Multitimbral Sound Module

Valerie Gordeski
Susan Hwang
Christopher Sheehan
What is it?

- Reads a song from ROM or keyboard for live play or later playback
- Interpolates stored note samples to obtain the correct pitch
- Shapes the pitch to imitate the sounds of a piano, violin and a flute
- Plays a magnificent trio (Bach? Mozart?)
The Block Diagram

SONG ROM

Major FSM

Decoding unit

Flute

Violin

Piano

enable

done

DAC control

Note ROM

Interpolator

ADSR
Frontend Block Diagram

- **Decoder**
  - Input: kclk, kdata
  - Output: note

- **Recorder**
  - Input: note

- **Play Live**
  - Inputs: Instrument 1, Instrument 2, Instrument 3
  - Output: note

- **Playback**
  - Input: message
  - Output: message
  - Sub-blocks: Playback RAM, Song ROM

- **MAJOR FSM**
**Encoding**

**Message Encoding**

<table>
<thead>
<tr>
<th>MSB (2)</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Pass</td>
<td>Voice 1</td>
<td>Voice 2</td>
<td>Voice 3</td>
</tr>
</tbody>
</table>

**Note Encoding**

<table>
<thead>
<tr>
<th>Type/ Instrument (2)</th>
<th>Note On/OFF (1)</th>
<th>Pitch Value (5 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (voice 1)</td>
<td>1 (on)</td>
<td>0 (lower C)</td>
</tr>
<tr>
<td>01</td>
<td>0 (off)</td>
<td>Nothing (ignored)</td>
</tr>
<tr>
<td>00 (pass)</td>
<td>Time value (6 bits)</td>
<td></td>
</tr>
</tbody>
</table>
A = 440Hz

Sampled Waveform in ROM

Generate an “A” one octave higher

Generate a higher “E”

A = 880Hz

E = 659Hz

Constant sampling rate, Variable Pointer

?
Linear Interpolation

\[
ROM[K + F] = (1 - F)ROM[K] + (F)ROM[K + 1]
\]

\[
ROM[2 + 0.5] = (1 - 0.5)ROM[2] + (0.5)ROM[2 + 1]
\]
Interpolation: Block Diagram

Instrument

play

note_val

Note Lookup Table

increment

FSM

reset

start

busy

val_a

val_b

fract

result

Sample ROM

rom_addr

rom_data

Interpolator
ADSR: An Overview

- Determines what “instrument” we hear
- Typical Envelope:

http://www.teachnet.ie/amhiggins/lesson6.html#
ADSR: The Diagram

- Clock
- Reset
- Enable

ADSR FSM

- Counter
- ADKPTR
- DecayPTR
- SuspTR

Envelope ROM

- q1
- q2

Multiplier

- Result
- Busy

Inter_output

Enable
Future Extensions

- Real MIDI keyboard input
- Expand number of instruments to full ensemble, including percussion (gated noise)
- Secure Digital card for extended music
- Dynamic shaping of the note depending on the input: staccato (fast attack versus slow attack), vibrato, trills etc.