

Vircon32

CONSOLA VIRTUAL DE 32 BITS



Especificación del sistema

Parte 5: El chip de sonido (SPU)

Documento con fecha 2023.12.31

Escrito por Carra

¿Qué es esto?

Este documento es la parte número 5 de la especificación del sistema Vircon32. Esta serie de documentos define el sistema Vircon32, y provee una especificación completa que describe en detalle sus características y comportamiento.

El principal objetivo de esta especificación es definir un estándar de lo que es un sistema Vircon32, y cómo debe implementarse un sistema de juego para que se considere conforme a él. Además, al ser Vircon32 es un sistema virtual, un importante objetivo adicional de estos documentos es proporcionar a cualquiera el conocimiento para crear sus propias implementaciones de Vircon32.

Sobre Vircon32

El proyecto Vircon32 fue creado de forma independiente por Carra. El sistema Vircon32 y su material asociado (incluyendo documentos, software, código fuente, arte y cualquier otro elemento relacionado) son propiedad del autor original.

Vircon32 es un proyecto libre y de código abierto en un esfuerzo por promover que cualquiera pueda jugar a la consola y crear software para ella. Para obtener información más detallada al respecto, se recomienda consultar los textos de licencia incluidos en cada uno de los programas disponibles.

Sobre este documento

Este documento se proporciona bajo la Licencia de Atribución Creative Commons 4.0 (CC BY 4.0). Puede leerse el texto completo de la licencia en el sitio web de Creative Commons: <https://creativecommons.org/licenses/by/4.0/>

Índice

La parte 5 de la especificación define el chip de sonido de la consola (SPU). Este documento describirá el comportamiento de este chip, sus puertos de control, los comandos de sonido que proporciona y el proceso que usa para producir la salida de audio.

1 Introducción	3
2 Conexiones externas	3
3 Conceptos funcionales	4
4 Efectos de audio	7
5 Generación de sonido	9
6 Variables internas	11
7 Puertos de control	15
8 Ejecución de comandos	20
9 Generación de la señal de audio	22
10 Respuestas a señales de control	23

1 Introducción

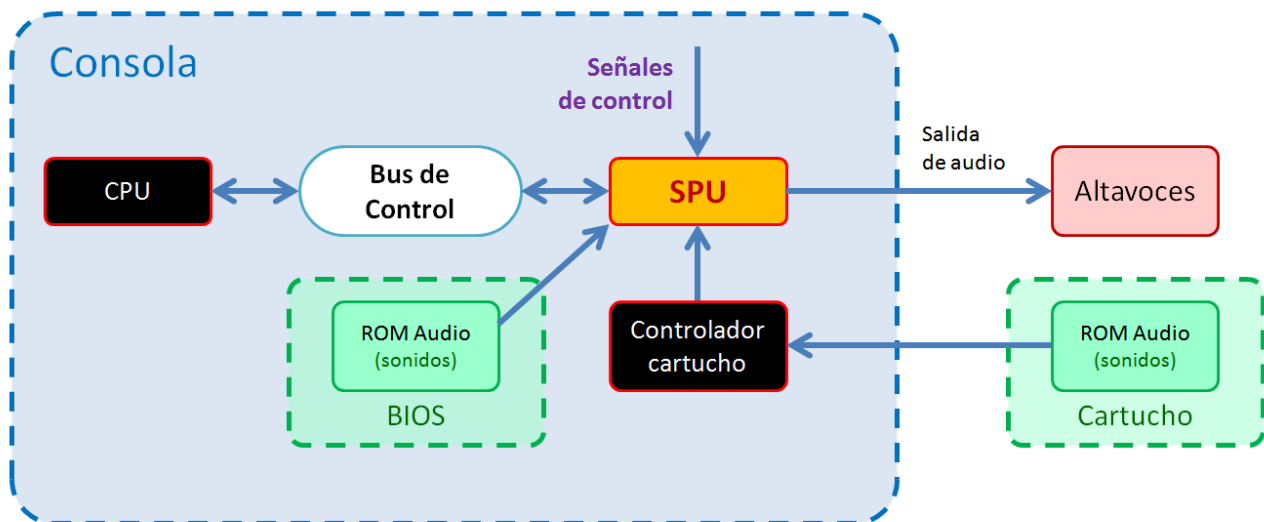
La SPU es el chip de audio de la consola Vircon32. Es responsable de crear todo el sonido que se puede escuchar por los altavoces. Para ello reproducirá los sonidos contenidos tanto en la BIOS como en el cartucho y les aplicará efectos de audio básicos.

Este chip de sonido es bastante simple: cuenta con una matriz de 16 canales de sonido, y cada canal puede reproducir uno de los sonidos disponibles en cada momento. Para producir la salida se mezcla el sonido de todos los canales que estén reproduciendo.

La actividad de los canales de sonido se controla mediante comandos de sonido del procesador. Aparte de esto, hay algunos efectos de audio que se pueden aplicar a la reproducción de los canales. Esto se tratará en secciones posteriores.

2 Conexiones externas

Como la SPU es sólo uno de los chips que forman la consola, no puede funcionar aislado. Esta figura muestra todas las comunicaciones de la SPU con otros componentes. Como se ve, la SPU tiene conexiones con todas las ROMs de audio disponibles para poder reproducir los sonidos que contienen. Obsérvese que, por claridad, los diagramas de la consola de la parte 2 de la especificación omiten a propósito estas conexiones.



Cada una de estas conexiones se explicará individualmente en las secciones siguientes.

2.1 Señales de control

Como todos los componentes de la consola, la SPU recibe las señales de reset, frame y ciclo. Las respuestas a esas señales se detallan en la sección 10 de este documento.

2.2 Bus de Control

La SPU está conectada como dispositivo esclavo al Bus de Control, con ID de dispositivo = 3. Esto permite al maestro del bus (la CPU) pedir operaciones de lectura o escritura en los puertos de control que expone la SPU. La lista de puertos de la SPU y sus propiedades se detallarán en secciones posteriores.

2.3 Chip de la BIOS

La BIOS, que siempre está presente, contiene una ROM de audio con exactamente 1 sonido. La SPU puede acceder a ese sonido a través de su pista de sonido con ID = -1.

2.4 Controlador de cartucho

Al igual que con la BIOS, la SPU puede acceder a cualquier sonido presente en el cartucho a través de las pistas de sonido con IDs de 0 a 1023. Pero la conexión debe usar el controlador de cartucho como proxy, ya que puede no haber cartucho presente. Incluso si hay cartucho, cada uno puede tener un número diferente de sonidos (de 0 a 1024).

