PlayStation

Software Development Seminar

Sound (Advanced)

Contents

- libsnd tips
- libspu tips
- Common to libsnd and libspu
- SPU streaming overview
- Synchronizing with other functions

libsnd tips

Compressing transfer processing and SEQ setting

```
SsVabTransBody (...);
SsVabTransCompleted (SS_WAIT_COMPLETED);
SsSeqOpen(...);
```

↓

```
SsVabTransBody (...);
SsSeqOpen(...);
SsVabTransCompleted (SS_WAIT_COMPLETED);
```

libsnd tips

Types of functions related to processing for each tick

- Functions which operate by calling SsUtFlush()

SsSetAutoKeyOffMode,
SsSetMono, SsSetStereo,
SsSetNoiseOn, SsSetNoiseOff,
SsUtAutoPan, SsUtAutoVol,
SsUtChangeADSR, SsUtChangePitch
SsUtKeyOn, SsUtKeyOnV
SsUtKeyOff, SsUtKeyOffV
SsUtPitchBend, SsUtSetDetVVol
SsUtSetVVol,
SsVoKeyOn, SsVoKeyOff,

libsnd tips

Types of functions related to processing for each tick

- Functions which do not operate unless SsSeqCalledTbyT() is called

SsIsEos, SsPlayBack,
SsSepOpen, SsSepClose,
SsSepPause, SsSepPlay,
SsSepReplay, SsSepSetAccelerando,
SsSepSetCrescendo, SsSepSetDecrescendo,
SsSepSetRitardando, SsSepSetVol,
SsSepStop, SsSeqClose,
SsSeqOpen, SsSeqPause,
SsSeqPlay, SsSeqReplay,
SsSeqSetAccelerando, SsSeqSetCrescendo,
SsSeqSetDecrescendo, SsSeqSetNext,
SsSeqSetRitardando, SsSeqSetVol,
SsSeqStop, SsSetLoop,
SsSetMarkCallback, SsSetNext,
SsSetTempo

libsnd tips

Types of functions related to processing for each tick

- Functions which are processed immediately even if SsUtFlush() or SsSeqCalledTbyT() are not called

All remaining functions

libsnd tips

VH information editing

Use SsUtGetVagAddr(), SsUtGetVagAtr(), and SsUtSetVagAtr() to edit VH information

libspu tips

Bundling key ON/OFF

```
SpuSetKey (SPU_ON, SPU_00CH);
SpuSetKey (SPU_ON, SPU_01CH);
SpuSetKey (SPU_ON, SPU_02CH);
```

↓

```
SpuSetKey (SPU_ON, SPU_00CH |
                   SPU_01CH |
                   SPU_02CH);
```

libspu tips

Use in VB libspu

Convert VABSPLIT.EXE with -v option

→ Output to .h file at the relative address corresponding to the location of VAG in VAB

Use the SpuSetVoiceAttr() attribute to specify the absolute address with the SpuMalloc() return value added

Common to libsnd and libspu

Initialization procedure

- Calling the SpuInit() in SsInit()

→ It isn't necessary to call SpuInit()

- SpuInitMalloc()

Set so that 32 blocks can be reserved in SsInit()

→ If the total exceeds 32 when using VAB with SpuMalloc(), it is necessary to have SpuInitMalloc() provide a control table region greater than 32 immediately after SsInit().
(The control table must be reserved in the user region.)

Common to libsnd and libspu

Voice division

Divide voices in each library using SsSetReservedVoice()

