InfiniteReality™ Graphics – Power Through Complexity

John Montrym

Brian McClendon

Advanced Systems Division

Outline

Goals

Architecture

Performance

Verification

Project Statistics

Major Application Areas

Visual Simulation

Professional Film / Video

Industrial Design

Image processing

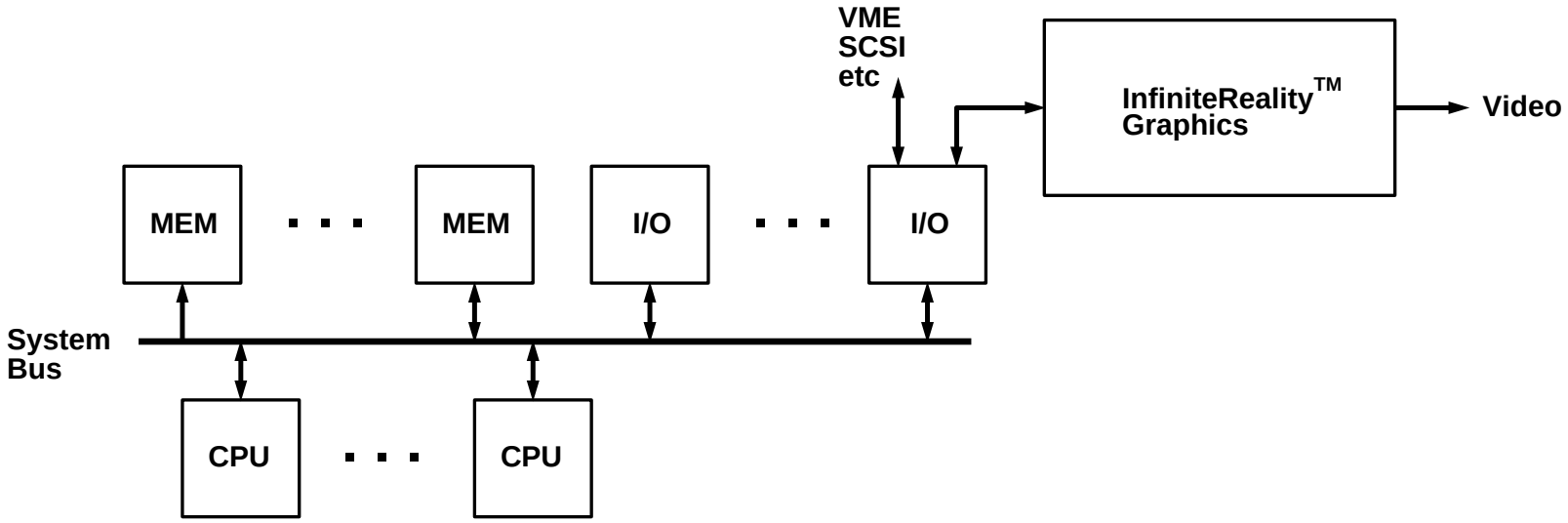
Goals

Each key feature addresses more than one application area:

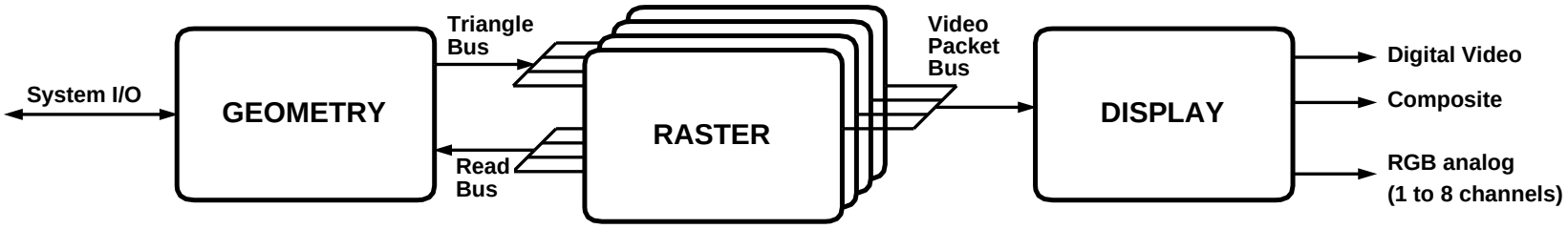
- **Textured antialiased pixel fill rate − never enough**
- **Geometry transform rate**
- **Multisample antialiasing**
- **Large resident textures**
- **Large virtual textures**
- **Texture paging**
- **Multiple independent video output channels**
- **Dynamic variable output resolution**
- **User−programmable video**
- **Fully OpenGL™ graphics library compatible**