2.5 Altavoces

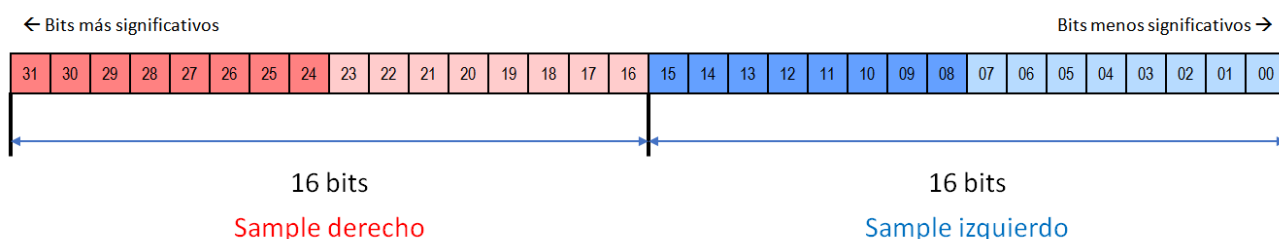
La SPU debe ser capaz de generar sonido para todos los canales en reproducción, hacer la mezcla y enviar el resultado a los altavoces a la frecuencia correcta para que se pueda oír. Esto se hace a través de la conexión de salida de audio. En la sección 9 se explica cómo.

3 Conceptos funcionales

Antes de explicar las funciones sonoras de la SPU o las variables internas que les afectan, debemos presentar algunos conceptos básicos en los que se basa el diseño de la SPU.

3.1 Samples de sonido

La unidad mínima de información de audio que maneja la SPU es el sample. Representa un único valor discreto en la secuencia de audio, para los altavoces o para un sonido. Cada sample se representa como un único valor de 32 bits que codifica valores de 16 bits para los canales izquierdo y derecho, como ya se mostró en la parte 2 de la especificación:



3.2 Sonidos de la SPU

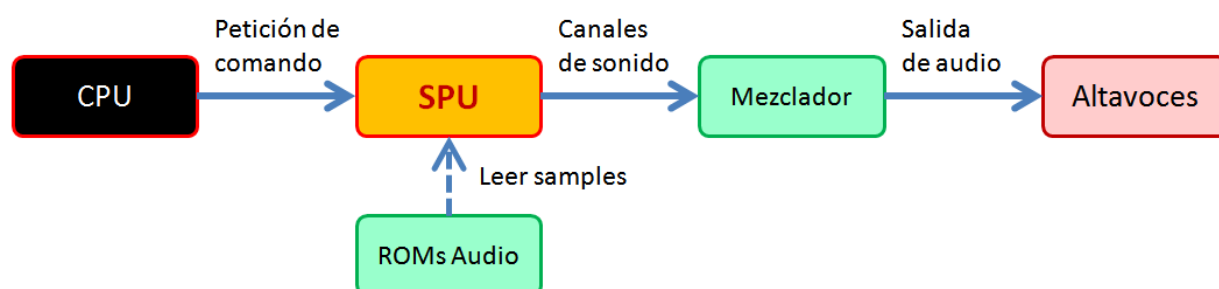
Un sonido de SPU es una secuencia de samples, guardados con el mismo formato y frecuencia de reproducción que la señal de salida de audio de la consola. Es decir: samples estéreo de 16 bits reproducidos a 44100 samples por segundo. Un sonido puede tener cualquier tamaño, desde 1 sample hasta el tamaño completo de la ROM de audio que lo contiene. Los samples de un sonido se numeran con posiciones empezando en 0. Todas las funciones de reproducción de la SPU necesitan un sonido para generar audio.

La SPU identifica y accede a sus sonidos disponibles a través de una matriz de pistas de sonido numeradas. Hay 1024 pistas utilizables por el cartucho (con IDs de sonido de 0 a 1023), y una pista adicional con ID = -1 para el sonido de la BIOS.

3.3 Canales de sonido

Un canal de sonido es un generador de audio básico al que se puede asignar un sonido de SPU y usará sus samples para crear una reproducción de salida. Los canales admiten los comandos básicos de control (reproducir, pausar/reanudar y detener). La SPU tiene una matriz de 16 canales de sonido, y cada uno se identifica con un ID numérico de 0 a 15.

Los canales de sonido usan un formato de reproducción idéntico al de la señal de salida de audio de la consola. Eso permite a la SPU mezclar la salida de todos los canales que haya reproduciendo en una única salida. Para cada nuevo frame, la SPU generará el número necesario de samples para ese tiempo y los almacenará en un buffer. Luego se enviará continuamente la secuencia de samples de audio de salida a los altavoces con la frecuencia correcta. Esto hará que todos los sonidos de los canales puedan oírse. En conjunto, el funcionamiento de la SPU con canales de sonido es el siguiente:



3.4 Conexión a ROMs de audio

La SPU en sí misma no contiene sonidos, así que necesita leer los sonidos guardados en la BIOS y los cartuchos conectándose a sus respectivas ROMs de audio. Una ROM de audio es una región de memoria de sólo lectura que contiene una secuencia de sonidos. Cada uno de estos sonidos puede tener cualquier tamaño posible, desde 1 sample hasta toda la ROM de audio. Las ROMs de audio tienen límites de tamaño: La SPU sólo puede

manejar ROMs de audio de tamaños de hasta 256 x 1024 x 1024 samples para cartuchos, y de hasta 1024 x 1024 samples para la BIOS.

Cuando se conecta un cartucho que contiene N sonidos, las N primeras pistas a partir de la ID = 0 se asignan a cada uno de los sonidos, por orden. El resto de pistas hasta la ID = 1023 quedarán sin usar y no serán accesibles. La pista con ID = -1 se asigna al sonido de la BIOS, que se garantiza que existe y es único.

Este proceso de conexión ocurre cada vez que se inserta un nuevo cartucho. Aún así, en sistemas Vircon32, los cartuchos sólo pueden insertarse con la consola apagada. Por lo tanto, en las implementaciones, los procedimientos necesarios para que la SPU establezca esta conexión se pueden retrasar hasta el siguiente encendido de la consola.

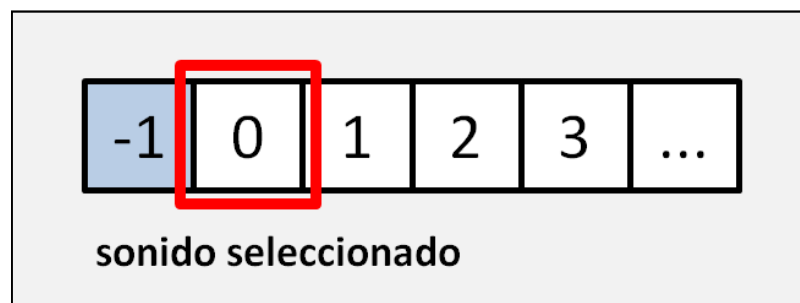
Depende de la implementación decidir cómo establecer conexión con las ROMs de audio, y localizar y leer sus sonidos. Por ejemplo, se puede leer todos los sonidos de las ROMs de audio de antemano al establecer la conexión. Otra opción sería mantener punteros a los sonidos y hacer que la SPU lea los valores de los samples sobre la marcha.

