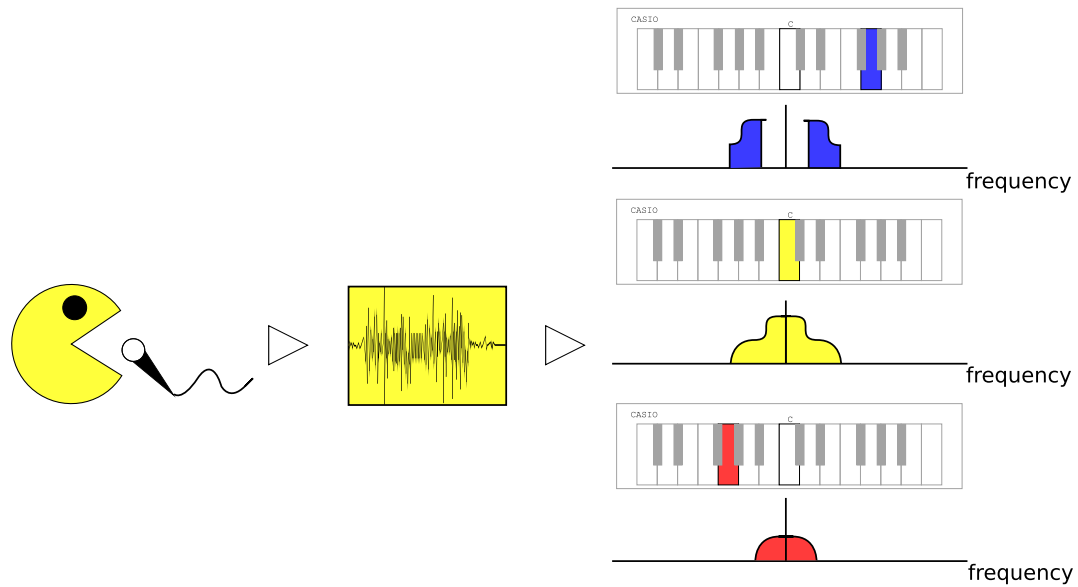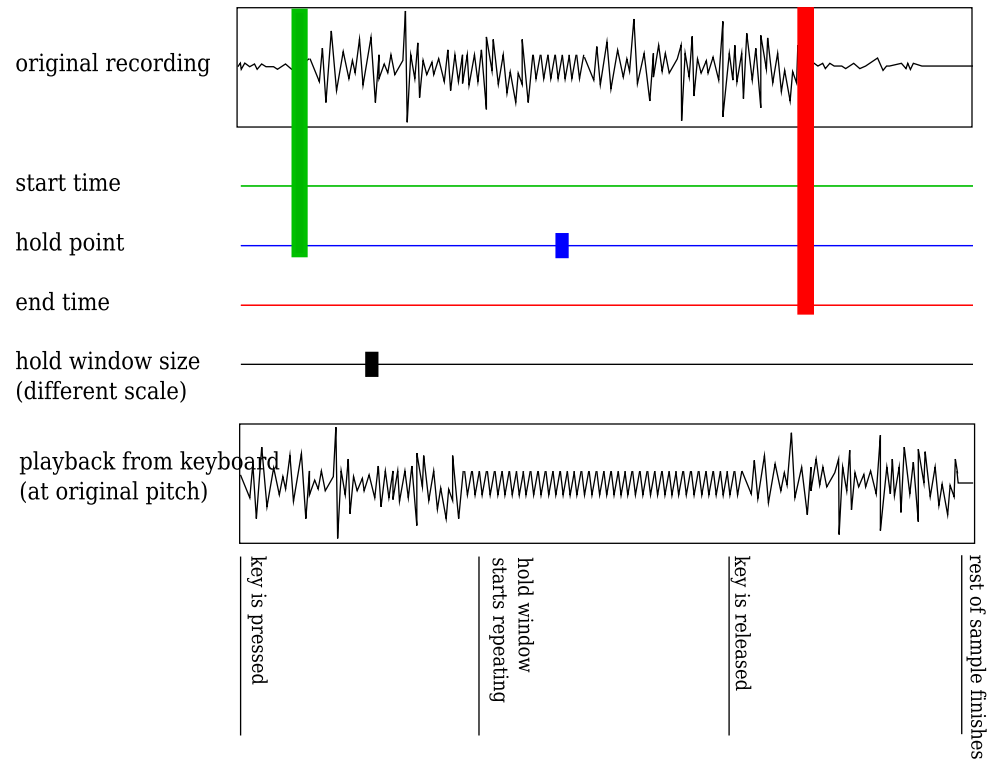# The Wumpus

