes de entrada, por el "alto" declaración y para CTRL-STOP y el fin de terminación de programa del texto. ONEFLG es primero puesto en el cero y luego, para el "fin" declaración sólo, todo yo los amortiguadores de la / O estoy cerrado (6C1CH). La colocación coetánea del texto de programa es colocada en SAVTXT y OLDTXT y el número actual de la línea en OLDLIN para el uso por cualquier subsiguiente "CONT" declaración. La impresora es cerrada (7304H), un CR LF emitió para la pantalla (7323H) y el par de registro que HL se dispuso a apuntar para el "Break" mensaje en 3FDCH. Para el "fin" declaración y fin de control de casos del texto luego transfiere para el punto Mainloop "Oklahoma" (411EH). Para el caso CTRL-STOP las transferencias de control hasta el fin del manipulador de error (40FDH) para exteriorizar el mensaje "Break".

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6424H

Éste es el "CONT" manipulador de declaración. A menos que son cero, en cuyo caso uno "no puede mantener" error es generado, los contenidos de OLDTXT son colocados en par de registro HL y esos de OLDLIN en CURLIN. El control luego regresa al Runloop para ejecutar en la colocación vieja del texto de programa. Un programa no puede ser mantenido después de que CTRL-STOP haya estado acostumbrado a separarse de DENTRO DE una declaración, por la rutina del estándar CKCNTC, en vez de entre declaraciones.

La dirección ... 6438H

Éste es el "TRON" manipulador de declaración, TRCFLG es poco cero simplemente hecho.

La dirección ... 6439H

Éste es el "TROFF" manipulador de declaración, TRCFLG es cero simplemente hecho.

La dirección ... 643EH

Éste es el "trueque" manipulador de declaración. El primer Variable está ubicado (5EA4H) y sus contenidos copiados para SWPTMP. La posición de este Variable y del fin del Storage Variable Area es por ahora ahorrada. El segundo Variable está luego ubicado (5EA4H) y su tipo se comparó con eso de la primera parte. Si los tipos desencajan un "bodorrio de tipo" error es generado (406DH). El fin coetáneo del Storage Variable Area es entonces comparado con el fin viejo y una "llamada de función ilegal" error generado (475AH) si difieren. Finalmente los contenidos del segundo Variable son emulados para la posición del primer Variable (2EF3H) y los contenidos de SWPTMP para la posición del segundo Variable (2EF3H).

Los cheques realizados por el manipulador quieren decir que el segundo Variable, si es simple y no un Array, siempre debe estar de existencia antes de que un "trueque" Declaración sea encontrado o un error será generado. La razón para esto es eso, suponiendo que el primer Variable fue un Array, luego la creación de un segundo (simple) Variable movería al Array Storage Area arriba invalidando su posición ahorrada. Note ese el caso perfectamente legal de un Variable primero simplista y un Variable recién segundo simplista creado es también rechazado.

La dirección ... 6477H

Éste es el "ERASE" manipulador de declaración. SUBFLG está primero listo a 01H, a controlar la rutina Variable de búsqueda, y el Array localizado (5EA4H). Todo el siguiente Arrays está emocionado que el descendente y STREND se sedimentan para su valor nuevo, inferior. El texto de programa es entonces a cuadros y, si una coma siguiera, controla transferencias de regreso al principio del manipulador.

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 64A7H

Estos cheques de rutina ya sea el carácter a cuya dirección están suministrados en par de registro HL son caja alta alfabética, si así es que devuelve Flag Carolina Del Norte.

La dirección ... 64AFH

Éste es el manipulador "cristalino" de declaración. Si ninguno de los operands son presente transferencias de control para la rutina cristalina en la carrera (62A1H) para quitar a todo Variables actual. De otra manera el operand de espacio de la cuerda es evaluado (4756H) seguido por la parte superior optativa de memoria operand (542FH). La parte superior de valor de memoria es comprobada y una "llamada de función ilegal" error generado (475AH) si está menos de 8000H o más gran que F380H. El espacio requerido por ahí lo yo los amortiguadores de la / O (267 bytes cada uno) y la String Storage Area es sustraído desde lo alto de valor de memoria y uno "por la memoria" que el error generaron (6275H) si que hay menos de faltante de 160 bytes para la base del Storage Variable Area. Asumir todo es adecuadamente HIMEM, MEMSIZ y STKTOP son determinados para sus valores nuevos y la reanudación restante de punteros de almacenamiento por la rutina cristalina en la carrera (62A1H). Lo que yo el almacenamiento del amortiguador de la / O soy reubicado (7E6BH) y que el manipulador termina.

Desafortunadamente la computación de MEMSIZ y STKTOP, cuándo una parte superior nueva de memoria está especificada, es incorrecta resultando en la parte superior de la String Storage Area siendo determinado byte demasiado alto. Esto puede verse con lo siguiente donde una cuerda ilegal es aceptada:

```
10 ACLARAN 200, y HF380
20 A$ STRING$ (201, "uno")
30 PRINT FRE ()
```

Porque debería haber una instrucción adicional DEC HL en 64EBH los valores nuevos de MEMSIZ y STKTOP son inicialmente determinado byte demasiado alto. Cuándo la rutina cristalina en la carrera es llamado MEMSIZ es emulado en FRETOP, la parte superior del String Storage Area, lo cual resulta en este byte del mismo de ser demasiado alto igualmente. Aunque MEMSIZ y STKTOP son correctamente recomputados cuando los punteros del archivo son vueltos a arrancar, FRETOP queda con su valor incorrecto. Cuando el "FRE" declaración es ajustado en línea treinta, y colección de basura de la cuerda iniciada, FRETOP está recuperado para su valor correcto pero, porque la cuerda se derrama el String Storage Area por un byte, la cantidad de espacio libre exhibido es - 1 byte. Correctamente ajustar todos los punteros de sistema cualquier alteración de la parte superior de memoria debería ser seguido inmediatamente por otra declaración "clara" sin operands.

La dirección ... 6520H

Esta rutina computa la diferencia entre los contenidos de pares de registro HL y Delaware. Es un duplicado del capítulo corto de código de 64ECH para 64F1H y está completamente sin uso.

- 163 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6527H

Éste es el "siguiente" manipulador de declaración. El texto más pretencioso está presente en la declaración que el lazo Variable es localizado (5EA4H), de otra manera una dirección predeterminada de cero está ocupada. La pila es entonces registrada para el "FOR"

correspondiente bloque de parámetro (3FE2H). Si ningún bloque de parámetro es encontrado, entonces o si un "GOSUB" bloque de parámetro es encontrado primero, entonces uno "siguiente sin PUES" error es generado (405BH). Asumir el bloque de parámetro es encontrado el capítulo intervector de pila, conjuntamente con cualquier "FOR" bloques puede contener, es descartado. El tipo del lazo Variable es luego tomado del bloque de parámetro y examinado para determinar la precisión precisada durante subsiguientes operaciones.

El valor de PASO es tomado del bloque de parámetro y se suma a (3172H, 324EH o 2697H) los contenidos coetáneos del lazo Variable que está entonces actualizado. El valor nuevo es comparado (2F4DH, 2F21H o 2F5CH) con el valor de terminación del bloque de parámetro para determinar si el lazo ha terminado (65B6H). El lazo terminará para un PASO positivo si el valor nuevo del lazo es MAYOR que el valor de terminación. El lazo terminará para un paso negativo si el valor nuevo del lazo está MENOS DEL valor de terminación. Si el lazo no ha terminado el texto original de programa número de colocación y de la línea es tomado del parámetro bloquee y controle transferencias para el Runloop (45FDH). Si el lazo ha terminado, entonces el bloque de parámetro es descartado de la pila y, a menos que el más texto de programa está presente adentro cuáles transferencias de control de regreso al principio del manipulador, controlan transferencias para el Runloop para ejecutar la siguiente declaración (4601H).

La dirección ... 65C8H

Esta rutina es usada por el Evaluador Expression para encontrar la relación (< > =) entre dos operands de la cuerda. La dirección del primer descriptor de la cuerda es suministrada en la pila Z80 y la dirección del segundo en DAC. El resultado es devuelto en A de registro y las banderas por lo que respecta a las rutinas numéricas de relación:

Ate con una cuerda a 1 = String 2 ... UN = 00H, Flag Z, Carolina Del Norte
Ate con una cuerda a 1 < String 2 ... UN = 01H, Flag NZ, Carolina Del Norte
Ate con una cuerda a 1 > String 2 ... NZ A=FFH, Flag, C

La comparación comienza en el primer carácter de cada cuerda y continúa hasta que los dos personajes difieren o uno de los instrumentos de cuerda está exhausto. El control luego regresa al Evaluador Expression (4F57H) para colocar el resultado verdadero o falso y numérico en DAC.

La dirección ... 65F5H

Esta rutina es usada por el Evaluador Factor para ejercer al "OCT\$" función para un operand contenido en DAC. El número es primero convertido a forma textual en FBUFFR (371EH) y luego la cuerda de resultado es creada (6607H).

- 164 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 65FAH

Esta rutina es usada por el Evaluador Factor para ejercer al "HEX\$" función para un operand contenido en DAC. El número es primero convertido a forma textual en FBUFFR (3722H) y luego la cuerda de resultado es creada (6607H).

La dirección ... 65FFH

Esta rutina es usada por el Evaluador Factor para ejercer al "N\$" función para un operand

contenido en DAC. El número es primero convertido a forma textual en FBUFFR (371AH) y luego la cuerda de resultado es creada (6607H).

La dirección ... 6604H

Esta rutina es usada por el Evaluador Factor para ejercer al "STR\$" función para un operand contenido en DAC. El número es primero convertido a forma textual en FBUFFR (3425H) luego analizado para determinar su largo y dirección (6635H). Después de comprobar eso suficiente espacio está disponible (668EH) la cuerda es copiada para el String Storage Area (67C7H) y el resultado que el descriptor creó (6654H).

La dirección ... 6627H

Esta primera parte de rutina inspecciona que hay suficiente espacio en el String Storage Area para la cuerda a cuyo largo está suministrado en A de registro (668EH). El largo de la cuerda y la dirección donde la cuerda será colocada en el String Storage Area es entonces copiado para DSCTMP.

La dirección ... 6636H

Esta rutina es usada por el Evaluador Factor para analizar la cadena de caracteres a cuya dirección está suministrado en par de registro HL. La cadena de caracteres es escandida un carácter (00H o ") que termina es encontrado. El largo y dirección de puesta en marcha son luego colocados en DSCTMP (662AH) y el control cae en la rutina de creación de descriptor.

La dirección ... 6654H

Esta rutina es usada por las funciones de la cuerda para crear un descriptor de resultado. El descriptor es emulado de DSCTMP para la siguiente colocación disponible en TEMPST y su dirección colocada en DAC. A menos que TEMPST está lleno, en cuál el caso una "fórmula de la cuerda también complica" el error es generado, TEMPST es aumentado por tres bytes y la rutina termina.

La dirección ... 6678H

Esta rutina exterioriza el mensaje, o cuerda, quién es aquel cuya dirección es suministrada en par de registro HL. La cuerda es analizada (6635H)

- 165 -

5. EL INTÉRPRETE DE BASIC ROM

Y su almacenamiento liberado (67D3H). Los personajes sucesivos son luego tomados de la cuerda y exhibidos, por la rutina del estándar OUTDO, hasta que la cuerda está agotada.

La dirección ... 668EH

Esta rutina inspecciona que hay espacio en la String Storage Area para añadir la cuerda a cuyo largo está suministrados en A de registro. En las proposiciones del par de registro de la salida Delaware para la dirección de puesta en marcha en la String Storage Area donde la cuerda debería estar posada. El largo de la cuerda es primero sustraído de la posición en uso en estos momentos y libre contenida en FRETOP. Esto es entonces comparado con STKTOP, la posición admisible mínima para el almacenamiento de la cuerda, para determinar si hay espacio para la cuerda. Si así es que FRETOP está actualizado

con la colocación nueva y la rutina termina. Si hay espacio insuficiente para la cuerda luego la cobranza de basura es iniciada (66B6H) para probar y eliminar cualquier instrumentos de cuerda muertos. Si, después de que la colección de basura, allí no fuera quieta lo suficientemente no espacio uno "por espacio de la cuerda" error es generado.

La dirección ... 66B6H

Éste es el colector de basuras de la cuerda, su función es eliminar cualquier instrumentos de cuerda muertos del String Storage Area. El problema básico con cuerda Variables, a distinción de numéricos, es que sus largos varían. Si los cuerpos de la cuerda se guardaron con su Variables en el Storage Variable Area aun tales declaraciones aparentemente simples como A\$ "X" tomaría, entonces el movimiento de miles de bytes de memoria y despacio ejecución acelera dramáticamente. El método usado por el Interpreter para vencer este problema es conservar la cuerda que los cuerpos distancian de los Variables. Así los instrumentos de cuerda son guardados en el String Storage Area y cada agarres Variables un tres descriptor de byte conteniendo el largo y dirección de la cuerda asociada. Cuandoquiera que una cuerda es asignada a un Variable se agrega simplemente para el montón de instrumentos de cuerda existentes en el String Storage Area y el descriptor de la Variable se alteró. Ningún intento está hecho para eliminar cualquier cuerda previa perteneciendo al Variable, reestructurando el montón, como esto arrasaría cualquier ganancias de rendimiento específico.

Si suficientes asignaciones Variables están hechas, entonces es inevitable que el String Storage Area se llenará. En un programa típico muchos de estos instrumentos de cuerda serán sin uso, ese es el resultado de asignaciones previas. La colección de basura es el proceso por medio de lo que estos instrumentos de cuerda muertos están removidos. Cada cuerda Variable en la memoria, el inclusivo Arrays y el presente local Variables durante la evaluación de usuario definieron funciones, es examinado hasta que lo uno sea encontrado quién es aquel cuya cuerda se guarda más alto en el montón. Esta cuerda es luego movida a la parte superior del String Storage Area y los contenidos Variables modificados para apuntar hacia la posición nueva. El dueño de la siguiente cuerda más alta es entonces encontrado y el proceso repetido hasta cada cuerda perteneciendo a un Variable ha sido compactado.

- 166 -

5. EL INTÉRPRETE DE BASIC ROM

Si un número grande de Variables está presente, entonces la basura la colección puede tomar un tiempo apreciable. El proceso puede ser visto en el trabajo con el siguiente programa que repetidamente asigna a la cuerda "AAAA" a cada elemento del Array A\$. El programa correrá a toda velocidad para que las primeras doscientas cincuenta asignaciones y luego la pausa eliminen lo cincuenta instrumentos de cuerda muertos. Unas adicionalmente cincuenta asignaciones luego pueden estar hechas antes de que una más colección de basura sea requerida:

```
10 ACLARAN 1000
20 OSCURECEN a A$ (200)
30 FOR N = 0 PARA 200
40 A$ (N) = STRING$ (4, "uno")
50 ESCRIBEN EN LETRAS DE IMPRENTA "";
60 SIGUIENTE N
70 GOTO 30
```

El String Storage Area está también acostumbrado a mantener los instrumentos de cuerda intermedios producidos durante la evaluación de expresión. Porque tantas funciones de la cuerda

toman discusiones múltiples, las tomas "MID\$" tres por ejemplo, la gestión de resultados intermedios es un problema principal. Para ocuparse de eso una vía de entrada estándar para colocar en serie resultados es tomado a todo lo largo del Interpreter. Un productor de una cuerda simplemente añade el cuerpo de la cuerda para el montón en el String Storage Area, añade el descriptor para el montón de descriptor en TEMPST y coloca la dirección del descriptor en DAC. Depende del usuario del resultado para liberar este almacenamiento (67D0H) una vez que ha procesado la cuerda. Esta regla se aplica a todas las partes del sistema, de los manipuladores individuales de función de regreso a través del Evaluatodor Expression para los manipuladores de declaración, con sólo dos excepciones.

La primera excepción ocurre cuando el Evaluatodor Factor encuentra una cuerda explícitamente declarada, como "algo" en el texto de programa. En este caso no hay que copiar la cuerda para el String Storage Area como el original saciará.

La segunda excepción ocurre cuando el Evaluatodor Factor encuentra una referencia para un Variable. En este caso no hay que colocar una copia del descriptor en TEMPST como uno ya existe dentro del Variable.

La dirección ... 6787H

Esta rutina es usada por el Evaluatodor Expression para concatenar dos operands de la cuerda. Controle transferencias aquí cuando una señal es encontrada después de un operand de la cuerda así la primera acción tomado debe ir a traer el segundo operand de la cuerda por el Evaluatodor Factor (4DC7H). Los largos son luego tomados de ambos descriptores de la cuerda y se agregan conjuntamente para comprobar el largo de la cuerda combinada. Si esto es mayor que doscientos cincuenta y cinco personajes una "cuerda demasiado largo" error es generada. Después de comprobar ese espacio está disponible en el String Storage Area (6627H) el almacenamiento de ambos operands es liberado (67D6H). La primera cuerda es entonces copiada para el String Storage Area (67BFH)

- 167 -

5. EL INTÉRPRETE DE BASIC ROM

Y se sigue por lo segundo (67BFH). El descriptor de resultado es creado (6654H) y control se traslada de regreso al Evaluatodor Expression (4C73H) '

La dirección ... 67D0H

Esta rutina libera cualquier almacenamiento ocupado por la cuerda cuyo descriptor la dirección es contenida en DAC. La dirección del descriptor es tomada de DAC y examinada para determinar si es eso del último descriptor en TEMPST (67EEH), en caso de que no la rutina termina. TEMPPT diferente es disminuido por limpieza total de bytes de tres este descriptor de TEMPST. La dirección del cuerpo de la cuerda es luego tomada del descriptor y comparada con FRETOP para la sede si ésta es la cuerda mínima en el String Storage Area, en caso de que no la rutina termina. De otra manera el largo de la cuerda se agrega para FRETOP, lo cual está entonces actualizado con esto valor nuevo, liberando el almacenamiento ocupado por el cuerpo de la cuerda.

La dirección ... 67FFH

Esta rutina es usada por el Evaluatodor Factor para ejercer al "LEN" función para un operand contenido en DAC. El almacenamiento de operand's es liberado (67D0H) y el largo de la cuerda tomado del descriptor y colocado en DAC como un entero (4FCFH).

La dirección ... 680BH

Esta rutina es usada por el Evaluador Factor para ejercer al "ASC" función para un operand contenido en DAC. El almacenamiento de operand's es liberado y la cuerda que el largo examinó (6803H), si es cero una "llamada ilegal de función" error es generado (475AH). De otra manera el primer is. de carácter tomado de la cuerda y colocado en DAC como un entero (4FCFH).

La dirección ... 681BH

Esta rutina es usada por el Evaluador Factor para ejercer al "CHR\$" función para un operand contenido en DAC. Después de comprobar eso suficiente espacio está disponible (6625H) el operand es convertido a un solo entero de byte (521FH). Este carácter es luego colocado en el String Storage Area y el resultado que el descriptor creó (6654H).

La dirección ... 6829H

Esta rutina es usada por el Evaluador Factor para ejercer al "STRING\$" función. Después de revisar en busca del carácter manifiesto del paréntesis el largo que el operand es evaluado y colocado en registro E (521CH). El segundo operand es entonces evaluado (4C64H). Si es numérico, entonces sea convertido a un solo entero de byte (521FH) y esté posado en A de registro. Si es una cuerda el primer carácter es tomada de ella y está posado en A de registro (680FH). El control luego las caídas en el "SPACE\$" función para crear la cuerda de resultado.

- 168 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6848H

Esta rutina es usada por el Evaluador Factor para ejercer al "SPACE\$" función para un operand contenido en DAC. El operand es primero convertido a un solo entero de byte en registro E (521FH). Después de comprobar eso suficiente espacio está disponible (6627H) el número requerido de espacios es copiado para el String Storage Area y el resultado que el descriptor creó (6654H).

La dirección ... 6861H

Esta rutina es usada por el Evaluador Factor para ejercer al "LEFT\$" función. Primer la dirección de descriptor de operand y el operand de segundo de entero son suministrados en la pila Z80. El tamaño de la rebanada es tomado de la pila (68E3H) y comparado para el largo de la cuerda de la fuente. Si el largo de la cuerda de la fuente está menos de la rebanada tamaño, entonces lo reemplaza como el largo para extraer. Después de comprobar eso suficiente espacio está disponible (668EH) el número requerido de personajes es emulado desde el principio de la fuente forma una serie para el String Storage Area (67C7H). El almacenamiento de la cuerda de la fuente está entonces liberado (67D7H) y el resultado que el descriptor creó (6654H).

La dirección ... 6891H

Esta rutina es usada por el Evaluador Factor para ejercer al "RIGHT\$" función. Primer la dirección de descriptor de operand y el operand de segundo de entero son suministrados en la pila Z80. El tamaño de la rebanada es tomado de la pila (68E3H) y sustraído del largo de la cuerda de la fuente para determinar la colocación de puesta en marcha de la rebanada. El control luego se traslada para el "LEFT\$" rutina para extraer la rebanada (6865H).

La dirección ... 689AH

Esta rutina es usada por el Evaluador Factor para ejercer al "MID\$" función. Primer la dirección de descriptor de operand y el operand de segundo de entero son suministrados en la pila Z80. La colocación de puesta en marcha es tomada de la pila (68E6H) e inspeccionada, si es cero una "llamada ilegal de función" error es generado (475AH). El tamaño optativo de la rebanada es entonces evaluado (69E4H) y control se traslada para el "LEFT\$" rutina para extraer la rebanada (6869H).

La dirección ... 68BBH

Esta rutina es usada por el Evaluador Factor para ejercer al "VAL" función para un operand contenido en DAC. El largo de la cuerda es tomado del descriptor (6803H) e inspeccionado, si es cero que es colocado en DAC como un entero (4FCFH). El largo se agrega luego a la dirección de puesta en marcha del cuerpo de la cuerda para dar la posición del carácter inmediatamente después de él.

Esto está por ahora repuesto con un byte de cero y la cuerda es convertida a forma numérica en DAC (3299H). El original

- 169 -

5. EL INTÉRPRETE DE BASIC ROM

El carácter es entonces recuperado y la rutina termina. El delimitador temporal de byte de cero es menester porque los instrumentos de cuerda están apiñados conjuntamente en el String Storage Area, sin él el convertidor numérico seguiría corriendo en tener éxito ata con una cuerda.

La dirección ... 68E3H

Esta rutina es usada por el "LEFT\$", "MID\$" y "RIGHT\$" funcionan manipuladores al cheque que el siguiente carácter del texto de programa es ")" y luego a hacer estallar un operand del Z80 apila en par de registro Delaware.

La dirección ... 68EBH

Esta rutina es usada por el Evaluador Factor para ejercer al "INSTR" función. El primer operand, cuál puede ser la puesta en marcha colocación o la fuente forman una serie, es evaluado (4C62H) y su tipo probado. Si es la fuente forme una serie, entonces una colocación de puesta en marcha de incumplimiento de uno es tomada. Si es el operand de colocación de puesta en marcha su valor es comprobado y el operand de la cuerda de la fuente evaluado (4C64H). La cuerda del patrón es entonces evaluada (4C64H) y el almacenamiento de ambos operands liberados (67D0H). El largo de la cuerda del patrón es comprobado y, si cero, la colocación de puesta en marcha es colocada en DAC (4FCFH). La cuerda del patrón es entonces a cuadros en contra de personajes sucesivos de la cuerda de la fuente, comenzando en la puesta en marcha sitúa, hasta que un encuentro es encontrado o la cuerda de la fuente está agotada. Con una búsqueda exitosa la colocación de carácter de la subcuerda es colocada en DAC como un entero (4FCFH), de otra manera un resultado de cero es devuelto.

La dirección ... 696EH

Éste es el "MID\$" manipulador de declaración. Después de revisar en busca del carácter manifiesto del paréntesis el destino Variable es hallado (5EA4H) y comprobado para asegurar que es una cuerda el tipo (3058H).

La dirección del cuerpo de la cuerda es luego tomada del Variable y examinada para determinar si está dentro del Program Text Area, como sería el caso para una cuerda explícitamente declarada. Si esto es el caso el cuerpo de la cuerda es copiado para el String Storage Area

(6611H) y un descriptor nuevo copiado para el Variable (2EF3H). Esto está hecho para evitar modificar el texto de programa. La colocación de puesta en marcha es entonces evaluada (521CH) y comprobada, si es cero una "llamada ilegal de función" error es generado (475AH). El operand optativo de largo de la rebanada es evaluado (69E4H) seguido por la cuerda de reposición (4C5FH) cuyo almacenamiento está entonces liberado (67D0H). Los personajes son entonces copiados de la cuerda de reposición para la cuerda de destino hasta ya sea la rebanada largo es completado o la cuerda de reposición está agotada.