3.5 Elementos seleccionados

La SPU organiza sus elementos de audio (sonidos y canales) en conjuntos. La generación de sonido puede usar todos los canales y sonidos que necesite a la vez, pero otras funciones de la SPU operan con un solo elemento de cada conjunto en un momento dado.

Sonido seleccionado

Todas las pistas de sonido asignadas forman un rango continuo de IDs de sonido usables como se ve en esta imagen. Para decidir qué sonido usar al ejecutar comandos o aplicar variables internas, la SPU siempre considera uno de esos IDs como “seleccionado”. El sonido seleccionado por defecto es el de la BIOS, ya que es el único que se garantiza que siempre existe.



Canal seleccionado

Del mismo modo, los canales de sonido disponibles forman un rango de IDs de canal usables de 0 a 15. Para decidir qué canal usar al ejecutar comandos o aplicar variables internas, la SPU siempre considera uno de esos IDs como “seleccionado”. El canal seleccionado por defecto es el primero, con ID de canal = 0.

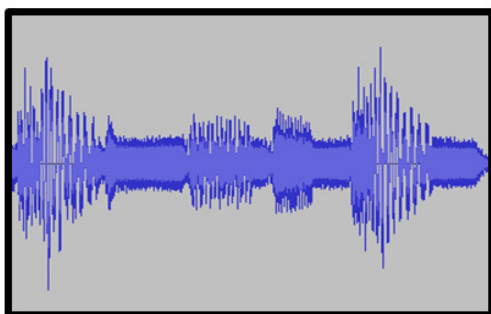
4 Efectos de audio

Para tener mayor flexibilidad y características sonoras más avanzadas, se pueden aplicar distintos efectos para configurar cada canal de sonido de la SPU. Estos efectos, cuando están activados, modifican la forma en que un canal reproduce su sonido asignado y, por tanto, cambian el resultado que se escucha por los altavoces.

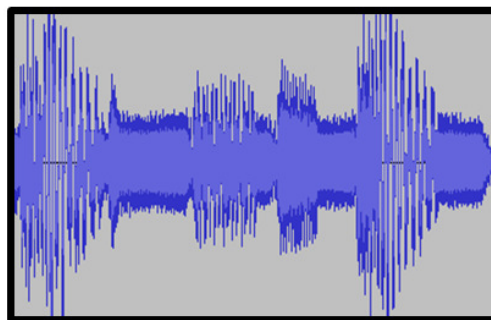
4.1 Efecto de volumen

Al reproducir un sonido, se puede configurar un canal de la SPU para cambiar el volumen de reproducción. El parámetro de volumen del canal actúa como un factor de escala lineal. Como resultado la onda de salida se escala. Un ejemplo podría ser el siguiente:

Sin modificar



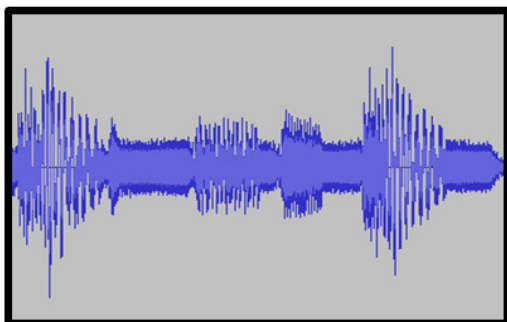
Volumen = 1.8



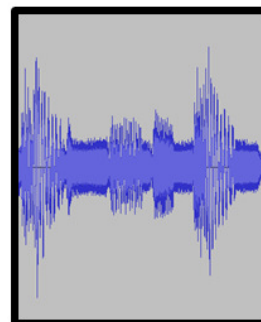
4.2 Efecto de cambio de velocidad

Otro parámetro configurable de los canales de la SPU es su velocidad de reproducción. El parámetro de velocidad de un canal se usará para determinar la velocidad de avance a lo largo de los samples de su sonido asignado, dando como resultado un cambio tanto en la velocidad como en el tono. Un ejemplo de velocidad de modificada sería el siguiente:

Sin modificar



Velocidad = 2.0



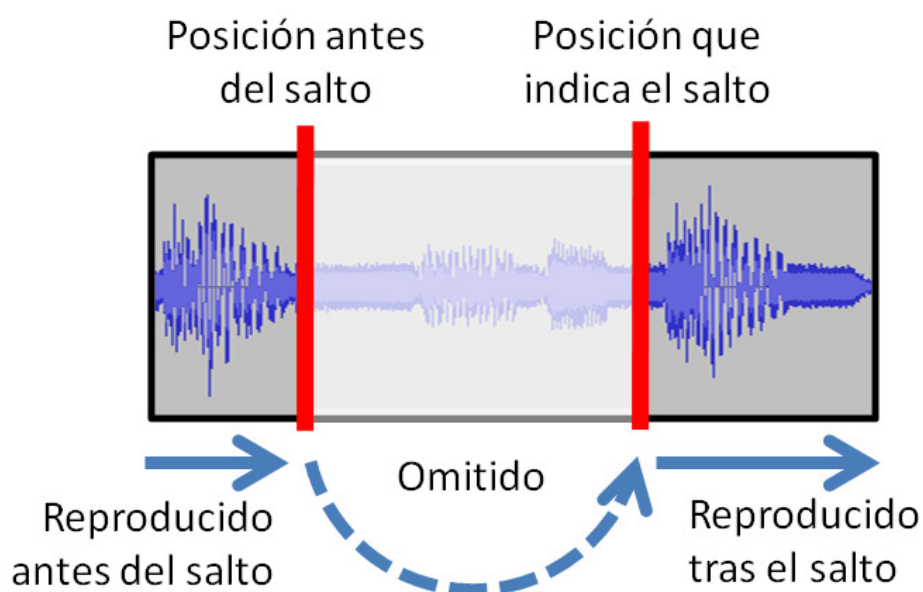
4.3 Reproducción en bucle

Activar este efecto hace que un canal haga un bucle sobre una región definida de su sonido asignado. El inicio y fin de la región en bucle son configurables para cada sonido de la SPU. Reproducir un sonido con bucle puede tener 3 etapas, como se muestra aquí:



4.4 Saltos de reproducción

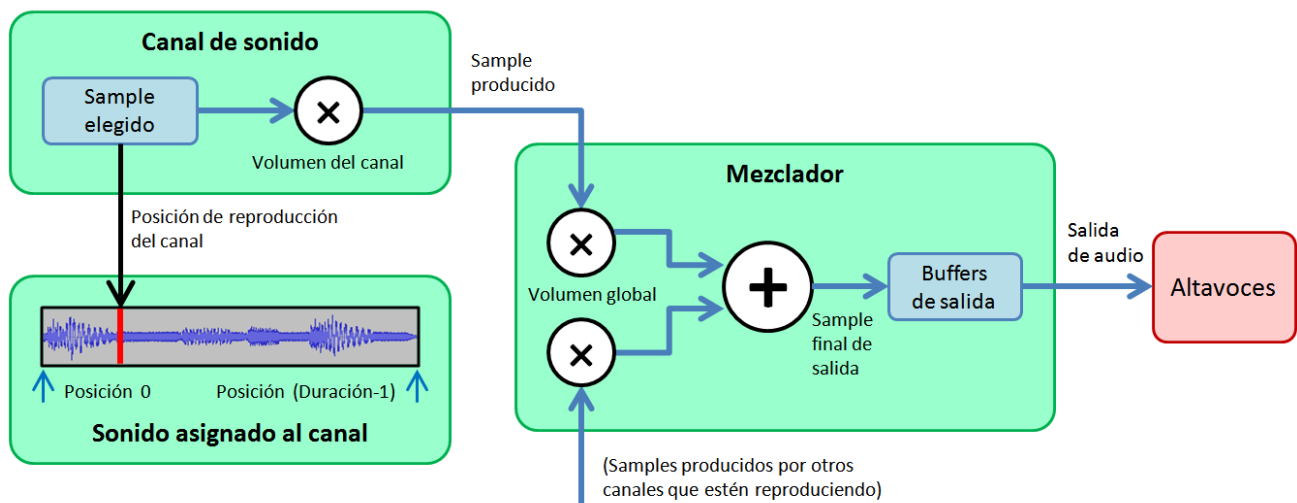
Los canales reproducen su sonido asignado siguiendo una posición de reproducción que avanza continuamente a lo largo del sonido. Los canales permiten modificar esta posición de reproducción con comandos del programa, dando lugar a saltos. Por ejemplo, para un salto hacia adelante, la reproducción de un sonido sucedería como en esta imagen.



5 Generación de sonido

La generación de sonido es el proceso usado para producir nuevos samples de audio de salida a partir de los sonidos SPU fuente. Para ello la SPU produce primero samples de salida separados de cada canal en reproducción, según determine su configuración. Luego esos samples de canales se mezclan en un único sample de salida global. Por último, se guarda una secuencia de samples de salida mezclados y se envían a los altavoces a través de la señal de audio, a la velocidad necesaria para reproducir a 44100 Hz.

El proceso global de generación de sonido puede resumirse en este diagrama:



5.1 Proceso general y temporización

El proceso de generación de sonido se activa al principio de cada frame. En cada ejecución este proceso producirá los samples de salida necesarios para 1 frame, es decir, una secuencia ordenada de $44100 / 60 = 735$ samples. Para ello, se repiten 735 veces los siguientes pasos:

1. Generar el sample actual para cada canal de sonido que esté reproduciendo.
2. Aplicar la lógica de mezclado para unir todos los samples de los canales en un único sample de salida.
3. Guardar ese sample en un buffer de salida, listo para ser enviado a reproducción.
4. Avanzar todos los canales que estén reproduciendo.

Cada uno de estos pasos se detalla en las secciones siguientes.

Una consecuencia importante de generar sonido sólo al principio de los frames es que el estado de los sonidos y canales entre esos instantes no es relevante para generar sonido. Por ejemplo: supongamos que en un frame un programa pone el canal 2 a volumen 0.5. Más tarde, en ese mismo frame, el volumen del canal 2 cambia a 0.8. En ese caso:

- Si la generación de sonido fuera continua, el canal 2 habría generado algunos samples de salida con volumen 0.5, y después otros samples con volumen 0.8.
- Sin embargo, como el sonido sólo se genera entre frames, el canal 2 generará todos los samples de ese frame usando el volumen final de 0.8.

Del mismo modo, si un canal se pone a reproducir y luego se para en el mismo frame, no generará ningún sonido en ese frame porque al llegar un nuevo frame estará detenido.

5.2 Generar samples de salida de los canales

Los canales que estén en estado de reproducción simplemente tomarán de su sonido asignado el sample correspondiente a su posición de reproducción actual. Ese sample se multiplicará por el valor actual del volumen de reproducción del canal y el valor resultante se tomará como sample de salida del canal.

Métodos de interpolación

Como las velocidades de reproducción pueden variar, las posiciones de reproducción de los canales no siempre serán enteras. Esto obliga a las implementaciones a adoptar alguna lógica para decidir cómo deben elegir los canales el sample que escogen.

El método de referencia para interpolación de sonido en la SPU de Vircon32 es el del “vecino más próximo”. Sin embargo, las implementaciones son libres de elegir otros métodos de interpolación para las posiciones de reproducción no enteras. Cualquier método es aceptable siempre que no distorsione los sonidos originales. Esto significa que las interpolaciones de audio basadas en suavizar en la forma de onda original, como la lineal o la cúbica, pueden considerarse conformes con Vircon32.

5.3 Mezclar samples de los canales

Las salidas producidas por todos los canales que se estén reproduciendo en ese momento se multiplicarán primero por el factor de volumen global de la SPU. Después se sumarán todos los valores resultantes. Este resultado se tomará como el sample de salida final.

Obsérvese que el orden de las operaciones afectará al resultado. El orden indicado aquí (sumar DESPUÉS de multiplicar) se elige para evitar la saturación de la señal de audio si se reproducen varios canales a la vez y el volumen global necesita modularlos a la baja.

5.4 Guardar los samples mezclados de salida

Para garantizar la sincronización correcta entre la generación y la salida de sonido se necesitará algún tipo de buffer de almacenamiento. Su función es retener los samples de salida generados hasta que llegue el momento de que la señal de salida de audio envíe cada sample individual a los altavoces. Las posibles estrategias para ello se explorarán en la sección 9.

5.5 Avanzar los canales tras cada sample

Una vez producido su sample de salida actual, el canal necesita que su posición de reproducción avance. Si la velocidad no fuera configurable, la posición del canal avanzaría 1 sample por cada sample de salida (misma velocidad de reproducción). Al incluir una velocidad que actúa como factor multiplicador, el avance tras cada sample de salida es:

$$\text{Posición de reproducción del canal} += \text{Velocidad de reproducción del canal}$$

Procesado de bucles

Después del avance el canal debe aplicar la lógica de bucle. Esto sólo se hará si el bucle del canal está activado, y sólo si la configuración del bucle del sonido es válida. Para que sea válida, su posición de fin de bucle debe ser mayor que la de su inicio de bucle.