## Sampling Synthesizer Keyboard

Damon Vander Lind and Mark Tobenkin

The user records a sample into memory.
Pressing different keys on a MIDI keyboard replays the sample with shifts in frequency.
Up to ten notes can be played simultaneously.

November 16th, 2005

# Start, End, Vowel Hold

original recording

start time

hold point

end time

hold window size
(different scale)

playback from keyboard
(at original pitch)

key is pressed

hold window
starts repeating

key is released

rest of sample finishes

November 16th, 2005

# User Interface
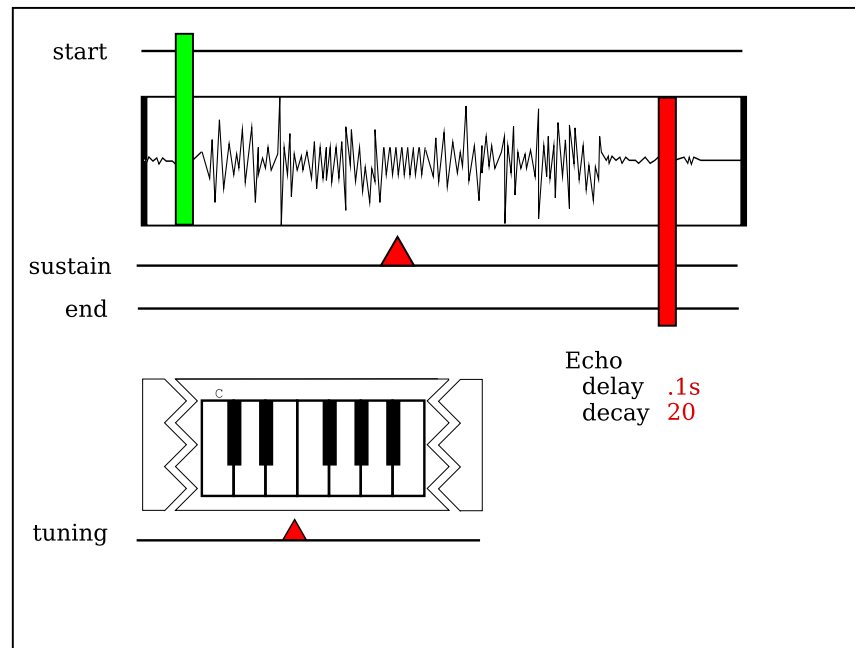
A external monitor allows a user to fine tune their playback.
    The user can crop the beginning and end of their sample.
    The key which the original sample plays at can be adjusted by tuning.