La dirección ... 69E4H

Esta rutina es usada por las funciones diversas de la cuerda para evaluar a un operand optativo (521CH) y regresar el resultado en registro

- 170 -

5. EL INTÉRPRETE DE BASIC ROM

E. Si ningún operand está presente, entonces un valor predeterminado de 255 es devuelto.

La dirección ... 69F2H

Esta rutina es usada por el Evaluador Factor para ejercer al "FRE" función para un operand contenido en DAC. Si el operand es numérico, entonces la sola diferencia de precisión entre el Z80 Stack Pointer y los contenidos de STREND es colocada en DAC (4FC1H). Si el operand es una cuerda escriba a máquina, entonces su almacenamiento es liberado (67D3H) y colección de basura iniciada (66B6H). La sola diferencia de precisión entre los contenidos de FRETOP y esos de STKTOP son luego colocados en DAC (4FC1H).

La dirección ... 6A0EH

Esta rutina es usada por el archivo yo los manipuladores de la / O para analizar un filespec tan " UNO:FILENAME.BAS ". El filespec consta de tres partes, el dispositivo, el filename y la extensión de tipo.

En las proposiciones del par de registro de entrada HL para el principio del filespec en el texto de programa. En registro de la salida la D sostiene el código del dispositivo, el filename está de cero de colocaciones para siete de FILNAM y la extensión de tipo en colocaciones ocho para diez. Cualquier colocaciones sin uso se llenan de espacios.

La cuerda del filespec es evaluada (4C64H) y su almacenamiento liberado (67D0H), si la cuerda es de largo de cero un "mal nombre del archivo" error es generado (6E6BH). El nombre del dispositivo es personajes a los que se se analizó gramaticalmente (6F15H) y sucesivos y tomados del filespec y está posado en FILNAM hasta que la cuerda esté agotada, uno "" El carácter es encontrado o FILNAM está lleno. Un "mal nombre del archivo" error es generado (6E6BH) si el filespec contiene cualquier caracteres de control, esto es esos quién es aquel cuyo valor es más pequeño que 20H. Si el filespec contiene una extensión de tipo un "mal nombre del archivo" error es generada (6E6BH) si es más largo que tres personajes o si el filename es más largo que ocho personajes. Si ninguna extensión de tipo está presente que el filename puede ser, entonces cualquier largo, personajes adicionales está simplemente ignorado.

La dirección ... 6A6DH

Esta rutina es usada por el archivo yo los manipuladores de la / O a localizar lo que yo / O modero FCB cuyo número es suministrado en A de registro. El número del amortiguador es primero dado jaque en contra de MAXFIL y un "mal número del archivo" error generado (6E7DH) si es demasiado grande. De otra manera la dirección requerida es tomada del bloque del puntero

del archivo y está posado en par de registro HL y el modo del amortiguador tomado de byte 0 del FCB y colocado en A de registro.

La dirección ... 6A9EH

Esta rutina es usada por el archivo y los manipuladores de la / O a evaluar uno que yo amortiguador de la / O número y para localizar a su FCB. Cualquier carácter es saltado (4666H) y el amortiguador que el número evaluó (521CH). El FCB no está ubicado (6A6DH) y un "archivo que no" el error "manifiesto" generó

- 171 -

5. EL INTÉRPRETE DE BASIC ROM

(6E77H) si el byte de modo del amortiguador es cero. De otra manera la dirección FCB es colocada en PTRFIL para reencauzar la salida Interpreter's.

La dirección ... 6AB7H

Éste es el manipulador "accesible" de declaración. El filespec es analizado (6A0EH) y cualquier siguiente modo convertido al byte correspondiente de modo, éstos son: "Para la ENTRADA" (01H), "para la SALIDA" (02H) y "pues ANEXA" (08H). Si ningún modo es explícitamente, entonces declaró que el modo aleatorio (04H) es asumido. Los "de AS - los personajes son a los que se dio jaque y el número del amortiguador evaluado (521CH), si éste es cero un "Mal número del archivo" el error es generado (6E7DH). El FCB está luego ubicado (6A6DH) y un "archivo ya" el error "manifiesto" generó (6E6EH) si el byte de modo del amortiguador es cualquier cosa aparte del cero.

El código del dispositivo es colocado en byte 4 del FCB, la función manifiesta despachada (6F8FH) y la salida de Interpreter puesto a cero para la pantalla (4AFFH).

La dirección ... 6B24H

Esta rutina es usada por el archivo y los manipuladores de la / O a cerrar lo que yo / O modero quién es aquel cuyo número es suministrado en A de registro. El FCB está ubicado (6A6DH) y, proveyó el amortiguador está funcionando, la función cercana despachada (6F8FH) y el amortiguador lleno con pone en el cero (6CEAH). PTRFIL y el byte de modo FCB son entonces puestos en el cero para poner a cero la salida de Interpreter para la pantalla.

La dirección ... 6B5BH

Ésta es la "carga", "MERGE" y "filespec ANDADO" manipulador de declaración. El filespec es analizado (6A0EH) y luego, para "la CARGA" y "anda" único, el texto de programa examinado para determinar si la "interrogación" andada en automóvil opción es especificada. Yo el amortiguador de la / O soy abierto para la entrada (6AFAH) y el primer byte de FILNAM determinado para FFH si andado en automóvil es requerido. Para "la CARGA" y "la CARRERA" sólo cualquier texto de programa es entonces aclarado por el manipulador "nuevo" (6287H) de declaración.

Como esto pondrá a cero la salida de Interpreter para la pantalla el amortiguador FCB está otra vez localizada y colocada en PTRFIL (6AAAH).

El control luego se traslada directamente para el Interpreter Mainloop (4134H) para que el texto de programa sea cargado como si escrito a máquina del teclado. Noto que ningún error inspeccionando de cualquier tipo es efectuado en los datos leídos.

La dirección ... 6BA3H

Éste es el "SAVE" manipulador de declaración. El filespec es analizado (6A0EH) y el texto de programa examinado para determinar si el sufijo de la "A" de ASCII está presente. Esto tiene sólo importancia bajo el BASIC Disk, da lo mismo en una máquina estándar MSX. Yo el amortiguador de la / O 0 soy abierto para la salida (6AFAH) y el control transfiere para la "lista" manipulador de declaración (522EH) devolver el texto de programa. Noto que ninguna información de comprobación de error de cualquier tipo acompaña el texto.

- 172 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6BDAH

Esta rutina es usada por el archivo que yo los manipuladores de la / O para devolver el dispositivo codifica para lo actualmente activo yo el amortiguador de la / O. La dirección FCB es tomada de PTRFIL que luego el código del dispositivo llevado de byte 4 del FCB y colocado en A de registro.

La dirección ... 6BE7H

Esta rutina es usada por el archivo yo los manipuladores de la / O para realizar una operación en un número de yo / que O modera. La dirección de la rutina pertinente es suministrada en par de registro BC y la cuenta del amortiguador en A de registro. Por ejemplo si el par de registro AC contuvo 6B24H y la A de registro contuvo amortiguadores del 03H 3, entonces 2, 1 y 0 estarían cerrados. La rutina tiene una función ligeramente diferente si es introducida con FLAG NZ. En este caso lo yo que el amortiguador de la / O numera está ocupado secuencialmente del texto de programa y evaluado (521CH) antes de que la operación sea realizada, un caso típico podría tener # "1 años de edad, # 2".

La dirección ... 6C14H

Éste es el manipulador "cercano" de declaración. El par de registro AC está colocado para 6B24H, la A de registro está cargada de los contenidos de MAXFIL y el número requerido de amortiguadores cerrados (6BE7H).

La dirección ... 6C1CH

Esta rutina es usada por el archivo yo los manipuladores de la / O a cerrar cada yo el amortiguador de la / O. El par de registro AC está colocado para 6B24H, la A de registro está cargada de los contenidos de MAXFIL y todos los amortiguadores cerrados (6BE7H).

La dirección ... 6C2AH

Éste son los "LFILES" manipulador de declaración. PRTFLG es poco cero hecho, a dirigir salida para la impresora, y controlar cae en los "archivos" manipulador de declaración.

La dirección ... 6C2FH

Éste son los "FILES" manipulador de declaración, una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 6C35H

Controle transferencias aquí del "PUT" general y "traiga a" los manipuladores (7758H) cuando el texto de programa contiene cualquier cosa aparte de un "bloque de objetos movibles" señal.

Uno "secuencial yo / la O sólo" el error soy generado (6E86H) en una máquina estándar MSX.

La dirección ... 6C48H

Esta rutina es usada por el archivo yo los manipuladores de la / O para

- 173 -

5. EL INTÉRPRETE DE BASIC ROM

Secuencialmente devuelva el carácter abastecido en A de registro. El carácter es colocado en C de registro y la salida secuencial que la función despachó (6F8FH).

La dirección ... 6C71H

Esta rutina es usada por el archivo yo los manipuladores de la / O para secuencialmente introducir en la computadora un solo carácter. La función secuencial de entrada es despachada (6F8FH) y el carácter regresó en A de registro, FLAG C indica una condición EOF (el Fin De File).

La dirección ... 6C87H

Esta rutina es usada por el Evaluador Factor para ejercer al "INPUT\$" función. El texto de programa es comprobado para el "\$" y "(" los personajes y el operand de largo evaluados (521CH). Si uno yo el número del amortiguador de la / O estoy presente es evaluado, entonces el FCB localizó (6A9EH) y el byte de modo examinado. Una "entrada después del fin" el error es generada (6E83H) si el amortiguador no está de entrada o modo aleatorio. Después de comprobar eso suficiente espacio está disponible (6627H) el número requerido de personajes es secuencialmente introduzca en la computadora (6C71H), o cobrado por la rutina del estándar CHGET, y copiado para el String Storage Area. Finalmente el descriptor de resultado es creado (6654H).

La dirección ... 6CEAH

Esta rutina es usada por el archivo yo los manipuladores de la / O para llenar el amortiguador cuyo FCB pone la dirección es contenido en PTRFIL con doscientos cincuenta y seis pone en el cero.

La dirección ... 6CFBH

Esta rutina es usada por el archivo yo los manipuladores de la / O para devolver, en registro para HL, la dirección de puesta en marcha del amortiguador cuyo FCB dirige la palabra a es contenida en PTRFIL. Esto justamente implica añadir nueve para la dirección FCB.

La dirección ... 6D03H

Esta rutina es usada por el Evaluador Factor para ejercer al "LOC" función para lo yo el amortiguador de la / O cuyo número soy contenido en DAC. El FCB está ubicado (6A6AH) y la función LOC despachada (6F8FH). Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 6D14H

Esta rutina es usada por el Evaluador Factor para ejercer al "LOF" función para lo yo el amortiguador de la / O cuyo número soy contenido en DAC. El FCB está ubicado (6A6AH) y la

función LOF despachada (6F8FH). Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

- 174 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6D25H

Esta rutina es usada por el Evaluador Factor para ejercer al "EOF" función para lo yo el amortiguador de la / O cuyo número soy contenido en DAC. El FCB está ubicado (6A6AH) y la función EOF despachada (6F8FH).

La dirección ... 6D39H

Esta rutina es usada por el Evaluador Factor para ejercer a los "FPOS" función para lo yo el amortiguador de la / O cuyo número soy contenido en DAC. El FCB está ubicado (6A6AH) y la función FPOS despachada (6F8FH). Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 6D48H

Controle transferencias para esta rutina cuando el Interpreter Mainloop encuentra una declaración directa, eso no es uno con ningún número de la línea. La rutina del estándar ISFLIO se usa primero para determinar si una "carga" declaración es activa. Si la entrada proviene de, entonces las transferencias de control del teclado para la ejecución Runloop apuntan (4640H) para ejecutar la declaración. Si la entrada viene del amortiguador del casete 0 está cerrado (6B24H) y una "declaración directa en archivo" que el error generó (6E71H). Esto podría dar con una máquina estándar MSX tampoco a través de un error del casete o intentando para cargar un archivo del texto sin números de la línea.

La dirección ... 6D57H

Esta rutina está usada por la "entrada", "entrada de la LÍNEA" y manipuladores "estampados" de declaración para revisar en busca de la presencia de un carácter en el texto de programa. Si uno es encontrado lo yo número del amortiguador de la / O soy evaluado (521BH), entonces el FCB localizó y su dirección colocada en PTRFIL (6AAAH). El byte de modo del FCB es entonces comparado con el modo que el número suministró por el manipulador de declaración en registro C, si no hacen juego con un "mal número del archivo" error es generado (6E7DH). Con "impresión" los modos admisibles son salida, al azar y anexan. Con "entrada" y "entrada de la LÍNEA" los modos admisibles son entrada y al azar. No note eso en una máquina estándar MSX no que todos estos modos son a los que se dio soporte en niveles inferiores. El alguna suerte de error consecuentemente será generado en una posterior etapa para modos ilegales.

La dirección ... 6D83H

Esta rutina es usada por la "entrada" manipulador de declaración para introducir en la computadora una serie de uno yo el amortiguador de la / O. Un regreso es primero establecido a la "lectura /entrada" manipulador de declaración (4BF1H). Los personajes que delimitan la entrada forman una serie, coma y espacio para una Variable numérico y la coma sólo para una cuerda Variable, son colocados en D de registros y E y transferencias de control para la "línea INTRODUCEN EN LA COMPUTADORA" rutina (6DA3H).

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6D8FH

Éste es la "entrada de la LÍNEA" manipulador de declaración cuando la entrada está de uno y o el amortiguador de la / O. El número del amortiguador es evaluados, el FCB localizados y el modo comprobados (6D55H). La Variable a la que asignar es luego hallada (5EA4H) y su tipo a cuadros para asegurarlo es un tipo de la cuerda (3058H). Un regreso es establecido al manipulador "dejado" (487BH) de declaración para realizar la asignación y la cuerda de entrada cobrada.

Los personajes son secuencialmente entrada (6C71H) y colocado en BUF hasta que el delimitador correcto es encontrado, EOF es alcanzado o BUF se llena (6E41H). Cuando la condición que termina es cumplida y la asignación es para un Variable numérico la cuerda es convertida a forma numérica en DAC (3299H). Cuando la asignación es para una cuerda Variable la cuerda es analizada y el resultado que el descriptor creó (6638H).

Para "la ENTRADA de la LÍNEA" todos los personajes son aceptados hasta que un código de RETORNO DE CARRO sea cumplido. Note que si este código de RETORNO DE CARRO es precedido por un LF código luego no funcionará como un delimitador sino meramente será aceptado como parte de la cuerda. Para "la ENTRADA" para un Variable numérico conduciendo espacios es desnudados luego personajes aceptados hasta que un código de RETORNO DE CARRO, un espacio o una coma sea alcanzado. Note que por lo que respecta a "la ENTRADA de la LÍNEA" un código de RETORNO DE CARRO no funcionará como un delimitador cuando precedido por un código LF. En este caso de cualquier forma que el código de RETORNO DE CARRO no será colocado en BUF pero ignorado. Para "la ENTRADA" para una conducción de la cuerda Variable los espacios son devastados luego personajes aceptados hasta un RETORNO DE CARRO o la coma es alcanzada. Note que por lo que respecta a "la ENTRADA de la LÍNEA" un código de RETORNO DE CARRO no funcionará como un delimitador cuando precedido por un código LF. En este caso sin embargo ningún código será colocado en BUF ambos están ignorados. Un modo alternativo es introducido cuando el primer carácter lee, después de cualquier espacios, lo hace un carácter doble de cita. En este caso todos los personajes serán aceptados, y se guardarán en BUF, hasta que otro delimitador doble de cita sea leído.

Una vez que la cuerda de entrada ha sido aceptada el delimitador que termina es examinado para ver si cualquier acción especial es precisada con relación a personajes de arrastre. Si la cuerda de entrada se deslindó por un carácter doble de cita o un espacio luego cualquier subsiguientes espacios serán leídos adentro e ignorados hasta que un carácter de poco espacio es encontrado. Si este carácter es una coma o código de RETORNO DE CARRO luego es aceptada e ignorada. De otra manera una función del putback es despachada (6F8FH) para devolver el carácter para lo yo el amortiguador de la / O. Si la cuerda de entrada se deslindó por un RETORNO DE CARRO código luego el siguiente carácter es leído adentro e inspeccionado. Si éste es un LF código, entonces será aceptado pero ignorado. Si no es un código LF luego una función del putback es despachada (6F8FH) para devolver el carácter para lo yo el amortiguador de la / O.

La dirección ... 6E6BH

Éste es un archivo de grupo de diez yo / que la O relacionó generadores de error. La E de registro está cargada del código pertinente de error y el control

5. EL INTÉRPRETE DE BASIC ROM

Las transferencias para el manipulador de error (406FH):

ADDR. EI ERROR

El nombre del archivo del 6E6BH Bad
El 6E6EH File ya claro
La declaración del 6E71H Direct en archivo
El 6E74H File no encontrado
El 6E77H File cerrado
El excedente del 6E7AH Field
El número del archivo del 6E7DH Bad
El error del 6E80H Internal
El 6E83H Input después del fin
El 6E86H Sequential yo / la O sólo

La dirección ... 6E92H

Éste es el "BSAVE" manipulador de declaración. El filespec es analizado (6A0EH) y la dirección de principio evaluada (6F0BH). La dirección de alto es entonces evaluado (6F0BH) y colocado en SAVEND seguido por la dirección optativa (6F0BH) de entrada que es colocada en SAVENT. Si ninguna dirección de entrada existe, entonces la dirección de principio está ocupada en lugar de eso. El código del dispositivo es comprobado para asegurar que es CAS, en caso de que no un "mal nombre del archivo" error es generado (6E6BH), y los datos escritos para casete (6FD7H). Note tan no moderador está involucrado, datos están escritos directamente para el casete, y ninguna información de comprobación de error acompaña los datos.

La dirección ... 6EC6H

Éste es el "BLOAD" manipulador de declaración. El filespec es analizado (6A0EH) y RUNBNF hizo poco cero si la "interrogación" andada en automóvil opción está presente en el texto de programa. El offset optativo de carga, con un valor predeterminado de cero, es entonces evaluado (6F0BH) y el código del dispositivo a cuadros para asegurar que sea CAS, en caso de que no un "mal nombre del archivo" error es generado (6E6BH). Los datos no son entonces leído directamente de casete (7014H), al igual que con "BSAVE" error moderador o inspeccionando está involucrado.

La dirección ... 6EF4H

Controle transferencias para esta rutina cuando el manipulador de declaración "BLOAD" ha completado cargar datos en la memoria. Si RUNBNF es amortiguador de cero 0 está cerrado (6B24H) y controle ingresos para el Runloop. Amortiguador diferente 0 es cerrado (6B24H), una dirección del remitente de 6CF3H es establecida (esta rutina justamente hace estallar el puntero del texto de programa de vuelta a par de registro HL y regresa al Runloop) y el control se traslada para la dirección contenida en SAVENT.

La dirección ... 6F0BH

Esta rutina es usada por el "BLOAD" y "BSAVE" manipuladores para evaluar una dirección operand, el resultado es devuelto en par de registro Delaware. El operand es evaluado (4C64H) luego se convierte para uno

5. EL INTÉRPRETE DE BASIC ROM

El entero (5439H).

La dirección ... 6F15H

Esta rutina es usada por el analizador del filespec para analizar gramaticalmente un nombre del dispositivo como "CAS". En registro de entrada que las proposiciones del par HL para el principio de la cuerda del filespec y la E de registro contienen su largo. Si ningún nombre del dispositivo está presente, entonces el dispositivo predeterminado que el código (CAS=FFH) es devuelto en A de registro con FLAG Z. Si un nombre legal del dispositivo está presente, entonces su código es devuelto en A de registro con FLAG NZ.

El filespec es examinado uno a uno. El carácter es encontrado luego el nombre comparado con cada uno del dispositivo legal nombra en la mesa del dispositivo en 6F76H. Si un encuentro es encontrado, entonces el código del dispositivo es tomado de la mesa y regresado en A de registro. Si ningún encuentro es encontrado, entonces las transferencias de control para el ROM externo investigan rutina (55F8H). Noto que cualquier personajes de la letra minúscula son puestos para la caja alta para propósitos de comparación. Así el crt y CRT, por ejemplo, es el mismo dispositivo.

La dirección ... 6F76H

Esta mesa es usada por el analizador gramatical de nombre del dispositivo, contiene los cuatro nombres del dispositivo y códigos disponibles en una máquina estándar MSX:

CAS ... FFH LPT ... FEH CRT ... FDH GRP ... FCH

La dirección ... 6F87H

Esta mesa es usada por el repartidor de función (6F8FH), contiene la dirección de la mesa de desciframiento de función para cada uno de los dispositivos MSX de cuatro estándar:

CAS ... 71C7H LPT ... 72A6H CRT ... 71A2H GRP ... 7182H

La dirección ... 6F8FH

Éste es el archivo y el repartidor de función de la / O. En conjunción con la estructura del amortiguador de Interpreter que le provee un dispositivo consistente, método independiente de introducir en la computadora o devolver datos. La función requerida al código está suministrada en A de registro y la dirección del amortiguador FCB en registroorean a HL.

El código del dispositivo es tomado de byte 4 del FCB y examinado para determinar si es uno de los cuatro dispositivos estándar, en caso de que no el control transfiere para el repartidor externo (564AH) de función ROM. De otra manera la dirección de la mesa de desciframiento de función del dispositivo es tomada de la mesa en 6F87H, la dirección de la función requerida tomado de ella y el control transferido para el manipulador pertinente de función.

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 6FB7H

Éste es el "CSAVE" manipulador de declaración. El filename es evaluado (7098H) seguido por la tasa optativa de baudio operand (7A2DH). El escollo de la identificación está luego escrito a casete (7125H) con un byte del filetype de D3H. Los contenidos del Program Text Area están escritos directamente para casete como un soltero que los datos bloquean (713EH). Noto que ninguna información de comprobación de error acompaña los datos.

La dirección ... 6FD7H

Las transferencias de control para esta rutina del "BSAVE" manipulador de declaración para escribir un bloque de memoria para el casete. El escollo de la identificación está primero escrito a casete (7125H) con un byte del filetype de D0H. El motor se enciende luego y un encabezado pequeño escrito para casete (72F8H) La dirección de puesta en marcha es a la que se hizo estallar de la pila Z80 y está escrito para casete LSB primero, el segundo MSB (7003H). La dirección de alto es tomada de SAVEND y está escrito para casete LSB primero, el segundo MSB (7003H). La dirección de entrada es tomada de SAVENT y está escrito para casete LSB primero, el segundo MSB (7003H). El área requerido de memoria está luego escrito al byte del mismo del casete a la vez (72DEH) y el motor del casete se desactivó por la rutina del estándar TAPOOF. Noto que ninguna información de comprobación de error acompaña los datos.

La dirección ... 7003H

Esta rutina escribe los contenidos de par de registro HL para casete con L de registro primero (72DEH) y el segundo de la H de registro (72DEH).

La dirección ... 700BH

Esta rutina lee dos bytes de casete y lugares lo de primero en registro L (72D4H), el segundo en registro H (72D4H).

La dirección ... 7014H

Las transferencias de control para esta rutina del "BLOAD" manipulador de declaración para cargar datos del casete en la memoria. El casete es leído hasta que un bloque de la identificación con un tipo del archivo de D0H y el filename correcto sea encontrado (70B8H). El encabezado del bloque de datos está luego ubicado en el casete (72E9H). El valor de offset es al que se hizo estallar de la pila Z80 y se suma a la dirección de principio del casete (700BH). La dirección de alto es leída de casete (700BH) y el offset además de eso igualmente. La dirección de entrada es leída de casete (700BH) y coloca a en SAVENT en caso andado en automóvil es precisado. Los bytes sucesivos de datos son entonces leídos de casete (72D4H) y están posado en la memoria, al principio dirigen la palabra inicialmente, hasta que la dirección de alto es alcanzada. Finalmente el motor está apagado por la rutina del estándar TAPIOF y el control transfiere para el "BLOAD" punto de terminación (6EF4H).

- 179 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 703FH

¿Éste es el "CLOAD" y "CLOAD"? El manipulador de declaración. El texto de programa es primero a cuadros para una señal "estampada" de arrastre (91H) que es cómo lo "? El carácter es tokenized. El filename es entonces evaluado (708CH) y el casete leído hasta un bloque de la identificación con un filetype de D3H y el filename correcto es encontrado (70B8H). Para "CLOAD" una operación "nueva" es entonces realizada (6287H) para borrar el texto de programa en uso en estos momentos. ¿Para "CLOAD"? Todos los punteros en el Program Text Area se convierten para revestir números (54EAH) para hacer juego con los datos del casete.

El encabezado del bloque de datos está ubicado en el casete y los bytes sucesivos de datos leyeron de casete y colocado en la memoria o se compararon con los contenidos coetáneos (715DH) de memoria. Cuándo el bloque de datos ha sido completamente leído que el mensaje "Oklahoma" es exhibido (6678H) y el control se traslada directamente hasta el fin del Interpreter Mainloop (4237H) para volver a arrancar los punteros Variables de almacenamiento. ¿Para "CLOAD"? La lectura del bloque de datos terminará si el byte del casete no equivale al byte del texto de programa en la memoria. Si la dirección donde esto ocurrió está por encima del fin del Program Text Area, entonces luego el manipulador termina con un mensaje "muy bueno" como antes. De otra manera uno "verifica error" es generado.

