Sound/Music RAM

This is a list of the RAM locations used by the sound/music program. The code that uses this data is in bank C5. This page also lists RAM locations used by the SPC-700 sound chip.

$1300-$1400: Sound/Music RAM

```
    $1300 SPC Command (byte 0)
    $1301 SPC Command (byte 1)
    $1302 SPC Command (byte 2)
    $1303 SPC Command (byte 3)
    $1304 Current Song Command (byte 0)
    $1305 Current Song Command (byte 1)
    $1306 Current Song Command (byte 2)
    $1307 Current Song Command (byte 3)
    $1308 Previous Song Command (byte 0)
    $1309 Previous Song Command (byte 1)
    $130A Previous Song Command (byte 2)
    $130B Previous Song Command (byte 3)

 ++$1310 Pointer to instrument brr data
  +$131C Pointer to next instrument (brr data) in SPC
```

SPC-700 RAM Map

$0000-$00FF: SPC Direct Page 0

```
  +$00 Song Script Offset
  +$02 Song Script Pointer (2 bytes per voice)
  +$12 Sound Effect Song Script Pointer (2 bytes per voice)
   $22 Key On (bitmask) -> DSP $4C
   $23 Key On (for paused song, bitmask)
   $24 Key Off (bitmask) -> DSP $5C (set when note duration counter is $02)
   $25 Note Duration Counter
   $26 Pointer to Loop Count (+$F920/+$F940)
   $35 Voice Duration Counter (sound effect)
   $36 Pointer to Loop Count (sound effect, +$F920/+$F940)
  +$45 Current Tempo in beats per minute (high byte active)
   $47 Tempo Counter
   $48 Sound Effect Tempo Counter (always 120)
   $49 Tempo Envelope Counter
  +$4A Tempo Envelope Change Rate
  +$4C Echo Volume (high byte active) -> DSP $2C & $3C
  +$4E Echo Volume Envelope Change Rate
   $50 Echo Volume Envelope Counter
```

```
    $51 Song Volume
    $52 Enabled Voices (bitmask, cleared if voice has no song script)
    $53 Enable Echo (bitmask)
    $54 Enable Echo (bitmask, sound effect)
    $55 Enable Noise (bitmask)
    $56 Enable Noise (bitmask, sound effect)
    $57 Enable Pitch Mod (bitmask)
    $58 Enable Pitch Mod (bitmask, sound effect)
    $59 Disable Key Off at End of Note (bitmask, used for ties, slurs, and
drum rolls)
    $5A Disable Key Off at End of Note (bitmask, sound effect)
    $5B Enable Slur (bitmask, set before first slurred note)
    $5C Enable Slur (bitmask, sound effect)
    $5D Activate Slur (bitmask, set after first slurred note)
    $5E Activate Slur (bitmask, sound effect)
    $5F Enable Drum Roll (bitmask, disables key off at end of note)
    $60 Enable Drum Roll (bitmask, sound effect)
    $61 Ignore Song Volume (bitmask)
    $62 Noise Clock Value
    $63 Noise Clock Value (sound effect)
    $64 -
$65-$74 Filter Data (8 values, 2 bytes each, high bytes active) -> DSP $0F-
$7F
   +$75 Echo Feedback (signed, high byte active) -> DSP $0D
    $77 Filter Envelope Counter
    $78 Echo Feedback Envelope Counter
   +$79 Echo Feedback Envelope Change Rate
    $7B Output Code
    $7C -
    $7D -
    $7E -
    $7F -
    $80 new echo delay value
    $81 echo buffer wait time
    $82 -
    $83 Game Sound effect (type 2) voices (bitmask)
    $84 System sound effect (type 1) voices (bitmask)
    $85 ----v??m
        v: enable master volume envelope output mode (interrupt command
$80/$81)
        m: enable mono mode (interrupt command $F3)
    $86 weft----
        w: enable waveform output mode (interrupt command $FF/$01)
        e: disable echo (interrupt command $FF/$02)
        f: fast forward
