

Formato SNA (Snapshot)

El formato SNA es uno de los más utilizados para snapshots del ZX Spectrum. Su popularidad se debe a su simplicidad y soporte extendido en emuladores. Sin embargo, tiene una limitación conocida: dos bytes de la memoria pueden corromperse debido a que el contador de programa (PC) se almacena temporalmente en la pila durante el proceso de guardar el snapshot.

Este problema surge porque el valor del registro **PC** se "empuja" en la pila, sobrescribiendo los dos bytes en la posición donde está el puntero de la pila (**SP**). Aunque en la mayoría de los casos esto no afecta el programa guardado, puede haber situaciones donde el espacio de la pila esté completamente en uso, lo que podría causar corrupción en la memoria. Una solución sugerida para mitigar este problema es sustituir los bytes corrompidos por ceros y luego incrementar el puntero de la pila.

Cuando se restaura un snapshot, el programa reanuda su ejecución mediante un comando **RETN**, que devuelve el control al punto donde se guardó el snapshot. El registro **IFF2** indica el estado de habilitación de las interrupciones; cuando está activo, significa que las interrupciones estaban habilitadas en el momento de la captura.

Estructura del archivo SNA (48K)

Desplazamiento	Tamaño	Descripción
0	1 byte	Registro I
1	8 bytes	Registros alternos: HL' , DE' , BC' , AF'
9	10 bytes	Registros principales: HL , DE , BC , IY , IX
19	1 byte	Registro de interrupciones (el bit 2 contiene IFF2 , 1 = habilitado, 0 = deshabilitado)
20	1 byte	Registro R
21	4 bytes	Registros AF , SP
25	1 byte	Modo de interrupción (IM) (0 = IM0, 1 = IM1, 2 = IM2)
26	1 byte	Color del borde (0 a 7)
27	49152 bytes	Volcado de RAM desde la dirección 16384 hasta 65535

Tamaño total: 49179 bytes

Formato SNA (128K)

El formato SNA para los modelos de 128K y +2 es una extensión del formato de 48K, incluyendo los bancos adicionales de memoria RAM. Además, corrige el problema relacionado con el registro **PC**, que ahora se almacena en una variable adicional en lugar de ser empujado en la pila.

Desplazamiento	Tamaño	Descripción
0	27 bytes	Cabecera SNA (idéntica al formato de 48K)
27	16 KB	Banco de RAM 5 (paginado en la dirección 0xC000)
16411	16 KB	Banco de RAM 2
32795	16 KB	Banco de RAM actualmente paginado
49179	2 bytes	Registro PC
49181	1 byte	Estado del puerto 0x7FFD (control de paginación)
49182	1 byte	ROM TR-DOS está paginada (1) o no (0)
49183	16 KB	Bancos de RAM restantes en orden ascendente (0, 1, 3, 4, 6, 7)

Tamaño total: 131103 o 147487 bytes, dependiendo de la cantidad de bancos guardados.

Detalles adicionales:

- El tercer banco de RAM guardado siempre es el banco actualmente paginado, incluso si es el banco 5 o 2, lo que significa que un mismo banco puede aparecer dos veces en el snapshot.
- Los bancos restantes se guardan en orden ascendente. Por ejemplo, si el banco 4 está paginado en el momento de la captura, el archivo contendrá los bancos 5, 2 y 4, seguidos de los bancos 0, 1, 3, 6 y 7. Por otro lado, si el banco 5 está paginado en el momento de la captura, el archivo contendrá los bancos 5, 2 y 5, seguidos de los bancos 0, 1, 3, 4, 6 y 7.

Casos donde se incluye la ROM

En una nueva variante del formato SNA para **48K**, si el tamaño del archivo es de **65563 bytes** en lugar de los **49179 bytes** estándar, significa que el snapshot incluye un volcado de la ROM de 16KB del ZX Spectrum. Este bloque extra de ROM se almacena inmediatamente después de la cabecera y antes del volcado de la memoria RAM desde la dirección **16384** hasta **65535**. (Esta variante fue introducida por Juan José Ponteprino en el proyecto ESpectrum.)

En este caso, la estructura sería la siguiente:

Desplazamiento	Tamaño	Descripción
0	27 bytes	Cabecera SNA
27	16 KB	Volcado de la ROM
16411	48 KB	Volcado de la RAM desde la dirección 0x4000 hasta 0xFFFF

Tamaño total: 65563 bytes

Formato SP

Overview

El formato SP es utilizado para almacenar programas en la memoria del ZX Spectrum. Este formato incluye información sobre el estado del procesador y los registros del Z80, así como el estado de la memoria y otras configuraciones relevantes.