Architecture

System Block Diagram



Graphics Subsystem Block Diagram



1, 2 or 4 RM boards

Operations

Per−vertex

Transform object space to screen space

- model−view & projection matrices
- perspective division

Clip−test

Compute phong model lighting

Compute generated texture coordinates

Per−primitive

Clip to clip volume

Imaging

Convolution

LUT

Histogram

Implementation

MIMD array of 4 Geometry Engine™ Processors

Work is assigned to least−busy processor

Output is re−assembled in original order and sent to Raster Subsystem through large FIFO

Hard problems

- State management
- Context switching

Geometry Engine™ Processor

Custom processor designed because DSP & processor vendors were not pushing floating point hard enough

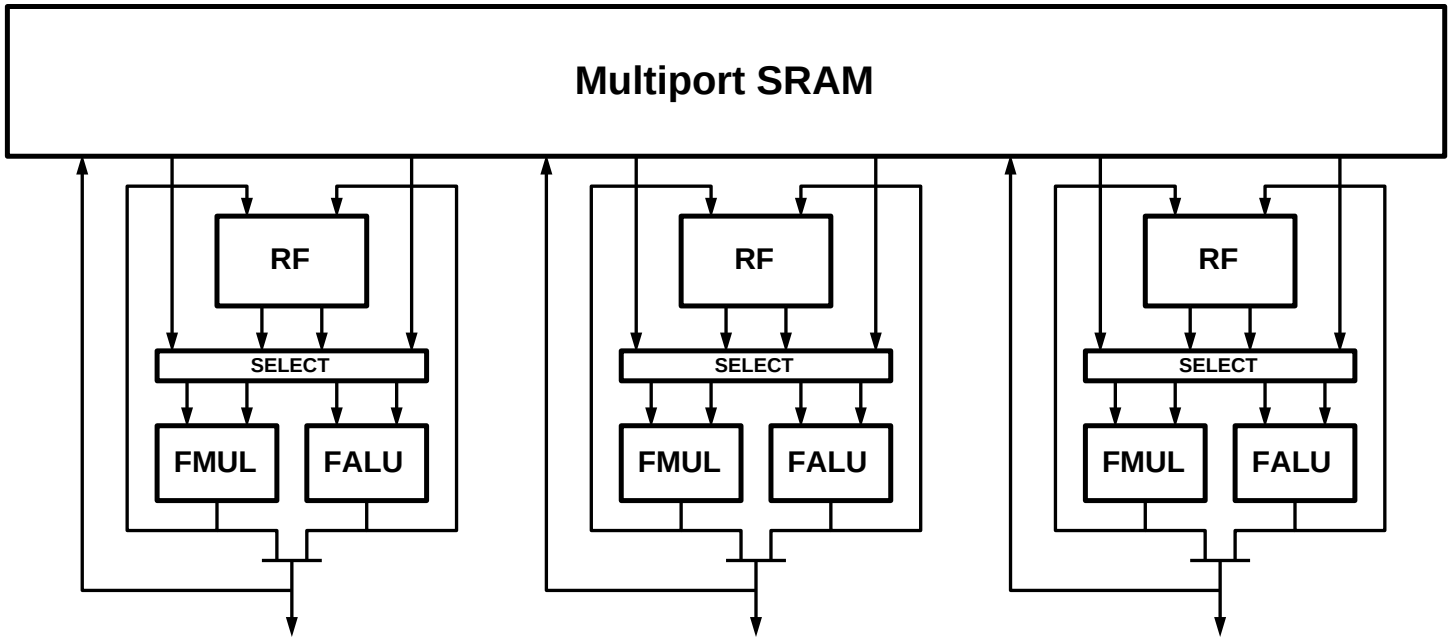
SIMD best way to get MFLOPS / $

Tuned memory system essential to keep FPUs busy

Custom support logic accelerates graphics−specific operations

540 MFLOPS peak (3 SIMD FP cores @ 90 MHz)

195 bit microinstruction



Hardware Tradeoffs

We used a quick−and−dirty scheduler to evaluate performance of key code fragments on alternative architectures by varying:

- FP MUL & ALU pipeline depth versus clock speed
- Register file size & number of ports
- SRAM partitions & number of ports
- Address unit resources
- Penalty for double−precision operations

Microcode Implementation

Tuned for real scenes

- Short triangle−strips common (4−12 vertices)
- Frequent mode changes

Requires careful mapping of data to SIMD processors

Modular microcode avoids steep performance "cliff"

Assembler used for critical microcode

Proprietary scheduler used for the remaining 80 percent. 10x productivity improvement over hand−packed assembly

Microcode width compression method acheived 2.7x reduction with less than 1.5% impact on performance and code length.

Triangle Bus

T−bus is the natural recombination point after transformation and screen−space projection.

400 MB/sec available given compatibility constraints.

High triangle rate goal forced us to minimize traffic at T−bus.

T−bus moves transformed vertices so triangle re−assembly and setup must be performed in the raster subsystem.

Raster Subsystem

1, 2, or 4 Raster Manager (RM) boards

Efficiently scan converts points, lines, triangles into fragments

Per−fragment calculations:

- **subpixel coverage mask**
- **color**
- **depth**
- **perspective−correct texture lookup**
- **mip−map & bicubic filtering (1D, 2D, 3D)**
- **texture application**
- **fog**
- **line & area stipple**

Framebuffer operations per−sample (1,4, or 8 samples per pixel):

- **stencil & depth tests**
- **blend function**
- **logic operation**

Texture Memory

Peak performance defines total memory bandwidth:

2 byte/sample * 8 sample/pix * 960 M pix/s => 15.36 GB/s

Texel re−use hides page miss & refresh overhead

128 x16−wide SDRAMs with individual address and data buses

One copy of texture memory per RM board

Texture capacity determined by available SDRAMs

64 MB available to the user with 16 Mb parts

Framebuffer Memory

320 processors; each owns one 256Kx32 SGRAM

Fine−grain tiling for best load balancing

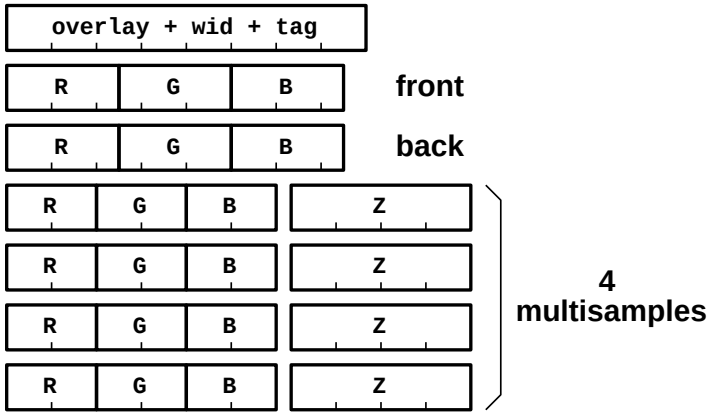
Choice of 256, 512, or 1024 bits per pixel