November 16th, 2005

# Block Diagram

## The Wumpus
Sampling Synthesizer Keyboard
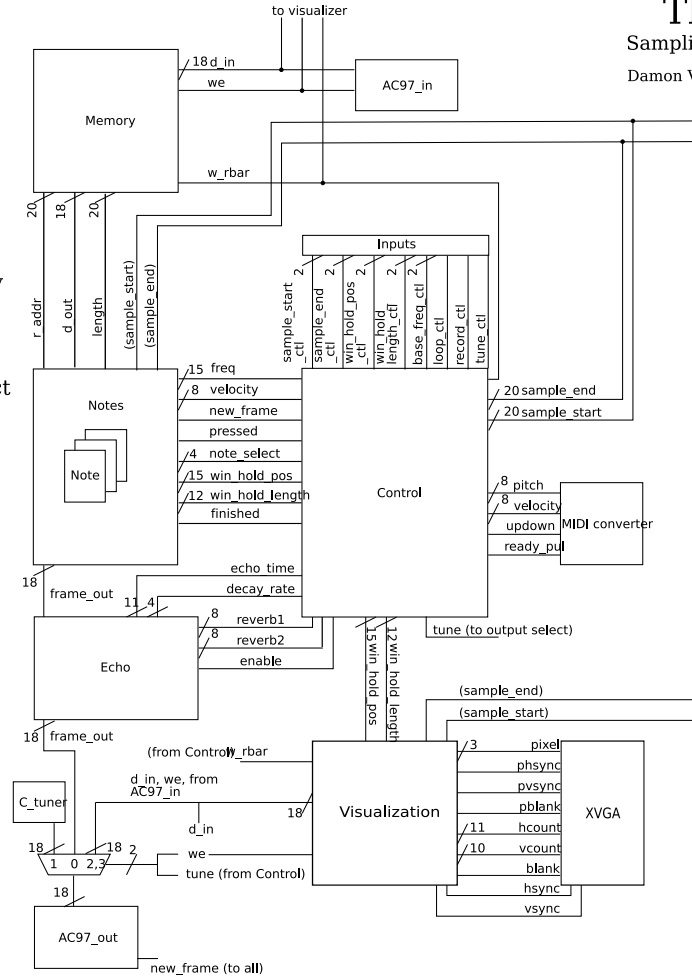
Damon Vander Lind and Mark Tobenkin

During recording, a new sound goes
directly into sample from AC97_in

During keyboard playback, individual NOTE's in
NOTES play back the sample scaled in frequency
to match the notes pressed on the keyboard

During loop playback, the sample is replayed on
loop scaled by base frequency. pressing sin_select
toggles a sin wave at middle C (for tuning your
sample)



to visualizer

Memory

18 d_in
we

AC97_in

w_rbar

20 r_addr
18 d_out
20 length
(sample_start)
(sample_end)

Inputs

sample_start_ctl 2
sample_end_ctl 2
win_hold_pos_ctl 2
win_hold_length_ctl 2
base_freq_ctl 2
loop_ctl
record_ctl
tune_ctl

15 freq
8 velocity
new_frame
pressed
4 note_select
15 win_hold_pos
12 win_hold_length
finished

Notes

Note

Control

20 sample_end
20 sample_start

8 pitch
8 velocity
updown
ready_pul

MIDI converter

18 frame_out

echo_time
decay_rate
11 4
8 reverb1
8 reverb2
enable

Echo

tune (to output select)

15 win_hold_pos
12 win_hold_length

(sample_end)
(sample_start)

18 frame_out

(from Control) w_rbar

d_in, we, from
AC97_in

C_tuner

d_in

18

we
tune (from Control)

18
1 0 2,3
18 2

18

AC97_out

new_frame (to all)

Visualization

3 pixel
phsync
pvsync
pblank
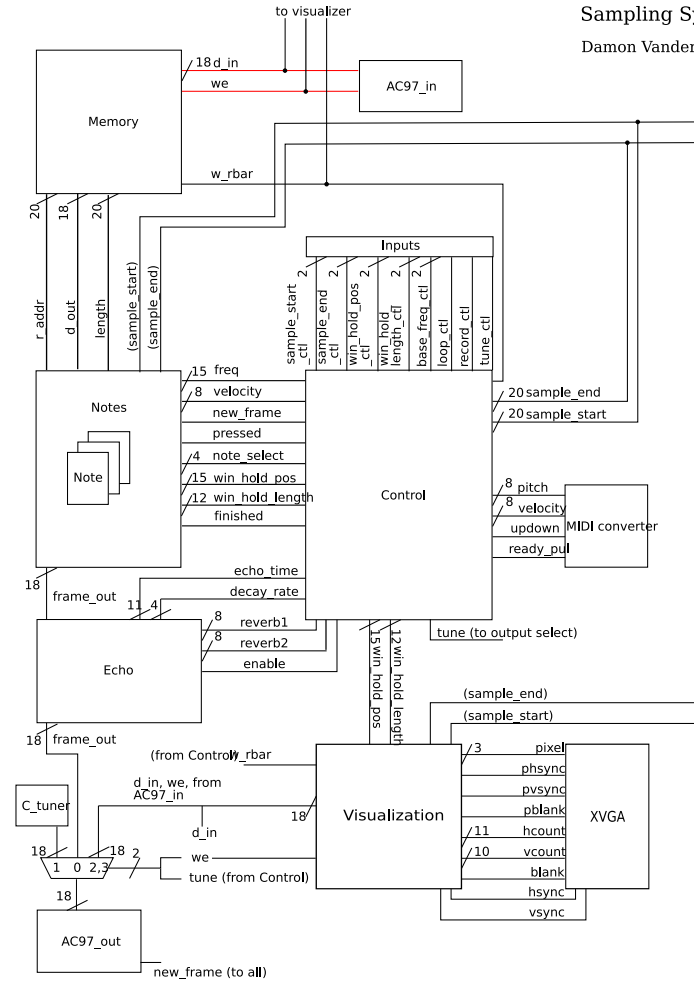11 hcount
10 vcount
blank
hsync
vsync