→ Restrain voice manager intervention into voices used by libspu

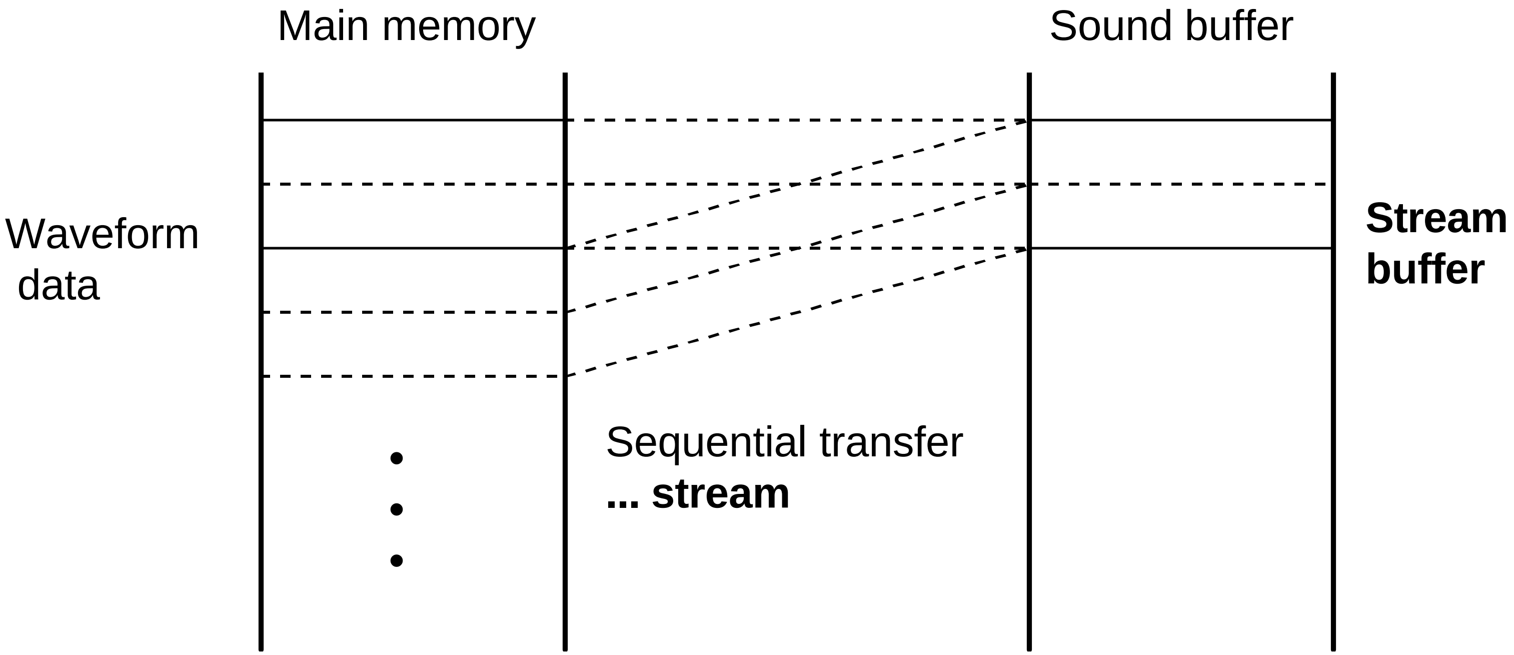
Reverb

libsnd uses libspu reverb functions as-is

→ Whichever is set is OK

SPU streaming

Use SPU to play large-size waveform data which can't be held in the sound buffer



Even if waveform data in main memory is continuous in units the size of the stream buffer, overall it can still be discontinuous

SPU streaming

Waveform data

VB file

Waveform data size

If you want to loop it, [use] an integral multiple
of 1/2 the stream buffer

Consuming one voice for one stream

→ Maximum 24 streams

Processing the stream destructively

Streaming can be used in multiple ways as data

SPU streaming

Basic operation

Pre-transfer

Prepare data 1/2 the size of the stream buffer

... timing is optional

Transfer

Sequential transfer of data 1/2 the size of the stream buffer

... Makes sequential transfer necessary

Transfer completion

By adding completion attributes when transferring, complete the "after next transfer is complete" stream

SPU streaming

Library-side requests in streaming processing

→ Callback function

- Pre-transfer completion callback
- Transfer completion callback
- Stream completion callback

SPU streaming

Operating modes

- Idle Status

  State in which the stream is not being processed

  → Load on CPU is 0

- Pre-transfer Status

  States during and after pre-transfer processing

  → The transition from in-processing to after-processing can be detected by pre-transfer completion callback