Total data bandwidth 76.8 GB/s

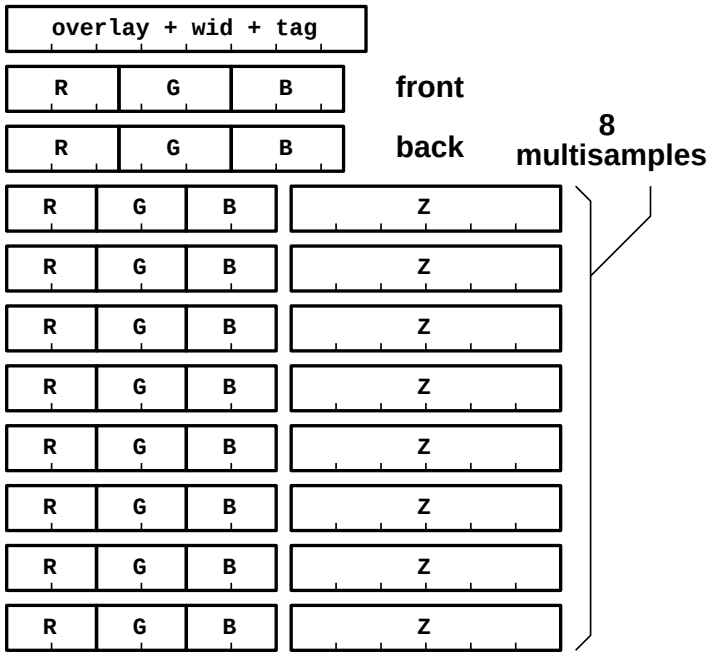
320 MB supports 2560x2048 pixels at 512 bits/pixel

