

# *PlayStation Musicians' Festival*



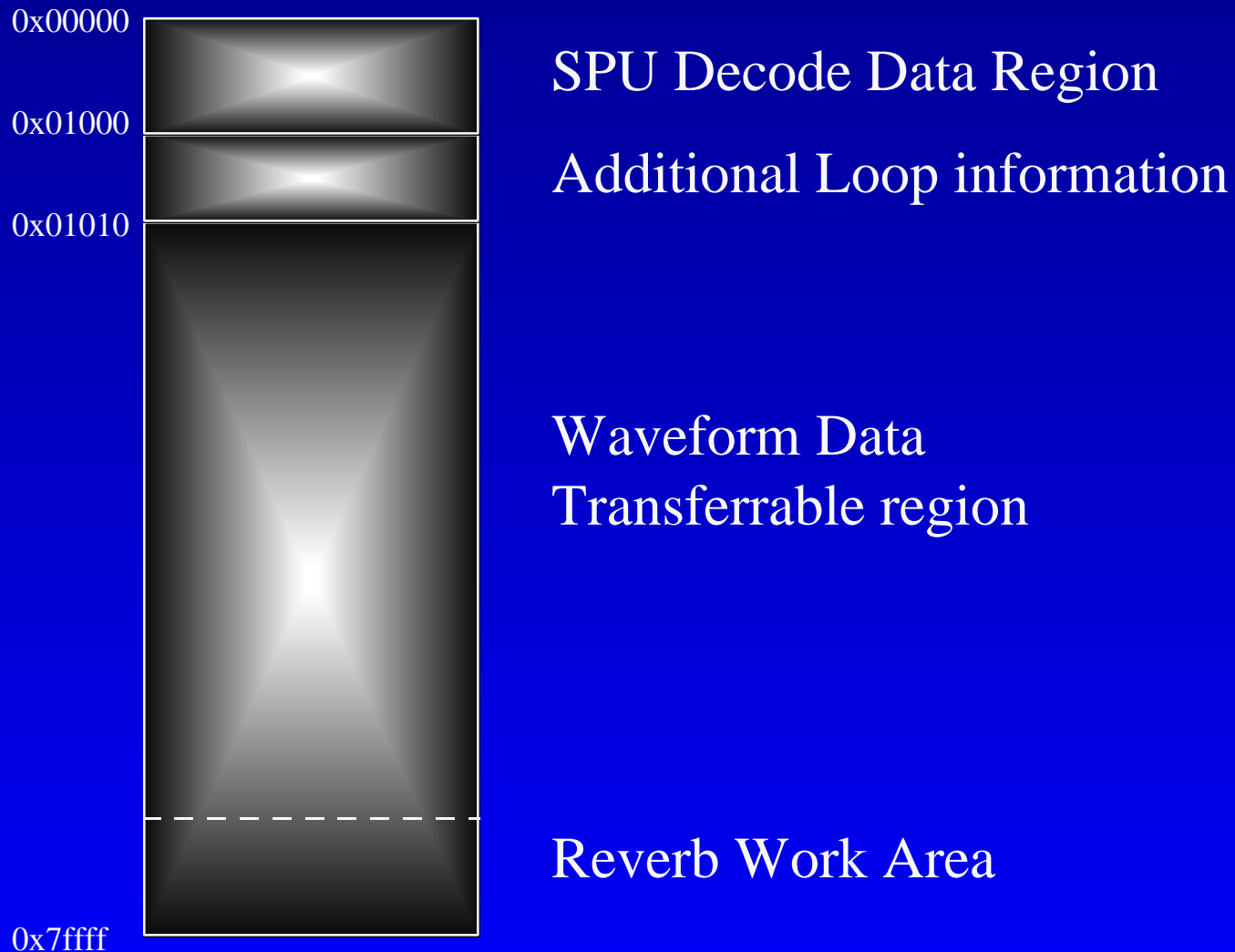
# *Maximizing voice usage*

- ▶ PlayStation voice allocation
  - First unused voice allocated
  - If all 24 voices busy, the oldest voice with the smallest envelope and the lowest priority (if lower than requested tone) is keyed off and allocated to new request