        t: update first/last 4 voices for waveform output mode (toggles
every frame)
    $87 Enable Echo (bitmask) -> DSP $4D
    $88 Enable Noise (bitmask) -> DSP $3D
```

```
    $89 Enable Pitch Mod (bitmask) -> DSP $2D
    $8A DSP Flags -> DSP $6C
        rmennnnn
        r: DSP reset
        m: mute all
        e: echo disable
        n: noise generator frequency
$8B-$93 scratchpad
    $8B interrupt command b0
    $8C interrupt command b1
    $8D interrupt command b2
    $8E interrupt command b3
    $8F bitmask for current voice
$98-$9B scratchpad

    $A0 Active Voices
    $A1 current octave
    $A2 Current Op Code/Note
    $A3 Voice Pointer
    $A4 Muted Voices (bitmask)

$A5-$BF Master Envelope Data (counters decrement every 39ms, 25.64 Hz)
-------
   +$A5 Master Volume (high byte active)
   +$A7 Sound Effect Volume (high byte active, affects game sound effects
only)
   +$A9 Master volume envelope change rate
   +$AB Sound Effect volume envelop change rate
    $AD Master volume envelope counter
    $AE -
    $AF Sound effect volume envelope counter
    $B0 -
   +$B1 Sound Effect Pan (high byte active)
   +$B3 Sound Effect Pan envelope change rate
    $B5 Sound Effect Pan envelope counter
   +$B7 Tempo Ratio (interrupt command $84, high byte active)
   +$B8 Tempo Ratio Envelope Change Rate
    $BA Tempo Ratio Envelope Counter
   +$BB Master Pitch Multiplier (high byte active)
   +$BD Master Pitch Multiplier envelope change rate
    $BF Master Pitch Multiplier envelope counter
$C0-$C3 Pitch Calculation Data
-------
   +$C0 Calculated Frequency Value
   +$C2 Note Pitch Multiplier
    $C3
    $C4 Pointer to Song Start Offsets (+$1C02)
    $C6 Paused Song Index
    $C7 Current Song Index
    $C8 Master Envelope Frame Counter (8 * 4.875 milliseconds, clears every
39 milliseconds/25.64 Hz)
```

```
$C9-$D8 Calculated volume for each voice (2 bytes each, left then right,
used for waveform output mode)
    $D9 Paused Voices (bitmask)
    $DA Paused Game Sound Effects (bitmask)
    $DB Enable volume update in DSP (bitmask)
    $DC Enable pitch update in DSP (bitmask)
    $DD Enable Conditional Jump (bitmask)
$DE-$EF -

$F0-$FF Hardware Registers
-------
    $F0 (test)
    $F1 --ab-xyz
        a: clear port-0 and port-1
        b: clear port-2 and port-3
        x: start timer-0
        y: start timer-1
        z: start timer-2
    $F2 DSP Address
    $F3 DSP Data
    $F4 Port-0
    $F5 Port-1
    $F6 Port-2
    $F7 Port-3
    $F8 -
    $F9 -
    $FA Timer-0
    $FB Timer-1
    $FC Timer-2
    $FD Counter-0 (4.875 milliseconds)
    $FE Counter-1 (16 milliseconds)
    $FF Counter-2 (78.125 microseconds)
```

$0100-$01FF: SPC Direct Page 1

```
$00-$0F Filter Envelope Change Rate
    $10 Vibrato Delay (in ticks)
    $11 Vibrato Delay Counter
    $30 Tremolo Delay (in ticks)
    $31 Tremolo Delay Counter
    $50 Pitch Envelope Target (signed, in half steps, clears when a new note
is played)
    $51 Vibrato Amplitude (unsigned fraction, low 6 bits active)
    $70 mmaaaaaa
        m: tremolo mode (0/1 = positive, 2 = negative, 3 = balanced)
        a: tremolo amplitude (unsigned fraction, low 6 bits active)
    $71 (pansweep)
    $90 -ttt-vvv
```

```
        t: tremolo gain counter (doesn't work because of a bug)
        v: vibrato gain counter (linear envelope lasting 4 cycles when
vibrato begins, unused if vibrato delay is 0)
    $91 -
$B0-$FF CPU Stack
```