Framebuffer Pixel Format Examples

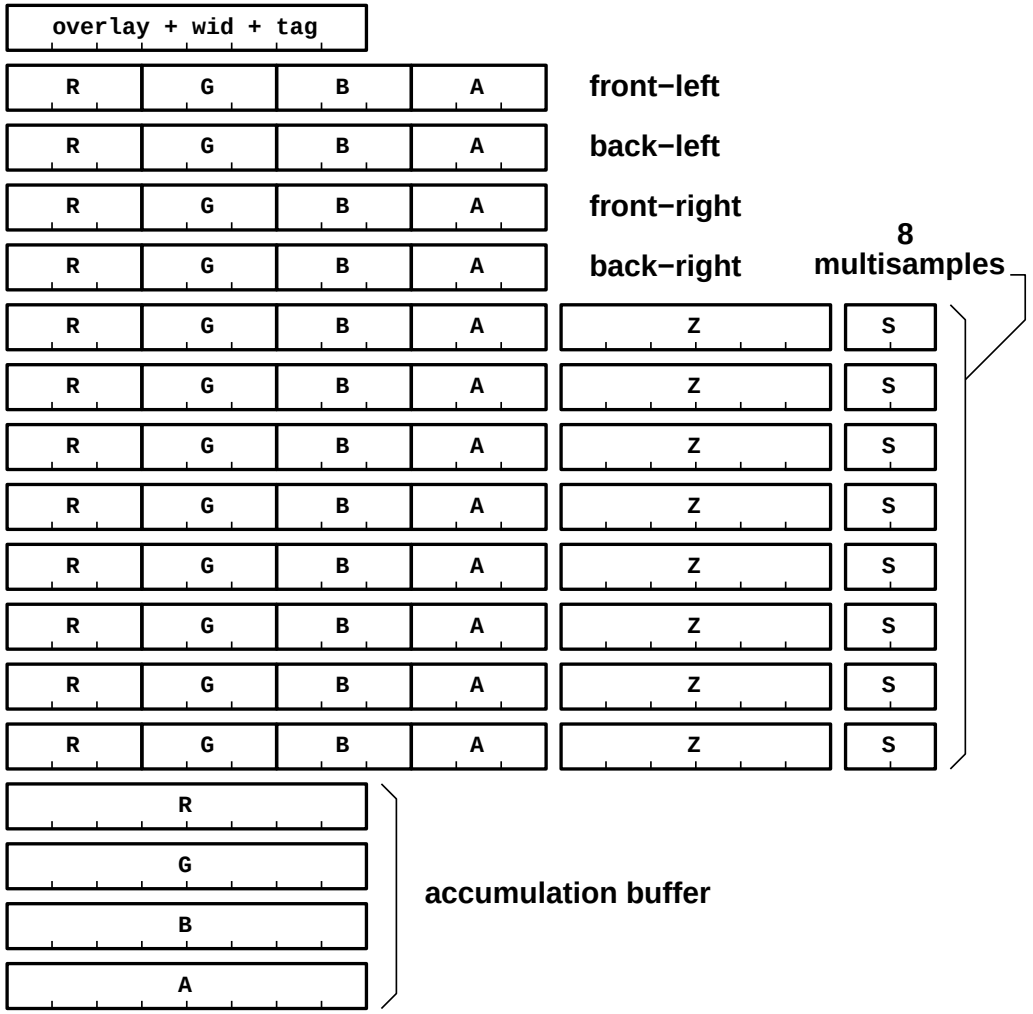
256 bits per pixel



512 bits



1024 bits



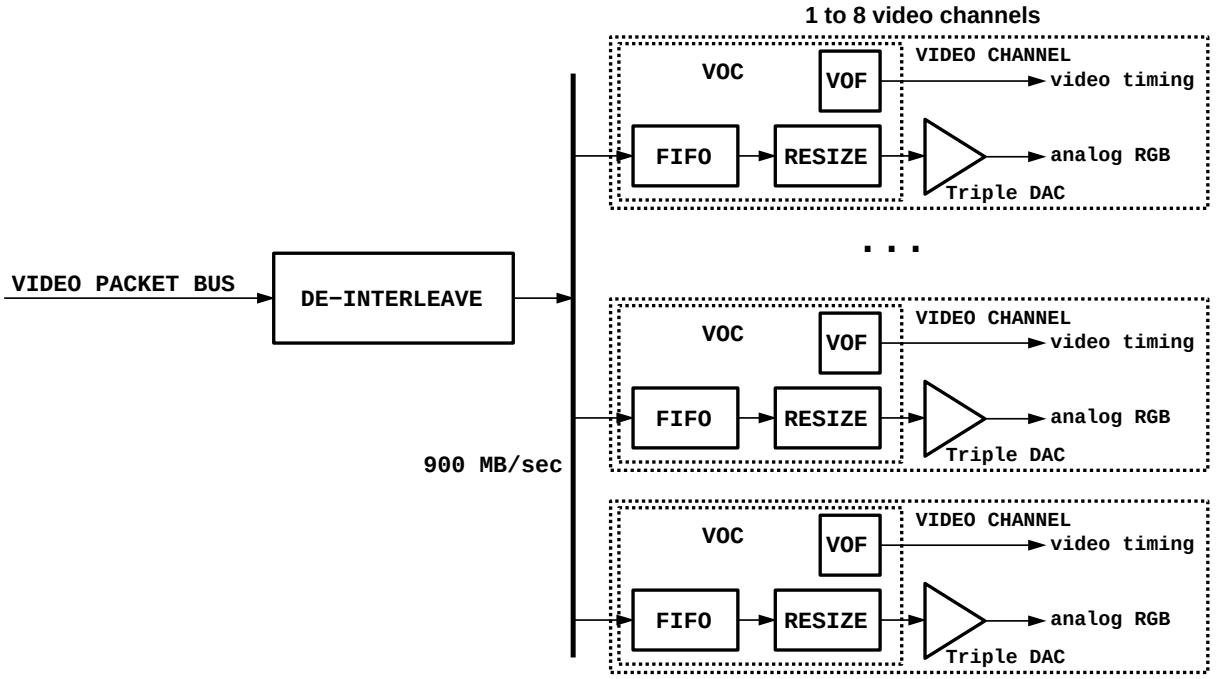
Display Subsystem

Multiple channel controllers independently request blocks of 160 pixels.

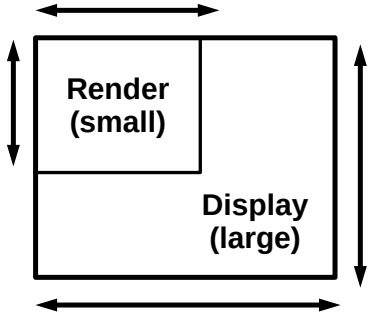
Video packets from raster subsystem are de−interleaved into horizontal pixel streams.

Resize engine performs magnification or minification.