Estructura del archivo SP

El archivo SP está estructurado de la siguiente manera:

Desplazamiento	Tamaño	Descripción
0	2 bytes	"SP" (0x53, 0x50) Signatura.
2	1 word	Longitud del programa en bytes (normalmente 49152 bytes).
4	1 word	Posición inicial del programa (normalmente posición 16384).
6	1 word	Registro BC del Z80.
8	1 word	Registro DE del Z80.
10	1 word	Registro HL del Z80.
12	1 word	Registro AF del Z80.
14	1 word	Registro IX del Z80.
16	1 word	Registro IY del Z80.
18	1 word	Registro BC' del Z80.
20	1 word	Registro DE' del Z80.
22	1 word	Registro HL' del Z80.
24	1 word	Registro AF' del Z80.
26	1 byte	Registro R (de refresco) del Z80.
27	1 byte	Registro I (de interrupciones) del Z80.
28	1 word	Registro SP del Z80.
30	1 word	Registro PC del Z80.
32	1 word	Reservado para uso futuro, siempre 0.
34	1 byte	Color del borde al comenzar.
35	1 byte	Reservado para uso futuro, siempre 0.
36	1 word	Palabra de estado codificada por bits. Formato:

Desplazamiento	Tamaño	Descripción
		Bit: 15-8 : Reservados para uso futuro.
		Bit: 7-6 : Reservados para uso interno, siempre 0.
		Bit: 5 : Estado del Flash: 0 - tinta INK, papel PAPER; 1 - tinta PAPER, papel INK.
		Bit: 4 : Interrupción pendiente de ejecutarse.
		Bit: 3 : Reservado para uso futuro.
		Bit: 2 : Biestable IFF2 (uso interno).
		Bit: 1 : Modo de interrupción: 0=IM1; 1=IM2.
		Bit: 0 : Biestable IFF1 (estado de interrupción): 0 - Interrupciones desactivadas (DI); 1 - Interrupciones activadas (EI).

Casos donde se incluye la ROM

En el caso de que el archivo SP incluya la ROM, tanto el campo "longitud del programa" (offset 2) como el campo "Posición inicial del programa" (offset 4) se establecen en 0.

Formato de Archivo Z80

El formato .z80 es, sin duda, el más ampliamente soportado por emuladores en todas las plataformas. Los archivos .z80 son instantáneas de memoria; contienen una imagen de los contenidos de la memoria del ZX Spectrum en un momento particular en el tiempo. Como resultado de esto, no pueden ser utilizados para reproducir la cinta original a partir de un archivo de instantánea, pero se cargan casi instantáneamente.

El formato .z80 fue desarrollado originalmente por Gerton Lunter para su emulador Z80, y se utilizan tres versiones del formato, como se guardan en las versiones Z80 1.45 (y anteriores), 2.x y 3.x (y posteriores). Para facilitar la notación, se denominarán versiones 1, 2 y 3 del formato respectivamente. También se han realizado varias extensiones al formato .z80 por otros emuladores.

Versión 1 del formato .z80 puede guardar solo instantáneas de 48K y tiene el siguiente encabezado:

Desplazamiento	Tamaño	Descripción
0	1 byte	Registro A
1	1 byte	Registro F
2	1 word	Par de registros BC (LSB, es decir, C primero)
4	1 word	Par de registros HL
6	1 word	Contador de programa
8	1 word	Puntero de pila
10	1 byte	Registro I (de interrupciones)
11	1 byte	Registro R (de refresco) (¡el bit 7 no es significativo!)
12	1 byte	Bit 0 : Bit 7 del registro R
		Bit 1-3: Color del borde
		Bit 4 : 1=SamRom de Basic activado
		Bit 5 : 1=Bloque de datos comprimido
		Bit 6-7: Sin significado
13	1 word	Par de registros DE
15	1 word	Par de registros BC'
17	1 word	Par de registros DE'
19	1 word	Par de registros HL'
21	1 byte	Registro A'
22	1 byte	Registro F'
23	1 word	Registro IY (nuevamente LSB primero)

Desplazamiento	Tamaño	Descripción
25	1 word	Registro IX
27	1 byte	Flip-flop de interrupción, 0=DI, de lo contrario EI
28	1 byte	IFF2 (no particularmente importante...)
29	1 byte	Bit 0-1: Modo de interrupción (0, 1 o 2)
		Bit 2 : 1=Emulación de Issue 2
		Bit 3 : 1=Frecuencia de interrupción doble
		Bit 4-5: 1=Sincronización de video alta
		3=Sincronización de video baja
		0,2=Normal
		Bit 6-7: 0=Joystick Cursor/Protek/AGF
		1=Joystick Kempston
		2=Joystick izquierdo Sinclair 2 (o definido por el usuario, para archivos .z80 versión 3)
		3=Joystick derecho Sinclair 2

Debido a la compatibilidad, si el byte 12 es 255, debe considerarse como 1.