La lógica de bucle necesita primero determinar si el último avance hizo que la posición de reproducción rebasara la posición final de bucle del sonido asignado. Cuando esto ocurra, la posición de reproducción “dará la vuelta” para volver al bucle. Pero, como la velocidad de reproducción puede ser mucho mayor que 1, el proceso debe tener en cuenta que:

- El final del bucle podría haberse rebasado por más de la propia longitud del bucle.
- La nueva posición correcta para un bucle continuo puede no ser el inicio del bucle.

En general, para garantizar un bucle correcto, la nueva posición debe establecerse así:

- 1) Duración bucle = Fin bucle - Inicio bucle + 1
- 2) Bucles completos pasados = floor((Posición repr. – Fin bucle) / Duración bucle)
- 3) Fragmento del bucle que sobra = Posición en bucle - (Duración bucle * Bucles completos pasados)
- 4) Posición repr. = Inicio bucle + Fragmento del bucle que sobra

Fin de la reproducción

Después de procesar el bucle si se necesita, el canal debe comprobar si la posición de reproducción es mayor que la posición del último de sample de su sonido asignado. En ese caso la reproducción del sonido ha terminado: el canal se detendrá y no generará más samples para el frame actual. Si una implementación necesita generar siempre 735 samples para un canal, puede hacerlo rellenando los samples restantes con ceros.

6 Variables internas

La SPU tiene un conjunto de variables que guardan distintos aspectos de su estado interno. Cada una se almacena como un valor de 32 bits, y todas se interpretan con los mismos formatos de datos (entero, float, etc.) descritos en la parte 2 de la especificación. La única excepción, como se verá luego, es la variable “Posición de canal”. A continuación enumeramos y detallamos todas las variables internas, organizadas en secciones.

6.1 Variables globales de la SPU

Volumen global	Valor inicial: 1.0
Formato: Float	Rango válido: De 0.0 a 2.0

Representa el nivel de volumen de salida actual de la SPU. Su valor actúa como un multiplicador lineal que aplica el mezclador a sus samples recibidos de todos los canales. En cada canal el multiplicador de volumen total es (Volumen global * Volumen de canal).

El uso más común del volumen global es evitar la saturación del audio al mezclar varios canales, usando valores menores que 1.

6.2 Elementos seleccionados

Sonido seleccionado	Valor inicial: -1
Formato: Entero	Rango válido: De -1 a 1023 (*)

Este valor es el ID numérico del sonido de SPU seleccionado actualmente. El sonido seleccionado es el que se usará en todas las operaciones relacionadas con sonidos. También es el sonido afectado por cualquier cambio de configuración de sonido.

(*) El límite superior viene dado por el cartucho conectado en ese momento. Si no hay cartucho, el sonido de la BIOS (ID = -1) es el único sonido seleccionable.

Canal seleccionado	Valor inicial: 0
Formato: Entero	Rango válido: De 0 a 15

Este valor es el ID numérico del canal de sonido de la SPU seleccionado actualmente. El canal seleccionado es el que se usará en todos los comandos de sonido de un sólo canal. También es el canal afectado por cualquier cambio de configuración de canal.

6.3 Configuración de cada sonido

Las variables listadas aquí son especiales. La SPU guarda una copia de cada una de estas variables por cada sonido existente en la SPU. Esto significa que puede haber hasta (1024+1) copias. La implementación puede decidir si todas ellas se guardan siempre, manteniendo inaccesibles las de las pistas de sonido no usadas, o si sólo se crean las necesarias cada vez que se inserta un nuevo cartucho.

Juntas, cada conjunto de estas variables describe la configuración actual para un único sonido de SPU (la sección 3 da una idea general sobre los sonidos de SPU). Como sólo un conjunto de estas variables es accesible en cada momento, a cada variable de esta sección se accede por puertos E/S a través de un “puntero”. Cuando cambia el sonido seleccionado esos puertos se redirigen a la copia de las variables para el nuevo sonido.

Duración de sonido	Valor inicial: (*)
Formato: Entero	Rango válido: De 1 a 268435456 (= 256*1024*1024)

Representa el número de samples de su sonido asociado.

(*): Su valor inicial ya está ajustado a la duración del sonido en samples cuando se conecta el cartucho o se inicia la consola. Nunca cambiará hasta que se retire el cartucho.

Bucle al reproducir sonido	Valor inicial: Falso
Formato: Booleano	Rango válido: Verdadero/Falso

Cuando este parámetro es verdadero, su sonido asociado está configurado para ser reproducido con bucle activado. Cuando un canal recibe un comando de reproducir, leerá esta variable de su sonido asignado y pondrá su variable “Bucle activado en canal” a ese mismo valor.

Inicio bucle de sonido	Valor inicial: 0
Formato: Entero	Rango válido: De 0 a 268435455 (*)

Marca la posición del primer sample incluido en el rango de bucle del sonido asociado. Si el bucle del canal está activado, cuando la reproducción sobrepase la posición final del bucle, la reproducción “dará la vuelta” hasta esta posición de inicio del bucle.

(*) Su valor máximo es la última posición de sonido, es decir, Duración de sonido – 1.

Fin bucle de sonido	Valor inicial: (*)
Formato: Entero	Rango válido: De 0 a 268435455 (**)

Marca la posición del último sample incluido en el rango de bucle del sonido asociado. Si el bucle del canal está activado, cuando la reproducción sobrepase esta posición final del bucle, la reproducción “dará la vuelta” hasta la posición de inicio del bucle.

(*) Su valor inicial es la última posición de sonido, es decir, Duración de sonido – 1.

(**) Su valor máximo también es la última posición de sonido, Duración de sonido – 1.

6.4 Configuración de cada canal de sonido

Estas variables funcionan de la misma forma que las de los sonidos: la SPU guarda una copia de cada una de estas variables por cada canal de sonido. Por tanto, hay 16 copias.

Juntas, cada conjunto de estas variables describe la configuración actual para un único canal de sonido (la sección 3 da una idea general sobre los canales de sonido de la SPU). Como sólo un conjunto de estas variables es accesible en cada momento, a cada variable

de esta sección se accede por puertos E/S a través de un “puntero”. Cuando el canal de sonido seleccionado cambia, todos esos puertos se redirigen a la copia de las variables para el nuevo canal.

Estado de canal	Valor inicial: 40h (Canal detenido)
Formato: Entero	Rango válido: Sólo los valores listados

Este valor se interpreta como el estado actual de reproducción de su canal de sonido asociado. Los valores posibles son los siguientes:

- 40h: Canal detenido
- 41h: Canal pausado
- 42h: Canal reproduciendo

Sonido asignado a canal	Valor inicial: -1
Formato: Entero	Rango válido: De -1 al último ID de sonido usado

Representa el ID del sonido actualmente asignado a su canal asociado. Si el canal recibe un comando de reproducir, el sonido se generará utilizando los samples del sonido de SPU asignado al canal.

