
Blitzen: Lightning Speed 3D Geometry Accelerator



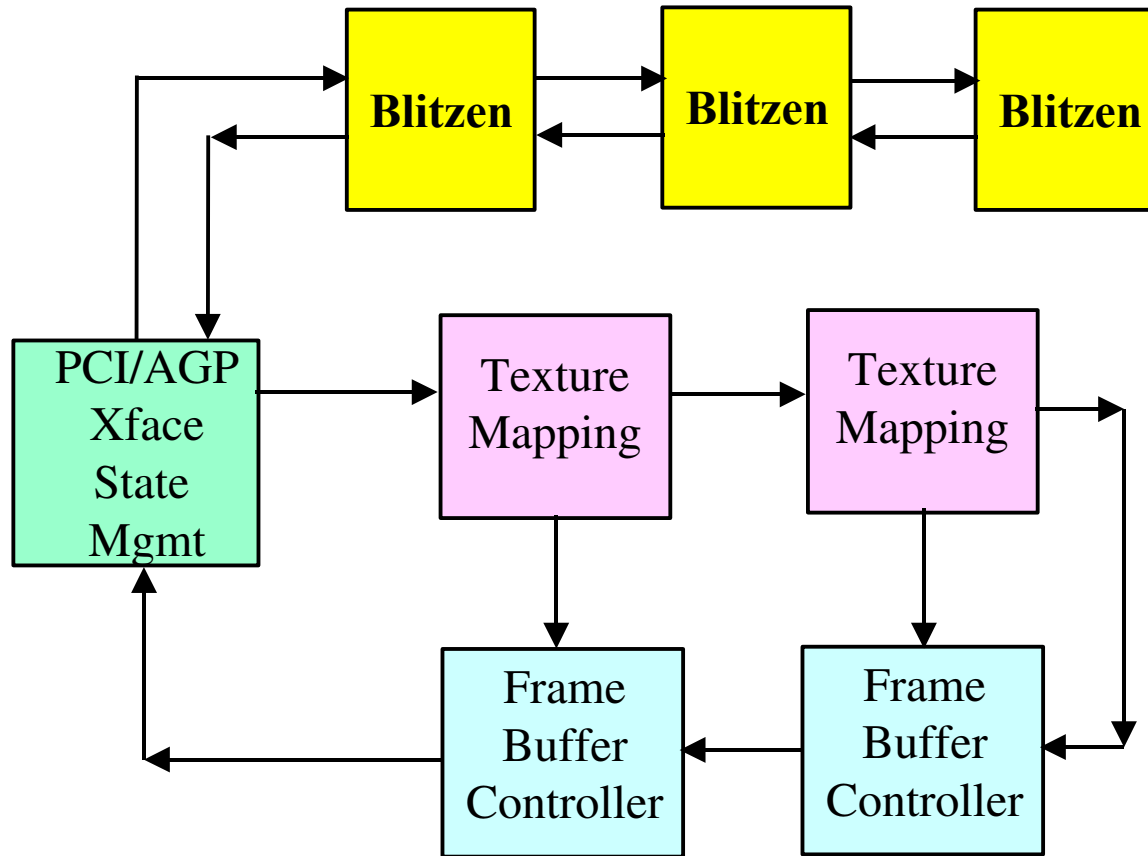
Alan Krech

Personal Systems Group
Hewlett-Packard Company

Blitzen Project Goals

- Provide World-class 3D Geometry Acceleration
- API Independence--OpenGL, Starbase, PEX
- Platform Independence--Unix, PC
- Low-cost
- Short Time-to-market
- Highly Scalable Architecture

System Block Diagram



Key Blitzen Features

- Geometric Transformations--Triangles, Quads, and Vectors
- Lighting (up to 8 sources directional or positional)
- View Volume Clipping
- Plane Equation Computation
- Texture Mapping Set-up