La dirección ... 708CH

Esta rutina es usada por el "CLOAD" y "CSAVE" manipuladores de declaración para evaluar un filename en el texto de programa. Los dos manipuladores usan proposiciones diferentes de entrada a fin de que un filename nulo está permitido para "CLOAD" pero no para "CSAVE". La cuerda del

filename es evaluada (4C64H), su almacenamiento liberado (680FH) y la primera parte que seis personajes emularon para FILNAM. Si el filename es más largo que seis personajes el exceso está ignorado. Si el filename es más corto que seis personajes luego FILNAM es acolchado con espacios.

La dirección ... 70B8H

Esta rutina es usada por el "CLOAD" y "BLOAD" manipuladores de declaración y para el repartidor el claro funciona (cuándo el dispositivo es CAS y el modo es entrada) localizar un bloque de la identificación en el casete. En la entrada el filename está en FILNAM y el tipo del archivo en registro C, D3H para un archivo de BASIC del tokenized (CLOAD), D0H para un EAH y archivo binario (BLOAD) para un archivo de ASCII (la CARGA o los datos).

El motor del casete se enciende y el casete leído hasta un encabezado es encontrado (72E9H). Cada escollo de la identificación es prefijado por diez personajes de tipo del archivo así es que los personajes sucesivos son leídos de casete (72D4H) y comparados al tipo requerido del archivo. Si los personajes de tipo del archivo no hacen juego con transferencias de control de regreso al principio de la rutina para encontrar el siguiente encabezado. De otra manera los siguientes seis personajes son leídos adentro (72D4H) y están posado en FILNAM. Si FILNAM no está lleno de espacios filename

- 180 -

5. EL INTÉRPRETE DE BASIC ROM

El encuentro es intentado y el bloque de la identificación ha sido encontrado. De otra manera los contenidos de FILNAM y FILNM2 son comparados para determinar si éste es el archivo requerido. Si el encuentro está sin éxito, entonces y el Interpreter está de modo directo, el mensaje "Skip" Es exhibido (710DH) seguido por el filename. El control luego se traslada de regreso al principio de la rutina para probar el siguiente

encabezado. Si el encuentro tiene éxito, entonces y el Interpreter está de modo directo, el mensaje "Found" Es exhibido (710DH) seguido por el filename y la rutina termina.

La dirección ... 70FFH

Éste es el texto puro que el mensaje "encontró" Terminado por un byte de cero.

La dirección ... 7106H

Éste es el mensaje de texto puro "Skip" Terminado por un byte de cero.

La dirección ... 710DH

A menos que CURLIN muestre al Interpreter estar de modo de programa que esta primera parte de rutina exterioriza (6678H) el mensaje a cuya dirección está suministrado en par de registro HL, seguido por los seis personajes contuvo en FILENAM2.

La dirección ... 7125H

Esta rutina es usada por el "CSAVE" y "BSAVE" manipuladores de declaración y para el repartidor el claro funciona (cuándo el dispositivo es CAS y el modo es salida) escribir un bloque de la identificación para casete. En la entrada el filename está en FILNAM y el filetype en A de registro, D3H para un archivo de BASIC del tokenized (CSAVE), D0H para un EAH y archivo binario (BSAVE) para un archivo de ASCII (SALVE o datos). El motor del casete se enciende y un encabezado largo escrito para casete (72F8H) El byte del filetype está escrito a casete (72DEH) que diez veces siguieron por los primeros seis personajes de FILNAM (72DEH). El motor del casete está apagado por la rutina del estándar TAPOOF y la rutina termina.

La dirección ... 713EH

Esta rutina es usada por el "CSAVE" manipulador de declaración para escribir al Program Text Area para casete como un soltero que los datos bloquean. Todos los punteros en el texto de programa se convierten de regreso a revestir números (54EAH) para independizar la dirección del texto. El motor del casete se enciende y un encabezado pequeño escrito para casete (72F8H) El Program entero Text a Area le es luego escrito para casete un byte a la vez (72DEH) y seguido con siete bytes de cero (72DEH) como un exterminador. El motor del casete es entonces revuelto apagado por la rutina del estándar TAPOOF y la rutina termina.

- 181 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 715DH

¿Esta rutina es usada por el "CLOAD" y "CLOAD"? Los manipuladores de declaración para leer unos datos del soltero el bloque en el Program Text Area o compararlo con los contenidos coetáneos. ¿En la entrada la A de registro contiene una bandera para distinguir entre las dos declaraciones, 00H para "CLOAD" y FFH para "CLOAD"? El motor del casete se enciende y el primer encabezado localizado (72E9H). Los personajes sucesivos son leídos de casete (72D4H) y están posado en el Program Text Area o comparados con los contenidos coetáneos. ¿Si la declaración actual es "CLOAD"? La rutina terminará con FLAG NZ si el carácter del casete no equivale al carácter de memoria. Los datos diferentes serán leídos hasta diez sucesivos pone en el cero es encontrado. Esta secuencia de pone en el cero está compuesto del último fin de la línea de programa de carácter de la línea, el enlace de fin y lo siete exterminador ponen en el cero adicionados por "CSAVE". ¿Note que la rutina probablemente terminará durante esta

secuencia, cuándo usada por "CLOAD"? Como la comparación de memoria está todavía de progreso. ¿Esto lleva las cuentas para la codificación más bien peculiar del "CLOAD"? El manipulador terminando condiciones.

La dirección ... 7182H

Esta mesa es usada por el repartidor al descifrar códigos de función para el dispositivo GRP. Contiene la dirección del manipulador para cada uno de los códigos de función, la mayoría está en generadores de error de hecho:

PARA FUNCIONAR

71B6H 0, abierto
71C2H 2, cerca
6E86H 4, aleatorio
7196H 6, salida secuencial
475AH 8, entrada secuencial
475AH 10, loc
475AH 12, lof
475AH 14, eof
475AH 16, fpos
475AH 18, putback

La dirección ... 7196H

Éste es el repartidor rutina secuencial de salida para el dispositivo GRP. SCRMOD es primero a cuadros y una "llamada de función ilegal" error generado (475AH) si la pantalla está ya sea modo del texto. El carácter a la salida es tomado de transferencias de registro de la C y de control para la rutina del estándar GRPPRT.

La dirección ... 71A2H

Esta mesa es usada por el REPARTIDOR del DISPOSITIVO al descifrar códigos de función para el dispositivo CRT. Contiene la dirección del manipulador para cada uno de los códigos de función, la mayoría está de hecho

- 182 -

5. EL INTÉRPRETE DE BASIC ROM

Los generadores de error:

PARA FUNCIONAR

71B6H 0, abierto
71C2H 2, cerca
6E86H 4, aleatorio
71C3H 6, salida secuencial
475AH 8, entrada secuencial
475AH 10, loc
475AH 12, lof
475AH 14, eof
475AH 16, fpos
475AH 18, putback

La dirección ... 71B6H

Ésta es la rutina de claro del repartidor para el CRT, LPT y dispositivos GRP. El modo requerido, en registro E, es comprobado y un "mal nombre del archivo" que el error generó (6E6BH) para la entrada o anexe.

La dirección FCB es luego colocada en PTRFIL, el modo en el byte 0 del FCB y la rutina termina. Noto que la instrucción Z80 RET al final de esta rutina (71C2H) es la repartidora cerca rutina para el CRT, LPT y dispositivos GRP.

La dirección ... 71C3H

Éste es el repartidor rutina secuencial de salida para el dispositivo CRT. El carácter a la salida es tomado de transferencias de registro de la C y de control para la rutina del estándar CHPUT.

La dirección ... 71C7H

Esta mesa es usada por el repartidor al descifrar códigos de función para el dispositivo CAS. Contiene la dirección del manipulador para cada uno de los códigos de función, varios es generadores de error:

PARA FUNCIONAR

71DBH 0, abierto
7205H 2, cerca
6E86H 4, aleatorio
722AH 6, salida secuencial
723FH 8, entrada secuencial
475AH 10, loc
475AH 12, lof
726DH 14, eof
475AH 16, fpos
727CH 18, putback

La dirección ... 71DBH

Ésta es la rutina de claro del repartidor para el dispositivo CAS. Lo

- 183 -

5. EL INTÉRPRETE DE BASIC ROM

La corriente yo el amortiguador de la / O sitio, mantenido en el byte 6 del FCB, y CASPRV, que mantiene cualquier carácter del putback son ambos puestos en el cero. El modo requerido, suministrado en registro E, es examinado y un "mal archivo" que el error de "nombre" generó (6E6BH) para anexe o modos aleatorios.

Para el modo de salida el bloque de la identificación está luego escrito a casete (7125H) mientras para el modo de entrada que el bloque correcto de la identificación está ubicado en el casete (70B8H). La dirección FCB es luego colocada en PTRFIL, el modo en el byte 0 del FCB y la rutina termina.

La dirección ... 7205H

Éste es el repartidor cerca rutina para el dispositivo CAS.

El byte 0 del FCB es examinado y, si el modo fuera introduce en la computadora, CASPRV es puesto en el cero y la rutina termina. Lo demás diferente de lo que yo el amortiguador de la / O soy llenado del fin de personajes del archivo (1AH) y lo yo los contenidos del amortiguador de la / O escrito para casete (722FH). CASPRV es entonces puesto en el cero y la rutina termina.

La dirección ... 722AH

Éste es el repartidor rutina secuencial de salida para el dispositivo CAS. El carácter a la salida es tomado de C de registro y mete en la siguiente colocación libre lo yo el amortiguador de la / O (728BH).

El byte 6 del FCB, lo yo el amortiguador de la / O sitio, es entonces incrementado. Si lo que yo el amortiguador de la / O sitio se ha enrollado redondee para poner en el cero esto manera que hay doscientos cincuenta y seis personajes en lo yo el amortiguador de la / O y eso tiene que estar escrita para casete. El motor del casete se enciende, un encabezado pequeño es escrito para casete (72F8H) seguido por lo que yo el amortiguador de la / O contenido (72DEH), y el motor está apagado por la rutina del estándar TAPOOF.

La dirección ... 723FH

Éste es el repartidor rutina secuencial de entrada para el dispositivo CAS. CASPRV es primero a cuadros (72BEH) para determinar si contiene un carácter que ha sido putback, en cuyo caso sus contenidos serán poco cero. Si así es que la rutina termina con el carácter en A de registro. De otra manera lo yo el amortiguador de la / O sitio es comprobado (729BH) para determinar si contiene cualquier caracteres. Si lo yo el amortiguador de la / O estoy vacío el motor del casete se enciende y el encabezado localizado (72E9H). Doscientos cincuenta y seis personajes son entonces leídos adentro (72D4H), el motor del casete cerrado por la rutina del estándar TAPION y lo yo el amortiguador de la / O sitio reanudación para poner en el cero. El carácter es luego tomado de la corriente yo colocación del amortiguador de la / O y la colocación incrementada. Finalmente el carácter es comprobado para ver si es el fin de carácter del archivo (1AH). Si no es, entonces la rutina termina con el carácter en Carolina Del Norte de registro de la A y de la BANDERA. De otra manera el fin de carácter del archivo es colocado en CASPRV, tan esa subsiguiente entrada secuencial demanda siempre devolverá el fin de condición del archivo, y la rutina termina con FLAG C.

- 184 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 726DH

Ésta es la rutina del eof del repartidor para el dispositivo CAS. El siguiente carácter es entrada (723FH) y colocado en CASPRV. Es entonces probado para el fin de código del archivo (1AH) y el resultado colocó en DAC como un entero, cero para FFFFH falso, para verdadero.

La dirección ... 727CH

Ésta es la rutina del putback del repartidor para el dispositivo CAS. El carácter es simplemente colocado en CASPRV para ser recogido en la siguiente petición secuencial de entrada.

La dirección ... 7281H

Esta rutina es usada por la función del espacio cerrado del repartidor a comprobar si hay cualquier personajes en lo yo el amortiguador de la / O y luego ponen en el cero lo yo el byte de colocación del amortiguador de la / O en el FCB.

La dirección ... 728BH

Esta rutina está usada por el repartidor función secuencial de salida para colocar el carácter en A de registro en lo yo el amortiguador de la / O en la corriente yo el amortiguador de la / O sitio, lo cual es entonces incrementada.

La dirección ... 729BH

Esta rutina está usada por el repartidor función secuencial de entrada para coleccionar el carácter en la corriente yo colocación del amortiguador de la / O, lo cual es entonces incrementada.

La dirección ... 72A6H

Esta mesa es usada por el repartidor al descifrar códigos de función para el dispositivo LPT. Contiene la dirección del manipulador para cada uno de los códigos de función, la mayoría está en generadores de error de hecho:

PARA FUNCIONAR

71B6H 0, abierto
71C2H 2, cerca
6E86H 4, aleatorio
72BAH 6, salida secuencial
475AH 8, entrada secuencial
475AH 10, loc
475AH 12, lof
475AH 14, eof
475AH 16, fpos
475AH 18, putback

La dirección ... 72BAH

Éste es el repartidor rutina secuencial de salida para el LPT

- 185 -

5. EL INTÉRPRETE DE BASIC ROM

El dispositivo. El carácter a la salida es tomado de transferencias de registro de la C y de control para la rutina del estándar OUTDLP.

La dirección ... 72BEH

Esta rutina está usada por el repartidor función secuencial de entrada para comprobar si un carácter del putback existe en CASPRV, y en caso de que no devolver Z Flag. CASPRV diferente es puesto en el cero y el carácter probado para ver si es el fin de carácter del archivo (1AH). En caso de que no regresa con el carácter en A de registro y FLAG NZ, Carolina Del Norte. De otra manera el fin de carácter del archivo es regresado a CASPRV y los ingresos de rutina con FLAG Z, C.

La dirección ... 72CDH

Esta rutina es usada por las funciones diversas del repartidor a comprobar si el modo en E de registro es anexo, si así es que un "mal nombre del archivo" error es generado (6E6BH).

La dirección ... 72D4H

Esta rutina es usada por las funciones diversas del repartidor para leer un carácter del casete. El carácter es leído por la rutina del estándar TAPIN y un "dispositivo yo que el error de la / O" generó (73B2H) si FLAG C es devuelto.

La dirección ... 72DEH

Esta rutina es usada por las funciones diversas del repartidor para escribir un carácter para casete. El carácter está escrito por la rutina del estándar TAPOUT y un "dispositivo yo que el error de la / O" generó (73B2H) si FLAG C es devuelto.

La dirección ... 72E9H

Esta rutina es usada por las funciones diversas del repartidor para encender el casete motor para la entrada. El motor se enciende por la rutina del estándar TAPION y un "dispositivo yo que el error de la / O" generó (73B2H) si FLAG C es devuelto.

La dirección ... 72F8H

Esta rutina es usada por las funciones diversas del repartidor para encender el casete motor para la salida, el control simplemente se traslada para la rutina del estándar TAPOON.

La dirección ... 7304H

Esta rutina está usada por el punto Interpreter Mainloop "Oklahoma", el "fin" manipulador de declaración y la rutina cristalina en la carrera para cerrar la impresora. PRTFLG es primero LPTPOS puestos en el cero y luego probados para ver si cualquier personajes han sido salida pero izquierda pendiendo en el amortiguador de la línea de la impresora. Si así un RETORNO DE CARRO, una secuencia LF es expedido para sonrojar la impresora y LPTPOS puesto en el cero.

- 186 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 7323H

Esta rutina emite un RETORNO DE CARRO, una secuencia LF para el dispositivo de salida en uso en estos momentos por la rutina del estándar OUTDO. LPTPOS o TTYPOS es entonces puesto en el cero dependiente ya sea la impresora o la pantalla es activa.

La dirección ... 7347H

Esta rutina es usada por el Evaluatodor Factor para ejercer al "INKEY\$" función. La condición del amortiguador del teclado es examinada por la rutina del estándar CHSNS. Si el amortiguador está vacío que la dirección de un descriptor nulo falso de la cuerda es devuelta en DAC. De otra manera el siguiente carácter es leído del amortiguador del teclado por la rutina del estándar CHGET. Después de comprobar eso suficiente espacio está disponible (6625H) el carácter es emulado para el String Storage Area y el resultado que el descriptor creó (6821H).

La dirección ... 7367H

Esta rutina es usada por la "lista" manipulador de declaración para devolver un carácter para el dispositivo de salida en uso en estos momentos por la rutina del estándar OUTDO. Si el carácter es un código LF luego un código de RETORNO DE CARRO es también expedido.

La dirección ... 7374H

Esta rutina es usada por el Interpreter Mainloop para coleccionar una línea de texto cuando la entrada está de uno yo el amortiguador de la / O en vez del teclado, esto es cuando una "carga" declaración es activa. Los personajes son secuencialmente entrada (6C71H) y colocado en BUF hasta que BUF se llena, un RETORNO DE CARRO es detectado o el fin de archivo es cumplido. Todos los personajes son aceptados con la excepción de los códigos LF que son limpiados con un filtro. Si BUF se llena o un RETORNO DE CARRO es detectado, entonces la rutina simplemente devuelve la línea para el Mainloop. Si el fin de archivo es alcanzado mientras algunos personajes están en BUF la línea es devuelta al Mainloop. Cuando el fin de archivo es cumplido sin personajes en BUF luego yo amortiguador de la / O 0 estoy cerrado (6D7BH) y FILNAM al que se dio jaque para determinar si andado en automóvil es requerido. Si no el control regresa al Interpreter "Oklahoma" apunte (411EH). De otra manera el sistema es aclarado (629AH) y control se traslada para el Runloop (4601H) para ejecutar el programa.

La dirección ... 73B2H

Éste es el "dispositivo yo" el generador de "error de la / O".

La dirección ... 73B7H

Éste es el manipulador "motor" de declaración. Si ningún operand es, entonces las transferencias presentes de control para la rutina del estándar STMOTR con FFH de adentro registran UNO. Si la señal "apagada" (EBH) sigue, entonces las transferencias de control con 00H de adentro registran UNO. Si el "ON" señal (95H) entiende, entonces las transferencias de control con 01H de adentro registran UNO.

- 187 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 73CAH

Éste es el manipulador "atinado" de declaración. El número de registro operand, cuál debe estar menos de catorce, es evaluado (521CH) y colocado en A de registro. El operand de datos es evaluado y la mordida reanudación 7 determinada, mordida 6 para evitar alterar al ayudante PSG yo los modos del puerto de la / O.

La dirección ... 73E4H

Éste es un solo espacio de ASCII usado por el "juego" manipulador de declaración para reemplazar un operand nulo de la cuerda con un que cuerda del espacio vacío de carácter.

La dirección ... 73E5H

Éste es el "juego" manipulador de declaración. La dirección del "juego" mesa de orden en 752EH es colocada en MCLTAB para el analizador gramatical de lenguaje de macro y PRSCNT puesto en el cero. La primera cuerda operand, cuál es obligatoria, es evaluada (4C64H), su almacenamiento liberado (67D0H) y su largo y dirección colocada en VCBA en los bytes 2, 3 y 4. El puntero de la pila del canal es en el que se puso las iniciales para VCBA + 33 y está posado en VCBA en los bytes 5 y 6 ' Si el más texto está presente en La declaración que este proceso

está repetido para B de voces y C hasta un máximo de tres operands ha sido evaluada, después de ésta un "error de Sintaxis" es generado (4055H). Si hay menos de tres presente de operands de la cuerda, entonces un fin de marca de la cola (FFH) es colocado en la cola (7507H) de cada voz sin uso. La A de registro es entonces puesta en el cero, a hacer una selección sonoriza UNO, y el control cae en el mainloop de juego.

La dirección ... 744DH

Éste es el mainloop de juego. El número de bytes libres en la cola en uso en estos momentos es comprobado (7521H) y, si menos de ocho resto de bytes, la siguiente voz es seleccionada (74D6H) para evitar en espera de la cola vaciarse. El largo restante de la cuerda del operand es luego tomado del amortiguador de voz en uso en estos momentos y, si los bytes de cero quedan ser a los que se analizó gramaticalmente, el lazo otra vez saltos para la siguiente voz (74D6H). De otra manera el largo coetáneo de la cuerda y dirección son tomados del amortiguador de voz y están posado en MCLLEN y MCLPTR para el analizador gramatical de lenguaje de macro. Los contenidos viejos de la pila son emulados del amortiguador de voz para la pila Z80 (6253H), MCLFLG es control y poco cero hecho se traslada para el analizador gramatical de lenguaje de macro (56A2H).

El analizador gramatical de lenguaje de macro normalmente escudriñará a lo largo de la cuerda, usando el "juego" manipuladores de orden de declaración, hasta que la cuerda esté agotada. Sin embargo, si una cola de música se llena arriba durante la generación de la nota, entonces una terminación anormal sea a la que se hizo retroceder al juego mainloop (748EH) a fin de que la siguiente voz puede ser tramitada fuera en espera de la cola para vaciarse. Cuando el control devuelve normalmente un fin de marca de la cola es colocado en la cola en uso en estos momentos

- 188 -

5. EL INTÉRPRETE DE BASIC ROM

(7507H) y PRSCNT es incrementado para mostrar el número de instrumentos de cuerda completados. Si el control regresa anormalmente luego cualquier cosa dejó en la pila Z80 es copiado en el amortiguador de voz (6253H) en uso en estos momentos.

Por la naturaleza recursiva del analizador gramatical de lenguaje de macro donde la "X" orden está involucrada puede haber un número de cuatro descriptores de la cuerda de byte, marcando el punto donde la cuerda original estaba suspendida, izquierda en la pila Z80 en la terminación. Salvo la pila los contenidos en el amortiguador de voz significan que pueden estar recuperados cuando el lazo tiene tiempo para esa voz otra vez. Note eso como haya sólo dieciséis bytes disponibles en cada amortiguador de voz una "llamada ilegal de función" error son generados (475AH) si demasiados restos de datos en la pila. Esto ocurrirá cuando una cola se llena hasta el tope y "X" múltiple, anidada órdenes existe, por ejemplo:

```
10 A$ "XB$""  
20 B$ "XC$""  
30 C$ "XD$""  
40 D$ STRING$ (150, "uno")  
50 JUEGAN CONTRA A$
```

Parece un problema no resuelto en el software leve en este pasaje como sólo quince bytes de datos de la pila están permitidos, en lugar de eso de dieciséis, antes de que un error sea generado.

Cuando los ingresos de control del analizador gramatical de lenguaje de macro registran UNO

es incrementado para seleccionar la siguiente voz para ir en procesión. Cuando todas las tres voces han sido INTFLG tratado es al que se dio jaque y, si CTRL-STOP ha sido detectado por ahí lo interrumpe transferencias del manipulador, de control para la rutina del estándar GICINI para detener toda música y terminar. Asumir mordió 7 de PRSCNT muestran así de ser el paso de primera parte a través del mainloop, esto es ninguna voz ha por ahora sido suspendida por una cola llena, PLYCNT es incrementado e interrumpe dequeueing echado a andar por la rutina del estándar STRTMS. PRSCNT es entonces a cuadros para determinar el número de instrumentos de cuerda completados por el analizador gramatical de lenguaje de macro. Si todo tres instrumentos de cuerda del operand han sido completados, entonces el manipulador termina, transferencias diferentes de control de regreso al principio del mainloop de juego para intentar de nuevo cada voz.

La dirección ... 7507H

Esta rutina es usada por el "juego" manipulador de declaración para colocar un fin de marca de la cola (FFH) en la cola en uso en estos momentos por la rutina del estándar PUTQ. Si la cola está llena, entonces espera hasta que el espacio se haga disponible.

La dirección ... 7521H

Esta rutina es usada por el "juego" manipulador de declaración a comprobar cuánto restos de espacio en la cola en uso en estos momentos por la rutina del estándar LFTQ. Si menos de ocho bytes se queda (el paquete posible más grande de datos de música tiene siete bytes de largo), entonces FLAG C es devuelto.