XVGA

November 16th, 2005

# Record Mode

The incoming recording is clocked into Memory, and a mux switches the output sound to be the input sound

## The Wumpus
### Sampling Synthesizer Keyboard
Damon Vander Lind and Mark Tobenkin

to visualizer

Memory

18 d_in
we
w_rbar

AC97_in

20 r_addr
18 d_out
20 length
(sample_start)
(sample_end)

Inputs

sample_start_ctl 2
sample_end_ctl 2
win_hold_pos_ctl 2
win_hold length_ctl 2
base_freq_ctl 2
loop_ctl
record_ctl
tune_ctl

Notes

Note

15 freq
8 velocity
new_frame
pressed
4 note_select
15 win_hold_pos
12 win_hold_length
finished

Control

20 sample_end
20 sample_start

8 pitch
8 velocity
updown
ready_pul

MIDI converter

echo_time
decay_rate

18 frame_out
11 4
8 reverb1
8 reverb2
enable

Echo

15 win hold pos
12 win hold length

tune (to output select)

(sample_end)
(sample_start)

18 frame_out

(from Control) w_rbar

d_in, we, from AC97_in

C_tuner

d_in

18

Visualization

3 pixel
phsync
pvsync
pblank
11 hcount
10 vcount
blank
hsync
vsync

XVGA

18
1 0 2,3
18 2

we
tune (from Control)

18

AC97_out

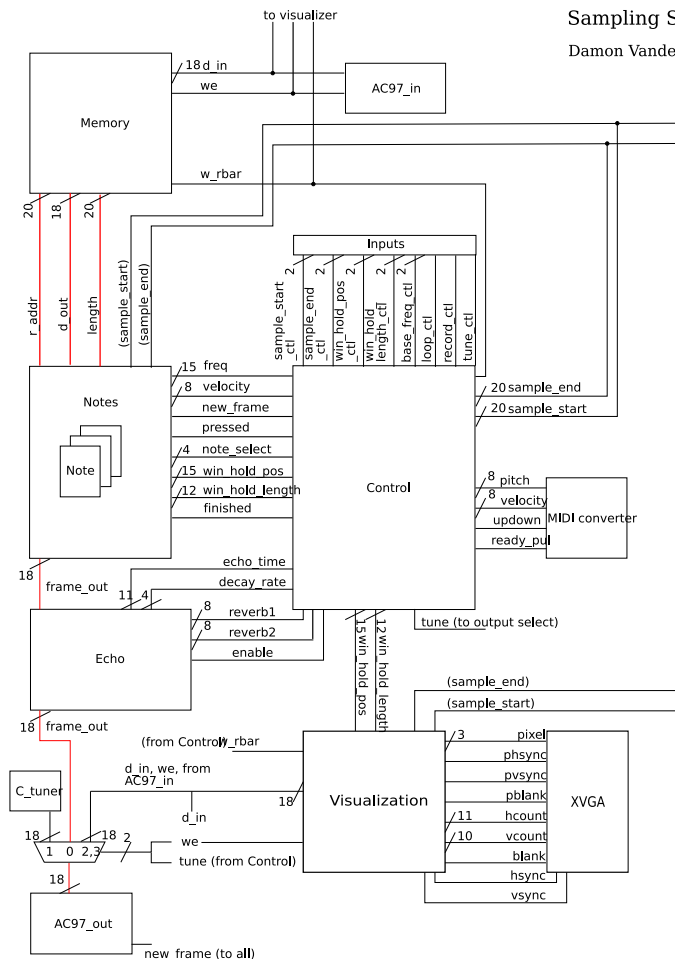new_frame (to all)

