A 3-D Rendering System

6.111 Final Project
Ben Hebert & Mayur Desai
Spring 2005
Project Overview

• Implement a 3D Model Rendering System
• Two Components:
  – Rendering and Shading of Models
  – Output Video to VGA Monitor
• Initial Implementation: Simple monochrome polygons
• Add features sequentially: camera control, shading, lighting, etc.
Block Diagram of Rendering Engine
Rendering Engine

- Transforms the 3-D model into the sequence of pixels that comprise an image.

- Rendering consists of a series of matrix transformations to the 3-D Model.

- Sequential design that first renders the vertices of a model and then shades it.
Rendering (cont.)

• Highly Pipelined Design

• Can implement more modules in parallel with the shader unit.

• Camera Control is implemented by a matrix transformation based upon input.
3D Model Format

- Model format is Alias Wavefront:

  g tetrahedron
  v 1.00 1.00 1.00
  v 2.00 1.00 1.00
  v 1.00 2.00 1.00
  v 1.00 1.00 2.00
  f 1 3 2
  f 1 4 3
  f 1 2 4
  f 2 3 4

Simple 3-D Tetrahedron Model in OBJ Format
Video Component

• Responsible for transmitting frames from Rendering Engine to D/A Video Converter

• Utilizes a double-buffering scheme with ZBT memories to allow for parallel rendering and video.

• Important synchronization issues
Block Diagram of Video Component

Pixel Clock (@ 40 Mhz for 800x600 60 FPS) → ADV7125 video DAC → DAC Input

Video Controller FSM

Input Select/Register

ZBT SRAM 1

ZBT SRAM 2

Memory Select

Rendering Engine and Control

3-D Model ROM

Camera control

Status signals (Start/Busy)

Data bus 1

Data bus 2

Addr 20

Ctrl 2

Data 36

Data 36

Ctrl 1

Ctrl 3
Double Buffering of Frames

- Write to ZBT Memory
- Reset
- Read frames from ZBT Memory 2
- Write to ZBT Memory 1
- Read frames from ZBT Memory 2
- Write to ZBT Memory 1

States:
- start
- busy
- !busy

Flow:
- From Write to ZBT Memory to start
- From start to Read frames from ZBT Memory 2
- From Read frames from ZBT Memory 2 to Write to ZBT Memory 1
- From Write to ZBT Memory 1 to !busy
- From !busy to start

Conditions:
- !busy
- busy
Project Timeline

Phase I
• Is there video output (without rendered model) 4/22
• Is the initial transform functional 4/24
• Is there video output showing rendered model 4/27
• Simple Shader (Vertices -> White Dots) 4/27
• Polygon Shader (including color) 5/1

Phase II
• Camera Control 5/1
• Lighting 5/4
• Debugging/Additional Features 5/9