$0200-$19FF: SPC Code

```
$178F-$17A8 Note Pitch Multipliers ($1000 * 2 ^ ((x - 12) / 12))
$17A9-$17D0 Filter Data (8 bytes each)
$17D1-$17DE Note Durations
$17DF-$17FE Pointers to System Sound Effect Scripts
$17FF-$1880 System Sound Effect Data
$1881-$18F8 Jump Table for Op Codes
$18F9-$1934 Number of Bytes for each Op Code
$1935-$1954 Jump Table for Interrupts $10-$1F
$1955-$1974 Jump Table for Interrupts $80-$8F
$1975-$1994 Jump Table for Interrupts $F0-$FF
$1995-$19A5 Pointers to DSP Registers
$19A6-$19B6 Pointers to DSP Data in dp 0
$19B7-$19FF -
```

$1A00-$F5FF: Misc. Data

```
$1A00-$1A7F Instrument Pitch Multipliers (2 bytes each, high byte then low
byte)
$1A80-$1AFF ADSR Data (2 bytes each, gdddaaa then sssrrrrr)
$1B00-$1BFF Pointers to BRR Waveform Data (4 bytes each, start then loop
start)
$1C00-$2BFF Song Scripts
$2C00-$2FFF Pointers to Game Sound Effects (4 bytes each, voice A then voice
B)
$3000-$47FF Game Sound Effect Data
$4800-$7CFF BRR Data
$7D00-$F5FF Echo Buffer (assuming max echo delay value of 15, 240ms)
```

$F600-$FFBF: Voice Data

```
     $F600 Octave
     $F601 Instrument
    +$F620 Voice Volume (upper byte active)
    +$F640 Voice Envelope Change Rate
    +$F660 Pan (0100 = left, 8000 = center, FF00 = right, upper byte
active)
    +$F680 Pan Envelope Change Rate
     $F6A0 Volume Envelope Counter (crescendo/decrescendo)
     $F6A1 Pan Envelope Counter (panslide)
```

```
    $F6C0 Vibrato Cycle Duration (wave period = 4.875ms * this value * 2)
    $F6C1 Vibrato Cycle Counter
    $F6E0 Tremolo Cycle Duration
    $F6E1 Tremolo Cycle Counter
    $F700 (pansweep)
    $F701 (pansweep)
    $F720 Pitch Envelope Duration
    $F721 Transpose (in half steps, signed)
   +$F740 Instrument Pitch Multiplier
    $F760 Detune
    $F761 Absolute Pitch
   +$F780 (pansweep)
   +$F7A0 (pansweep)
   +$F7C0 Maximum Vibrato Change Rate (amplitude / cycle duration)
   +$F7E0 Vibrato Change Rate (signed)
   +$F800 Maximum Tremolo Change Rate (amplitude / cycle duration)
   +$F820 Tremolo Change Rate (signed)
   +$F840 Vibrato Value (signed fraction, high byte active)
   +$F860 Tremolo Value (signed fraction, high byte active)
   +$F880 (pansweep)
   +$F8A0 (pansweep)
   +$F8C0 Calculated Vibrato Value (added directly to frequency)
   +$F8E0 Calculated Frequency Value
   +$F900 ADSR Data
    $F920 Repeat Count (8 voices, 4 bytes per voice, counts up)
    $F940 Loop Counter (16 voices, 4 bytes per voice, counts down)
   +$F980 Loop Start Script Pointer (16 voices, 8 bytes per voice)
$FA00-$FDFF Voice Data for Paused Song
$FE00-$FEFF Saved dp 0 (first $80 bytes)
$FF00-$FF9F Saved dp 1 (first $A0 bytes)
```