November 16th, 2005

# Playback Mode

When a key is pressed, it is entered into a queue in Control. Key presses are read out of the queue into Note's as they become availible, unless too much time elapses
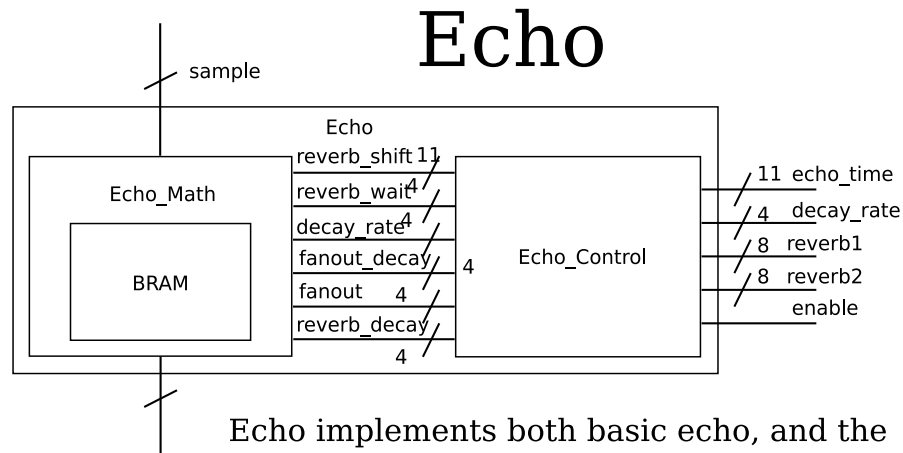
For each key pressed, The sample, from start to end, is read out of Sample, and processed through a Note and Echo before being fed into AC97_out.

to visualizer

**Memory**
18 d_in
we
w_rbar

AC97_in

20  18  20
r_addr  d_out  length  (sample_start)  (sample_end)

**Inputs**

sample_start_ctl  2
sample_end_ctl  2
win_hold_pos_ctl  2
win_hold_length_ctl  2
base_freq_ctl  2
loop_ctl
record_ctl
tune_ctl

**Notes**

Note

15  freq
8  velocity
new_frame
pressed
4  note_select
15  win_hold_pos
12  win_hold_length
finished

**Control**

20 sample_end
20 sample_start

8  pitch
8  velocity
updown
ready_pul

**MIDI converter**

18  frame_out
echo_time
decay_rate
11  4
8  reverb1
8  reverb2
enable

**Echo**

tune (to output select)

18  frame_out

15 win_hold_pos  12 win_hold_length

(sample_end)
(sample_start)

**C_tuner**

(from Control) w_rbar
d_in, we, from AC97_in

18
d_in
we
tune (from Control)

**Visualization**

3  pixel
phsync
pvsync
pblank
11  hcount
10  vcount
blank
hsync
vsync

**XVGA**

18  18  2
1  0  2,3
18

**AC97_out**

new_frame (to all)