Volumen de canal	Valor inicial: 1.0
Formato: Float	Rango válido: De 0.0 a 8.0

Representa el nivel de volumen de salida actual de su canal asociado. Su valor actúa como un multiplicador lineal aplicado a los samples de salida producidos por este canal.

Velocidad de canal	Valor inicial: 1.0
Formato: Float	Rango válido: De 0.0 a 128.0

Es la velocidad de reproducción actual de su canal de sonido asociado. Con velocidad 1.0 se reproduce sin cambios, y otros valores se interpretan como un multiplicador. Por ejemplo, una velocidad de 2.0 hace que la posición del canal avance 2 samples de sonido por cada sample de salida. Para un valor de 0.0 la reproducción no avanzará en absoluto.

Bucle activado en canal	Valor inicial: Falso
Formato: Booleano	Rango válido: Verdadero / Falso

Determina si la reproducción en bucle está activada actualmente para su canal de sonido asociado. Cuando sea verdadero, ese canal seguirá reproduciendo la región en bucle de su sonido asociado.

Posición de canal	Valor inicial: 0.0
Formato: Double	Rango válido: De 0.0 a (*)

Representa la posición de reproducción actual de su canal asociado. Se interpreta como una posición de sample dentro de su sonido asociado. Cuando este canal genere su siguiente sample de salida, tomará el sample de esta posición en su sonido asociado.

El formato de esta variable es una excepción: un double IEEE de 64 bits. Es así porque debe soportar un rango amplio de posiciones, y a la vez conservar suficiente precisión para valores no enteros. Aún así los puertos de control sólo manejan valores de 32 bits, por lo que las lecturas y escrituras manejarán la posición como un entero de 32 bits.

(*): El límite superior de esta variable es diferente para cada canal, y se actualiza cada vez que el sonido asociado cambia para ese canal. Ese límite superior se fijará en la posición del último sample del nuevo sonido asociado, es decir, su duración de sonido - 1.

7 Puertos de control

Esta sección detalla el conjunto de puertos de control expuestos por la SPU a través de su conexión como dispositivo esclavo al bus de control de la CPU. Todos los puertos expuestos y sus propiedades básicas se enumeran en la siguiente tabla:

Lista de puertos de control expuestos			
Dirección externa	Dirección interna	Nombre del puerto	Acceso L/E
300h	00h	Comando	Sólo Escritura
301h	01h	Volumen Global	Lectura y Escritura
302h	02h	Sonido Seleccionado	Lectura y Escritura
303h	03h	Canal Seleccionado	Lectura y Escritura
304h	04h	Duración de Sonido	Sólo Lectura
305h	05h	Bucle al Reproducir Sonido	Lectura y Escritura
306h	06h	Inicio Bucle de Sonido	Lectura y Escritura
307h	07h	Fin Bucle de Sonido	Lectura y Escritura
308h	08h	Estado de Canal	Sólo Lectura
309h	09h	Sonido Asignado a Canal	Lectura y Escritura
30Ah	0Ah	Volumen de Canal	Lectura y Escritura
30Bh	0Bh	Velocidad de Canal	Lectura y Escritura
30Ch	0Ch	Bucle Activado en Canal	Lectura y Escritura
30Dh	0Eh	Posición de Canal	Lectura y Escritura

7.1 Acciones en peticiones de lectura/escritura de puertos

Los puertos de control de la SPU no son simples registros hardware. Los efectos causados por una petición de lectura/escritura a un puerto concreto pueden ser distintos de lectura o escritura de valores. Esta sección detalla cómo se comporta cada puerto de la SPU.

Obsérvese que, además de las acciones realizadas, será necesario dar una respuesta de éxito/fallo a la petición, como parte de la comunicación del bus de control. Si no se indica lo contrario, siempre se asumirá que la respuesta es de éxito. Cuando la respuesta sea de fallo, la SPU no realizará más acciones y la CPU activará un error hardware.

Puerto “Comando”

En peticiones de **lectura**:

Como este puerto es de sólo escritura, se dará una respuesta de fallo al bus de control.

En peticiones de **escritura**:

La SPU realizará el proceso de ejecución de comando detallado en la sección 8. En todos los casos se dará respuesta de éxito a la petición, aún si el comando no se reconoce.

Puerto “Volumen Global”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Volumen global”.

En peticiones de **escritura**:

La SPU comprobará si el valor recibido está fuera del rango de la variable interna “Volumen global” y, si lo está, lo ajustará a los límites del rango. El valor resultante sobrescribirá la variable interna “Volumen global”. Esto hará que la siguiente generación de sonido aplique inmediatamente el nuevo volumen global.

Puerto “Sonido Seleccionado”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Sonido seleccionado”.

En peticiones de **escritura**:

La SPU comprobará si el valor recibido corresponde a un ID de sonido actualmente válido. Si no lo es la petición será ignorada. Para valores válidos, la SPU sobrescribirá la variable

interna “Sonido seleccionado” con el valor recibido. Después redirigirá todos los puertos de configuración de sonido para apuntar a las variables del nuevo sonido seleccionado.

Puerto “Canal Seleccionado”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Canal seleccionado”.

En peticiones de **escritura**:

La SPU comprobará si el valor recibido corresponde a un ID de canal válido. Si no lo es la petición será ignorada. Para valores válidos, la SPU sobrescribirá la variable interna “Canal seleccionado” con el valor recibido. Después redirigirá todos los puertos de configuración de canal para apuntar a las variables del nuevo canal seleccionado.

Puerto “Duración de Sonido”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Duración de sonido” asociada al ID de sonido actualmente seleccionado.

En peticiones de **escritura**:

Como este puerto es de sólo lectura, se dará una respuesta de fallo al bus de control.

Puerto “Bucle al Reproducir Sonido”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Bucle al reproducir sonido” asociada al ID de sonido actualmente seleccionado.

En peticiones de **escritura**:

La SPU usará el valor recibido para sobrescribir la variable interna “Bucle al reproducir sonido” asociada al ID de sonido seleccionado actualmente. Esto hará que la siguiente generación de sonido aplique inmediatamente la nueva configuración de bucle del sonido.

Puerto “Inicio Bucle de Sonido”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Inicio bucle de sonido” asociada al ID de sonido actualmente seleccionado.

En peticiones de escritura:

La SPU comprobará si el valor recibido está fuera del rango de la variable interna “Inicio bucle de sonido” para el ID de sonido seleccionado actualmente y, si lo está, lo ajustará a los límites del rango. El valor resultante sobrescribirá la variable interna “Inicio bucle de sonido” asociada a este sonido. Esto hará que la siguiente generación de sonido aplique inmediatamente el nuevo inicio de bucle del sonido.