$FFC0-$FFFF: SPC IPL-ROM

Other SPC Data

Note Lengths

```
0: Whole Note                C0
1: Half Note                 60
2: Half Note Triplet         40
3: Dotted Quarter Note       48
4: Quarter Note              30
5: Quarter Note Triplet      20
6: Dotted Eighth Note        24
7: Eighth Note               18
8: Triplet                   10
```

```
9: Sixteenth Note              0C
A: Sixteenth Note Triplet      08
B: Thirty-second Note          06
C: Thirty-second Note Tripet   04
D: Sixty-fourth Note           03
```

Note Names

```
0: C        00-0D
1: C#/Db    0E-1B
2: D        1C-29
3: D#/Eb    2A-37
4: E        38-45
5: F        46-53
6: F#/Gb    54-61
7: G        62-6F
8: G#/Ab    70-7D
9: A        7E-8B
A: A#/Bb    8C-99
B: B        9A-A7
C: Tie      A8-B5
D: Rest     B6-C3
```

SPC Command Codes

```
C4: xx       $11D3 Set Voice Volume to xx (00-7F)
C5: xx yy    $11E4 Set Voice Volume w/ Envelope (yy: volume, xx: envelope
duration)
C6: xx       $1236 Set Voice Pan to (xx: 01 = Left, 40 = Center, 7F = Right,
top bit inactive)
C7: xx yy    $1246 Set Voice Pan w/ Envelope (yy: 01 = Left, 40 = Center, 7F
= Right, top bit inactive, xx = envelope duration)
C8: xx yy    $1266 Change Pitch w/ Envelope (xx: envelope duration, yy:
change in pitch, signed)
C9: xx yy zz $12E3 Enable Vibrato (xx: delay in ticks, yy: cycle duration,
??zzzzzz: amplitude, max 1/4 step)
CA:          $138C Disable Vibrato
CB: xx yy zz $1396 Enable Tremolo (xx: delay in ticks, yy: cycle duration,
??zzzzzz: amplitude, max 50%)
CC:          $13EA Disable Tremolo
CD: xx yy    $13F4 Enable Pansweep (xx: delay in ticks, yy: cycle duration)
CE:          $144A Disable Pansweep
CF: xx       $14CB Set Noise Clock (00-1F)
D0:          $148D Enable Noise
D1:          $14BB Disable Noise
D2:          $14D9 Enable Pitch Modulation
D3:          $14F2 Disable Pitch Modulation
D4:          $1464 Enable Echo
```

```
D5:          $147D Disable Echo
D6: xx       $1460 Set Octave to xx
D7:          $1456 Increment Octave
D8:          $145C Decrement Octave
D9: xx       $1275 Set Transpose
DA: xx       $1271 Add to Transpose
DB: xx       $174E Set Detune
DC: xx       $1502 Set Instrument
DD: xx       $1538 Set ADSR Attack Value (0-15)
DE: xx       $1568 Set ADSR Decay Value (0-7)
DF: xx       $157B Set ADSR Sustain Value (0-7)
E0: xx       $158D Set ADSR Release Value (0-31)
E1:          $159D Reset ADSR Default Values
E2: xx       $16C0 Loop Start (loop xx+1 times)
E3:          $16F0 Loop End
E4:          $15B4 Enable Slur (key on at beginning of first note only, no
key off at end of note unless the next note is a rest)
E5:          $0FD3 Disable Slur
E6:          $15DE Enable Drum Roll (no key off at end of note unless the
next note is a rest)
E7:          $0FD3 Disable Drum Roll
E8: xx       $174B Add to Note Duration (xx = duration in ticks)
E9: xx       $1602 Play Game Sound Effect (voice A)
EA: xx       $1606 Play Game Sound Effect (voice B)
EB:          $1770 End of Script
EC:          $1770 End of Script
ED:          $1770 End of Script
EE:          $1770 End of Script
EF:          $1770 End of Script
F0: xx       $11AF Set Tempo
F1: xx yy    $11B8 Set Tempo w/ Envelope
F2: xx       $1205 Set Song Echo Volume
F3: xx yy    $1212 Set Song Echo Volume w/ Envelope
F4: xx       $11D0 Set Song Volume
F5: xx yyyy  $1665 Jump to yyyy When Loop Count Reaches xx
F6: xxxx     $164A Jump to xxxx
F7: xx yy    $1279 Set Echo Feedback to yy over xx frames
F8: xx yy    $129B Set Filter ------yy
F9:          $1752 Increment Output Code
FA:          $1755 Clear Output Code
FB:          $15B0 Ignore Song Volume
FC: xxxx     $1759 Conditional Jump to xxxx
FD:          $1770 End of Script
FE:          $1770 End of Script
FF:          $1770 End of Script
```