- 189 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 752EH

Esta mesa contiene las cartas válidas de orden y las direcciones asociadas para el "juego" órdenes de declaración. Esas órdenes que toman un parámetro, y consecuentemente han mordido 7 colocado en la mesa, es mostrado con un asterisco:

CMD PARA

Un 763EH

B 763EH

C 763EH

D 763EH

E 763EH

F 763EH

G 763EH

M * 759EH

V * 7586H

S * 75BEH

N * 7621H

O * 75EFH

* el 75FCH de interrogación

T * 75E2H

L * 75C8H

X 5782H

La dirección ... 755FH

Esta mesa es usada por la "A" de declaración de "juego" para "G" manipulador de orden para traducir un número de la nota de cero a catorce para un offset en la mesa de divisor de tono en 756EH. La nota misma, en vez del número de la nota, es mostrada debajo con cada valor de offset:

16 ... Uno
18 ... Uno
20 ... Uno + O B
22 ... B O C
00 ... B
00 ... C
02 ... C + O D
04 ... D
06 ... D + O E-
08 ... E O F
10 ... E
10 ... F
12 ... F + O G
14 ... G
16 ... G

La dirección ... 756EH

Esta mesa contiene las doce constantes de divisor PSG requeridas producir los tonos de octava 1. Para cada constante

- 190 -

5. EL INTÉRPRETE DE BASIC ROM

La nota correspondiente y frecuencia son exteriorizadas:

3421 ... C 32.698 Hz
3228 ... C + 34.653 Hz
3047 ... D 36.712 Hz
2876 ... D + 38.895 HZ
2715 ... E 41.201 Hz
2562 ... F 43.662 Hz
2419 ... F + 46.243 Hz
2283 ... G 48.997 Hz
2155 ... G + 51.908 Hz
2034 ... Un 54.995 Hz
1920 ... Un + 58.261 Hz
1812 ... B 61.773 Hz

La dirección ... 7586H

Esto es el "juego" manipulador de orden de declaración "V". El parámetro, con un valor predeterminado de ocho, es colocado en byte 18 del amortiguador de voz en uso en estos momentos sin alterar bit 6 de los contenidos existentes. Ninguno de los datos de música son generados.

La dirección ... 759EH

Esto es el "juego" manipulador de orden de declaración "M". El parámetro, con un valor predeterminado de doscientos cincuenta y cinco, es comparado con el período existente de modulación contenido en bytes 19 y 20 del amortiguador de voz en uso en estos momentos. Si son lo mismo la rutina termina sin acción. De otra manera el período nuevo de modulación es colocado en el amortiguador de voz y set 6 mordido en el byte 18 del amortiguador de voz a señalar que el valor nuevo debe ser incorporado en el siguiente paquete de datos de música producido.

Ninguno de los datos de música son generados.

La dirección ... 75BEH

Esto es el "juego" manipulador de orden de declaración "S". El parámetro es colocado en byte 18 del amortiguador de voz en uso en estos momentos y el bit 4 del mismo byte se sedimentó señalar que el valor nuevo debe ser incorporado en el siguiente paquete de datos de música producido. Ninguno de los datos de música son generados. Por las características PSG la forma y parámetros de volumen son mutuamente exclusivos así es que el mismo byte de los amortiguadores de voz sirve para ambos.

La dirección ... 75C8H

Esto es el "juego" manipulador de orden de declaración "L". El parámetro, con un valor predeterminado de cuatro, es colocado en byte 16 del amortiguador de voz en uso en estos momentos donde es usado en la computación de tener éxito duraciones de la nota. Ninguno de los datos de música son generados.

La dirección ... 75E2H

Esto es el "juego" manipulador de orden de declaración "T". Lo

- 191 -

5. EL INTÉRPRETE DE BASIC ROM

El parámetro, con un valor predeterminado de ciento veinte, es colocado en byte 17 del amortiguador de voz en uso en estos momentos donde será usado en la computación de tener éxito duraciones de la nota datos de música ho es generado.

La dirección ... 75EFH

Esto es el "juego" manipulador de orden de declaración "O". El parámetro, con un valor predeterminado de cuatro, es colocado en byte 15 del amortiguador de voz en uso en estos momentos donde es usado en la computación de tener éxito frecuencias de la nota. Ninguno de los datos de música son generados.

La dirección ... 75FCH

Éste es la "interrogación" de declaración de "juego" manipulador de orden. El parámetro de largo, con un valor predeterminado de cuatro, se queda en par de registro Delaware y un valor de divisor de tono de cero colocado en par de registro HL. El valor existente de volumen es tomado de byte 18 del amortiguador de voz en uso en estos momentos, por ahora repuesto con un valor de cero y un control transferido para el generador de la nota (769CH).

La dirección ... 7621H

Esto es el "juego" manipulador de orden de declaración "N". El parámetro obligatorio es primero examinado, si es cero un descanso es generado (760BH). Si es mayor que noventa y seis que un error "ilegal de llamada de función" es generado (475AH). De otra manera doce son

repetidamente sustraídos del número de la nota hasta el poco flujo para sacar un número de octava de uno a nueve en E de registro y un número de la nota de cero a once en C de registro. El control luego se traslada para el generador de la nota (7673H).

La dirección ... 763EH

Ésta es la "A" de declaración de "juego" para "G" manipulador de orden. La carta de la nota se convierte primero en un número de la nota de cero para catorce, ésta expandió rango siendo menester por la redundancia implícita en la notación. La mesa en 755FH se usa luego para obtener el offset en la mesa de divisor de tono y la constante divisionista para la nota colocada en par de registro Delaware. El valor de octava es tomado de byte 15 del amortiguador de voz en uso en estos momentos y la constante divisionista disminuida a la mitad hasta que la octava correcta sea cumplida. El operand de la cuerda es entonces examinado directamente (56EEH) para determinar si una nota de arrastre largo parámetro existe.

Si así es que es convertido (572FH) y colocado en C de registro. Si ningún parámetro existe, entonces el largo predeterminado es tomado de byte 16 del amortiguador de voz en uso en estos momentos. La duración de la nota es entonces computada de:

$$\text{La duración (Interrumpen hace tictac)} = 12,000 / (\text{LENGTH} * \text{TEMPO})$$

Con el largo normal el valor (4) y el tiempo aprecian (120) éste da una duración de la nota de veinticinco interrupciones tictacs de 20 ms cada uno o 0.5 segundos. El operand de la cuerda es entonces examinado

- 192 -

5. EL INTÉRPRETE DE BASIC ROM

(56EEH) para rezagarse "" Los personajes y, para cada uno, la duración multiplicado por uno y una mitad. Finalmente la duración resultante es comprobada y, si está menos de cinco interrupciones tictacs, es reemplazado con un valor de cinco. Así la nota más pequeña que puede ser generada en una máquina UK es 0.10 segundos cualquier cosa el tiempo o el largo de la nota.

El paquete de datos de música, cuál estarán tres, cinco de o siete bytes de largo, es luego ensamblado en bytes 8 para 14 del amortiguador de voz en uso en estos momentos antes de colocar eso en la cola. La duración es colocada en bytes 8 y 9 del amortiguador de voz. El volumen y el byte de la bandera son tomados de byte 18 y colocados en el byte 10 del amortiguador de voz con el set 7 mordido para indicar un cambio de volumen para lo interrumpa sacar de la cola rutina. Si mordió 6 del byte del libro están colocados luego el período de modulación es tomado de bytes 19 y 20 y se suma al paquete de datos en los bytes 11 y 12. Si el valor de divisor de tono es poco cero luego se agrega para el paquete de datos en los bytes 11 y 12 (sin un período de modulación) o los bytes 13 y 14 (con un período de modulación). Finalmente la cuenta de byte es mixta en los tres añicos más altos de byte 8 del amortiguador de voz para completar la preparación del paquete de datos de música.

Si el valor de divisor de tono es cero, entonces indicando un descanso, los contenidos de SAVVOL están recuperados para el byte 18 del amortiguador estático. El paquete de datos de música es luego colocado en la cola en uso en estos momentos por la rutina del estándar PUTQ y el número de permanecer libre de bytes inspeccionó (7521H). Si menos de ocho bytes se queda, entonces controle transferencias directamente para el "juego" el manipulador de declaración (748EH), el control diferente regresa normalmente para el analizador gramatical de lenguaje de macro.

La dirección ... 7754H

Ésta es la sola constante de precisión 12,000 usada en la computación de duración de la nota.

La dirección ... 7758H

Éste es el manipulador "puesto" de declaración. La B de registro está colocada para 80H y el control cae en el "GET" manipulador de declaración.

La dirección ... 775BH

Éste es el "GET" manipulador de declaración. La B de registro es puesta en el cero, para distinguir a "GET" de "PUT" y la siguiente señal de programa examinada. El control luego se traslada para el "bloque de objetos movibles PUESTO" manipulador de declaración (7AAFH) o el manipulador de declaración de BASIC Disk "GET/PUT" (6C35H).

La dirección ... 7766H

Éste es el "LOCATE" manipulador de declaración. Si una coordenada de la columna está presente, entonces es evaluada (521CH) y colocada en registro D, de otra manera la columna en uso en estos momentos es tomada de CSRX. Si

- 193 -

5. EL INTÉRPRETE DE BASIC ROM

Una coordenada de la fila está presente que es evaluada (521CH) y colocada en registro E, de otra manera la fila en uso en estos momentos es tomada de CSRY. Si un operand del interruptor del cursor existe, entonces es evaluada (521CH) y A de registro cargada con 78H para un cero operand (COMPLETAMENTE) y 79H para cualquier poco cero operand (ADELANTE). El cursor es entonces cambiado devolviendo a ESC, 78H/79H, "5" por la rutina del estándar OUTDO. La fila y coordenadas de la columna son colocadas en par de registro HL y la colocación del cursor set por la rutina del estándar POSIT.

La dirección ... 77A5H

Éste es el "alto ADELANTE / FUERA DE / el ALTO" el manipulador de declaración. La dirección de byte de estado TRPTBL del dispositivo es colocada en transferencias de registro del par HL y de control para el "ON/OFF/STOP" rutina (77CFH).

La dirección ... 77ABH

Éste es el "bloque de objetos movibles ADELANTE / FUERA DE / el ALTO" el manipulador de declaración. La dirección de byte de estado TRPTBL del dispositivo es colocada en transferencias de registro del par HL y de control para el "ON/OFF/STOP" rutina (77CFH).

La dirección ... 77B1H

Éste es el "intervalo ADELANTE / FUERA DE / el ALTO" el manipulador de declaración. Como no hay "el INTERVALO" específico la señal (controle transferencias aquí cuando un "INT" señal es encontrado) un cheque es primero hecho en el texto de programa para los personajes "E" y "interrogación" luego el "VAL" señal (94H). La dirección de byte de estado TRPTBL del dispositivo es colocada en transferencias de registro del par HL y de control para el "ON/OFF/STOP" rutina (77CFH).

La dirección ... 77BFH

Éste es el "STRIG ADELANTE / FUERA DE / el ALTO" el manipulador de declaración. El número del gatillo, de cero para cuatro, es evaluado (7C08H) y la dirección del TRPTBL del dispositivo que el byte de estado colocó en par de registro HL. El "ON/OFF/STOP" señal es examinado y el byte de estado TRPTBL modificado consecuentemente (77FEH). El control luego se traslada directamente para el Runloop (4612H) para evitar prueba para antes de interrumpen hasta el fin de la siguiente declaración.

La dirección ... 77D4H

Ésta es la "llave (n) ON / FUERA DE / el ALTO" el manipulador de declaración. El número crucial, de uno para diez, es evaluado (521CH) y la dirección del TRPTBL de los dispositivos que el byte de estado colocó en par de registro HL. El "ON/OFF/STOP" señal es examinado y el byte de estado TRPTBL modificado consecuentemente (77FEH). El bit 0 del byte de estado TRPTBL, el ON mordió, es entonces copiado en la entrada correspondiente en FNKFLG para el uso durante lo interrumpen transferencias del keyscan y de control directamente para el Runloop (4612H).

- 194 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 77FEH

Esta rutina revisa en busca de la presencia de uno de lo interrumpa cambiar señales y transfiere control para la rutina apropiada: "Adelante" (631BH), "fuera" (632BH) o "alto" (6331H). Si ninguna señal está presente, entonces un "error de sintaxis" es generado (4055H).

La dirección ... 7810H

Esta rutina es usada por ahí lo "en DEVICE GOSUB" manipulador de declaración (490DH) para comprobar el texto de programa para una señal del dispositivo. A menos que ninguna de las señales del dispositivo son presentes, en cuyo caso la C Flag es devuelta, TRPTBL del dispositivo que el número de entrada es devuelto en B de registro y el operand admisible máximo de número de la línea incluyen C de registro:

LOS NÚMEROS DEL DISPOSITIVO TRPTBL#LINE

LLAVE 00 10
ALTO 10 01
EL BLOQUE DE OBJETOS MOVIBLES 11 01
STRIG 12 05
EL INTERVALO 17 01

Adicionalmente, para "el INTERVALO" sólo, el operand de intervalo es evaluado (542FH) y colocado en INTVAL e INTCNT.

La dirección ... 785CH

Esta rutina es usada por ahí lo "en DEVICE GOSUB" manipulador de declaración (490DH) para colocar la dirección de una línea de programa en TRPTBL. El número de entrada TRPTBL, suministrado en registro B, es multiplicado por tres y se suma a la mesa bajo para apuntar hacia

la entrada pertinente. La dirección, suministrado en par de registro Delaware, está luego posada allí el segundo LSB de primera parte, MSB.

La dirección ... 786CH

Éste es el manipulador "crucial" de declaración. Si el siguiente carácter es, entonces cualquier cosa aparte de la "lista" control de la señal (93H) se traslada para la "n CRUCIAL" manipulador de declaración (78AEH). Cada uno de los instrumentos de cuerda de diez funciones de crucial son luego tomados de FNKSTR y exhibidos por la rutina del estándar OUTDO con un RETORNO DE CARRO, LF (7328H) después de cada uno. El carácter DEL (7FH) o cualquier carácter de control más pequeño que 20H es reemplazado con un espacio.

La dirección ... 78AEH

Ésta es la "n CRUCIAL", "teclea a (n) ON/OFF/STOP", "crucial de ADELANTE" y "teclea FUERA DE" manipulador de declaración. Si el siguiente personaje del texto de programa está encendido "(" controle transferencias para la "llave (n) / FUERA DE / el ALTO" el manipulador de declaración (77D4H). Si es, entonces un "ON" control de la señal (95H) se traslada para la rutina del estándar DSPFNK y si es una señal "apagada" (EBH) para la rutina del estándar ERAFNK. De otra manera la función

- 195 -

5. EL INTÉRPRETE DE BASIC ROM

El número crucial es evaluado (521CH) y lo crucial la dirección FNKSTR colocó en par de registro Delaware ' El operand de la cuerda es evaluado (4C64H) y su almacenamiento liberado (67D0H) ' Hasta quince personajes es emulado de la cuerda para FNKSTR y las colocaciones sin uso acolchadas con bytes de cero. Si un byte de cero es encontrado en el operand forme una serie, entonces una "llamada ilegal de función" error es generada (475AH). El control luego se traslada para la rutina del estándar FNKSB para actualizar el despliegue de la llave de función si es posibilitado.

La dirección ... 7900H

Esta rutina es usada por el Evaluador Factor para ejercer la función "de tiempo". Los contenidos de MOMENTITO son colocados en DAC como un solo número de precisión (3236H).

La dirección ... 790AH

Esta rutina es usada por el Evaluador Factor para ejercer al "CSRLIN" función. Los contenidos de CSRY son decremented y colocado en DAC como un entero (2E9AH).

La dirección ... 7911H

Éste es el manipulador de declaración "de tiempo". El operand es evaluado (542FH) y colocado en el MOMENTITO.

La dirección ... 791BH

Esta rutina es usada por el Evaluador Factor para ejercer el "juego" función. El operand numérico de selección del canal es evaluado (7C08H). Si esto es ponga en el cero, entonces los contenidos de MUSICF son colocados en DAC como un entero de cero de valor o FFFFH. De otra manera el número del canal se usa para hacer una selección el bit apropiado de MUSICF y esto es luego convertido a un entero como antes.

La dirección ... 7940H

Esta rutina es usada por el Evaluador Factor para ejercer la "vara" función para un operand contenido en DAC. El número de la vara es comprobado (521FH) y pasado para la rutina del estándar GTSTCK en A de registro. El resultado es colocado en DAC como un entero (4FCFH).

La dirección ... 794CH

Esta rutina es usada por el Evaluador Factor para ejercer al "STRIG" función para un operand contenido en DAC. El número del gatillo es comprobado (521FH) y pasado para la rutina del estándar GTTRIG en A de registro. El resultado es colocado en DAC como un entero de cero de valor o FFFFH.

La dirección ... 795AH

Esta rutina es usada por el Evaluador Factor a aplicar lo

- 196 -

5. EL INTÉRPRETE DE BASIC ROM

"PDL" funcione para un operand contenido en DAC. El número del remo es comprobado (521FH) y pasado para la rutina del estándar GTPDL en A de registro. El resultado es colocado en DAC como un entero (4FCFH).

La dirección ... 7969H

Esta rutina es usada por el Evaluador Factor para ejercer la "almohadilla" función para un operand contenido en DAC. El número de la almohadilla es comprobado (521F) y pasado para la rutina del estándar GTPAD en A de registro. El resultado es colocado en DAC como un entero para almohadillas 1, 2, 5 o 6. Para almohadillas 0, 3, 4 o 7 que el resultado es colocado en DAC como un entero de cero de valor o FFFFH.

La dirección ... 7980H

Éste es el "color" manipulador de declaración. Si un operand de color del primer plano existe, entonces es evaluado (521CH) y colocado en registro E, de otra manera el color actual del primer plano es tomado de FORCLR. Si un operand de color de fondo existe, entonces es evaluado (521CH) y colocado en registro D, de otra manera el color de fondo actual es tomado de BAKCLR. Si un operand fronterizo de color existe, entonces es evaluado (521CH) y colocado en BDRCLR. El color del primer plano es colocado en FORCLR y ATRBYT, el color de fondo en las transferencias BAKCLR y de control para la rutina del estándar CHGCLR para modificar al VDP.

La dirección ... 79CCH

Éste es la "pantalla" manipulador de declaración. Si un operand de modo existe, entonces es evaluado (521CH) y pasado para la rutina del estándar CHGMOD en A de registro. Si un operand de tamaño de bloque de objetos movibles existe, entonces es evaluado (521CH) y colocado en añicos 0 y 1 de RG1SAV, la copia Workspace Area de VDP Mode Register 1. Los parámetros de bloque de objetos movibles VDP son entonces aclarados por la rutina del estándar CLRSPPR. Si un operand crucial de chasquido existe, entonces es evaluado (521CH) y colocado en CLIKSW, cero para desactivar el chasquido y el poco cero para posibilitarlo. Si un operand de tasa de baudio existe, entonces es evaluado y la tasa de baudio se sedimentase (7A2DH). Si un operand de modo de la impresora existe, entonces es evaluado (521CH) y colocado en NTMSXP, cero para una impresora MSX y poco ponga en el cero para

una impresora general de propósito.

La dirección ... 7A2DH

Esta rutina se usa para ajustar la tasa de baudio del casete. El operand es evaluado (521CH) y cinco bytes copiados de CS1200 o CS2400 al PUNTO BAJO según el caso.

La dirección ... 7A48H

Éste es el "bloque de objetos movibles" manipulador de declaración. Si el siguiente carácter es, entonces cualquier cosa aparte de un "\$" control se traslada para el "bloque de objetos movibles ON/OFF/STOP" manipulador de declaración (77ABH). SCRMOD es entonces a cuadros y una "llamada ilegal de función" que el error generó (475AH)

- 197 -

5. EL INTÉRPRETE DE BASIC ROM

Si la pantalla está de Modo del 40x24 Text. El número del patrón de bloque de objetos movibles es evaluado y su posición en el VRAM que el Sprite Pattern Table obtuvo (7AA0H). El operand de la cuerda es entonces evaluado (4C5FH) y su almacenamiento liberado (67D0H). El tamaño de bloque de objetos movibles, obtenido por la rutina del estándar GSPSIZ, es comparado con el largo de la cuerda y, si la cuerda es más corta que el bloque de objetos movibles, la entrada Sprite Pattern Table es primero llenada de pone en el cero por la rutina del estándar FILVRM. Los personajes son entonces copiados del cuerpo de la cuerda para el Sprite Pattern Table por la rutina del estándar LDIRVM hasta que la cuerda está agotada o el bloque de objetos movibles está lleno. Si la cuerda es más larga que el bloque de objetos movibles tamaño cualquier personajes excedentes está ignorado.

La dirección ... 7A84H

Esta rutina es usada por el Evaluador Factor para ejercer al "SPRITE\$" función. El número del patrón de bloque de objetos movibles es evaluado y su posición en el VRAM que el Sprite Pattern Table obtuvo (7A9FH).

El tamaño de bloque de objetos movibles, obtenido por la rutina del estándar GSPSIZ, es luego colocado en par de registro BC para controlar el número de bytes emulados. Después de comprobar eso suficiente espacio está disponible en el String Storage Area (6627H) el patrón de bloque de objetos movibles es copiado de VRAM por la rutina del estándar LDIRMV y el resultado que el descriptor creó (6654H). Note eso como ningún cheque esté hecho en el modo de la pantalla durante esta función algunos efectos secundarios interesantes pueden ser encontrados, pueden ver debajo.

La dirección ... 7A9FH

Esta rutina es usada por el "SPRITE\$" declaración y la función para localizar un patrón de bloque de objetos movibles en el VRAM Sprite Pattern Table.

El operand de número del patrón es evaluado (7C08H) y pasado para la rutina del estándar CALPAT en A de registro. La dirección del patrón es colocada en par de registro Delaware y la rutina termina.

Noto que ningún cheque está hecho en la magnitud de número del patrón para el bloque de objetos movibles discrepante dimensiona. El patrón numera hasta doscientos cincuenta y cinco está aceptado aun adentro modo de bloque de objetos movibles del 16x16 cuándo el número máximo del patrón debería tener sesenta y tres años de edad. Como un resultado del que VRAM se ocupa más gran que 3FFFH se producirá cuál envolverá alrededor de en VRAM bajo. Con el "SPRITE\$" declaración esto corromperá al Character Generator Table, por ejemplo:


```
10 OCULTAN 3,2
20 SPRITE$ (0) = STRING$ (32,255)
30 PONEN BLOQUE DE OBJETOS MOVIBLES 0, (0,0),, 0
40 SPRITE$ (65) = STRING$ (32,255)
50 GOTO 50
```

Lo antedicho mete un bloque de objetos movibles real en la izquierda máxima de la pantalla y luego usa una declaración ilegal en línea 40 para corromper al VRAM justamente a la derecha de él. El "SPRITE\$" función también puede ser manipulado de este modo y, ya que no hay cheque de modo de la pantalla, hasta treinta y dos bytes del Name Table puede ser leído en 40x24

- 198 -

5. EL INTÉRPRETE DE BASIC ROM

El texto Mode, por ejemplo:

```
10 OCULTAN 0,2
20 PRINT"something "
30 A$ SPRITE$ (64)
40 ESCRIBEN EN LETRAS DE IMPRENTA a A$
```

La dirección ... 7AAFH

Éste es el "bloque de objetos movibles GET/PUT" manipulador de declaración, el control es transferido aquí del "GET/PUT" general manipulador de declaración (775BH). La B de registro es primero a cuadros para hacer seguro que la declaración es "PUT" y una "llamada de función ilegal" error generado (475AH) si de otra manera. SCRMOD es entonces a cuadros y una "llamada de función ilegal" error generado (475AH) si la pantalla está de Modo del 40x24 Text. El número de bloque de objetos movibles operand, de cero para treinta y uno, es evaluado (521CH) y pasado para la rutina del estándar CALATR para localizar el bloque de atributo de cuatro bytes en el Sprite Attribute Table. Si un operand coordinado existe, entonces es evaluado y la coordenada de la X colócase en par de registro BC, la Y coordinada en par de registro Delaware (579CH).

La coordenada de la Y a LSB le es escado para el byte 0 del bloque de atributo en VRAM por la rutina del estándar WRTVRM. El bit 7 de la coordenada de la X es entonces examinado para determinar si es negativa, esto es fuera de la izquierda lateral de la pantalla. Si tan treinta y dos se agrega para la coordenada de la X y la B de registro está colocada para 80H para se sedimentó, entonces el reloj temprano mordió en el bloque de atributo. Para el ejemplo una X coordinada de - 1 (FFFFH) se variaría para + 31 con un reloj temprano. La coordenada de la X a LSB le es luego escado para el byte 1 del bloque de atributo por la rutina del estándar WRTVRM.

El byte 3 del bloque de atributo es leído adentro por la rutina del estándar RDVRM, el bit temprano nuevo del reloj es en el que se mezcló y es luego escrito hacia atrás para VRAM por la rutina del estándar WRTVRM.

Si un operand de color está presente, entonces es evaluado (521CH), el byte 3 del bloque de atributo es leído adentro por la rutina del estándar RDVRM que el código nuevo de color es mixto en el cuatro bits mínimo y eso es contestado por escrito a VRAM por la rutina del estándar WRTVRM.

Si un operand de número del patrón existe, entonces es evaluado (521CH) y comprobado para la

magnitud en contra del bloque de objetos movibles actual que el tamaño proveyó por la rutina del estándar GSPSIZ. El número admisible máximo del patrón es doscientos cincuenta y cinco para bloques de objetos movibles del 8x8 y sesenta y tres para bloques de objetos movibles del 16x16. El número del patrón le es escado para el byte 2 del bloque de atributo por la rutina del estándar WRTVRM y el manipulador termina.

La dirección ... 7B37H

Éste es el "VDP" manipulador de declaración. El número de registro operand, de cero para siete, es evaluado (7C08H) seguido por el datos operand (521CH). El número de registro es al que se estuvo agotado registro C, el valor de datos en B de registro y el control transferido para la rutina del estándar WRTVDP.