Después de este bloque de encabezado de 30 bytes, los 48K de memoria del Spectrum siguen en un formato comprimido (si se establece el bit 5 del byte 12). El método de compresión es muy simple: reemplaza las repeticiones de al menos cinco bytes iguales por un código de cuatro bytes ED ED xx yy, que significa "byte yy repetido xx veces". Solo se codifican secuencias de longitud al menos 5. La excepción son las secuencias que consisten en ED; si se encuentran, incluso dos ED se codifican en ED ED 02 ED. Finalmente, cada byte que sigue directamente a un único ED no se toma en un bloque, por ejemplo, ED 6*00 no se codifica en ED ED ED 06 00, sino en ED 00 ED ED 05 00. El bloque se termina con un marcador de fin, 00 ED ED 00.

Las versiones 2 y 3 de los archivos .z80 comienzan con los mismos 30 bytes de encabezado que los archivos de la versión 1. Sin embargo, los bits 4 y 5 del byte de bandera ya no tienen significado, y el contador de programa (byte 6 y 7) es cero para señalar un archivo de versión 2 o 3.

Después de los primeros 30 bytes, sigue un encabezado adicional:

Desplazamiento	Tamaño	Descripción
* 30	1 word	Longitud del bloque de encabezado adicional (ver más abajo)
* 32	1 word	Contador de programa
* 34	1 byte	Modo de hardware (ver más abajo)
* 35	1 byte	Si está en modo SamRam, estado bitwise de 74ls259.

Desplazamiento	Tamaño	Descripción
		Por ejemplo, bit 6=1 después de un OUT 31,13 (=2*6+1)
		Si está en modo 128, contiene el último OUT a 0x7ffd
		Si está en modo Timex, contiene el último OUT a 0xf4
* 36	1 byte	Contiene 0xff si el rom de la Interfaz I está paginado
		Si está en modo Timex, contiene el último OUT a 0xff
* 37	1 byte	Bit 0: 1 si la emulación del registro R está activada
		Bit 1: 1 si la emulación de LDIR está activada
		Bit 2: AY sonido en uso, incluso en máquinas de 48K
		Bit 6: (si se establece el bit 2) Emulación del Fuller Audio Box
		Bit 7: Modificar hardware (ver más abajo)
* 38	1 byte	Último OUT al puerto 0xffffd (número de registro del chip de sonido)
* 39	16 bytes	Contenidos de los registros del chip de sonido
55	1 word	Contador de estado T bajo
57	1 byte	Contador de estado T alto
58	1 byte	Byte de bandera utilizado por Spectator (emulador QL spec.)
		Ignorado por Z80 al cargar, cero al guardar
59	1 byte	0xff si el Rom de MGT está paginado
60	1 byte	0xff si el Rom de Multiface está paginado. Siempre debe ser 0.
61	1 byte	0xff si 0-8191 es ROM, 0 si RAM
62	1 byte	0xff si 8192-16383 es ROM, 0 si RAM
63	5 words	5 x asignaciones de teclado para joystick definido por el usuario
73	5 words	5 x palabra ASCII: teclas correspondientes a las asignaciones anteriores
83	1 byte	Tipo de MGT: 0=Disciple+Epson, 1=Disciple+HP, 16=Plus D
84	1 byte	Estado del botón de inhibición de Disciple: 0=fuera, 0xff=adentro
85	1 byte	Bandera de inhibición de Disciple: 0=rom paginable, 0xff=no
** 86	1 byte	Último OUT al puerto 0x1ffd

El valor de la palabra en la posición 30 es 23 para archivos de versión 2, y 54 o 55 para la versión 3; los campos marcados '*' son los que están presentes en el encabezado de la versión 2. El byte final (marcado '**') solo está presente si el valor en la posición 30 es 55.

En general, los campos tienen el mismo significado en archivos de versión 2 y 3, con la excepción del byte 34:

Valor	Significado en v2	Significado en v3
0	48k	48k
1	48k + If.1	48k + If.1
2	SamRam	SamRam
3	128k	48k + M.G.T.
4	128k + If.1	128k
5	-	128k + If.1
6	-	128k + M.G.T.

(Como un dato curioso, la documentación para las versiones 3.00 a 3.02 de Z80 tenía las entradas para 'SamRam' y '48k + M.G.T.' en la segunda columna de la tabla anterior intercambiadas; además, los bytes 61 y 62 del formato estaban documentados incorrectamente hasta la versión 3.04. Las instantáneas producidas por las versiones anteriores de Z80 siguen lo anterior; solo que la documentación estaba equivocada).