SPU streaming

Operating modes

- Transfer status

  State in which streaming is operating

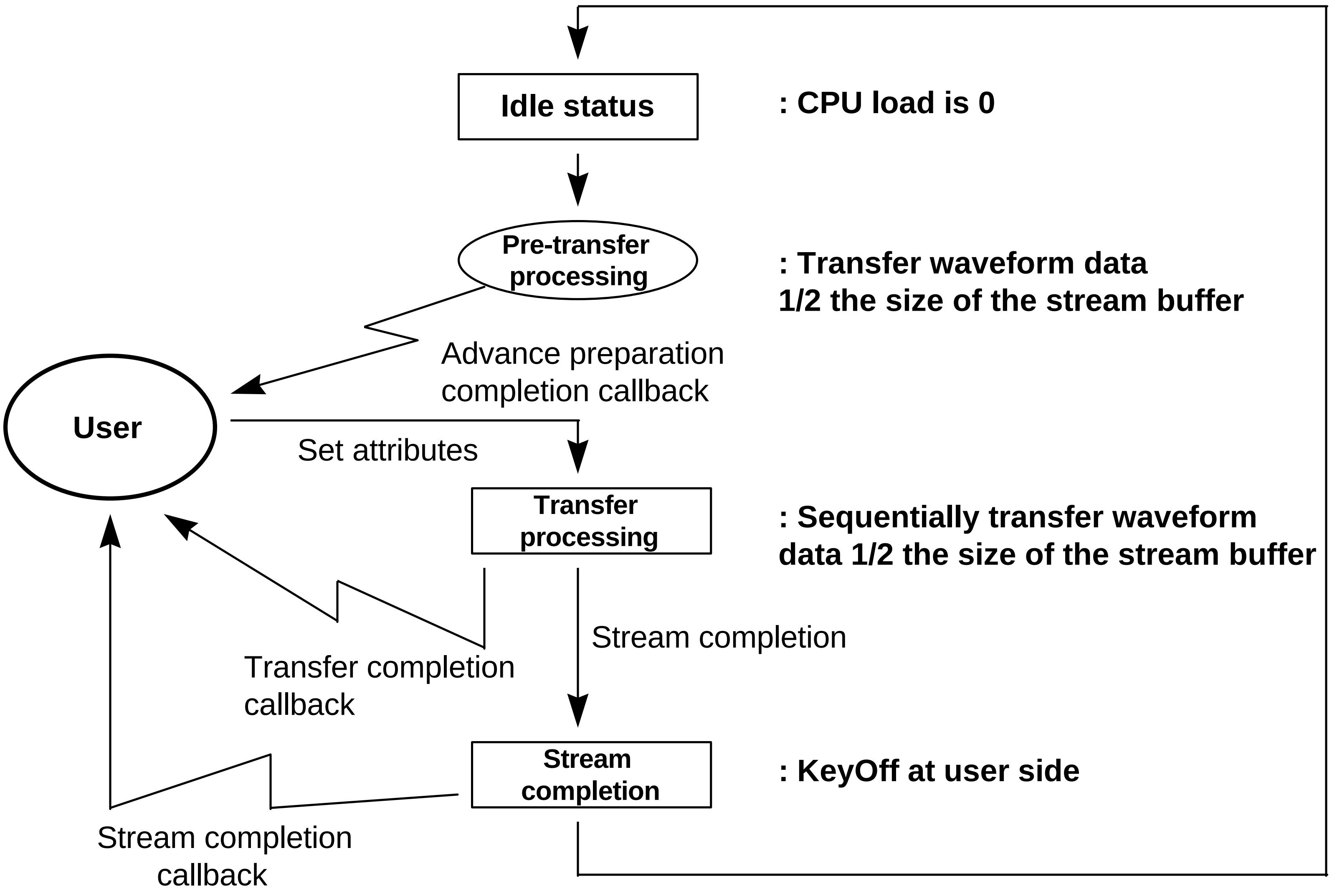
  → Setting attributes for the next transfer is done by transfer completion callback

- End of operation status

  State in which completion processing of all operating streaming is performed. Status is changed to idle

  → The transition from end of operation status to idle status can be detected by streaming completion callback

SPU streaming operation overview


Idle status
: CPU load is 0
Pre-transfer processing
: Transfer waveform data 1/2 the size of the stream buffer
Advance preparation completion callback
User
Set attributes
Transfer processing
: Sequentially transfer waveform data 1/2 the size of the stream buffer
Stream completion
Transfer completion callback
Stream completion
: KeyOff at user side
Stream completion callback

Synchronizing with other functions

In almost all cases, sound is dominant and other processing is subordinate

→ It is necessary to know the status of all sounds

- Know the envelope status
- Know the play location using SPU IRQ
- Check status using SpuReadDecodedData()
- Use of SPU streaming

Synchronizing with other functions - Knowing the status of sound

- Decide by getting the envelope status

Summary: SpuGetKeyStatus(), SpuGetAllKeysStatus()

Details: SpuGetVoiceAttr() +
SpuVoiceAttr() structure member envx

Synchronizing with other functions - Knowing the status of sound

- Know the play location using SPU IRQ

  - Can check the detailed settings and speed of response to some extent

  - Can know only one voice

Synchronizing with other functions - Knowing the status of sound

- Check status using SpuReadDecodedData()

  - Know status of voice 1 and voice 3

  - Overhead transfers done in background

Synchronizing with other functions - Knowing the status of sound

- Use of SPU streaming
  - Synchronize with other functions by timing each callback's processing
  - Possibility of time lag due to block size

Synchronizing with other functions

Situations in which sound is subordinate and other processing is dominant

- Sync with CD streaming, etc.

  ex: Method of synchronizing with frame

  - Process sound using value of frame counter

- TOD animation

  ex: Synchronize with animation using user-defined packet

  - Routines that interpret TOD animation are developed as TOD sample sources