- 199 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 7B47H

Esta rutina es usada por el Evaluatodor Factor para ejercer al "VDP" función. El número de registro operand, de cero para ocho, es evaluado (7C08H) y añadido para RGOSAV para hallar la imagen correspondiente de registro en el Workspace Area. La imagen de registro VDP es entonces leída y colocada en DAC como un entero (4FCFH).

La dirección ... 7B5AH

Éste es el manipulador "bajo" de declaración. El número de la mesa VDP operand, de cero para diecinueve, es evaluado (7C08H) seguido por la dirección bajo operand (4C64H). Después de comprobar que la dirección bajo está menos de 4000H (7BFEH) el número de la mesa VDP se usa para hallar la entrada asociada en el filtrado posponga en 7BA3H. La dirección bajo es ANDed con la máscara y una "llamada de función ilegal" error generado (475AH) si cualquier añicos ilegales están colocados. El número de la mesa VDP se agrega luego a TXTNAM para localizar la dirección bajo en uso en estos momentos en el Workspace Area y la nueva dirección bajo acomodada allí. El número de la mesa VDP está dividido a las cinco para determinar cuál de los cuatro modos de la pantalla la mesa pertenece para. Si esto equivale al modo coetáneo de la pantalla la dirección bajo nueva es también escrito para el VDP (7B99H).

La dirección ... 7B99H

Esta rutina es usada por el manipulador "bajo" de declaración para actualizar las direcciones de la base VDP. El modo coetáneo de la pantalla, en A de registro, es examinado y el control se traslada para el SETTXT, SETT32, SETGRP o rutina del estándar SETMLT según el caso. Noto que ésta no es una inicialización llena VDP y que la cuatro mesa en uso en estos momentos dirige la palabra (NAMBAS, CGPBAS, PATBAS y ATRBAS) cuál son los unos realmente usó por las rutinas de la pantalla, no está actualizada. Esto puede ser demostrado con lo siguiente, donde el Interpreter continúa devolver para el viejo VRAM Name Table:

```
10 OCULTAN 0
20 BASAN (0) = y H400
30 PRINT"something "
40 FOR N = 1 TO 2000:DESPUÉS
50 BASAN (0) = 0
```

Note también que esta rutina contiene un problema no resuelto en el software. Mientras SETTXT sirve correctamente para 40x24 Text Mode, SETGRP sirve para 32x24 Text Mode y

SETMLT para Graphics Mode y el Multicolor Mode.
Cualquier "base" declaración por consiguiente debería ser inmediatamente seguido por una "pantalla" declaración para realizar una inicialización llena.

La dirección ... 7BA3H

Esta mesa de filtrado es usada por el manipulador "bajo" de declaración para asegurar que sólo las direcciones legales de la base VDP son aceptadas. Lo

- 200 -

5. EL INTÉRPRETE DE BASIC ROM

El número de la mesa y la variable correspondiente Workspace Area son exteriorizados con cada uno máscara:

La MESA de la MÁSCARA

03FFH 00, TXTNAM
003FH 01, TXTCOL
07FFH 02, TXTCGP
007FH 03, TXTATR
07FFH 04, TXTPAT
03FFH 05, T32NAM
003FH 06, T32COL
07FFH 07, T32CGP
007FH 08, T32ATR
07FFH 09, T32PAT
03FFH 10, GRPNAM
1FFFH 11, GRPCOL
1FFFH 12, GRPCGP
007FH 13, GRPATR
07FFH 14, GRPPAT
03FFH 15, MLTNAM
003FH 16, MLTCOL
07FFH 17, MLTCGP
007FH 18, MLTATR
07FFH 19, MLTPAT

La dirección ... 7BCBH

Esta rutina es usada por el Evaluador Factor para ejercer la función "bajo". El número de la mesa VDP operand, de cero para diecinueve, es evaluado (7C08H) y añadido para TXTNAM para localizar la dirección bajo Area Workspace requerida. Esto es luego colocado en DAC como un solo número de precisión (3236H).

La dirección ... 7BE2H

Éste es el "VPOKE" manipulador de declaración. El operand de la dirección VRAM es evaluado (4C64H) y comprobado para asegurar que está menos de 4000H (7BFEH). El operand de datos es entonces evaluado (521CH) y pasado para la rutina del estándar WRTVRM en A de registro para escribir para la dirección requerida.

La dirección ... 7BF5H

Esta rutina es usada por el Evaluador Factor para ejercer al "VPEEK" función para un operand contenido en DAC. El operand de la dirección VRAM es comprobado para asegurar está menos de 4000H (7BFEH). VRAM es entonces leído por la rutina del estándar RDVRM y el resultado colocó en DAC como un entero (4FCFH).

La dirección ... 7BFEH

Esta rutina convierte a un operand numérico en DAC a un entero (2F8AH) y coloca ella en par de registro HL. Si el operand es

- 201 -

5. EL INTÉRPRETE DE BASIC ROM

El igual para o más gran que 4000H, y así fuera del rango admisible VRAM, una "llamada ilegal de función" error sea generada (475AH).

La dirección ... 7C08H

Esta rutina evalúa (521CH) un parenthesized operand numérico y la devuelve como un entero en A de registro. Si el operand es mayor que el máximo valor admisible inicialmente abastecido en A de registro una "llamada ilegal de función" error es generada (475AH).

La dirección ... 7C16H

Éste es el "DSKO\$" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C1BH

Éste es el manipulador "determinado" de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C20H

Éste es el "nombre" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C25H

Éste es la "presa" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C2AH

Éste es el "IPL" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C2FH

Éste es la "copia" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C34H

Éste es el "CMD" manipulador de declaración. Una "llamada ilegal de función" error es

generada (475AH) en una máquina estándar MSX.

La dirección ... 7C39H

Esta rutina es usada por el Evaluador Factor para ejercer al "DSKF" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

- 202 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 7C3EH

Esta rutina es usada por el Evaluador Factor para ejercer al "DSKI\$" función. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C43H

Esta rutina es usada por el Evaluador Factor para ejercer al "ATTR\$" función. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C48H

Éste es el "LSET" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C4DH

Éste es el "RSET" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX. L

La dirección ... 7C52H

Éste es el "campo" manipulador de declaración. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C57H

Esta rutina es usada por el Evaluador Factor para ejercer al "MKI\$" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C5CH

Esta rutina es usada por el Evaluador Factor para ejercer al "MKS\$" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C61H

Esta rutina es usada por el Evaluador Factor para ejercer al "MKD\$" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C66H

Esta rutina es usada por el Evaluatodor Factor para ejercer al "CVI" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

- 203 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 7C6BH

Esta rutina es usada por el Evaluatodor Factor para ejercer a los "CVS" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C70H

Esta rutina es usada por el Evaluatodor Factor para ejercer al "CVD" función para un operand contenido en DAC. Una "llamada ilegal de función" error es generada (475AH) en una máquina estándar MSX.

La dirección ... 7C76H

Esta rutina completa la inicialización que se impulsa arriba. En este punto el Workspace entero Area es puesto en el cero y sólo EXPTBL y SLTTBL han sido inicializados. Una pila temporal está colocada en F376H y todos los ciento doce ganchos (560 bytes) llenados de Z80 RET opcodes (C9H). HIMEM está listo para F380H y la RAM mínima que la posición encontró (7D5DH) y colocó en RAÍZ. Los cien cuarenta y cuatro bytes de datos comenzando en 7F27H son emulados para el Workspace Area de F380H para F40FH Los instrumentos de cuerda de la llave de función son en los que se puso las iniciales por la rutina del estándar INIFNK, ENDBUF y NLOONLY son puestos en el cero y una coma es colocada en BUFMIN y unos dos puntos en KBFMIN. La dirección del carácter MSX ROM determinado es tomada de 0004H de posiciones y 0005H y colocada en CGPNT + 1 y PRMPRV está dispuesto a apuntar hacia PRMSTK. Los valores falsos son colocados en STKTOP, MEMSIZ y VARTAB (sus valores correctos no son conocidos aún), uno que yo el amortiguador de la / O soy ubicado (7E6BH) y el Z80 SP se sedimentó (62E5H). Un byte de cero es colocado en la base de RAM, TXTTAB está listo para la siguiente posición y un "NEW" ejecutado (6287H).

El VDP es entonces inicializado por el INITIO, INIT32 y rutinas del estándar CLRSPR, las coordenadas del cursor son determinadas para hacer avanzar con el remo 11, columna 10 y el signo en el mensaje "el sistema MSX etcétera" Es exhibido (6678H). Después de un tres segundo retraso una búsqueda es efectuada para cualquier extensión ROMs (7D75H) y un más "NEW" ejecutados (6287H) en caso un programa de BASIC ha sido andado de ROM.

Finalmente el mensaje de la identificación "etc MSX BASIC" Es exhibido (7D29H) y el control transfiere para el 411FH del punto Interpreter Mainloop "Oklahoma".

La dirección ... 7D29H

Esta rutina es usada durante que se energiza arriba para posibilitar el despliegue de la llave de función, colocar en la pantalla 40x24 Text Mode por la rutina del estándar INITXT, y exteriorizar (6678H) el mensaje de la identificación "MSX BASIC etcétera". La cantidad de memoria libre es entonces computada sustrayendo los contenidos de VARTAB de los contenidos de STKTOP y

exhibido (3412H) seguido por los "bytes gratuitamente" el mensaje.

- 204 -

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 7D5DH

Esta rutina es usada durante que se energiza arriba para encontrar la posición mínima de la RAM. Iniciando en EF00H cada byte es probado hasta que uno sea encontrado que escribió o una dirección de 8000H es alcanzada. La dirección bajo, redondeada arriba para lo más cercano 256 la demarcación de byte, es devuelta en par de registro HL.

La dirección ... 7D75H

Esta rutina es usada durante que se energiza arriba para realizar una búsqueda de la extensión ROM. Manda a llamar 1 y 2 (4000H para BFFFH) de cada ranura son examinadas y los resultados colocados en SLTATR. Una extensión ROM tiene los dos personajes de la identificación "AB" en los primeros dos bytes para distinguirlo de RAM. La información acerca de sus propiedades está también presente en los primeros dieciséis bytes como sigue:

```
+
| El 10-15 Reservado De | Byte
| BASIC Text Address MSB | Byte 9
| BASIC Text Address LSB | Byte 8
| DISPOSITIVO Address MSB | Byte 7
| DISPOSITIVO Address LSB | Byte 6
| DECLARACIÓN Address MSB | Byte 5
| DECLARACIÓN Address LSB | Byte 4
| INICIALICE a Address MSB | Byte 3
| INICIALICE a Address LSB | Byte 2
| 42H (' B ') | Byte 1
| 41H (' UNO ') | Byte 0
+
```

Figura 48: ROM Header

Cada página en una ranura dada es examinada leyendo los primeros dos bytes (7E1AH) y revisando en busca del "AB" personajes. Si un ROM está presente que la dirección de inicialización es leída (7E1AH) y el control pasó para ella por la rutina del estándar CALSLT. Con un juego ROM no puede haber regreso para el BASIC de este punto. La "llamada" extendió dirección del manipulador de declaración es entonces leída (7E1AH) y el bit 5 de B de registro colocada si es válida, ese es poco cero. La dirección extendida del manipulador del dispositivo es leída (7E1AH) y el bit 6 de B de registro colocada si es válida. Finalmente la dirección del texto de programa de BASIC es leída (7E1AH) y el bit 7 de B de registro colocada si es válida. La B de registro es entonces copiada para la colocación pertinente en SLTATR y la búsqueda continuó hasta ningún más resto de ranuras.

SLTATR es entonces examinado para cualquier extensión ROM enlosado como contener texto de programa de BASIC. Si uno es encontrado, entonces su colocación en SLTATR es convertida a un Slot Idaho (7E2AH) y el ROM permanentemente cambió de decisión adentro por la rutina del estándar ENASLT. VARTAB está listo para C000H, en su estado actual sabido qué tan grande el Program que Text Area es, TXTTAB es determinado para 8008H y BASROM hizo poco cero para desactivar la llave CTRL-STOP. El sistema es aclarado (629AH) y control se traslada para el Runloop (4601H) para ejecutar el programa de BASIC.

5. EL INTÉRPRETE DE BASIC ROM

La dirección ... 7E1AH

Esta rutina se usa para leer dos bytes de posiciones sucesivas en una extensión ROM. La inicial a la dirección está suministrada en par de registro HL y el Slot Idaho de adentro registra C. Los bytes son leídos por la rutina del estándar RDSLT y regresados en par de registro Delaware. Si ambos son, entonces

s el cero FLAG Z es devuelto.

La dirección ... 7E2AH

Estos conversos de rutina en quienes la colocación SLTATR abasteció registran B en el correspondiente Slot Idaho en C de registro y la dirección de la base ROM en H de registro. La colocación es primero modificado a fin de que echa a andar de 0 para 63 en vez de 64 para 1, a fin de que la información requerida está presente en la forma:

```

7 6 5 4 3 2 1 0
+-----+ | 0 | 0 PSLOT# |SSLOT# |PAGE
+

```

Figura 49

Los añicos 0 y 1 son intercambiados en los añicos dos más altos de H de registro para formar la dirección. Los añicos 4 y 5 son intercambiados para añicos 0 y 1 de C de registro para formar el número Primario Slot. Los añicos 2 y 3 son intercambiados para añicos 2 y 3 de C de registro para formar el número Secundario Slot y mordieron 7 de la entrada correspondiente EXPTBL copiada para bit 7 de C de registro.

La dirección ... 7E4BH

Éste son los "MAXFILES" manipulador de declaración. Como el control se traslada aquí cuando un "MAX" señal (CDH) es detectado que el texto de programa es primero a cuadros para un "FILES" de arrastre señal (B7H). La cuenta del amortiguador operand, de cero para quince, es luego evaluado (521CH) y cualquier amortiguadores existentes cerrado (6C1CH). El número requerido de yo que los amortiguadores de la / O son ubicados (7E6BH), el sistema es aclarado (62A7H) y el control se traslada directamente para el Runloop (4601H).

La dirección ... 7E6BH

Esto es lo yo la rutina de afijación del amortiguador de la / O. Es usado durante que se energiza arriba y por los "MAXFILES" y los manipuladores "cristalinos" de declaración para ubicar almacenamiento para el número de yo / que la O modera suministró en A de registro. Doscientos sesenta y siete bytes son sustraídos de los contenidos de HIMEM para cada amortiguador para producir un valor nuevo MEMSIZ. El tamaño del String existente Storage Area (inicialmente doscientos bytes) es computado sustrayendo los contenidos viejos de STKTOP de los contenidos viejos de MEMSIZ, éste es entonces sustraído del valor nuevo MEMSIZ para producir el valor nuevo STKTOP. Uno más bytes es

5. EL INTÉRPRETE DE BASIC ROM

Sustraída para la pila Z80 y uno "por la memoria" el error generado (6275H) si esta dirección está más abajo del principio del Storage Variable Area. De otra manera la cuenta del amortiguador es colocada en MAXFIL y MEMSIZ y STKTOP determinado para sus valores nuevos. La dirección del remitente de la persona que llama es al que se hizo estallar, el Z80 SP determinado para la colocación nueva y la dirección de regreso empujada hacia atrás encima de la pila.

FILTAB está luego listo al principio de lo que yo el bloque del puntero del amortiguador de la / O y cada puntero a punto para el FCB asociado. Finalmente la dirección de yo el amortiguador de la / O 0, "la CARGA" de Interpreter y "salvo" amortiguador, es colocada en NULBUF y la rutina termina.

La dirección ... 7ED8H

Éste es el mensaje de texto puro "que el sistema MSX" terminó por un byte de cero

La dirección ... 7EE4H

Éste es la "versión" de mensaje de texto puro "1.0" RETORNO DE CARRO, LF terminó por un byte de cero.

La dirección ... 7EF2H

Éste es el mensaje de texto puro "que MSX BASIC" terminó por un byte de cero.

La dirección ... 7EFDH

Éste es el "derecho de autor" de mensaje de texto puro "1983 por Microsoft" RETORNO DE CARRO, LF terminó por un byte de cero.

La dirección ... 7F1BH

Éste es el mensaje de texto puro "que los Bytes liberan" terminado por un byte de cero.

La dirección ... 7F27H

Este bloque de cien bytes de cuarenta y cuatro datos se usa para inicializar el área Workspace de F380H para F40FH.

La dirección ... 7FB7H

Este siete parche de byte apuros un problema no resuelto en el software en el dispositivo externo analizando gramaticalmente rutina (55F8H). Revisa en busca de un nombre del dispositivo de largo de cero en A de registro y lo cambia para uno si es necesario.

La dirección ... 7FBEH

Este capítulo del ROM está sin uso y llenado con bytes de cero.

6. El MAPA de MEMORIA

Un máximo de 32 el KB de RAM está disponible para el BASIC Interpreter para mantener el texto de programa, el BASIC Variables, la pila Z80, lo que yo / la O modero y el área de trabajo interno. Un mapa de memoria de estas áreas en la condición que se impulsa arriba es mostrado debajo:

```

+
| El área de trabajo Area
HIMEM = F380H
+
| Yo El Amortiguador / O 1
+
| FCB 1
+
| Yo El Amortiguador / O 0
NULBUF = F177H
+
| FCB 0
+
| F277H (FCB 1)
+
| F16AH ----- | F16EH (FCB 0)
FILTAB =
+
| El 00H
+
MEMSIZ = F168H
FRETOP = F168H --- + | String Storage Area
|
STKTOP = F0A0H
+
| Z80 Stack
|
+

STREND = 8003H
+
| El área De Almacenamiento De Conjunto Imponente
ARYTAB = 8003H
+
| El área Variable De Almacenamiento
VARTAB = 8003H
+
| El área Del Texto De Programa
TXTTAB = 8001H
+
| El 00H
+

```

Figura 50: El 8000H de memoria Map para FFFFH

El Program Text Area está compuesto de rayado de programa del tokenized almacenado en la orden de número de la línea y terminado por un fin de cero enlace, cuando en la condición "nueva" sólo el enlace de fin está presente. El byte de cero en 8000H es un fin falso de línea que el carácter necesitó sincronizar el Runloop al principio de un programa.

6. El MAPA de MEMORIA

El Variable y Array Storage Areas están compuestos de cuerda o el numérico Variables y Arrays almacenado en la orden en la cual son encontrados en el texto de programa. La velocidad de ejecución se mejora marginalmente si Variables es pronunciado antes de que Arrays en un programa como éste reduzca la cantidad de memoria para estar emocionado arriba.

La pila Z80 es situada inmediatamente debajo del String Storage Area, la estructura de la parte superior de la pila es exteriorizada debajo:

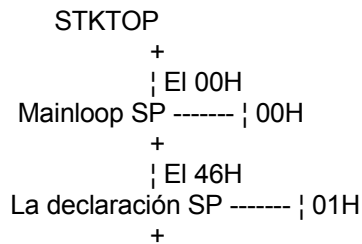
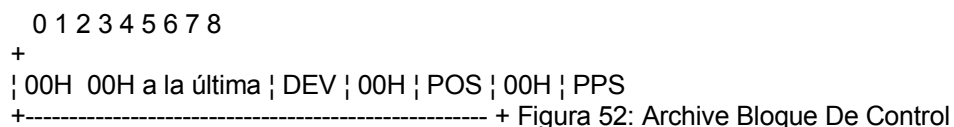


Figura 51: La Parte Superior De La Pila Del Z80

Cuandoquiera que la colocación de la pila está alterada, como resultado de un "claro" o "MAXFILES" bytes de declaración, de dos cero son primero empujado para funcionar como un exterminador durante "para" o "GOSUB" búsquedas del bloque de parámetro. Pretencioso ninguno de los bloques de parámetro están presentes que el Z80 SP por consiguiente estará en casa de STKTOP-2 dentro del Interpreter Mainloop y de STKTOP-4 cuando el control se traslada del Runloop para un manipulador de declaración.

El String Storage Area está compuesto de los cuerpos de la cuerda asignados a Variables o Arrays. Durante la evaluación de expresión un número de resultados intermedios de la cuerda también pueden estar por ahora presentes bajo el montón permanente de la cuerda. El byte de cero siguiendo al String Storage Area es un delimitador temporal para el "VAL" función.

La región entre el String Storage Area e HIMEM es usada para yo almacenamiento del amortiguador de la / O. Yo el amortiguador de la / O 0, el "SAVE" y "carga" moderamos, es siempre presente pero el número de usuario amortiguadores es determinado por los "MAXFILES" declaración. Cada uno que yo el amortiguador de la / O de un 9 byte FCB, quién es aquel cuya dirección está contenida en la mesa bajo FCB 0, seguida por un amortiguador de datos de 256 bytes. El FCB contiene el estado de lo yo el amortiguador de la / O como debajo:



El byte A LA ÚLTIMA sostiene el modo del amortiguador, el byte DEV el código del dispositivo, el byte DE PUNTO DE VENTA la colocación coetánea en el amortiguador (0 para 255) y el byte PPS la colocación "estampada". El resto del FCB es no acostumbrado en una máquina estándar MSX.

6. El MAPA de MEMORIA

El área de trabajo Area

El capítulo del Workspace Area de F380H para FD99H sostiene las variables BIOS / Interpreter. Estos están listados en las siguientes páginas en forma estándar de lenguaje de ensambla:

F380H RDPRIM: FUERA (0A8H), UNO; Coloque a Slot Primario nuevo
F382H LD E, (HL); Lea la memoria
F383H JR WRPRM1; Restaura a viejo Slot Primario

Esta rutina es usada por la rutina del estándar RDSLT para hacer cambiar al Primario Slots y leer un byte de memoria. El Registro Primario nuevo de la Ranura sedimentándose es suministrado en A de registro, la D vieja de registro de establecimiento y el byte leen devueltos en E de registro.

F385H WRPRIM: FUERA (0A8H), UNO; Coloque a Slot Primario nuevo
F387H LD (HL), E; Escriba para la memoria
F388H WRPRM1: LD A, D; Envejezca sedimentándose
F389H OUT (0A8H), UNO; Restaura a viejo Slot Primario
F38BH RET

Esta rutina es usada por la rutina del estándar WRSLT para hacer cambiar al Primario Slots y escribir un byte para la memoria. El Registro Primario nuevo de la Ranura sedimentándose le es suministrado en A de registro, el trasfondo viejo en D de registro y el byte para escribir en E de registro.

F38CH CLPRIM: FUERA (0A8H), UNO; Coloque a Slot Primario nuevo
F38EH EX AF AF, AF '; Haga un intercambio para AF para la llamada
F38FH LLAME CLPRM1; Hágalo
F392H EX AF AF, AF '; Haga un intercambio para AF
F393H HAGA ESTALLAR a AF; Envejezca sedimentándose
F394H OUT (0A8H), UNO; Restaura a viejo Slot Primario
F396H EX AF AF, AF '; Haga un intercambio para AF
F397H RET
F398H CLPRM1: JP (IX)

Esta rutina es usada por la rutina del estándar CALSLT para hacer cambiar al Primario Slots y llamar una dirección. La Ranura Primaria nueva Registro sedimentándose es abastecido en A de registro, el trasfondo viejo en la pila Z80 y la dirección para hacer venir registro parean a IX.

F39AH USRTAB: DEFW 475AH; USR 0
F39CH DEFW 475AH; USR 1
F39EH DEFW 475AH; USR 2
F3A0H DEFW 475AH; USR 3
F3A2H DEFW 475AH; USR 4
F3A4H DEFW 475AH; USR 5
F3A6H DEFW 475AH; USR 6
F3A8H DEFW 475AH; USR 7
F3AAH DEFW 475AH; USR 8
F3ACH DEFW 475AH; USR 9

Estas diez variables contienen al "USR" direcciones de función. Sus valores son determinados para el Interpreter "la Llamada ilegal de función" el error

6. El MAPA de MEMORIA

El generador en.

F3AEH LINL40: DEFB 37

Esta variable contiene la anchura de la pantalla del 40x24 Text Mode. Su valor acometió contra que se energiza arriba y después sólo alterado por la "anchura" declaración.

F3AFH LINL32: DEFB 29

Esta variable contiene la anchura de la pantalla del 32x24 Text Mode. Su valor acometió contra que se energiza arriba y después sólo alterado por la "anchura" declaración.

F3B0H LINLEN: DEFB 37

Esta variable contiene la anchura coetánea de la pantalla de modo del texto. Su valor es determinado de LINL40 o LINL32 cuandoquiera que el VDP es inicializado para un modo del texto por el INITXT o las rutinas estándar INIT32.

F3B1H CRTCNT: DEFB 24

Esta variable contiene el número de filas en la pantalla. Su valor acometió energizar y después inalterado.

F3B2H CLMLST: DEFB 14

Esta variable contiene el número mínimo de columnas que todavía debe estar disponible en una línea que unos datos ítem sean " PRINT"ed, si menos espacio está disponible un RETORNO DE CARRO, LF es expedido primera parte. Su valor acometió contra que se energiza arriba y después sólo alterado por la "anchura" y "pantalla" declaraciones.

