Landscape Generator

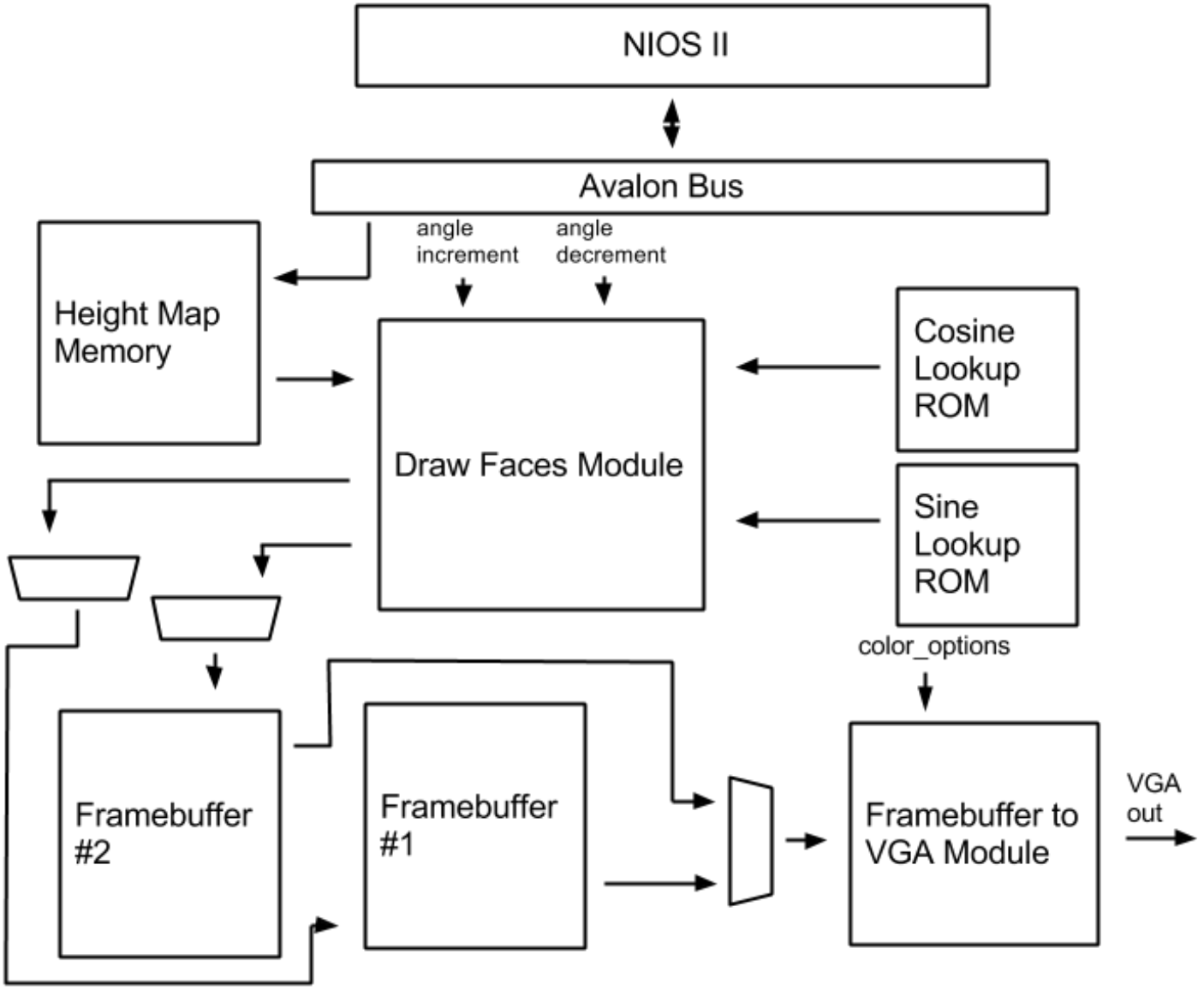
CSEE 4840

Calvin Hu

Overview

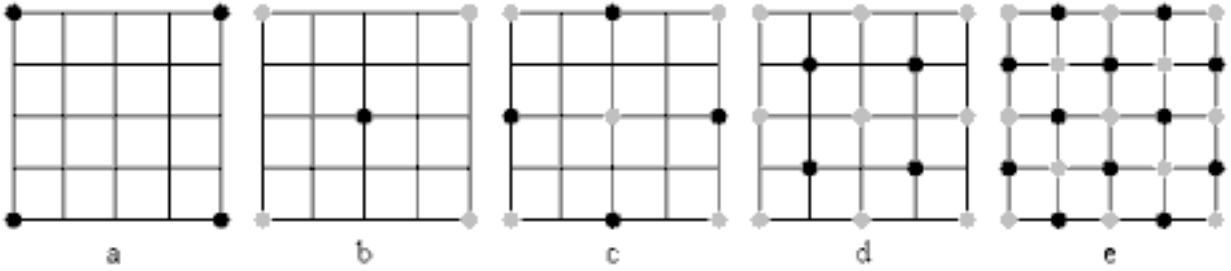
Goal: Create heightmap using diamond square algorithm and display on monitor.

- Generate heightmap
- Rotate vertices
- Backface cull vertices
- Z-sort/Grid sort
- Draw wireframe lines


NIOS II
Avalon Bus
angle increment
angle decrement
Height Map Memory
Draw Faces Module
Cosine Lookup ROM
Sine Lookup ROM
color_options
Framebuffer #2
Framebuffer #1
Framebuffer to VGA Module
VGA out

Diamond-Square Algorithm

- Use to generate terrain like heightmaps.
- Setup square grid of pixels with 2^n + 1 rows and columns
- Find average at center of "square" and "diamond" and add random value.
- Iterate and repeat until grid is filled with values.



Backface Culling

- Find orientation of face to determine whether to draw.
- Cross product of two edges in the face.
- Find dot product with camera vector (in this case, <0,0,1>
- If dot product is negative, face is facing away from camera, and can be culled.

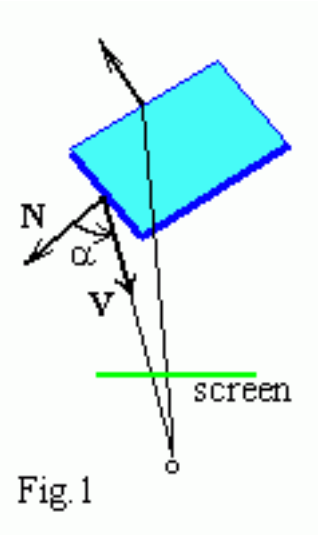


Fig. 1

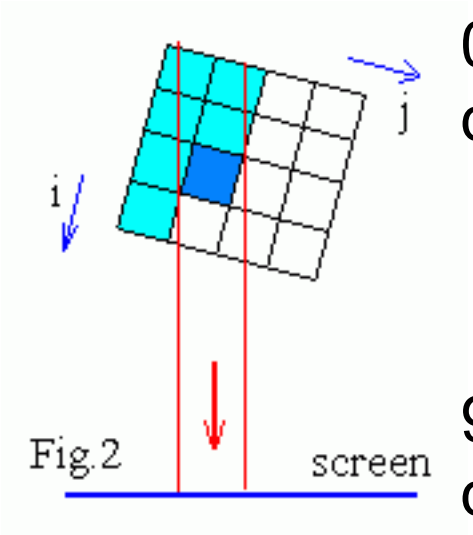
Rotation

- To rotate vertices, represent coordinates in column matrix and multiply with the corresponding rotation matrix.
- Take advantage of 9 bit multipliers in DE2.
- Read cosine/sine values from lookup ROM.

$$\begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_y \\ \mathbf{d}_z \end{bmatrix} = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Grid/Z-sorting

- Because we know the model is organized as a grid, drawing the faces in different orders depending on the orientation relative to the screen.



0 - 90 degrees

0	2
1	4

180 - 270 degrees

2	4
0	1

90 - 180 degrees

4	1
2	0

270 - 360 degrees

1	0
4	2

Memory Units

- 2 Framebuffers: 2-bit word, 76800 words
  - 4 colors
- Heightmap RAM: 36-bit word, 1024 words
  - 1 36-bit coordinate per vertex
- Sine, Cosine lookup roms: 20-bit word, 181 words
  - Each address represents the value of sine/cosine of address in degrees

Fixed Numbers

36 bit signed (2's complement) fixed point to represent vertex coordinates

Integer	Fractional
35 to 18	17 to 0

Used to handle trigonometric values/rotation.

Draw Lines and VGA output

- Uses provided Bresenham Line Algorithm component.
- Output to VGA at 640x480 60Hz

Lessons Learned

- Start early and plan well before executing.
- VHDL is much harder to debug than C.
- Don't work alone.