SPC Interrupt Codes

```
00:                   No Interrupt
10: xx yy      $0A1E Load New Song (x: song number, y: master volume)
11: xx yy      $0A26 Load New Song (pause current song)
14: xx yy      $0A1B Long New Song (alternate start position)
15: xx yy      $0A23 Long New Song (alternate start position, pause current
song)
18: xx yy      $0B3D Game Sound Effect (x: sound effect number, y: pan value
[$80 = center])
20:                   Cursor (select)
21:                   Cursor (move/cancel)
22:                   Error
23:                   Ring (Success)
24:                   Delete/Erase
28:                   Ba-ding high (character becomes active in battle)
29:                   Ba-ding low (character controlled by player 2 becomes
active in battle)
2C:                   Ching
30-3F                 Quick Load Song $00-$0F at full volume (scpu commands, no
effect in spc)
80: xx yy      $0C6A Set master/sound effect volume to yy (w/ envelope xx)
81: xx yy      $0C6A Set master volume to yy (w/ envelope xx)
82: xx yy      $0C6A Set sound effect volume to yy (w/ envelope xx)
83: xx yy      $0CE5 Set sound effect pan to yy (w/ envelope xx)
84: xx yy      $0D09 Set Tempo Ratio (yy = envelope duration, xx = tempo
ratio, signed fraction)
85: xx yy      $0D35 Change Pitch
89:            $0F9C Enable Conditional Jump (used by Phantom Train and
Dancing Mad)
F0:            $0D6F Stop song & sound effect
F1:            $0D6F Stop song
F2:            $0D6F Stop sound effect
F3: xx         $0D61 Enable/Disable Mono Mode (x: 0 = disable mono mode, 1 =
enable mono mode)
F4: xx         $0DC3 Mute Voices (x: voices to mute, bitmask)
F5: xx         $0DCA Pause/Unpause music (0 = unpause, 1 = pause)
F6: xx         $0F83 Enable/Disable Fast Forward (0 = disable, 1 = enable)
FC: xx         $1002 Set Echo Delay to xx
FD: xx yy      $0FE8 Set DSP Register xx with data yy
FE: xx         $0E38 Transfer Data from SCPU
                     00: no transfer
                     01: transfer one byte at a time
                     02: transfer two bytes at a time
                     03: transfer three bytes at a time
                     07: move chunk
FF: xx yy      $0FA3 Reset Codes
                     01: enable/disable waveform output mode (y: 0 = disable,
1 = enable)
                     02: enable/disable echo (y: 0 = disable, 1 = enable)
                     F0: reset spc
```

Signed Fractions

```
$80: 1/2x
$A0: 5/8x
$C0: 3/4x
$E0: 7/8x
$00:   1x
$20: 5/4x
$40: 3/2x
$60: 7/4x
$7F:   2x
```

Song Format

```
2 bytes preceding data in the ROM are the length of the song
   +$00 Song Start Address (HiROM)
   +$02 Song End Address (HiROM)
$04-$13 Voice Start Addresses (HiROM)
$14-$23 Voice Start Addresses (HiROM, alternate start position)
$24-EOF Song Data
```

From:
https://www.ff6hacking.com/wiki/ - **ff6hacking.com wiki**

Permanent link:
https://www.ff6hacking.com/wiki/doku.php?id=ff3:ff3us:doc:asm:ram:sound_music_ram&rev=1457822863

Last update: **2019/02/12 13:05**