F3B3H TXTNAM: DEFW 0000H; La Base De La Mesa De Nombre

F3B5H TXTCOL: DEFW 0000H; La Base De La Mesa De Color

F3B7H TXTCGP: DEFW 0800H; La Base Del Patrón De Carácter

F3B9H TXTATR: DEFW 0000H; La Base De Atributo De Bloque De Objetos Movibles

F3BBH TXTPAT: DEFW 0000H; La Base Del Patrón De Bloque De Objetos Movibles

Estas cinco variables contienen las direcciones de la base del 40x24 Text Mode VDP. Sus valores acometió contra que se energiza arriba y después sólo alterado por la declaración "bajo".

F3BDH T32NAM: DEFW 1800H; La Base De La Mesa De Nombre

F3BFH T32COL: DEFW 2000H; La Base De La Mesa De Color

F3C1H T32CGP: DEFW 0000H; La Base Del Patrón De Carácter

F3C3H T32ATR: DEFW 1B00H; La Base De Atributo De Bloque De Objetos Movibles

F3C5H T32PAT: DEFW 3800H; La Base Del Patrón De Bloque De Objetos Movibles

Estas cinco variables contienen las direcciones de la base del 32x24 Text Mode VDP. Sus valores acometió contra que se energiza arriba y después sólo alterado por la declaración "bajo".

6. EL MAPA de MEMORIA

F3C7H GRPNAM: DEFW 1800H; La Base De La Mesa De Nombre
F3C9H GRPCOL: DEFW 2000H; La Base De La Mesa De Color
F3CBH GRPCGP: DEFW 0000H; La Base Del Patrón De Carácter
F3CDH GRPATR: DEFW 1B00H; La Base De Atributo De Bloque De Objetos Movibles
F3CFH GRPPAT: DEFW 3800H; La Base Del Patrón De Bloque De Objetos Movibles

Estas cinco variables contienen las direcciones de la base Graphics Mode VDP. Sus valores acometió contra que se energiza arriba y después sólo alterado por la declaración "bajo".

F3D1H MLTNAM: DEFW 0800H; La Base De La Mesa De Nombre
F3D3H MLTCOL: DEFW 0000H; La Base De La Mesa De Color
F3D5H MLTCGP: DEFW 0000H; La Base Del Patrón De Carácter
F3D7H MLTATR: DEFW 1B00H; La Base De Atributo De Bloque De Objetos Movibles
F3D9H MLTPAT: DEFW 3800H; La Base Del Patrón De Bloque De Objetos Movibles

Estas cinco variables contienen las direcciones de la base Multicolour Mode VDP. Sus valores acometió contra que se energiza arriba y después sólo alterado por la declaración "bajo".

F3DBH CLIKSW: DEFB 01H

Esta variable controla lo interrumpe chasquido de la llave del manipulador:
El 00H = Off, NZ = On. que Su valor es contra el que se acometió.

F3DCH CSRY: DEFB 01H

Esta variable contiene la fila coordinada (de 1 para CTCNT) del cursor de modo del texto.

F3DDH CSRX: DEFB 01H

Esta variable contiene la columna coordinada (de 1 para LINLEN) del cursor de modo del texto. Noto que las coordenadas del cursor BIOS para la casa sitúan son 1,1 cualquier cosa la anchura de la pantalla.

F3DEH CNSDFG: DEFB FFH

Esta variable contiene la condición coetánea del despliegue de la llave de función: El 00H = Off, NZ = On.

F3DFH RG0SAV: DEFB 00H
F3E0H RG1SAV: DEFB F0H
F3E1H RG2SAV: DEFB 00H
F3E2H RG3SAV: DEFB 00H
F3E3H RG4SAV: DEFB 01H
F3E4H RG5SAV: DEFB 00H
F3E5H RG6SAV: DEFB 00H
F3E6H RG7SAV: DEFB F4H

Estas ocho variables imitan la condición del ocho VDP que se escribe sólo Mode Registers. Los valores exteriorizados son para 40x24 Text Mode.

6. El MAPA de MEMORIA

F3E7H STATFL: DEFB CAH

Esta variable está continuamente actualizada por ahí lo interrumpe manipulador con los contenidos del VDP Status Register.

F3E8H TRGFLG: DEFB F1H

Esta variable está continuamente actualizada por ahí lo interrumpe manipulador con la condición de las cuatro entradas del gatillo de la palanca de control y la llave de espacio.

F3E9H FORCLR: DEFB 0FH; El blanco

Esta variable contiene el color actual del primer plano. Su valor acometió contra que se energiza arriba y después sólo alterado por el "color" declaración. El color del primer plano es usado por la rutina del estándar CLRSPR para colocar el color de bloque de objetos movibles y por la rutina del estándar CHGCLR para incrustar al color 1 pixel en los modos del texto. También funciona como los gráficos color de tinta en su estado actual copiado para ATRBYT por la rutina del estándar GRPPRT y usado a todo lo largo del Interpreter como el valor predeterminado para cualquier operand optativo de color.

F3EAH BAKCLR: DEFB 04H; Azul profundo

Esta variable contiene el color de fondo actual. Su valor acometió contra que se energiza arriba y después sólo alterado por el "color" declaración. El color de fondo es usado por la rutina del estándar de Lingüísticas Computacionales para despejar la pantalla en los modos de gráficos y por la rutina del estándar CHGCLR a incrustar al color 0 pixel en los modos del texto.

F3EBH BDRCLR: DEFB 04H; Azul profundo

Esta variable contiene el color fronterizo actual. Su valor acometió contra que se energiza arriba y después sólo alterado por el "color" declaración. Mientras color fronterizo es usado por la rutina del estándar CHGCLR en 32x24 Text Mode, Graphics Mode y Mode Multicolor para se sedimentó color fronterizo.

F3ECH MAXUPD: DEFB C3H
F3EDH DEFW 0000H

Estos dos bytes son suplidos por la "línea" manipulador de declaración para forjar a un Z80 JP para el RIGHTC, LEFTC, UPC o rutinas del estándar DOWNC.

F3EFH MINUPD: DEFB C3H
F3F0H DEFW 0000H

Estos dos bytes son suplidos por la "línea" manipulador de declaración para forjar a un Z80 JP para el RIGHTC, LEFTC, UPC o rutinas del estándar DOWNC.

6. El MAPA de MEMORIA

F3F2H ATRBYT: DEFB 0FH

Esta variable contiene el color de tinta de gráficos usado por el SETC y rutinas del estándar NSETCX.

F3F3H HACE COLA: DEFW F959H

Esta variable contiene la dirección de los bloques de control para las tres colas de música. Su valor acometió energizar y después inalterado.

F3F5H FRCNEW: DEFB FFH

¿Esta variable contiene una bandera para distinguir las dos declaraciones en el "CLOAD/CLOAD"? El manipulador de declaración: ¿00H = CLOAD, FFH=CLOAD?.

F3F6H SCNCNT: DEFB 01H

Esta variable es utilizada como un mueble mostrador por lo interrumpe manipulador para monitorear la tasa en la cual el teclado escudriña es realizado.

F3F7H REPCNT: DEFB 01H

Esta variable es utilizada como un mueble mostrador por lo interrumpe manipulador para monitorear la llave repite tasa.

F3F8H PUTPNT: DEFW FBF0H

Esta variable contiene la dirección de la colocación puesta en KEYBUF.

F3FAH GETPNT: DEFW FBF0H

Esta variable contiene la dirección de lo obtiene colocación en KEYBUF.

F3FCH CS1200: DEFB 53H; LO recicle mitad 1
F3FDH DEFB 5CH; LO recicle mitad 2
F3FEH DEFB 26H; El ciclo de Hawaii la mitad 1
F3FFH DEFB 2DH; El ciclo de Hawaii la mitad 2
F400H DEFB 0FH; La cuenta de ciclo del encabezado

Estas cinco variables contienen los 1200 parámetros del casete de baudio. Sus valores acometió energizar y después inalterado.

F401H CS2400: DEFB 25H; LO recicle mitad 1
F402H DEFB 2DH; LO recicle mitad 2
F403H DEFB 0EH; El ciclo de Hawaii la mitad 1
F404H DEFB 16H; El ciclo de Hawaii la mitad 2
F405H DEFB 1FH; La cuenta de ciclo del encabezado

Estas cinco variables contienen los 2400 parámetros del casete de baudio. Sus valores acometió energizar y después inalterado.

6. El MAPA de MEMORIA

F406H MUJA: DEFB 53H; LO recicle mitad 1
F407H DEFB 5CH; LO recicle mitad 2
El ALTO F408H: DEFB 26H; El ciclo de Hawaii la mitad 1
F409H DEFB 2DH; El ciclo de Hawaii la mitad 2
El ENCABEZADO F40AH: DEFB 0FH; La cuenta de ciclo del encabezado

Estas cinco variables contienen los parámetros coetáneos del casete.
Sus valores son determinados para 1200 el baudio en energizar y después sólo alterado por el "CSAVE" y "ocultan" declaraciones.

F40BH ASPCT1: DEFW 0100H

Esta variable contiene lo recíproco del "círculo" predeterminado proporción dimensional multiplicado por 256. Su valor acometió energizar y después inalterado.

F40DH ASPCT2: DEFW 01C0H

Esta variable contiene el "círculo" predeterminado proporción dimensional multiplicado por 256. Su valor acometió energizar y después inalterado. La proporción dimensional está presente en dos formas a fin de que el "círculo" manipulador de declaración puede seleccionar a lo apropiado inmediatamente en vez de necesitar examinar y posiblemente recíprocar él según es la caja con un operand en el texto de programa.

F40FH ENDPRG: DEFB ""
F410H DEFB 00H
F411H DEFB 00H
FE12H DEFB 00H
F413H DEFB 00H

Estos cinco bytes forman una línea falsa de programa. Sus valores acometió energizar y después inalterado. La línea existe en caso un error ocurre en el Interpreter Mainloop antes de que cualquier texto del tokenized esté disponible en KBUF. Si uno "en ERROR GOTO" es activo a esta hora luego provee algún texto para el "curriculum vitae" declaración para terminar adelante.

F414H ERRFLG: DEFB 00H

Esta variable es usada por el manipulador de error Interpreter para ahorrar el número de error.

F415H LPTPOS: DEFB 00H

Esta variable es usada por el "LPRINT" manipulador de declaración para sostener la colocación coetánea de la cabeza de la impresora.

F416H PRTFLG: DEFB 00H

Esta variable determina si la rutina del estándar OUTDO dirige su salida para la pantalla o para la impresora: El 00H = Screen, 01H Printer.

F417H NTMSXP: DEFB 00H

Esta variable determina si la rutina del estándar OUTDO reemplazará headered que los personajes de gráficos dirigieron para la impresora con espacios: El 00H = Graphics, NZ = Spaces. Su valor acometió contra que se energiza arriba y después sólo alterado por la "pantalla" declaración.

F418H RAWPRT: DEFB 00H

Esta variable determina si la rutina del estándar OUTDO modificará control y headered que los personajes de gráficos dirigieron para la impresora: El 00H = Modify, NZ = Raw. Su valor acometió energizar y después inalterado.

F419H VLZADR: DEFW 0000H

F41BH VLZDAT: DEFB 00H

Estas variables contienen la dirección y valor de cualquier carácter por ahora removido por el "VAL" función.

F41CH CURLIN: DEFW FFFFH

Esta variable contiene el número actual de la línea Interpreter. Un valor de FFFFH denota modo directo.

F41EH KBFMIN: DEFB ""

Este byte provee un prefijo falso para el texto del tokenized contenido en KBUF. Su función es similar a que de ENDPRG sino sirve para la situación donde un error ocurre dentro de una declaración directa.

F41FH KBUF: DEFS 318

Este amortiguador contiene la forma del tokenized de la línea de entrada cobrada por el Interpreter Mainloop. Cuando una declaración directa es ejecutada los contenidos de esta forma del amortiguador el texto de programa.

F55DH BUFMIN: DEFB

Este byte provee un prefijo falso para el texto contenido en BUF. Se usa para sincronizar la "entrada" manipulador de declaración como comienza a analizar el texto de entrada.

F55EH BUF: DEFS 259

Este amortiguador contiene el texto cobrado de la consola por la rutina del estándar INLIN.

F661H TTYPOS: DEFB 00H

¡Esta variable es usada por el manipulador "estampado" de declaración para sostener la colocación coetánea de la pantalla (el Teletipo!).

F662H DIMFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada por el manipulador "débil" de declaración para controlar la operación de la rutina variable de búsqueda.

F663H VALTYP: DEFB 02H

Esta variable contiene el código de tipo del operand actualmente contenido en DAC: El entero, 3 = String, 4 = el Solo Precision, 8 = el Doble Precision.

F664H DORES: DEFB 00H

Esta variable es normalmente cero pero es determinada para impedir la tokenización de palabras claves no citadas siguiendo a una "DATA" valor simbólico.

F665H DONUM: DEFB 00H

Esta variable es normalmente cero pero es determinada cuando una constante numérica sigue uno del palabra clave GOTO, GOSUB, LUEGO, etc., Y debe ser tokenized para la forma especial del operand de número de la línea.

F666H CONTXT: DEFW 0000H

Esta variable es usada por la rutina del estándar CHRGTTR para ahorrar la dirección del carácter siguiendo una constante numérica en el texto de programa.

F668H CONSAV: DEFB 00H

Esta variable es usada por la rutina del estándar CHRGTTR para ahorrar la señal de una constante numérica encontrada en el texto de programa.

F669H CONTYP: DEFB 00H

Esta variable es usada por la rutina del estándar CHRGTTR para ahorrar el tipo de una constante numérica encontrada en el texto de programa.

F66AH CONLO: DEFS 8

Este amortiguador es usado por la rutina del estándar CHRGTTR para ahorrar el valor de una constante numérica encontrada en el texto de programa.

F672H MEMSIZ: DEFW F168H

Esta variable contiene la dirección de la parte superior del String Storage Area. Su valor acometió contra que se energiza arriba y después sólo alterado por el "claro" y "MAXFILES" declaraciones.

F674H STKTOP: DEFW F0A0H

Esta variable contiene la dirección de la parte superior de la pila Z80. Su valor acometió contra que se energiza arriba para MEMSIZ-200 y después sólo

Alterado por el "claro" y "MAXFILES" declaraciones.

F676H TXTTAB: DEFW 8001H

Esta variable contiene la dirección del primer byte del Program Text Area. Su valor acometió energizar y después inalterado.

F678H TEMPPT: DEFW F67AH

Esta variable contiene la dirección de la siguiente posición libre en TEMPST.

F67AH TEMPST: DEFS 30

Este amortiguador se usa para almacenar descriptores de la cuerda. Funciona como una pila con productores de la cuerda empujando sus resultados y los consumidores de la cuerda haciéndoles estallar.

F698H DSCTMP: DEFS 3

Este amortiguador es usado por las funciones de la cuerda a mantener un descriptor de resultado mientras está siendo construido.

F69BH FRETOP: DEFW F168H

Esta variable contiene la dirección de la siguiente posición libre en el String Storage Area. Cuándo el área es FRETOP vacío es igual a MEMSIZ.

F69DH TEMP3: DEFW 0000H

Esta variable sirve para almacenamiento temporal por partes diversas del Interpreter.

F69FH TEMP8: DEFW 0000H

Esta variable sirve para almacenamiento temporal por partes diversas del Interpreter.

F6A1H ENDFOR: DEFW 0000H

Esta variable es usada por el "FOR" manipulador de declaración para sostener el fin de dirección de declaración durante construcción de un bloque de parámetro.

F6A3H DATLIN: DEFW 0000H

Esta variable contiene el número de la línea de los "datos" coetáneos ítem en el texto de programa.

F6A5H SUBFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada por el "ERASE"

- 218 -

6. El MAPA de MEMORIA

"Para", "FN" y "DEF FN" manipuladores para controlar el procesamiento de subíndices por la variable investigan rutina.

F6A6H FLGINP: DEFB 00H

Esta variable contiene una bandera para distinguir las dos declaraciones en la "lectura /entrada" manipulador de declaración: = la ENTRADA del 00H, NZ=READ.

F6A7H TEMP: DEFW 0000H

Esta variable sirve para almacenamiento temporal por partes diversas del Interpreter.

F6A9H PTRFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada si cualquier operands de número de la línea en el Program Text Area han sido convertidos a punteros.

F6AAH AUTFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada cuando "el AUTOMÓVIL" el modo se enciende.

F6ABH AUTLIN: DEFW 0000H

Esta variable contiene el "automóvil" en uso en estos momentos número de la línea.

F6ADH AUTINC: DEFW 0000H

Esta variable contiene el "automóvil" en uso en estos momentos incremento de número de la línea.

F6AFH SAVTXT: DEFW 0000H

Esta variable está actualizada por el Runloop al principio de cada declaración con la posición en uso en estos momentos en el texto de programa. Es usado durante la recuperación de error para colocar a ERRTXT para el "curriculum vitae" manipulador de declaración y OLDTXT para el "CONT" manipulador de declaración.

F6B1H SAVSTK: DEFW F09EH

Esta variable está actualizada por el Runloop al principio de cada declaración con el Z80 actual SP para el error que la recuperación tiene en mente.

F6B3H ERRLIN: DEFW 0000H

Esta variable es usada por el manipulador de error para mantener el número de la línea de la línea de programa generando un error.

F6B5H PUNTEE: DEFW 0000H

Esta variable está actualizada por el Mainloop y el manipulador de error con el número actual de la línea para el uso con lo "" El parámetro.

F6B7H ERRTXT: DEFW 0000H

Esta variable está actualizada de SAVTXT por el manipulador de error para el uso por el "curriculum vitae" manipulador de declaración.

F6B9H ONELIN: DEFW 0000H

Esta variable está lista por ahí lo "en ERROR GOTO" manipulador de declaración con la dirección de la línea de programa a ejecutar cuando un error ocurre.

F6BBH ONEFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada por el manipulador de error cuando el control se traslada para uno "en ERROR GOTO" declaración. Esto debe impedir un lazo desarrollándose si un error ocurre dentro de las declaraciones de recuperación de error.

F6BCH TEMP2: DEFW 0000H

Esta variable sirve para almacenamiento temporal por partes diversas del Interpreter.

F6BEH OLDLIN: DEFW 0000H

Esta variable contiene el número de la línea de la línea de programa que termina. Está colocado por el "fin" y "el ALTO" manipuladores de declaración para el uso con el "CONT" declaración.

F6C0H OLDTXT: DEFW 0000H

Esta variable contiene la dirección de la declaración de programa que termina.

F6C2H VARTAB: DEFW 8003H

Esta variable contiene la dirección del primer byte del Storage Variable Area.

F6C4H ARYTAB: DEFW 8003H

Esta variable contiene la dirección del primer byte del Array Storage Area.

F6C6H STREND: DEFW 8003H

Esta variable contiene la dirección del byte siguiendo el área Array Storage.

F6C8H DATPTR: DEFW 8000H

Esta variable contiene la dirección de los "datos" coetáneos ítem en el texto de programa.

- 220 -

6. El MAPA de MEMORIA

F6CAH DEFTBL: DEFB 08H; uno

F6CBH DEFB 08H; B
F6CCH DEFB 08H; C
F6CDH DEFB 08H; D
F6CEH DEFB 08H; E
F6CFH DEFB 08H; F
F6D0H DEFB 08H; G
F6D1H DEFB 08H; H
F6D2H DEFB 08H; yo
F6D3H DEFB 08H; J
F6D4H DEFB 08H; K
F6D5H DEFB 08H; L
F6D6H DEFB 08H; M
F6D7H DEFB 08H; N
F6D8H DEFB 08H; O
F6D9H DEFB 08H; P
F6DAH DEFB 08H; Q
F6DBH DEFB 08H; interrogación
F6DCH DEFB 08H; S
F6DDH DEFB 08H; T
F6DEH DEFB 08H; U
F6DFH DEFB 08H; V
F6E0H DEFB 08H; W
F6E1H DEFB 08H; X
F6E2H DEFB 08H; Y
F6E3H DEFB 08H; Z

Estas veintiseis variables contienen el tipo predeterminado para cada grupo de BASIC Variables. Sus valores son determinados para duplicar precisión en "NEW" que se energiza arriba, y "despejarse" y después alterados sólo por el "DEF" grupo de declaraciones.

F6E4H PRMSTK: DEFW 0000H

Esta variable contiene la dirección bajo del "FN" previo bloque de parámetro en la pila Z80. Es usado durante colección de basura de la cuerda para hacer de bloque para bloquear en la pila.

F6E6H PRMLN: DEFW 0000H

Esta variable contiene el largo del "FN" actual bloque de parámetro en PARM1.

F6E8H PARM1: DEFS 100

Este amortiguador contiene las Variables locales formando parte del "FN" función actualmente siendo evaluado.

F74CH PRMPRV: DEFW F6E4H

Esta variable contiene la dirección del "FN" previo bloque de parámetro. Es de hecho una constante usada asegurar que la colección de basura de la cuerda empieza con el bloque de parámetro en uso en estos momentos antes de proceder para esos en la pila.

- 221 -

6. El MAPA de MEMORIA

F74EH PRMLN2: DEFW 0000H

Esta variable contiene el largo del bloque de parámetro "FN" forjándose en PARM2

F750H PARM2: DEFS 100

Este amortiguador se usa para construir al local Variables poseído por el "FN" actual función.

F7B4H PRMFLG: DEFB 00H

Esta variable es usada durante una búsqueda Variable para indicar si Variables local o global está siendo examinado.

F7B5H ARYTA2: DEFW 0000H

Esta variable es usada durante una búsqueda Variable para mantener la dirección de terminación del área de almacenamiento siendo examinada.

F7B7H NOFUNS: DEFB 00H

Esta variable es normalmente cero pero es determinada por el "FN" manipulador de función para indicar para la rutina variable de búsqueda que las Variables locales están presentes.

F7B8H TEMP9: DEFW 0000H

Esta variable sirve para almacenamiento temporal por partes diversas del Interpreter.

F7BAH FUNACT: DEFW 0000H

Esta variable contiene el número de actualmente activo " FN funciona.

F7BCH SWPTMP: DEFS 8

Este amortiguador se usa para mantener el primer operand en un "trueque" declaración.

F7C4H TRCFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada por el "TRON" manipulador de declaración para encender la instalación de la huella.

F7C5H FBUFFR: DEFS 43

Este amortiguador se usa para mantener el texto producido durante la conversión numérica de salida.

F7F0H DECTMP: DEFW 0000H

Esta variable sirve para almacenamiento temporal por el doble

- 222 -

6. El MAPA de MEMORIA

La rutina de división de precisión.

F7F2H DECTM2: DEFW 0000H

Esta variable sirve para almacenamiento temporal por la rutina doble de división de precisión.

F7F4H DECCNT: DEFB 00H

Esta variable es usada por la rutina doble de división de precisión para mantener el número de bytes de poco cero en el mantissa del segundo operand.

F7F6H DAC: DEFS 16

Este amortiguador funciona como el acumulador primario de Interpreter durante la evaluación de expresión.

F806H HOLD8: DEFS 65

Este amortiguador es usado por la rutina doble de multiplicación de precisión para sostener los múltiplos del primer operand.

F847H ARG: DEFS 16

Este amortiguador funciona como el acumulador secundario de Interpreter durante la evaluación de expresión.

F857H RNDX: DEFS 8

Este amortiguador contiene la precisión doble coetánea al azar número.

F85FH MAXFIL: DEFB 01H

Esta variable contiene el número de usuario actualmente ubicado que yo / la O moderador. Su valor es determinado para 1 en energizar y después sólo alterado por los "MAXFILES" declaración.

F860H FILTAB: DEFW F16AH

Esta variable contiene la dirección de la mesa del puntero para lo yo los FCBs del amortiguador de la / O.

F862H NULBUF: DEFW F177H

Esta variable contiene la dirección del primer byte del amortiguador de datos perteneciendo para yo el amortiguador de la / O 0.

F864H PTRFIL: DEFW 0000H

Esta variable contiene que la dirección de lo actualmente activo yo / la O moderador FCB.

- 223 -

6. El MAPA de MEMORIA

F866H FILNAM: DEFS 11

Este amortiguador mantiene un filename especificado por usuario. Son once personajes largos

para tener en cuenta lentes del archivo del disco como "FILENAME.BAS".

F871H FILNM2: DEFS 11

Este amortiguador tiene aplicación que un filename leyó de uno yo el dispositivo de la / O para la comparación con los contenidos de FILNAM.

F87CH NLONLY: DEFB 00H

Esta variable es normalmente cero pero es determinada durante un programa "carga". El bit 0 se usa para impedir yo amortiguador de la / O 0 estar cerrado durante cargar y el bit 7 para advertir el usuario que yo ser de amortiguadores de la / O cerré la plana si andado en automóvil es requerido.

F87DH SAVEND: DEFW 0000H

Esta variable es usada por el "BSAVE" manipulador de declaración para mantener la dirección de fin del bloque de memoria a ser ahorrada.