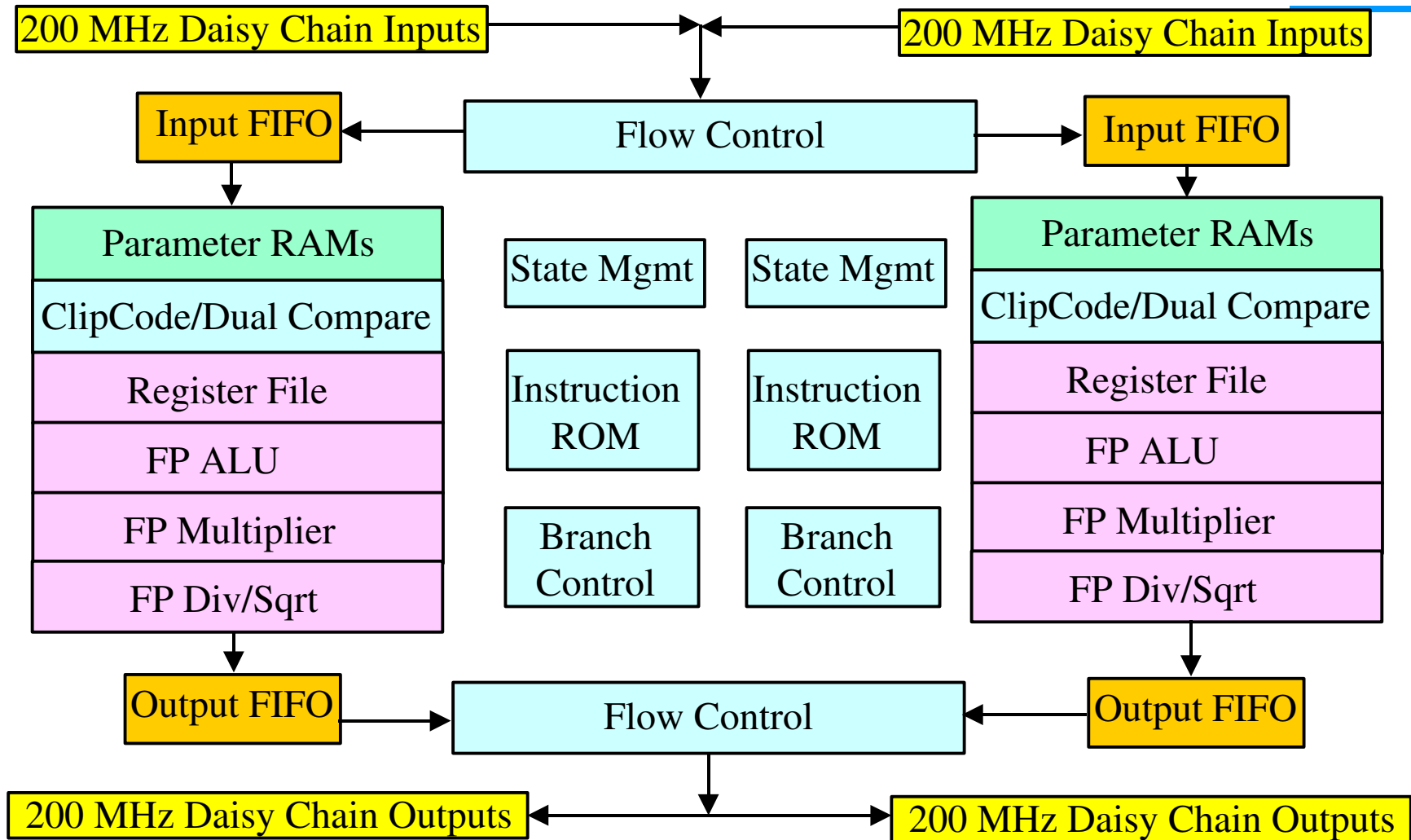
Additional Blitzen Features

- Fog and Depth Cueing
- Texture Coordinate Generation
- 6-plane Model Clipping
- Environment Mapping Set-up
- Front/Back Face Culling and Lighting
- Full-featured OpenGL Programming Interface and Starbase API Support
- Separate Specular Lighting Application