# *Maximizing voice usage*

- ▶ *Priority* in tone window
  - Set to 127 for most important tones - ensures that these will always be keyed on
  - Set to 0 for least important sounds
- ▶ What if you have 25 sounds which are priority 127?
  - Not available for MIDI; only for sound effects
  - Inform programmer and ask if `SsUtKeyOnV()` may be used

# SPU RAM map



# *Maximizing SPU RAM*

- ▶ 520,175 usable bytes - area allocated for reverb
- ▶ VAG format overview
- ▶ Reduce VAG size
  - 1) AIFF2VAG supports sampling rates as low as 5KHz
  - 2) Stripping VAGs of unneeded components

# *Maximizing SPU RAM*

## ▶ VAG format overview

- VAGs made up of 16 byte blocks
- 1st block always all zero data to avoid noise
- Proprietary format data blocks follow
- Compression ratio about 3.5 to 1 compared to AIFFs
- One-shot VAGs end with an “SPU IRQ Clear Block”
- Looping VAGs do not contain this block

# *Maximizing SPU RAM*

- ▶ Ask your programmer if they will be using any of the SPU IRQ functions
  - If not, can strip the last block off the VAG
  - Will work on Mac tool for stripping the block
  - Will save 16 bytes per VAG - up to 4064 bytes per VAB!

# *Maximizing SPU RAM*

## ▶ SPU Streaming

- Uses small SPU RAM buffer
- Playback of beginning of VAG started
- As playback continues, old data overwritten and playback of new data continues



# *Maximizing SPU RAM*

- ▶ Spu streaming perfect for things like sports commentators
  - Talk programmer out of using MIDI for speech
    - Excessive memory usage
  - Talk programmer out of using DA for speech
    - Longer seek times than VAG data (since DA files longer)
    - Disk space issues
  - XA also a possibility

# *Maximizing SPU RAM*

- ▶ Noise generator
  - Random noise generator takes up no SPU RAM
  - Outputs white noise

# *3D Sound*

- PlayStation plays only mono waveforms
- Need two tones for stereo
- More than likely need 2 VAGs for 3D sound
  - Use sparingly on most important sounds only

# *Cd audio*

- ▶ 2 types
  - XA and DA
- ▶ Why is it cool?
  - Does not enter main RAM or SPU RAM
  - Does not use a PlayStation voice
  - Can be stereo

# *Xa Audio*

- ▶ All channels of XA are stripped off together
- ▶ Can change channels during playback
  - Can make XA track adaptive to player performance
  - Can make XA track variable to defeat boredom
  - Must make a few natural gaps with very few instruments sounding
- ▶ Slightly reduced seek time over DA audio

# *DA audio*

- ▶ Best quality
- ▶ Least adaptable
- ▶ Longest seek times
- ▶ Most disk space usage

# Reverb

- ▶ Fixed format - not programmable
  - 9 types
  - Only one type may be applied at any given time
  - Can set reverb on/off for individual voices\CD input
- ▶ Designing with Sound Deli
  - Only tones with *mode* set to “04” will have reverb
  - Can also set on ADSR preview window

# *Reverb MIDI limitations*

- ▶ NRPN data 2 (“mode”) does not function properly
  - Does change reverb mode of individual tones
  - Also turns reverb on or off for all voices



# *Reverb MIDI limitations*

- ▶ Create two sets of tones - 1 with reverb, 1 without
  - Bloats VAB header size
- ▶ OR place a callback marker which alters the tone's *mode*
  - Adds CPU cycles
- ▶ Even with these workarounds cannot change reverb of currently sounding tone

# *Shifting VAG loop startpoint*

- ▶ Unlike other platforms, VAGs can only be set to loop during creation process
- ▶ Loop startpoints of looping VAGs can be shifted during playback
- ▶ Give your programmer the following info
  - Total samples contained in AIFF
  - Sample number of desired loop startpoint

# *Shifting VAG loop startpoint*

- ▶ Programmer info cont.
  - Sample number of original loop startpoint if:
    - The new startpoint is before the original startpoint OR
    - The original startpoint will be used again during playback