F87FH FNKSTR: DEFS 160

Este amortiguador contiene lo diez instrumentos de cuerda de la llave de función de dieciséis caracteres. Sus valores acometió contra que se energiza arriba y después sólo alterado por la declaración "crucial".

F91FH CGPNT: DEFB 00H; La ranura Idaho
F920H DEFW 1BBFH; La dirección

Estas variables contienen la posición del set de carácter copiado para el VDP por el INITXT y rutinas estándar INIT32. Sus valores son determinados para el conjunto de caracteres MSX ROM en energizar y después inalterado.

F922H NAMBAS: DEFW 0000H

Esta variable contiene la dirección de la base de modo del texto VDP Name Table en uso en estos momentos. Su valor es determinado de TXTNAM o T32NAM cuandoquiera que el VDP es inicializado para un modo del texto por el INITXT o las rutinas estándar INIT32.

F924H CGPBAS: DEFW 0800H

Esta variable contiene la dirección de la base de modo del texto VDP Character Pattern Table en uso en estos momentos. Su valor es determinado de TXTCGP o T32CGP cuandoquiera que el VDP es inicializado para un modo del texto por el INITXT o las rutinas estándar INIT32.

F926H PATBAS: DEFW 3800H

Esta variable contiene la Mesa VDP Sprite Pattern en uso en estos momentos

- 224 -

6. El MAPA de MEMORIA

Base dirección. Su valor es determinado de T32PAT, GRPPAT o MLTPAT cuandoquiera que el VDP es inicializado por el INIT32, INIGRP o rutinas del estándar INIMLT.

F928H ATRBAS: DEFW 1B00H

Esta variable contiene la dirección de la base VDP Sprite Attribute Table en uso en estos momentos. Su valor es determinado de T32ATR, GRPATR o MLTATR cuandoquiera que el VDP es inicializado por el INIT32, INIGRP o rutinas del estándar INIMLT.

F92AH CLOC: DEFW 0000H; La posición Pixel
F92CH CMASK: DEFB 80H; Pixel Mask

Estas variables contienen la dirección física del pixel en uso en estos momentos usada por el RIGHTC, LEFTC, UPC, TUPC, DOWNC, TDOWNC, FETCHC, STOREC, READC, SETC, NSETCX, SCANR y rutinas del estándar SCANL. CLOC mantiene la dirección del byte conteniendo el pixel en uso en estos momentos y CMASK define el pixel dentro de ese byte.

F92DH MINDEL: DEFW 0000H

Esta variable es usada por la "línea" manipulador de declaración para sostener la diferencia mínima entre las proposiciones de fin de la línea.

F92FH MAXDEL: DEFW 0000H

Esta variable es usada por la "línea" manipulador de declaración para sostener la diferencia máxima entre las proposiciones de fin de la línea.

EI ASPECTO F931H: DEFW 0000H

Esta variable es usada por el "círculo" manipulador de declaración para sostener la proporción dimensional coetánea. Esto es almacenado como un solo byte fragmento binario así es que una proporción dimensional de 0.75 se convertiría en 00C0H. El MSB es sólo requerido si la proporción dimensional es exactamente 1.00, ese es 0100H.

F933H CENCNT: DEFW 0000H

Esta variable es usada por el "círculo" manipulador de declaración para mantener la cuenta del punto del ángulo de fin.

F935H CLINEF: DEFB 00H

Esta variable es usada por el "círculo" manipulador de declaración para mantener las dos banderas de la línea. El bit 0 está colocado si una línea es requerida del ángulo de principio para el centro y el set 7 mordido si uno es requerido del ángulo de fin.

F936H CNPNTS: DEFW 0000H

Esta variable es usada por el "círculo" manipulador de declaración para mantener el número de proposiciones dentro de un segmento de cuarenta y cinco grados.

- 225 -

6. El MAPA de MEMORIA

F938H CPLOTF: DEFB 00H

Esta variable es normalmente cero pero es determinada por el "círculo" manipulador de

declaración si el ángulo de fin es más pequeño que el ángulo de principio. Se usa para determinar si los pixels deberían estar colocados "dentro de" los ángulos o "fuera de" ellos.

F939H CPCNT: DEFW 0000H

Esta variable es usada por el "CÍRCULO" manipulador de declaración para mantener la cuenta del punto dentro de la corriente cuarenta y cinco segmento de grado, ésta está de hecho la coordenada de la Y.

F93BH CPCNT8: DEFW 0000H

Esta variable es usada por el "círculo" manipulador de declaración para mantener la cuenta total del punto de la colocación presente.

F93DH CRCSUM: DEFW 0000H

Esta variable es usada por el "círculo" manipulador de declaración como el mueble mostrador de computación del punto.

F93FH CSTCNT: DEFW 0000H

Esta variable es usada por el "círculo" manipulador de declaración para mantener la cuenta del punto del ángulo de principio.

F941H CSCLXY: DEFB 00H

Esta variable es usada por el "círculo" manipulador de declaración como una bandera para determinar en cuál dirección el tropel elíptico debe ser aplicado: 00H = Y, 01H = X.

F942H CSAVEA: DEFW 0000H

Esta variable sirve para almacenamiento temporal por la rutina del estándar SCANR.

F944H CSAVEM: DEFB 00h

Esta variable sirve para almacenamiento temporal por la rutina del estándar SCANR.

F945H CXOFF: DEFW 0000H

Esta variable sirve para almacenamiento temporal por el "círculo" manipulador de declaración.

F947H CYOFF: DEFW 0000H

Esta variable sirve para almacenamiento temporal por el "círculo" manipulador de declaración.

- 226 -

6. El MAPA de MEMORIA

F949H LOHMSK: DEFB 00H

Esta variable es usada por la "pintura" manipulador de declaración para sostener la colocación del leftmost de una excursión LH.

F94AH LOHDIR: DEFB 00H

Esta variable es usada por la "pintura" manipulador de declaración para mantener la dirección de pintura nueva requerida por una excursión LH.

F94BH LOHADR: DEFW 0000H

Esta variable es usada por la "pintura" manipulador de declaración para sostener la colocación del leftmost de una excursión LH.

F94DH LOHCNT: DEFW 0000H

Esta variable es usada por la "pintura" manipulador de declaración para sostener el tamaño de una excursión LH.

F94FH SKPCNT: DEFW 0000H

Esta variable es usada por la "pintura" manipulador de declaración para mantener la cuenta de salto devuelta por la rutina del estándar SCANR.

F951H MOVCNT: DEFW 0000H

Esta variable es usada por la "pintura" manipulador de declaración para mantener la cuenta de movimiento devuelta por la rutina del estándar SCANR.

F953H PDIREC: DEFB 00H

Esta variable es usada por la "pintura" manipulador de declaración para sostener la dirección coetánea de pintura: El 40H = Down, C0H Up, 00H Terminate.

F954H LFPROG: DEFB 00H

Esta variable es normalmente cero pero es determinada por la "pintura" manipulador de declaración si ha habido cualquier hacia la izquierda progresiva.

F955H RTPROG: DEFB 00H

Esta variable es normalmente cero pero es determinada por la "pintura" manipulador de declaración si ha habido cualquier progreso de rightwards.

F956H MCLTAB: DEFW 0000H

Esta variable contiene la dirección de la mesa de orden para ser usada por el analizador gramatical de lenguaje de macro. El "empate" mesa está en 5D83H y el "juego" mesa en 752EH.

F958H MCLFLG: DEFB 00H

Esta variable es cero si el analizador gramatical de lenguaje de macro es

- 227 -

6. El MAPA de MEMORIA

Usados por el "empate", el manipulador de declaración y el poco cero si estuviera siendo usado "juegan".

F959H QUETAB: DEFB 00H; AQ Ponga colocación
F95AH DEFB 00H; AQ Obtenga colocación
F95BH DEFB 00H; La bandera AQ Putback
F95CH DEFB 7FH; AQ Size
F95DH DEFW F975H; AQ Address

F95FH DEFB 00H; BQ Ponga colocación
F960H DEFB 00H; BQ Obtenga colocación
F961H DEFB 00H; La bandera BQ Putback
F962H DEFB 7FH; BQ Size
F963H DEFW F9F5H; BQ Address

F965H DEFB 00H; CQ Ponga colocación
F966H DEFB 00H; CQ Obtenga colocación
F967H DEFB 00H; La bandera CQ Putback
F968H DEFB 7FH; CQ Size
F969H DEFW FA75H; CQ Address

F96BH DEFB 00H; RQ Ponga colocación
F96CH DEFB 00H; RQ Obtenga colocación
F96DH DEFB 00H; La bandera RQ Putback
F96EH DEFB 00H; RQ Size
F96FH DEFW 0000H; RQ Address

Estas veinticuatro variables forman los bloques de control para las tres colas de música (VOICAQ, VOICBQ y VOICCQ) y la cola del RS232. Los tres bloques de control de música son en los que se puso las iniciales por la rutina del estándar GICINI y después mantenidos por ahí lo interrumpen manipulador y la rutina del estándar PUTQ. El bloque de control del RS232 está sin uso en el MSX ROM actual.

F971H QUEBAK: DEFB 00H; El carácter AQ Putback
F972H DEFB 00H; El carácter BQ Putback
F973H DEFB 00H; El carácter CQ Putback
F974H DEFB 00H; El carácter RQ Putback

Estas cuatro variables se usan para crear que cualquier carácter no deseado devuelto a la cola asociada. Aunque la facilidad del putback es implementada en el MSX ROM está actualmente sin uso.

F975H VOICAQ: DEFS 128; Sonorice UNA cola
F9F5H VOICBQ: DEFS 128; La cola de la B de voz
FA75H VOICCQ: DEFS 128; La cola de la C de voz
FAF5H RS2IQ: DEFS 64; La cola del RS232

Estos cuatro amortiguadores contienen las tres colas de música y la cola del RS232, lo más reciente está sin uso.

FB35H PRSCNT: DEFB 00H

Esta variable es usada por el "juego" manipulador de declaración a contar

El número de operand completado forma una serie. El bit 7 está también colocado después de que cada uno de los tres operands han sido a los que se analizó gramaticalmente para impedir repetida activación de la rutina del estándar STRTMS.

FB36H SAVSP: DEFW 0000H

Esta variable es usada por el "juego" manipulador de declaración para salvar al Z80 SP antes de que el control se traslada para el analizador gramatical de lenguaje de macro. Su valor es comparado con el SP en el regreso para determinar si cualquier datos han quedado sobre la pila por una terminación llena en la cola por el analizador gramatical.

FB38H VOICEN: DEFB 00H

Esta variable contiene el número actual de voz siendo procesada por el "juego" manipulador de declaración. Los valores 0, 1 y 2 son propios de A de canales PSG, B y C.

FB39H SAVVOL: DEFW 0000H

Esta variable es usada por la "interrogación" de declaración de "juego" manipulador de orden a salvar el trasfondo actual de volumen mientras un descanso de amplitud de cero es generado.

FB3BH MCLLEN: DEFB 00H

Esta variable es usada por el analizador gramatical de lenguaje de macro para sostener el largo del operand de la cuerda siendo a la que se analizó gramaticalmente.

FB3CH MCLPTR: DEFW 0000H

Esta variable es usada por el analizador gramatical de lenguaje de macro para mantener la dirección del operand de la cuerda siendo a la que se analizó gramaticalmente.

FB3EH QUEUEN: DEFB 00H

Esta variable es usada por ahí lo interrumpe manipulador para mantener el número de la cola de música actualmente siendo procesado. Los valores 0, 1 y 2 son propios de A de canales PSG, B y C.

FB3FH MUSICF: DEFB 00H

Esta variable contiene banderas de tres bits se sedimentan por la rutina del estándar STRTMS a iniciar el procesamiento de una cola de música por lo interrumpe manipulador. Los añicos 0, 1 y 2 son propios de VOICAQ, VOICBQ y VOICCQ.

FB40H PLYCNT: DEFB 00H

Esta variable es usada por la rutina del estándar STRTMS para mantener el número de "juego" secuencias de declaración actualmente mantenidas en las colas de música. Es examinado cuando todo lo que tres fin de la cola demarca ha sido encontrado pues una secuencia a determinar si dequeueing debería ser vuelta a arrancar.

- 229 -

6. El MAPA de MEMORIA

FB41H VCBA: DEFW 0000H; El mueble mostrador de duración
FB43H DEFB 00H; El largo de la cuerda
FB44H DEFW 0000H; La dirección de la cuerda
FB46H DEFW 0000H; La dirección de datos de la pila
FB48H DEFB 00H; El largo del paquete de música
FB49H DEFS 7; El paquete de música
FB50H DEFB 04H; La octava
FB51H DEFB 04H; El largo
FB52H DEFB 78H; El tiempo
FB53H DEFB 88H; El volumen
FB54H DEFW 00FFH; El período del sobre
FB56H DEFS 16; Espacie para los datos de la pila

Este treinta y siete byte el amortiguador es usado por el "juego" manipulador de declaración para sostener los parámetros coetáneos para A de voz.

FB66H VCBB: DEFS 37

Este amortiguador es usado por el "juego" manipulador de declaración para sostener los parámetros coetáneos para la voz B, su estructura equivale a VCBA.

FB8BH VCBC: DEFS 37

Este amortiguador es usado por el "juego" manipulador de declaración para sostener los parámetros coetáneos para la voz C, su estructura equivale a VCBA.

FBB0H ENSTOP: DEFB 00H

Esta variable determina ya sea lo interrumpe manipulador ejecutará un principio caliente para el Interpreter al detectar el CÓDIGO de llaves, GRAPH, CTRL y CAMBIO deprimido conjuntamente: El 00H = Disable, NZ = Enable.

FBB1H BASROM: DEFB 00H

Esta variable determina ya sea el ISCNTC y rutinas del estándar INLIN responderá a la clave CTRL-STOP: El 00H = Enable, NZ = Disable. Se usa para impedir terminación de un BASIC ROM localizado durante la búsqueda que se impulsa arriba ROM.

FBB2H LINTTB: DEFS 24

Cada uno de estas veinticuatro variables son normalmente poco cero pero son puestos en el cero si los contenidos de la fila correspondiente de la pantalla se han derramado encima de la siguiente fila. Son mantenidos por los BIOS sino sólo realmente usado por la rutina del estándar INLIN (el editor de la pantalla) para discriminar entre rayado lógico y físico.

FBCAH FSTPOS: DEFW 0000H

Esta variable se usa para sostener las coordenadas del cursor en la entrada

- 230 -

6. El MAPA de MEMORIA

Para la rutina del estándar INLIN. Su función es restringir la extensión de volver hacia atrás realizado cuando el texto es coleccionado de la pantalla en la terminación.

FBCCH CURSAV: DEFB 00H

Esta variable se usa para sostener la pantalla que el carácter reemplazó por el cursor del texto.

FBCDH FNKSWI: DEFB 00H

Esta variable es usada por la rutina del estándar CHSNS a determinar si las llaves intercambiadas o no intercambiadas de función son actualmente exhibidas: El 00H = Cambió De Posición, 01H Unshifted.

FBCEH FNKFLG: DEFS 10

Cada uno de estas diez variables son normalmente cero pero son determinados para 01H si la llave asociada de función se ha encendido por una "llave (n) EN" declaración. Son usados por ahí lo interrumpe al manipulador para determinar si, en el modo de programa sólo, debería devolver una cadena de caracteres o debería actualizar la entrada asociada en TRPTBL.

FBD8H ONGSBF: DEFB 00H

Esta variable es normalmente cero pero es incrementada por ahí lo interrumpe manipulador cuandoquiera que un dispositivo ha logrado las condiciones necesarias para generar un programa interrumpe. Es usado por el Runloop a determinar si cualquier programa interrumpe está pendiente sin tener que registrar a TRPTBL.

FBD9H CLIKFL: DEFB 00H

Esta variable es usada internamente por ahí lo interrumpe manipulador para impedirlos chasquidos cruciales espurios cuándo los personajes múltiples restituidores de una sola depresión crucial como una llave de función.

FBD9H OLDKEY: DEFS 11

Este amortiguador es usado por ahí lo interrumpe manipulador para sostener la condición previa de la matriz del teclado, cada byte contiene una fila de llaves echando a andar con fila 0.

FBE5H NEWKEY: DEFS 11

Este amortiguador es usado por ahí lo interrumpe manipulador para sostener la condición coetánea de la matriz del teclado. Las transiciones de la llave son detectadas en contraste con los contenidos de OLDKEY después del cual OLDKEY está actualizado con la condición coetánea.

FBF0H KEYBUF: DEFS 40

Este amortiguador contiene los personajes del teclado descifrados producidos por lo interrumpe al manipulador. Noto que el amortiguador es organizado como

- 231 -

6. El MAPA de MEMORIA

Una cola circular conducida por GETPNT y PUTPNT y consecuentemente no tiene punto de partida fijo.

FC18H LINWRK: DEFS 40

Este amortiguador es usado por los BIOS para mantener una línea completa de personajes de la pantalla.

FC40H PATWRK: DEFS 8

Este amortiguador es usado por los BIOS para mantener un patrón 8x8 pixel.

La RAÍZ del FC48H: DEFW 8000H

Esta variable contiene la dirección de la posición de la RAM mínima usada por el Interpreter. Su valor acometió energizar y después inalterado.

FC4AH HIMEM: DEFW F380H

Esta variable contiene la dirección del byte siguiendo la posición de la RAM más alta usada por el Interpreter. Su valor acometió contra que se energiza arriba y después sólo alterado por la declaración "clara".

FC4CH TRPTBL: DEFS 3; LLAVE 1
FC4FH DEFS 3; LLAVE 2
FC52H DEFS 3; LLAVE 3
FC55H DEFS 3; LLAVE 4
FC58H DEFS 3; LLAVE 5
FC5BH DEFS 3; LLAVE 6
FC5EH DEFS 3; LLAVE 7
FC61H DEFS 3; LLAVE 8
FC64H DEFS 3; LLAVE 9
FC67H DEFS 3; LLAVE 10
FC6AH DEFS 3; ALTO
FC6DH DEFS 3; EI BLOQUE DE OBJETOS MOVIBLES
FC70H DEFS 3; STRIG 0
FC73H DEFS 3; STRIG 1
FC76H DEFS 3; STRIG 2
FC79H DEFS 3; STRIG 3
FC7CH DEFS 3; STRIG 4
FC7FH DEFS 3; EI INTERVALO
FC82H DEFS 3; Sin uso
FC85H DEFS 3; Sin uso
FC88H DEFS 3; Sin uso
FC8BH DEFS 3; Sin uso
FC8EH DEFS 3; Sin uso
FC91H DEFS 3; Sin uso
FC94H DEFS 3; Sin uso
FC97H DEFS 3; Sin uso

Estas veintiseis tres variables de byte sostienen la condición coetánea de lo interrumpa generar artificios. El primer byte de cada entrada contiene el estado del dispositivo (el bit 0 = de Adelante, mordido = Stop 1, mordió 2 = Event

- 232 -

6. El MAPA de MEMORIA

Activo) y está actualizado por ahí lo interrumpe al manipulador, el Runloop interrumpen

procesador y el "dispositivo 0 ON/OFF/STOP" y "el REGRESO" manipuladores de declaración. Los bytes dos restantes de cada entrada están listos por ahí lo "en DEVICE GOSUB" manipulador de declaración y contienen la dirección de la línea de programa para ejecutar en un programa interrumpa.

FC9AH RTYCNT: DEFB 00H

Esta variable está sin uso por el MSX ROM actual.

FC9BH INTFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada para 03H o 04H si el CTRL-STOP o llaves de ALTO es detectó por ahí lo interrumpe al manipulador.

FC9CH PADY: DEFB 00H

Esta variable contiene la Y coordinada del último punto detectado por un touchpad.

FC9DH PADX: DEFB 00H

Esta variable contiene la X coordinada del último punto detectado por un touchpad.

EI MOMENTITO del FC9EH: DEFW 0000H

Esta variable es continuamente incrementada por ahí lo interrumpe manipulador. Su valor puede ser determinado o puedo leer por la declaración "de tiempo" o puede funcionar.

FCA0H INTVAL: DEFW 0000H

Esta variable tiene aplicación que la duración de intervalo se sedimentó por ahí lo "en el INTERVALO" el manipulador de declaración.

FCA2H INTCNT: DEFW 0000H

Esta variable es continuamente decremented por lo interrumpe manipulador. Cuando el cero es cumplido su valor es puesto a cero de INTVAL y, si aplicable, un programa interrumpa generado. Noto que esta variable siempre cuenta sin distinción de ya sea un "intervalo EN" declaración es activo.

FCA4H LOWLIM: DEFB 31H

Esta variable se usa para sostener la duración admisible mínima del bit de principio tan determinada por la rutina del estándar TAPION.

FCA5H WINWID: DEFB 22H

Esta variable se usa para sostener la duración de discriminación de ciclo LO/HI tan determinada por la rutina del estándar TAPION.

- 233 -

6. El MAPA de MEMORIA

FCA6H GRPHED: DEFB 00H

Esta variable es normalmente cero pero es determinada para 01H por la rutina del estándar CNVCHR en la detección de un código gráfico del encabezado.

FCA7H ESCCNT: DEFB 00H

Esta variable es usada por el procesador de secuencia de rutina del estándar CHPUT ESC para contar parámetros de escapada.

FCA8H INSFLG: DEFB 00H

Esta variable es normalmente cero pero es determinada para FFH por la rutina del estándar INLIN cuando el modo de inserción está adelante.

FCA9H CSRSW: DEFB 00H

Si esta variable es, entonces ponga en el cero el cursor es sólo el rato exteriorizado que la rutina del estándar CHGET está esperando a un carácter del teclado. Si es poco ponga en el cero, entonces el cursor es permanentemente exhibido por la rutina del estándar CHPUT.

FCAAH CSTYLE: DEFB 00H

Esta variable determina el estilo del cursor: El 00H = Block, NZ = Underline.

FCABH CAPST: DEFB 00H

Esta variable es usada por ahí lo interrumpe manipulador para mantener el estado coetáneo del cerrojo de gorras: El 00H = Off, NZ = On.

FCACH KANAST: DEFB 00H

Esta variable se usa para sostener el estado del cerrojo del teclado Kana en máquinas japonesas y el estado TOTALMENTE crucial en máquinas europeas.

FCADH KANAMD: DEFB 00H

Esta variable sostiene un modo del teclado en máquinas japonesas sólo.

FCAEH FLBMEM: DEFB 00H

Esta variable es determinada por el archivo yo los generadores de error de la / O pero está otrora sin uso.

FCAFH SCRMOD: DEFB 00H

Esta variable contiene el modo coetáneo de la pantalla: El 0=40x24 Text Mode, Mode 1=32x24 Text Mode, 2 = Graphics, 3 = el Multicolor Mode.

FCB0H OLDSCR: DEFB 00H

Esta variable tiene aplicación el modo de la pantalla del último set de modo del texto.

FCB1H CASPRV: DEFB 00H

Esta variable se usa para crear que cualquier carácter devuelto a uno y/o el amortiguador de la / O por la función del putback del casete.

FCB2H BDRATR: DEFB 00H

Esta variable contiene el color de demarcación para la "pintura" manipulador de declaración. Su valor es determinado por la rutina del estándar PNTINI y es usado por el SCANR y rutinas del estándar SCANL.

FCB3H GXPOS: DEFW 0000H

Esta variable sirve para almacenamiento temporal de una coordenada de la X de gráficos.

FCB5H GYPOS: DEFW 0000H

Esta variable sirve para almacenamiento temporal de una coordenada de la Y de gráficos.

FCB7H GRPACX: DEFW 0000H

Esta variable contiene la coordenada coetánea de la X de gráficos para la rutina del estándar GRPPRT.

FCB9H GRPACY: DEFW 0000H

Esta variable contiene la coordenada coetánea de la Y de gráficos para la rutina del estándar GRPPRT.

FCBBH DRWFLG: DEFB 00H

Los bits 6 y 7 de esta variable son determinadas por el "empate" declaración "N" y "B" manipuladores de orden para revolver el modo asociado adelante.

FCBCH DRWSCL: DEFB 00H

Esta variable es usada por el "empate" manipulador de orden de declaración "S" para sostener el factor coetáneo de escala.

FCBDH DRWANG: DEFB 00H

Esta variable es usada por la "A" de declaración de "empate" manipulador de orden para mantener el ángulo actual.

FCBEH RUNBNF: DEFB 00H

Esta variable es normalmente cero pero es determinada por el "BLOAD" manipulador de declaración cuando una "interrogación" andada en automóvil parámetro es especificada.

FCBFH SAVENT: DEFW 0000H

Esta variable contiene al "BSAVE" y "BLOAD" dirección de entrada.

- 235 -

6. El MAPA de MEMORIA

FCC1H EXPTBL: DEFB 00H; La Ranura Primaria 0
FCC2H DEFB 00H; La Ranura Primaria 1
FCC3H DEFB 00H; La Ranura Primaria 2
FCC4H DEFB 00H; La Ranura Primaria 3

Cada uno de estas cuatro variables son normalmente cero pero son determinados para 80H durante la búsqueda que se impulsa arriba de la RAM si el Slot Primario asociado - se encuentra - es expandido.

