Sound Blocks
An interactive environment for real-time music creation
What does it do?

• Users create and modify sounds by manipulating blocks on a specially-designed table.

• Three kinds of blocks: source, local-effect, global-effect.
  – Source: sound file stored in ram to be played out of speakers in loop
  – Local: modifies source block sound, ex. Add reverb, band-pass filter
  – Global: modifies entire system

• Rotating effect blocks changes magnitude of effect

• Audio waveforms displayed on table.
How does it do it?

• The table surface is translucent.
• The bottom of each block has two different colored LEDs.
• A camera underneath the table looks for the LED colors.
• The audio and video is processed according to which colors the camera sees.
• The audio is played through speakers.
• A projector underneath the table displays the waveforms onto the table from below.
FROM CAMERA

adv7185

ntsc_decode

ycrcb_to_rgb

xy_generator

memory_control

identify

xy position registers

ZBT SRAM 1

ZBT SRAM 2

identify

[BLOCK DIAGRAM 1 – Identifying objects]
MEMORY_CONTROL FSM

STATE A
- WE_A = 0
- WE_B = 0
- read_select = 1
- ready_read = 0
- hcount & vcount ≠ 0
- address_A = +1
- address_B = +1
- end_frame = 0

STATE B
- WE_A = 0
- WE_B = 0
- read_select = 0
- ready_read = 0
- hcount & vcount = 0
- address_A = +1
- address_B = +1
- end_frame = 0

Transitions:
- State A to State B: hcount & vcount ≠ 0
- State B to State A: hcount & vcount = 0
- State A to State A: address_A = 0, address_B = 0, end_frame = 1
- State B to State B: address_A = 0, address_B = 0, end_frame = 1
- State A to State B: ready_data
- State B to State A: ready_data
- State A to State A: done_read
- State B to State B: done_read
- State A to State A: address_A = +1, ready_read = 1
- State B to State B: address_A = +1, ready_read = 1
- State A to State A: address_B = +1, ready_read = 1
- State B to State B: address_B = +1, ready_read = 1
FROM CAMERA

adv7185

ntsc_decode

ycrcb_to_rgb

xy_generator

memory_control

ZBT SRAM 1

ZBT SRAM 2

identify

xy position registers

BLOCK DIAGRAM 1 – Identifying objects
Identify Controller

[35:0] read_data_in

[35:24] read_data_in = x coordinate

[23:12] read_data_in = y coordinate

X coordinate

Y coordinate

ready_read done_read end_frame

[9:0] x_block_red
[9:0] x_block_green

... [9:0] x_marker_red
[9:0] x_marker_green

Running sum of y-coordinate (vcount)

Number of red pixels found

IDENTIFY BLOCK

[9:0] y_block_red
BLOCK DIAGRAM 2 – Interpreting the data

connections[23:0] : 000 000 000 001 000 000 111 010

source4
source3
source2
source1
effect4
effect3
effect2
effect1

connections

connection registers

[23:0] connections

initial value
orientation registers

[89:0] old_i

[89:0] new_i

current value
orientation registers

[89:0] current
BLOCK DIAGRAM 3 – Generating the audio output
done = 0
load sample

Local Effect 1
done = 0
modify sample

Local Effect 2
done = 0
modify sample

Local Effect 3
done = 0
modify sample

Local Effect 4
done = 0
modify sample

Do Nothing
done = 1
sample = 0
counter = 0

Done
done = 1
counter = +1

done = 1
sample = 0
counter = 0

counter = +1

connections[23:0] : 000 000 000 001 000 000 111 010
source4 source2 source1 effect4 effect3 effect2 effect1

next = 1
next = 1

000
000
011
100
011
100
010
010

011
100
011
100
001
001
111 (from any state)

AUDIO FSM
BLOCK DIAGRAM 4 – Generating the visual output