Puerto “Fin Bucle de Sonido”

En peticiones de lectura:

La SPU proveerá el valor actual de la variable interna “Fin bucle de sonido” asociada al ID de sonido actualmente seleccionado.

En peticiones de escritura:

La SPU comprobará si el valor recibido está fuera del rango de la variable interna “Fin bucle de sonido” para el ID de sonido seleccionado actualmente y, si lo está, lo ajustará a los límites del rango. El valor resultante sobrescribirá la variable interna “Fin bucle de sonido” asociada a este sonido. Esto hará que la siguiente generación de sonido aplique inmediatamente el nuevo final de bucle del sonido.

Puerto “Estado de Canal”

En peticiones de lectura:

La SPU proveerá el valor actual de la variable interna “Estado de canal” asociada al ID de canal actualmente seleccionado.

En peticiones de escritura:

Como este puerto es de sólo lectura, se dará una respuesta de fallo al bus de control.

Puerto “Sonido Asignado a Canal”

En peticiones de lectura:

La SPU proveerá el valor actual de la variable interna “Sonido asociado a canal” asociada al ID de canal actualmente seleccionado.

En peticiones de escritura:

La SPU comprobará primero el estado del ID de canal seleccionado, y si no es Detenido la petición será ignorada. Si el valor recibido no es un ID de sonido válido, la petición también se ignorará. En ambos casos no se realizará ningún otro procesado.

Si se procesa la petición, la SPU usará el valor recibido para sobrescribir la variable interna “Sonido asignado a canal”. Esto hará que la siguiente generación de sonido aplique inmediatamente el nuevo sonido asignado para el canal seleccionado.

Después la SPU hará este procesamiento para la variable “Posición de canal” del canal:

- Pondrá su valor a 0.
- Fijará su límite superior al número de samples del nuevo sonido – 1.

Puerto “Volumen de Canal”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Volumen de canal” asociada al ID de canal actualmente seleccionado.

En peticiones de **escritura**:

La SPU comprobará si el valor recibido está fuera del rango de la variable interna “Volumen de canal” y, si lo está, lo ajustará a los límites del rango. El valor resultante sobrescribirá la variable interna “Volumen global” para el ID de canal seleccionado actualmente. Esto hará que la siguiente generación de sonido aplique inmediatamente el nuevo volumen para el canal seleccionado.

Puerto “Velocidad de Canal”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Velocidad de canal” asociada al ID de canal actualmente seleccionado.

En peticiones de **escritura**:

La SPU comprobará si el valor recibido está fuera del rango de la variable interna “Velocidad de canal” y, si lo está, lo ajustará a los límites del rango. El valor resultante sobrescribirá la variable interna “Velocidad global” para el ID de canal seleccionado actualmente. Esto hará que la siguiente generación de sonido aplique inmediatamente la nueva velocidad para el canal seleccionado.

Puerto “Bucle Activado en Canal”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Bucle activado en canal” asociada al ID de canal actualmente seleccionado.

En peticiones de **escritura**:

La SPU usará el valor recibido para sobrescribir la variable interna “Bucle activado en canal” asociada al ID de canal seleccionado actualmente. Esto hará que la siguiente generación de sonido aplique inmediatamente la nueva configuración de bucle para el canal seleccionado.

Puerto “Posición de Canal”

En peticiones de **lectura**:

La SPU proveerá el valor actual de la variable interna “Posición de canal” asociada al ID de canal actualmente seleccionado. Como esta variable es un double, el valor provisto será el resultado de truncar ese valor a un entero de 32 bits.

En peticiones de **escritura**:

La SPU comprobará el número de samples para el sonido de la SPU asociado actualmente al ID de canal seleccionado. El rango permitido para esta operación de escritura se determinará como: [de 0 a número de samples - 1], ambos incluidos. Si el valor entero de 32 bits recibido está fuera del rango, se ajustará a los límites del rango. El valor resultante se convertirá a double y se usará para sobrescribir la variable interna “Posición del canal” asociada al ID de canal seleccionado actualmente. Esto hará que la siguiente generación de sonido aplique inmediatamente la nueva posición de reproducción para el canal seleccionado.

8 Ejecución de comandos

Los comandos de la SPU son instrucciones de control de reproducción de canales. Los puede solicitar la CPU enviando un valor al puerto “Comando” de la SPU. Esta sección describe el comportamiento de la SPU cuando recibe una petición de este tipo.

Hay 6 comandos distintos que la SPU puede ejecutar. Los 3 primeros operan sobre un único canal de sonido, mientras que los otros 3 afectan a todos los canales. Esta tabla muestra todos los comandos de la SPU junto con sus valores numéricos.

Nombre del Comando	Valor Numérico
Reproducir Canal Seleccionado	30h
Pausar Canal Seleccionado	31h
Detener Canal Seleccionado	32h
Pausar Todos los Canales	33h
Reanudar Todos los Canales	34h
Detener Todos los Canales	35h

8.1 Procesado común

El funcionamiento general descrito aquí es común a la ejecución de todos los comandos.

Como primer paso, la SPU comprobará si el valor que se pide escribir corresponde a uno de los comandos válidos enumerados antes. Si no es así la petición se ignorará y se detendrá su procesado.

8.2 Comando “Reproducir Canal Seleccionado”

La SPU pondrá el canal de sonido seleccionado en estado Reproduciendo. Si el estado anterior del canal era Detenido se hará lo siguiente para iniciar una nueva reproducción:

1. La posición del canal se pondrá a 0.
2. Su variable "Bucle activado en canal" se ajustará al mismo valor que la variable "Bucle al reproducir sonido" del sonido asociado. Esto aplicará automáticamente la configuración de bucle del sonido en el canal usado para reproducirlo.

Si el estado anterior del canal ya era Reproduciendo también se aplicarán esas 2 acciones, provocando un reinicio de la reproducción del sonido. Para un canal previamente en estado Pausado la posición no se alterará, para reanudar la reproducción del sonido.

8.3 Comando “Pausar Canal Seleccionado”

Si el canal de sonido actualmente está en estado Reproduciendo, la SPU cambiará su estado a Pausado. No se hará ninguna acción para otros estados anteriores.

8.4 Comando “Detener Canal Seleccionado”

La SPU pondrá el canal de sonido actualmente seleccionado en estado Detenido. Detener un canal ya detenido no tiene ningún efecto.

8.5 Comando “Pausar Todos los Canales”

La SPU aplicará el mismo procesado descrito para el comando “Pausar Canal Seleccionado”, pero aplicará las acciones a todos los canales en estado Reproduciendo.

8.6 Comando “Reanudar Todos los Canales”

La SPU aplicará el mismo procesado descrito para el comando “Reproducir Canal Seleccionado”, pero las acciones se aplicarán a todos los canales en estado Pausado.