FCC5H SLTTBL: DEFB 00H; La Ranura Primaria 0
FCC6H DEFB 00H; La Ranura Primaria 1
FCC7H DEFB 00H; La Ranura Primaria 2
FCC8H DEFB 00H; La Ranura Primaria 3

Estas cuatro variables multiplican por dos los contenidos del cuatro Slot Secundario posible Registers. Los contenidos de cada variable sólo deberían ser supuestos tan válidos si EXPTBL muestra al Slot Primario asociado ser expandido.

FCC9H SLTATR: DEFS 4; PS0, SS0
FCCDH DEFS 4; PS0, SS1
FCD1H DEFS 4; PS0, SS2
FCD5H DEFS 4; PS0, SS3

FCD9H DEFS 4; PS1, SS0
FCDDH DEFS 4; PS1, SS1
FCE1H DEFS 4; PS1, SS2
FCE5H DEFS 4; PS1, SS3

FCE9H DEFS 4; PS2, SS0
FCEDH DEFS 4; PS2, SS1
FCF1H DEFS 4; PS2, SS2
FCF5H DEFS 4; PS2, SS3

FCF9H DEFS 4; PS3, SS0
FCFDH DEFS 4; PS3, SS1
FD01H DEFS 4; PS3, SS2
FD05H DEFS 4; PS3, SS3

Estas sesenta y cuatro variables contienen los atributos de cualquier extensión que los ROMs encontraron durante la búsqueda que se impulsa arriba ROM. Las características de cada 16 KB ROM están codificadas en un solo byte así es que cuatro bytes son precisados para cada ranura posible. La codificación es:

Mordido programa de = BASIC de 7 sets
Mordido manipulador = Device de 6 sets
Mordido manipulador = Statement de 5 sets

Note que las entradas para página 0 (0000H para 3FFFH) y página 3 (C000H para FFFFH) siempre serán cero como sólo página 1 (4000H para 7FFFH) y página 2 (8000H para BFFFH) es de hecho examinado. La convención MSX es que los ROMs de la extensión de código de la máquina son colocados en la página 1 y programa de BASIC que los ROMs de adentro mandan a llamar 2.

6. El MAPA de MEMORIA

FD09H SLTWRK: DEFS 128

Este amortiguador provee dos bytes de área de trabajo local para cada uno de los sesenta y cuatro ROMs posibles de la extensión.

FD89H PROCNM: DEFS 16

Este amortiguador se usa para mantener un dispositivo o nombre de declaración para examen por una extensión ROM.

El DISPOSITIVO del FD99H: DEFB 00H

Esta variable se usa para pasar un código del dispositivo, de 0 para 3, para una extensión ROM.

Los Hooks

El capítulo del Workspace Area de FD9AH para FFC9H contiene ciento doce ganchos, cada uno del cual se llena de cinco opcodes Z80 RET en que se energiza arriba. Estos son llamados de posiciones estratégicas dentro del BIOS / Interpreter a fin de que el ROM puede estar extendido, particularmente a fin de que puede ser mejorado para el BASIC Disk. Cada gancho tiene suficiente cuarto para hacer una llamada lejana cumplir cualquier ranura:

RST 30H
DEFB Haga Una Ranura En Idaho
DEFW Address
ENRIE

Los ganchos están listados en las siguientes páginas conjuntamente con la dirección de las que ellos son llamadas y una nota del resumen en lo que se refiere a su función.

FD9AH HKEYI: DEFS 5; El manipulador del 0C4AH Interrupt
FD9FH HTIMI: DEFS 5; El manipulador del 0C53H Interrupt
FDA4H HCHPU: DEFS 5; La rutina del estándar 08C0H CHPUT
FDA9H HDSPC: DEFS 5; El cursor del 09E6H Display
FDAEH HERAC: DEFS 5; El cursor del 0A33H Erase
FDB3H HDSPF: DEFS 5; La rutina del estándar 0B2BH DSPFNK
FDB8H HERAF: DEFS 5; La rutina del estándar 0B15H ERAFNK
FDBDH HTOTE: DEFS 5; La rutina del estándar 0842H TOTEXT
FDC2H HCHGE: DEFS 5; La rutina del estándar 10CEH CHGET
FDC7H HINIP: DEFS 5; La copia del 071EH el símbolo se sedimentó para VDP
FDCCH HKEYC: DEFS 5; El decodificador del 1025H Keyboard
FDD1H HKYEA: DEFS 5; El decodificador del 0F10H Keyboard
FDD6H HNMI: DEFS 5; La rutina del estándar 1398H NMI
FDDBH HPINL: DEFS 5; La rutina del estándar 23BFH PINLIN
FDE0H HQINL: DEFS 5; La rutina del estándar 23CCH QINLIN
FDE5H HINLI: DEFS 5; La rutina del estándar 23D5H INLIN
FDEAH HONGO: DEFS 5; 7810H "en DEVICE GOSUB"
FDEFH HSKO: DEFS 5; 7C16H "DSKO\$"
FDF4H HSETS: DEFS 5; 7C1BH "se sedimentó"
FDF9H HNAME: DEFS 5; 7C20H "nombre"

6. El MAPA de MEMORIA

FD FEH HKILL: DEFS 5; 7C25H "presa"
FE03H HIPL: DEFS 5; 7C2AH "IPL"
FE08H HCOPY: DEFS 5; 7C2FH "cópiese"
FE0DH HCMD: DEFS 5; 7C34H "CMD"
FE12H HDSKF: DEFS 5; 7C39H "DSKF"
FE17H HDSKI: DEFS 5; 7C3EH "DSKI\$"
FE1CH HATTR: DEFS 5; 7C43H "ATTR\$"
FE21H HLSET: DEFS 5; 7C48H "LSET"
FE26H HRSET: DEFS 5; 7C4DH "RSET"
FE2BH HFIEL: DEFS 5; 7C52H "campo"
FE30H HMKI\$: DEFS 5; 7C57H "MKI\$"
FE35H HMK\$: DEFS 5; 7C5CH "MK\$"
FE3AH HMKD\$: DEFS 5; 7C61H "MKD\$"
FE3FH HCVI: DEFS 5; 7C66H "CVI"
FE44H HCVS: DEFS 5; 7C6BH "CVS"
FE49H HCVD: DEFS 5; 7C70H "CVD"
FE4EH HGETP: DEFS 5; El 6A93H Locate FCB
FE53H HSETF: DEFS 5; El 6AB3H Locate FCB
FE58H HNOFO: DEFS 5; 6AF6H "ábrase"
FE5DH HNULO: DEFS 5; 6B0FH "ábrase"
FE62H HNTFL: DEFS 5; El 6B3BH Close yo el amortiguador de la / O 0
FE67H HMERG: DEFS 5; 6B63H "MERGE/LOAD"
FE6CH HSAVE: DEFS 5; 6BA6H "salvo"
FE71H

HBINS: DEFS 5; 6BCEH "salvo"
FE76H HBINL: DEFS 5; 6BD4H "MERGE/LOAD"
FE7BH HFILE: DEFS 5; 6C2FH "archivos"
FE80H HDGET: DEFS 5; 6C3BH "GET/PUT"
FE85H HFILO: DEFS 5; La salida del 6C51H Sequential
Las CIERVAS del FE8AH: DEFS 5; La entrada del 6C79H Sequential
FE8FH HRSLF: DEFS 5; 6CD8H "INPUT\$"
FE94H HSAVD: DEFS 5; 6D03H "LOC", 6D14H "LOF"
; 6D25H "EOF", 6D39H "FPOS"
FE99H HLOC: DEFS 5; 6D0FH "LOC"
FE9EH HLOF: DEFS 5; 6D20H "LOF"
FEA3H HEOF: DEFS 5; 6D33H "EOF"
FEA8H HFPOS: DEFS 5; 6D43H "FPOS"
FEADH HBAKU: DEFS 5; 6E36H "revista INSUMO"
FEB2H HPARD: DEFS 5; El nombre del dispositivo del 6F15H Parse
FEB7H HNODE: DEFS 5; El nombre del dispositivo del 6F33H Parse
FEBCH HPOSD: DEFS 5; El nombre del dispositivo del 6F37H Parse
FEC1H HDEVN: DEFS 5; Este gancho no es usado.
FEC6H HGEND: DEFS 5; El 6F8FH yo el repartidor de función de la / O
FECBH HRUNC: DEFS 5; El 629AH claro en la carrera
FED0H HCLEA: DEFS 5; El 62A1H claro en la carrera
FED5H HLOPD: DEFS 5; El 62AFH claro en la carrera
FEDAH HSTKE: DEFS 5; El 62F0H Volvió A Arrancar pila
FEDFH HISFL: DEFS 5; La rutina del estándar 145FH ISFLIO
FEE4H HOUTD: DEFS 5; La rutina del estándar del 1B46H OUTDO
FEE9H HCRDO: DEFS 5; 7328H CR, LF a SUPERAR
FEEEH HDSKC: DEFS 5; 7374H Mainloop revista insumo
FEF3H HDOGR: DEFS 5; El empate del 593CH Line
FEF8H HPRGE: DEFS 5; El fin del 4039H Program
FEFDH HERRP: DEFS 5; El manipulador del 40DCH Error

FF02H HERRF: DEFS 5; El manipulador del 40FDH Error

- 238 -

6. El MAPA de MEMORIA

FF07H HREAD: DEFS 5; 4128H Mainloop "Oklahoma"
FF0CH HMAIN: DEFS 5; 4134H Mainloop
FF11H HDIRD: DEFS 5; 41A8H Mainloop dirija declaración
FF16H HFINI: DEFS 5; 4237H Mainloop terminó
FF1BH HFINE: DEFS 5; 4247H Mainloop terminó
FF20H HCRUN: DEFS 5; 42B9H Tokenize
FF25H HCRUS: DEFS 5; 4353H Tokenize
FF2AH HISRE: DEFS 5; 437CH Tokenize
FF2FH HNTFN: DEFS 5; 43A4H Tokenize
FF34H HNOTR: DEFS 5; 44EBH Tokenize
FF39H HSNFG: DEFS 5; 45D1H "para"
FF3EH HNEWS: DEFS 5; La declaración nueva 4601H Runloop
FF43H HGONE: DEFS 5; 4646H Runloop ejecute
FF48H HCHRG: DEFS 5; La rutina del estándar 4666H CHRGR
FF4DH HRETU: DEFS 5; 4821H "regrese"
FF52H HPRTF: DEFS 5; 4A5EH "escriba en letras de imprenta"
FF57H HCOMP: DEFS 5; 4A54H "escriba en letras de imprenta"
FF5CH HFINP: DEFS 5; 4AFFH "escriba en letras de imprenta"
FF61H HTRMN: DEFS 5; El error de "lectura /entrada" 4B4DH
FF66H HFRME: DEFS 5; El Evaluador del 4C6DH Expression
FF6BH HNTPL: DEFS 5; El Evaluador del 4CA6H Expression
FF70H HEVAL: DEFS 5; El Evaluador del 4DD9H Factor
FF75H HOKNO: DEFS 5; El Evaluador del 4F2CH Factor
FF7AH HFING: DEFS 5; El Evaluador del 4F3EH Factor
FF7FH HISMI: DEFS 5; 51C3H Runloop ejecute
FF84H HWIDT: DEFS 5; 51CCH "anchura"
FF89H HLIST: DEFS 5; 522EH "escore"
FF8EH HBUFL: DEFS 5; 532DH Detokenize
FF93H HFRQI: DEFS 5; El 543FH Convert para el entero
FF98H HSCNE: DEFS 5; El número del 5514H Line para puntero
FF9DH HFRET: DEFS 5; El descriptor del 67EEH Free
FFA2H HPTRG: DEFS 5; La búsqueda del 5EA9H Variable
FFA7H HPHYD: DEFS 5; La rutina del estándar 148AH PHYDIO
FFACH HFORM: DEFS 5; La rutina del estándar del FORMATO del 148EH
FFB1H HERRO: DEFS 5; El manipulador del 406FH Error
FFB6H HLPTO: DEFS 5; La rutina del estándar 085DH LPTOUT
FFBBH HLPTS: DEFS 5; La rutina del estándar 0884H LPTSTT
FFC0H HSCRE: DEFS 5; 79CCH "filtre"
FFC5H HPLAY: DEFS 5; 73E5H "juegue" declaración

El Workspace Area de FFCAH para FFFFH es no acostumbrado. (En MSX 1)

EI ÍNDICE

"EI SISTEMA DE FRENOS ABS" 97 CLIKFL 57, 231 ángulos 152 CLIKSW 57, 197, 212 ARG 90, 223 CLINEF 152, 225 el almacenamiento Array 155, 162, 208 Recortándole 70 ARYTAB 137, 208, 220 CLMLST 126, 211 "ASC" 168 "CLOAD" 180 ASPCT1 77, 215 CLOC 70, 225 ASPEC2 77, 215 "le cierran" 173 ASPECTO 150, 225 CLPRIM 31, 210 "ATN" 93 "Lingüísticas Computacionales" 41 ATRBAS 34, 225 CMASK 70, 225 ATRBYT 69, 72, 76, 144, 214 "CMD" 202 "ATTR\$" 203 CNPNTS 150, 225 AUTFLG 116, 219 CNSDFG 49, 212 AUTINC 116, 219 Coincidencia 9, 52 AUTLIN 116, 219 "predisponen" 197 "automóvil" 125 Colores 12, 39, 40, 77 CONLO 121, 217 BAKCLR 39, 213 códigos Control 44 "basan" 200, 201 CONSAV 121, 217 BASROM 32, 86, 230 "CONT" 162 tasa Baud 79, 197 CONTXT 121, 217 BDRATR 77, 235 CONTYP 121, 217 BDRCLR 40, 213 Coordinadas, gráfico 70, 143 "emiten un pip" 60 Coordinadas, el texto 42, 47, 51 "N\$" que 165 "emulan" 202 "BLOAD" 177 "Colorados" 92 MÁS BAJO 204, 232 CPCNT8 152, 226 Boundary inflexión 148 CPLOTF 150, 226 "BSAVE" 177 Crash 30 BUF 85, 127176, 216 CRCSUM 151, 226 CRTCNT 47, 51, 211 "la LLAMADA" 140 LED CS1200 197, 214 Caps 6, 57 CS2400 197, 214 CAPST 57, 234 "CSAVE" 179 CASPRV 184, 235 CSAVEA 78, 149, 226 la entrada Cassette 24, 81 CSAVEM 78, 149, 226 el motor Cassette 6, 65, 78 CSCLXY 150, 226 la salida Cassette 6, 79 "CSNG" 100 "CDBL" 100 "CSRLIN" 196 CENCNT 150, 225 CSRSW 45, 234 CGPBAS 34, 224 CSRX 43, 212 CGPNT 38, 204, 224, 256 CSRY 43, 212 el conjunto de caracteres 3884, 256 CSTCNT 150, 226 "CHR\$" 168 CSTYLE 45, 85, 234 "CINT" 99 CTRL-STOP 32, 33, 42, 57, 59 "rodean" 150 CURLIN 56, 115, 116, 216 "claro" 163 CURSAV 46, 231

EI ÍNDICE

El cursor 13, 32, 42, 43, 46 Expander 4 "CVD" 204 Extension ROM 30, 140, 178, 205 "CVI" 203 el Evaluador Expression 128 "CVS" 204 EXPTBL 5, 32, 236 CXOFF 152, 226 CYOFF 152, 226 el Evaluador Factor 129 FBUFFR 106, 137, 222 DAC 90, 223 "el CAMPO" 203 "datos" 123 File Control Block 66, 209 áreas Data 26 "archivos" 173 DATLIN 115, 128, 218 Filespec 171 DATPTR 161, 220 FILNAM 171, 180, 224 Totalmente le teclean 56, 58 FILNM2 181, 224 "DEFDBL" 121 FILTAB 208, 223 "DEFFN" que 134 "se vengan de" 101 "DEFINT" 121 FLGINP 127, 219 "DEFSNG" 121 "FN" 134 "DEFSTR" 121 FNKFLG 56, 231 DEFTBL 121, 221 FNKSTR 65, 224 "DEFUSR" 134 "PARA "119" LE SUPRIME "137 FORCLR 40, 213 Dequeueing 60" FPOS "175

DISPOSITIVO 142, 237" FRE "171 Device 140, 171, 178 FRETOP 159, 208, 218" DÉBIL "155
FSTPOS 84, 88, 230 DIMFLG 155, 217 FUNACT 159, 222 PUNTEAN 115, 219 direcciones
Function 110, 130" LE SACAN "153 despliegue crucial Function 49, 53 DRWANG 154, 235 llaves
Function 56, 58, 65 DRWFLG 154, 235 DRWSCL 154, 235" GET "193" DSKF "202 GETPNT 33,
59, 214" DSKI\$ " 203"GOSUB" 122 "DSKO\$" 202 "GOTO" 123

Los personajes gráficos 42 Editor 84 Graphic no devuelven a 69 llaves
Edit 85 GRPACX 69, 143, 235 123, 125 GRPACY 69, 143, 235 "le acaban" 161 GRPATR 36, 212
ENDFOR 119, 218 GRPHED 42, 234 ENSTOP 53, 230 GRPNAM 36, 212 "EOF" 175 GRPPAT
36, 212 "ERASE" 162 GXPOS 144, 235 "ERL" 130 GYPOS 144, 235 "ERR" 130 ERRFLG 115,
130, 215 ENCABEZADO 79, 215 ERRLIN 115, 125, 130, 219 "HEX\$" 165 "error" 125 HIMEM
163, 204206, 232 generadores Error 115, 176 Ganchos 30, 204, 237 el manipulador Error 115
mensajes Error 113 "la DECLARACIÓN IF" 125 ERRTXT 115, 125, 220 "INKEY\$" 187 ESCCNT
43, 234 "INP" 114 "EXP" 94 "introducen en la computadora" 127

- I2 -

EI ÍNDICE

"INPUT\$" 174 LPTPOS 83, 125, 132, 215 Input, teclado 53, 59 "LSET" 203 INSFLG 85, 234
"INSTR" 170 el analizador gramatical Macro 142 "INT" 101 Mainloop 116 INTCNT 52, 233
constantes Math 97 el modo Interrupt 32 operadores Math 112 Interrupts 9, 52, 56, 161 MAXDEL
147, 225 "el INTERVALO" 194 MAXFIL 171, 223 INTFLG 32, 57, 233 "MAXFILES" 206 INTVAL
52, 233 MAXUPD 147, 213 yo / O Buffer 66, 125, 130, 171 MCLFLG 142, 188, 227 yo / OEI
repartidor 178 MCLLEN 142, 188, 229 "IPL" 202 MCLPTR 142, 188, 229
MCLTAB 142, 188, 227 japonés 4, 26, 58, 88 MEMSIZ 159, 163, 208, 217
el MOMENTITO 52, 196, 233 "mancomunan" 172 Joystick 24, 52, 63 "MID\$" 169, 170
MINDEL 147, 225 KANAMD 33, 234 MINUPD 147, 213 KANAST 58, 234
"MKD\$" 203 Kansas City 79 "MKI\$" 203 KBUF 116, 118, 216 "MKS\$" que 203 "teclean" 195
MLTATR 36, 212 Keyboard 5, 53, 55, 66 MLTNAM 36, 212 KEYBUF 33, 231 MLTPAT 36, 212
Key dan un clic sobre 6, 57, 58 números Cruciales 187 "motores" 54 MOVCNT 149, 227 Palabras
Claves, el BASIC 110 MUSICF 52, 62, 229 "presa" 202 paquete Music 61, 193

"LEFT\$" 169 NAMBAS 34, 224 "LEN" 168 "nombran" 202 "le dejaron" 123 "nuevos" 159 "LFILES"
173 NEWKEY 53, 231 "la LÍNEA" 145 Newton-Raphson 93 "entrada de la LÍNEA" 126
"siguientes" 164 Line números 117, 118, 119, 122 NLONLY 159, 224 Enlaces 117 NOFUNS 135,
156, 222 LINL32 34, 211 NTMSXP 83, 197, 216 LINL40 34, 211 NULBUF 208, 223 LINLEN 34,
211 la salida Numérica 106 LINTTB 39, 84, 230 tipos Numéricos 104 LINWRK 46, 232
"escoran"137 "OCT\$" 164 "LLIST" 136 OLDKEY 33, 53, 231 "le cargan" 172 OLDLIN 115, 162,
220 "LOC" 174 OLDSCR 34, 136, 234 "LOCATE" 193 OLDTXT 60, 115, 159, 162, 220 "LOF" 174
"en" 124 "le ponen en bitácora" 93 ONEFLG 124, 159, 220 LOWLIM 81, 233 ONELIN 115, 124,
159, 220 "LPOS" 132 ONGSBF 56, 160, 231 "LPRINT" que 125 "abren" 172

- I3 -

EI ÍNDICE

"Fuera de" 114 "reanuda" 124 Salida, Interpreter que 82 "devuelven" 123 Output, pantalla 43, 69
RGOSAV 35, 212

"RIGHT\$" que 169 "acolchan" 197 "RND" 94 el Remo 25, 64 RNDX 94,
159, 223 PADX 65, 233 "RSET" 203 PADY 65, 233 "dirigen" 122 Page 3 RUNBNF 177, 235

"pintan" 148 Runloop 120 PARM1 135 156, 221 PARM2 135 222 "le salvan" 172 PATBAS 34, 224 SAVENT 179, 235 PATWRK 69, 232 SAVSTK 115, 120, 159, 219 "PDL" 196 SAVTXT 120, 219 "atisbo" 138 SCNCNT 52, 214 "juego" 188, 196 "pantalla" 197 PLYCNT 62189, 229 SCRMOD 34, 136, 234 "enseñan con el dedo a" 144 Slot Secundario 4, 30, 32 Punteros 118, 139 "le colocan" 202 "pinchazo" 138 "SGN" 98 "el PECADO" 95 Polynomial 92 "de punto de venta" 132 SKPCNT 149, 227 que se energizan arriba 32, 204 Slot Idaho 29, 140, 205 Precedence 112, 129 SLTATR 140, 205, 236 "preprogramado" 144 SLTTBL 32, 236 Primary Slot 3, 30, 32, 66 "suenan" 188 "estampado" 125 "SPACE\$" 169 Printer 33, 41, 83 "bloque de objetos movibles" 194, 197, 198 PRMLN 159, 221 Bloques De Objetos Movibles 9, 18, 37, 199 PRMLN2 159, 222 "SQR" 93 PRMSTK 159, 221 Stack le espacian 159 PROCNM 140, 237 rutinas Estándar 26 almacenamiento Program 117, 208 direcciones Statement 109, 136 PRSCNT 188 228 STATFL 52, 213 PRTFLG 83, i25, 215 "vara" 196 "PSET" 144 STKTOP 159, 163, 209, 217 PSG 21, 33, 34, 60, 190 "alto" 161, 194 PTRFIL 66, 125, 223 "STR\$" 165 "puesto" 193 STREND 137208, 220 PUTPNT 33, 59, 214 "STRIG" 194, 196 "STRING\$" 168 QUEBAK 68, 228 el almacenamiento String 133, 163, 166, 168 QUETAB 34, 67, 228 SUBFLG 155, 218 Queue 60, 67, 68, 189 "intercambian" 162 QUEUEN 62, 229 SWPTMP 162, 222 COLAS 68, 214 T32ATR 34, 211 RDPRIM 30 210 T32CGP 34, 211 "leen" 12; T32NAM 34, 211 "REM" 123 T32PAT 34, 211 "RENUM" que 138 "broncean" 93 REPCNT 53, 214 TEMPPT 159, 218 "restituyen" 161 TEMPST 124, 218

- 14 -

El ÍNDICE

"Cronométrele" 196 Señales 111, 117, 118, 121, 137 Touchpad 25, 64 TRCFLG 120, 222 TRGFLG 52, 213 "TROFF" 162 "TRON" 162 TRPTBL 32, 56, 160, 194, 232 TTYPOS 43, 125, 132, 216 TXTCGP 34, 211 TXTNAM 34 211 TXTTAB 117, 204, 208, 218 Tipos 140

"USR" 133 USRTAB 133, 210

"VAL" 169 VALTYP 104, 217 el almacenamiento Variable 135, 155, 208 "VARPTR" 130 VARTAB 137, 208, 220 VCBA 34, 61, 67, 230 "VDP" 199, 200 VDP Address Register 8, 40 VDP Mode Registers 9, Modes 35 VDP 10, 13, 34, 41 VDP Status Register 9, 52, 65 VDP Cronometrándole 39 "VPEEK" 201 "VPOKE" 201 VLZADR 115, 216 VLZDAT 115, 216

"Espera" 114 Wait condición 81 el principio Caliente 53 "la ANCHURA" 136 WINWID 81, 233 Workspace Area 32, 208 WRPRIM 30, 210

El Reloj Del Z80 81