Mixing Setup

Nathan Artz, Adam Goldstein, Matthew Putnam
The Big Picture

Memory Subsystem

Scratch Pad

Track 1

Keyboard

Pitch, Speed, and Beat Matching

Track 1

Effects

Track 1

Display

External Sound Source

For recording

Waveforms

Sound Output

Track 2

Effects

Track 2

+
Memory Subsystem

FLASH (x16 mode)
(Irrelevant inputs ignored)

Address[23:0]
Write Enable
Output Enable

Data Bus [15:0]
Loader

Last Read Sample [15:0]
Last Write Sample [15:0]
Next_Position_to_read[23:0]
Current_State == Write || Read

DataToWrite[15:0]
NextPositionToRead[23:0]

WriteDataExists?

Get_sample (0 for address, or 1 for Write)

Flash Controller

Last Read Sample [15:0]
Last Write Sample [15:0]
Current_State == Write || Read
Current_Song == Song0 || Song 1
(from loader 1 or loader 2)
Current_Write_Address
If(reset)
Erase

• Input Sample
  • Signals which sample to input (read or write sample)
• ReadDataExists / Write Data Exists
  • 1 when data exists, 0 when no data

Reset

Reset Done

WriteDataExists?

Data[23:0]

Data[15:0]

Reset

Address[23:0]

Sample[15:0]

WE = 1
Flash Controller: MIXING

Read from...
WE = 0
Addresses: 0 – 8388608
Reg Every_Xth_Sample
   Start & End_Read_Address_1 & 2
   Current_Read_Address_1 & 2
   Completed_Read?

Write to...
WE = 1
Addresses: 8388608 – 16177216
Reg Start_Write_Address
   Current_Address
   Completed_Write?

One Read Cycle
Completed
One Write Cycle
Completed

Spec
• Read (one cycle)
  • CurrentState == Read
  • Read Data Exists?
    • Get_sample Signal(0)
    • Read Address
    • Send back to Loader
    • Loader output (next clock cycle)
  • IF(WriteDataExists)
    • CurrentState -> Write
      • Get_sample signal (1)
      • Write Data (100 ns)

• Timings (for one second)
  • 24 Khz Sampling means 24,000 clock cycles needed per sample
    • Sample every 4.16*10^-5 seconds
  • Read = 25 ns ( < 100 ns)
  • Write < 100 ns

• Per cycle
  • Read X 2 (loader 1 and 2)
  • Write Sample (other loader)
  • About 300 ns
Write until full, then simply loop over and overwrite over old.
Song Recorder Specs

• Flash Controller will be ERASED before songs are loaded
• Songs are written starting at the beginning of the allotted possible address space for that Specified Address Field
• Specified Address Fields (in decimal):
  • Song 0: 0 - 5242880
  • Song 1: 5242880 – 10,485,760
  • RECORDING SPACE IS 10,485,760 – 16,777,216
The Keyboard

- Given a start pitch and an end pitch, outputs a ratio for frequency shifting
The Scratch Pad
The Scratch Pad

• The strategy: Notice *substantial* changes in speed (outside a few % of the current speed)
The Scratch Pad

• The strategy: Notice *substantial* changes in speed (outside a few % of the current speed)
• Convert the speed into a position in memory
The Scratch Pad

- The strategy: Notice *substantial* changes in speed (outside a few % of the current speed)
- Convert the speed into a position in memory
- If the speed is being provided externally (e.g. from the beat matcher), change the speed the motor is turning
Speed Changing

- bm_period[15:0]
- enc_period[15:0]
- speed_source

Speed

period[15:0]

Position

beat

next_pos[23:0]

position[23:0]
Speed Changing

```
- bm_period[15:0]
- enc_period[15:0]
- speed_source
  Speed
    period[15:0]
  Position
    beat
    next_pos[23:0]
    position[23:0]
```
Speed Changing

- \texttt{bm}\_\texttt{period}[15:0]
- \texttt{enc}\_\texttt{period}[15:0]
- \texttt{speed\_source}

\texttt{Speed} \rightarrow \texttt{Position}

- \texttt{beat}
- \texttt{next\_pos}[23:0]

\texttt{Period}[15:0]

\texttt{Position} \rightarrow \texttt{Position}[23:0]
Speed Changing

- \texttt{bm\_period[15:0]}
- \texttt{enc\_period[15:0]}
- \texttt{speed\_source}

\textbf{Speed}

\begin{align*}
\text{Period}[15:0] & \to \\
\text{Beat} & \downarrow \\
\text{Next_pos}[23:0] & \downarrow \\
\textbf{Position}
\end{align*}

\begin{align*}
\text{Period}[15:0] & \to \\
\text{Position}[23:0] & \downarrow \\
\end{align*}
• Immediately after starting to beat-match, hold position constant until the next beat so the beats are synchronized
Speed Changing

- Immediately after starting to beat-match, hold *position* constant until the next beat so the beats are synchronized

- Nathan’s sampling stuff
Frequency Changing
Frequency Changing
Frequency Changing

- For 512-sample windows, need ~1024 cycles for each window to STFT and ISTFT
- Ideally have 4 overlapping windows; can do with 2
Other Effects
Other Effects

• How they work: Echo, Reverb, Bandpass, Etc.
Other Effects

- How they work: Echo, Reverb, Bandpass, Etc.
Schedule
Schedule

November 18

Nathan: Record/Output a Single Speed
Matt: Convert Keyboard Signals to Pitches
Adam: Interfacing with Scratch Pad
Schedule

November 18

Nathan: Record/Output a Single Speed
Matt: Convert Keyboard Signals to Pitches
Adam: Interfacing with Scratch Pad

November 21

Nathan: Up/Downsampling, Outputting at Multiple Speeds
Matt: Echo, Reverb
Adam: Working Vocoder
Schedule

November 18
- Nathan: Record/Output a Single Speed
- Matt: Convert Keyboard Signals to Pitches
- Adam: Interfacing with Scratch Pad

November 21
- Nathan: Up/Downsampling, Outputting at Multiple Speeds
- Matt: Echo, Reverb
- Adam: Working Vocoder

November 26
- Adam: Beat Matching
- Matt: Visual Outputs
- Nathan: Integration
Schedule

November 18
Nathan: Record/Output a Single Speed
Matt: Convert Keyboard Signals to Pitches
Adam: Interfacing with Scratch Pad

November 21
Nathan: Up/Downsampling, Outputting at Multiple Speeds
Matt: Echo, Reverb
Adam: Working Vocoder

November 26
Adam: Beat Matching
Matt: Visual Outputs
Nathan: Integration

December 4
All: Integration, Visual Outputs