8.7 Comando “Detener Todos los Canales”

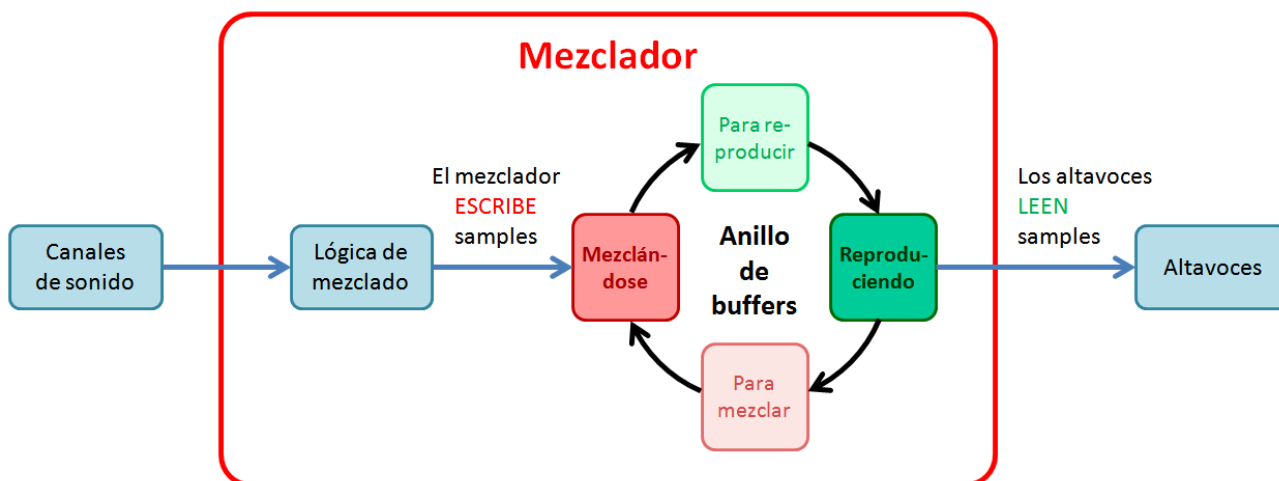
La SPU aplicará el mismo procesamiento descrito para el comando “Detener Canal Seleccionado”, pero las acciones se aplicarán a todos los canales.

9 Generación de la señal de audio

Como se vio en la sección 5, la SPU de Vircon32 genera audio frame a frame, gobernado por la temporización global de la consola a 60 Hz. Sin embargo, la reproducción en tarjetas de sonido y altavoces suele requerir que cada sample se envíe por separado a los altavoces en el momento correcto, a la velocidad de reproducción de 44100 Hz.

Para cubrir el lapso de tiempo entre la generación y la salida del sonido, los samples que se generan en cada frame tendrán que almacenarse en uno o más “buffers de espera” hasta que llegue el momento adecuado para que cada sample de SPU individual se envíe para su reproducción. Puede haber muchas estrategias posibles para esto, dependiendo de la implementación del sistema de audio y su formato para la señal de salida.

Una de las estrategias posibles, común en sistemas de audio por software, se llama "anillo de buffers". Consiste en tener varios buffers de salida, que cíclicamente se reproducen y rellenan con nuevos samples. Para asegurar un funcionamiento correcto, cada buffer debería guardar samples para 1 frame. Es decir, el tamaño del buffer será de $44100 / 60 = 735$ frames. La imagen muestra cómo podría implementarse un anillo de 4 buffers:



9.1 Buffers de sonido y latencia

En una implementación real la temporización global de 60 Hz de la consola podría no ser precisa. Así, estrategias como este anillo deberán usar los buffers garantizando que:

1. Los altavoces nunca se quedan sin samples que reproducir.
2. El mezclador siempre tiene un buffer libre para generar nuevo sonido.

En general, las estrategias con buffers de audio siempre se enfrentarán a un compromiso. Por un lado, usando más buffers se puede seguir cumpliendo estas 2 condiciones incluso con tiempos más imprecisos. Pero usar más buffers aumenta la latencia del sonido, ya que habrá colas de reproducción más largas.

9.2 Señal de salida de audio

Según el tipo de implementación, la señal de salida de audio tratará el estéreo de distinta forma. Algunas separan sus 2 canales, es decir: dividen los samples de 32 bits de la SPU en 2 secuencias de samples de 16 bits separados para la izquierda y la derecha. Otros sistemas usan una secuencia conjunta con samples identificados para cada altavoz.

Después la SPU enviará esa información a los altavoces a través de la señal de salida de audio. Si el formato/protocolo de la señal lo necesita, se añadirá información de tiempos o secuenciación para formar una transmisión de samples de audio válida.

Las implementaciones son libres de elegir el formato de comunicación y los conectores físicos para las señales de audio y vídeo. Pueden enviarse por separado o, como es habitual en la mayoría de sistemas actuales, usar un conector conjunto que envíe ambas a la pantalla.

10 Respuestas a señales de control

Como todos los componentes de la consola, cada vez que se produzca una señal de control, la SPU la recibirá y reaccionará para procesar ese evento. Para cada una de las señales de control, la SPU responderá realizando las siguientes acciones:

Señal de Reset:

- Si había algún sonido aún en reproducción, se detienen.
- Todas las variables internas de la SPU se establecen a sus valores iniciales. Esto incluye las variables de configuración para todos los sonidos y canales.
- Cualquier efecto adicional asociado a esos cambios en las variables internas se aplica inmediatamente, como se indica en las peticiones de escritura en puertos de control.
- Todos los samples en los buffers de salida se borran a 0.

Señal de Frame:

- La SPU genera salida de audio para 1 nuevo frame, como describe la sección 5.

Señal de Ciclo:

- La SPU no necesita reaccionar a esta señal, a menos que se requiera por detalles concretos de la implementación.

Además de reaccionar a las señales de control, la SPU también tendrá que realizar el siguiente procesamiento en respuesta a eventos a nivel de la consola:

Cuando se conecta un nuevo cartucho:

- La SPU localizará los sonidos contenidos en la BIOS y, si está presente, el cartucho.
- La SPU realizará el proceso de conexión descrito en la sección 3 para asignar una pista a cada sonido y acceder a la información de sus samples.
- El límite superior de la variable “Sonido seleccionado” se ajustará al ID de la última pista de sonido utilizada.
- Para cada pista de sonido usada, el valor de su variable “Duración de sonido” se ajustará al número de samples de su sonido asignado.
- Para cada pista de sonido usada, el límite superior de sus variables “Inicio bucle de sonido” y “Fin bucle de sonido” se ajustarán al número de samples de su sonido asignado – 1.

(Fin de la parte 5)