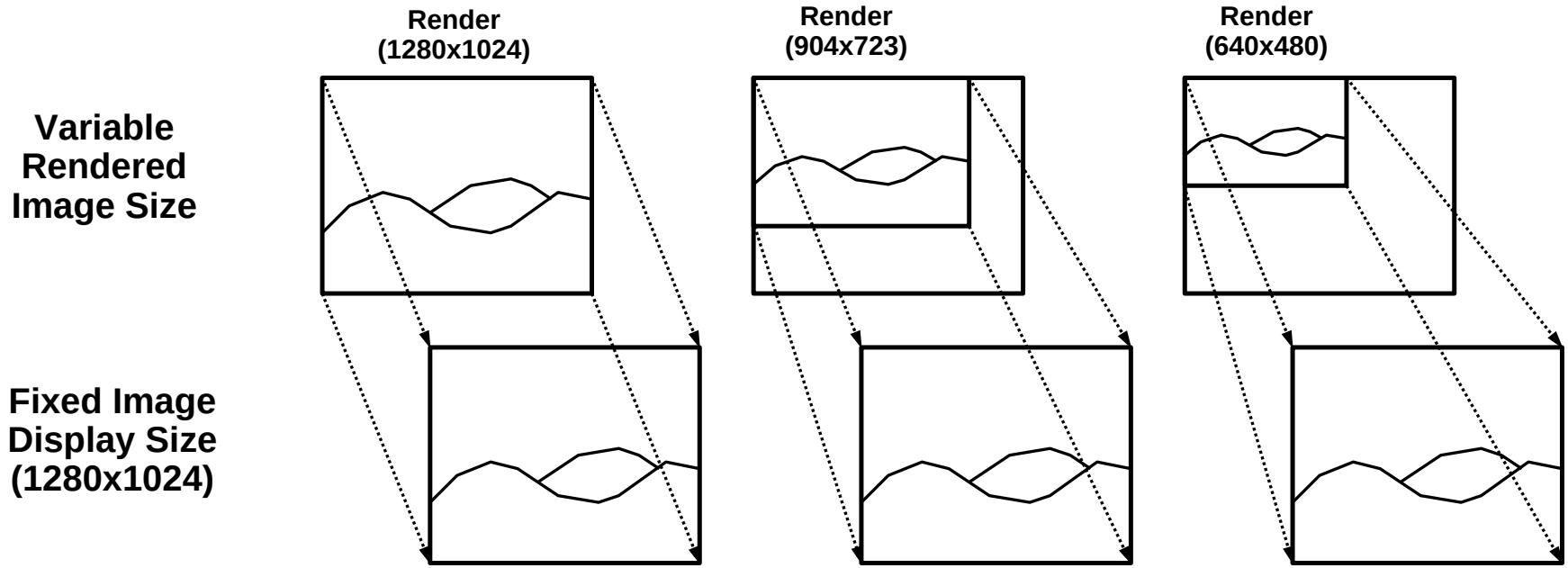
Video format compiler and combiner provided to customers.



Dynamic Variable Resolution



- Independent X and Y axis zoom factors
- Zoom factor specified each frame
- No additional latency due to zoom



Performance

11 M triangles / sec	**non−lighted, depth−buffered, antialiased triangle strip; 40 pixel triangles**
8.3 M triangles / sec	**textured, depth−buffered, antialiased triangle strip; 50 pixel triangles**
.80 G pixels / sec	**trilinear mip−map textured, 16−bit texel, depth−buffered**
.75 G pixels / sec	**trilinear mip−map textured, 16−bit texel, 4 sample antialiased depth−buffered**
6.4 G pixels / sec	**screen clear of 32−bit color & depth**
300 M displayed pixels / sec	**total over 1 to 8 output channels**

Verification

Full C−simulation of every component from system I/O interface to video DACs.

Simulation backplane allowed Verilog & VHDL models to be substituted into the C−sim as needed.

C−simulator was used to verify and debug operational microcode plus OS kernel, graphics library, and diagnostic software.

Project Statistics

55 engineers on hardware, software, verification

3 board designs

12 ASIC designs (process: 0.5 micron, 3LM, 3.3v)

430,000 lines of verilog

840,000 lines of C & microcode

2.7 million unique gates

251 million ASIC transistors and 596 million DRAM bytes in maximum system

System was demonstrated to customers 7 days after the last ASIC was received

First pass architecture: 9/93; First customer shipment: 3/96