# *Minimizing VAB size*

- ▶ VAB bodies (VAG data) are transferred to SPU RAM
- ▶ VAB headers are kept on main RAM
- ▶ How big are VAB headers?
  - Fixed size for master VAB info, program info, and VAG size info, regardless of how many programs, tones, or VAGs are used
  - Variable size for tone info
    - Fixed size (16 tones) for each program used

# *Minimizing VAB size*

- ▶ Reduce tone portion of VAB header by reducing programs used
  - Combine sound effects into single programs
    - Must have same pan and volume values at program level
    - Single-tone sound effects can have the same note range as other tones
    - Multi-tone sound effects should have different note ranges for programming ease

# *Minimizing VAB size*

- ▶ Reduce tone portion of VAB header cont.
  - Can also combine MIDI instruments
    - Instruments which need to key on at different times must have different note ranges
    - Must have same pan value and same volume value at program level

# *Pitch calculations*

- ▶ Center value actually unimportant
- ▶ Pitch calculation
  - $60 + \text{keyon\_note} - \text{tone\_center}$
  - Adjusted for the difference in `keyon_fine` and `tone_shift` accordingly
- ▶ Note range for tones can be expanded

# *Volume calculations*

- ▶ Input volume adjusted downward
  - Master VAB volume as % of max
  - Program volume as as % of max
  - Tone volume as as % of max
- ▶ Pan reduction to side away from pan effect
  - Program pan
  - Tone pan
  - Input pan



# *Volume calculations*

- ▶ Output as two volumes
  - voll and volr
- ▶ Exponential reduction
  - $lvol = voll * voll / 127$
  - $rvol = volr * volr / 127$

# *Master VAB pan*

- ▶ Used only in Sound Deli
- ▶ Substitutes for SEQ volume in order to test sounds without a sequence

# *Reducing SEQ size*

- ▶ SEQ data must reside on main memory
- ▶ Remove aftertouch
  - PlayStation doesn't deal well with aftertouch anyway

# *MIDI channel muting*

- ▶ `SsChannelMute()`
  - Blocks key on commands only
  - Key off commands, tempo changes, bank changes still processed
  - Allows for much more song variability
- ▶ Map each instrument to a separate channel
  - Allows for main instruments to always play
  - Can mute/unmute detail instruments adaptively

# *MIDI channel muting*

- ▶ Change SEQs during playback
  - Create 2 SEQs to switch between
  - Play both simultaneously, one muted, one unmuted
  - Reverse muting at specified times
  - Allows for largest variability

# *MIDI channel muting*

## ▶ 3: “Pack” SEQs

- What is “packing”
  - Map main instruments of song 1 to MIDI channel 0
  - Map detail instruments of song 1 to MIDI channel 1
  - Map main instruments of song 2 to MIDI channel 2
  - Map detail instruments of song 2 to MIDI channel 3
  - ...
  - Map detail instruments of song 8 to MIDI channel 15

# *MIDI channel muting*

- “Packing” benefits
  - Easy song switching
  - Allows access to more songs at a cheaper memory price
  - Saves up to 1204 bytes per SEQ
- “Packing” drawbacks
  - Eats up a few extra CPU cycles to process muted tempo changes etc.
  - Reduces individual song variability

# *MIDI channel muting*

- ▶ Combine “packing” and SEQ switching
  - Create 2 or more SEQs
  - Map as many detail and main instruments as desired
  - Use muting to turn instruments on and off and switch songs



# *MIDI callback messages*

- ▶ Use as timers
  - Start other SEQs playing
  - Switch SEQs
  - Mute/Unmute MIDI channels
  - Programmer defined tasks (graphical displays etc.)

# *Pitch LFO*

- ▶ Pitch LFO functions apply one waveform as an oscillator on another waveform
- ▶ Since no vibrato is currently available, use pitch LFO
- ▶ Use a small sine, triangle, or square wave
- ▶ Drawbacks
  - Not available for MIDI
  - Uses 2 voices
- ▶ Pitch LFO player for Mac

# *Pitchbend limitations*

- ▶ Must key on sound before pitchbend can take effect