# Echo

sample

**Echo**

Echo_Math

BRAM

reverb_shift 11

reverb_wait 4

decay_rate 4

fanout_decay

fanout 4

reverb_decay

4

4

**Echo_Control**

11 echo_time

4 decay_rate

8 reverb1

8 reverb2

enable

Echo implements both basic echo, and the more complicated reverb function. Reverb is set through a large number of variables controlling the creation of multiple echos at closely related echo_times. The control of this function is simplified in Echo_Control to correspond to two 8 bit numbers that make a basic echo when set to 0.
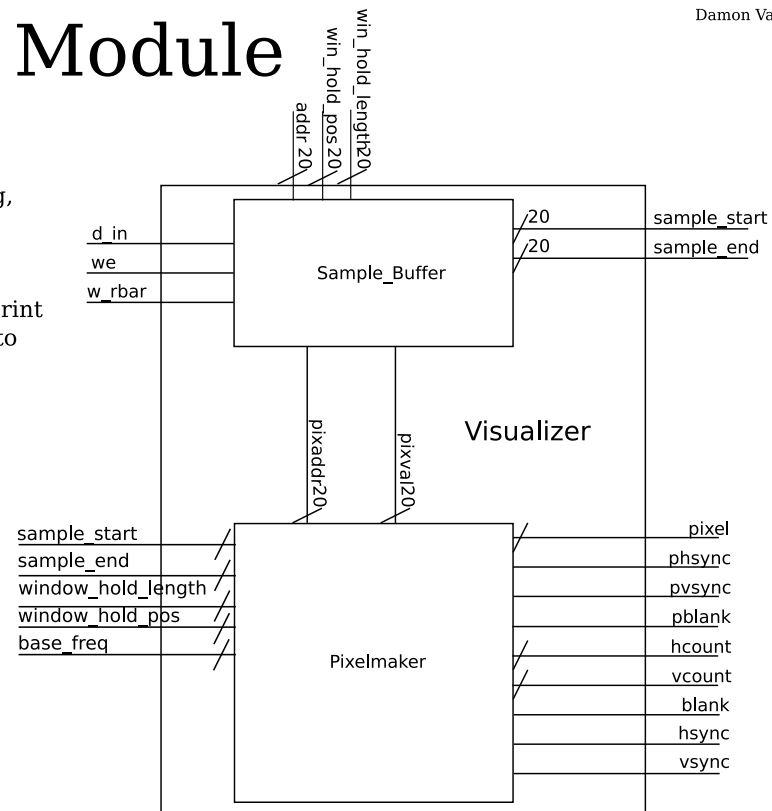
November 16th, 2005

# An Individual Note

The memory manager buffers three frames of memory for the Taylor series module. Taylor asks Memory_manager for a frame T frames away. T is mapped, in the case of window hold, to a cycle of memory addresses going from win_hold_pos to win_hold_pos + win_hold_length. In the case of length rescaling, a segment of memory addresses are repeated or deleted occasionally to lengthen or shorten the sample. Taylor does a second order approximation for frame_out.



November 16th, 2005
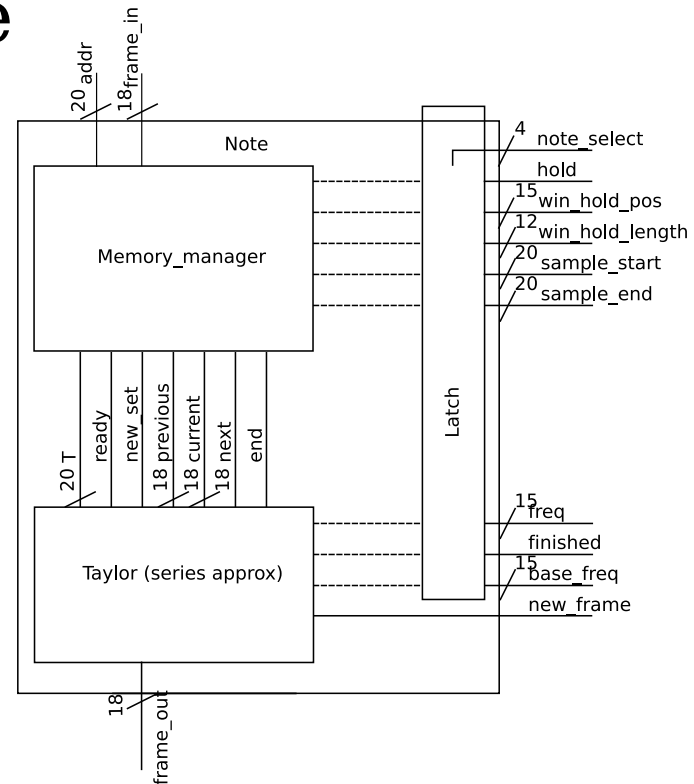
# Visualizer Module

The visualizer buffers a representation of the recording, and displays all other graphics through use of vectors in the Pixelmaker submodule. Sample_Buffer is accessed to print only the block of pixels meant to display the sound time series.