Otros emuladores han extendido el formato .z80 para soportar más tipos de máquinas:

Valor	Significado
7	Spectrum +3
8	[utilizado erróneamente por algunas versiones de ZX-Program para indicar un +3]
9	Pentagon (128K)
10	Scorpion (256K)
11	Didaktik-Kompakt
12	Spectrum +2
13	Spectrum +2A
14	TC2048
15	TC2068
128	TS2068

Mientras que la mayoría de los emuladores que usan estas extensiones escriben archivos de versión 3, algunos escriben archivos de versión 2, por lo que probablemente sea mejor asumir que cualquiera de estos valores puede aparecer en archivos de versión 2 o versión 3.

Si el bit 7 del byte 37 está activado, los tipos de hardware se modifican ligeramente: cualquier máquina de 48K se convierte en una máquina de 16K, cualquier máquina de 128K se convierte en una +2 y cualquier máquina +3 se convierte en una +2A.

El contador de estados T alto cuenta hacia arriba en módulo 4. Justo después de que la ULA genera su interrupción una vez cada 20 ms, el contador es 3 y se incrementa en uno cada 5 milisegundos emulados. En

estos intervalos de 1/200 s, el contador de estados T bajo cuenta hacia abajo desde 17471 a 0 (17726 en modos de 128K), lo que hace un total de 69888 (70908) estados T por cuadro.

Las 5 palabras ASCII (el byte alto siempre es 0) en 73-82 son las teclas correspondientes a las direcciones del joystick: izquierda, derecha, abajo, arriba, disparo respectivamente. Shift, Symbol Shift, Enter y Space se denotan por [, /, \ respectivamente. Los valores ASCII se utilizan solo para mostrar las teclas del joystick; la información en las 5 palabras de mapeo del teclado determina qué tecla está realmente presionada (y debe corresponder a los valores ASCII). El byte bajo está en el rango de 0-7 y determina la fila del teclado. El byte alto es un byte de máscara y determina la columna. Enter, por ejemplo, se almacena como 0x0106 (fila 6 y columna 1) y 'g' como 0x1001 (fila 1 y columna 4).

El byte 60 debe ser cero, porque el contenido de la RAM del Multiface no se guarda en el archivo de instantánea. Si el Multiface estaba paginado cuando se guardó la instantánea, el programa emulado probablemente se bloqueará al cargarse de nuevo.

Los bytes 61 y 62 son una función de las otras banderas, como el byte 34, 59, 60 y 83.

A continuación, siguen una serie de bloques de memoria, cada uno conteniendo los datos comprimidos de un bloque de 16K. La compresión se realiza de acuerdo con el antiguo esquema, excepto por el marcador de fin, que ahora está ausente. La estructura de un bloque de memoria es:

Byte	Longitud	Descripción
0	2	Longitud de los datos comprimidos (sin este encabezado de 3 bytes)
		Si la longitud = 0xffff, los datos tienen 16384 bytes de largo y no están comprimidos
2	1	Número de página del bloque
3	[0]	Datos

Las páginas están numeradas, dependiendo del modo de hardware, de la siguiente manera:

Página	En modo 48	En modo 128	En modo SamRam
0	ROM de 48K	ROM (basic)	ROM de 48K
1	ROM de Interface I, Disciple o Plus D, según configuración	(idem modo 48)	(idem modo 48)
2	-	ROM (reset)	ROM de samram (basic)
3	-	Página 0	ROM de samram (monitor,..)
4	8000-bfff	Página 1	Normal 8000-bfff
5	c000-ffff	Página 2	Normal c000-ffff
6	-	Página 3	Sombra 8000-bfff
7	-	Página 4	Sombra c000-ffff

Página	En modo 48	En modo 128	En modo SamRam
8	4000-7fff	Página 5	4000-7fff
9	-	Página 6	-
10	-	Página 7	-
11	ROM de Multiface	ROM de Multiface	-

En modo 48K, se guardan las páginas 4, 5 y 8. En modo SamRam, se guardan las páginas 4 a 8. En modo 128K, se guardan todas las páginas de 3 a 10. Las instantáneas de Pentagon son muy similares a las instantáneas de 128K, mientras que las instantáneas de Scorpion tienen las 16 páginas de RAM guardadas en las páginas de 3 a 18. No hay un marcador de fin.

Casos donde se incluye la ROM

En el modo 48k, aunque no es habitual, es posible guardar la pagina 0 (ROM de 48k).

Referencias

- World of Spectrum. Formats Reference. Recuperado de <https://worldofspectrum.org/faq/reference/formats.htm>.
- Documentación del Emulador de Spectrum por Pedro Gimeno.
- Lunter, G. (1988). Z80 Emulator Documentation.