November 16th, 2005

# Block Diagram

## The Wumpus
Sampling Synthesizer Keyboard
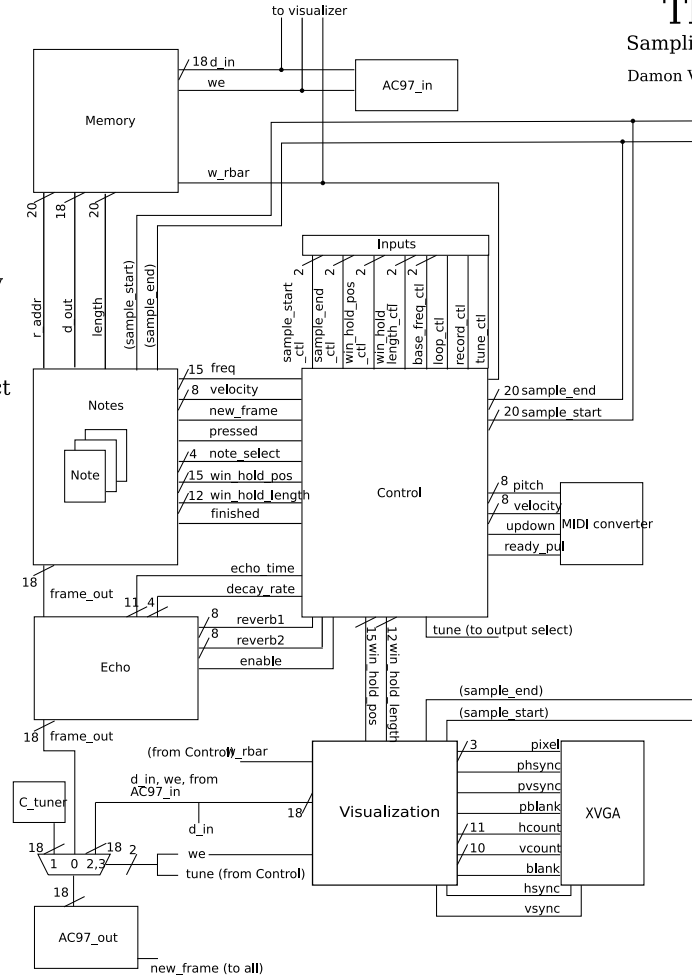
Damon Vander Lind and Mark Tobenkin

During recording, a new sound goes
directly into sample from AC97_in

During keyboard playback, individual NOTE's in
NOTES play back the sample scaled in frequency
to match the notes pressed on the keyboard

During loop playback, the sample is replayed on
loop scaled by base frequency. pressing sin_select
toggles a sin wave at middle C (for tuning your
sample)

to visualizer

Memory

18 d_in
we

AC97_in

w_rbar

20 r_addr
18 d_out
20 length
(sample_start)
(sample_end)

Inputs

sample_start_ctl 2
sample_end_ctl 2
win_hold_pos_ctl 2
win_hold_length_ctl 2
base_freq_ctl 2
loop_ctl
record_ctl
tune_ctl

15 freq
8 velocity
new_frame
pressed
4 note_select
15 win_hold_pos
12 win_hold_length
finished

Notes

Note

Control

20 sample_end
20 sample_start

8 pitch
8 velocity
updown
ready_pul

MIDI converter

echo_time
decay_rate
11 4
8 reverb1
8 reverb2
enable

tune (to output select)

18 frame_out

Echo

15 win hold pos
12 win hold length

(sample_end)
(sample_start)

18 frame_out

C_tuner

(from Control) w_rbar

d_in, we, from
AC97_in

18 d_in
we
tune (from Control)

Visualization

3 pixel
phsync
pvsync
pblank
11 hcount
10 vcount
blank
hsync
vsync

XVGA

18
1  0  2,3
18 2

18

AC97_out

new_frame (to all)

November 16th, 2005