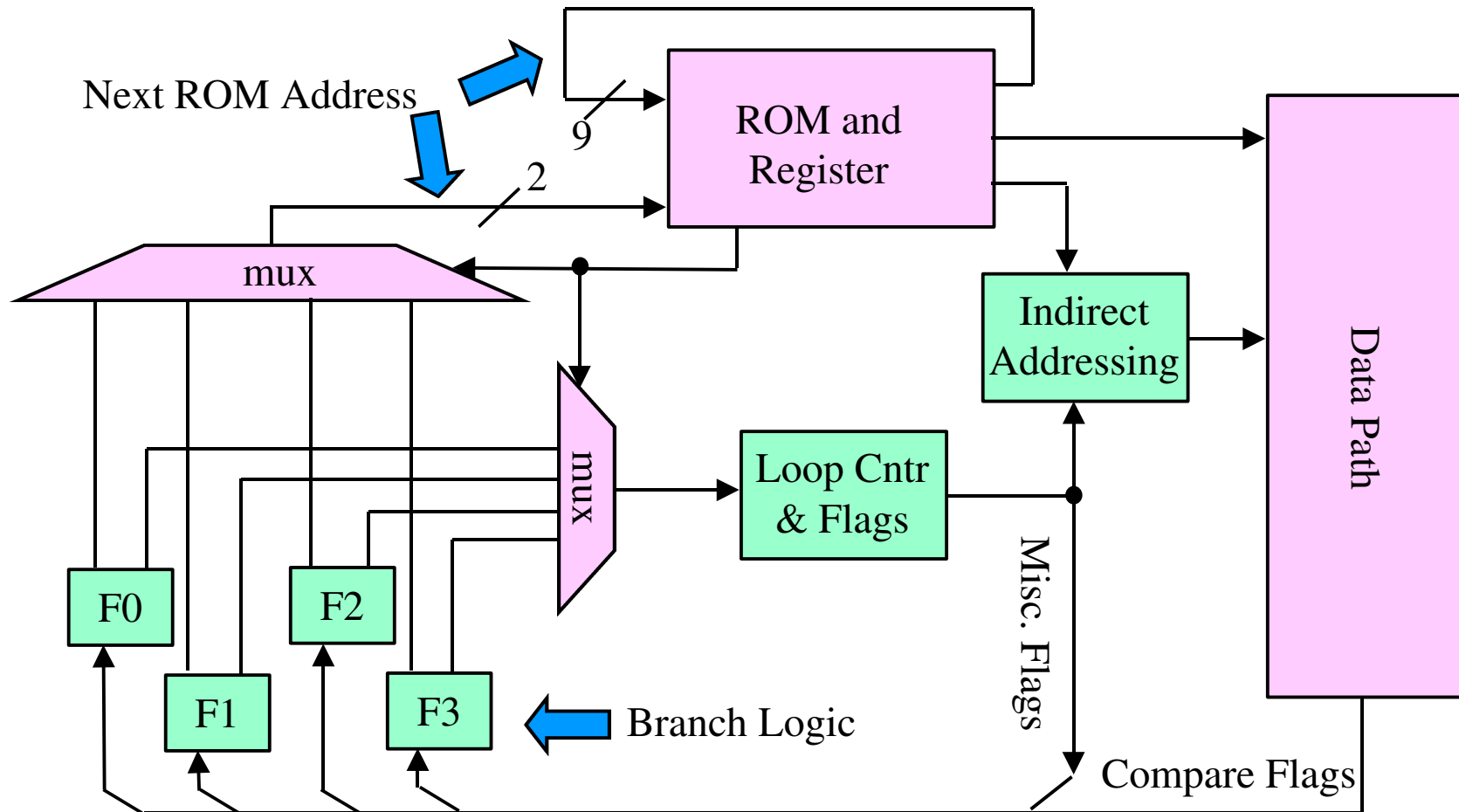
Blitzen Design

- ❑ Two Independent and Identical Slices
 - ✓ HP-PA Floating Point Core--ALU, Multiplier, Divider, and Register File
 - ✓ Wide-word Instruction ROM
 - ✓ Micro-instruction Branching Control
 - ✓ Multiple Double-buffered RAMs
 - ✓ OpenGL State Management Logic
 - ✓ Flow Control for Multiple Slices/Chips

Blitzen Block Diagram



Blitzen ROM Design



Blitzen ROM Design

- ❑ Up to 8-way No-penalty Branching
- ❑ Independent Branching Control
- ❑ Indirect Addressing for Code Reduction
- ❑ Looping Control For Use of Subroutines
- ❑ Global Flags with Feedback for “State Machine” Intercommunication
- ❑ Wide-word Instruction Controls Entire Data path

Why Not Use a CPU?

- ❑ Instruction Set Is Limited--fadd, fsub, fmpy, fmac, fdiv, etc.--inefficient algorithms.
- ❑ Data Origination/Destination Is Limited--fastest performance requires data reside only in register file.
- ❑ Very Complex Instructions Must Be Unwrapped--e.g. clip-code generation.
- ❑ Complex Branching Costly--wasted states.

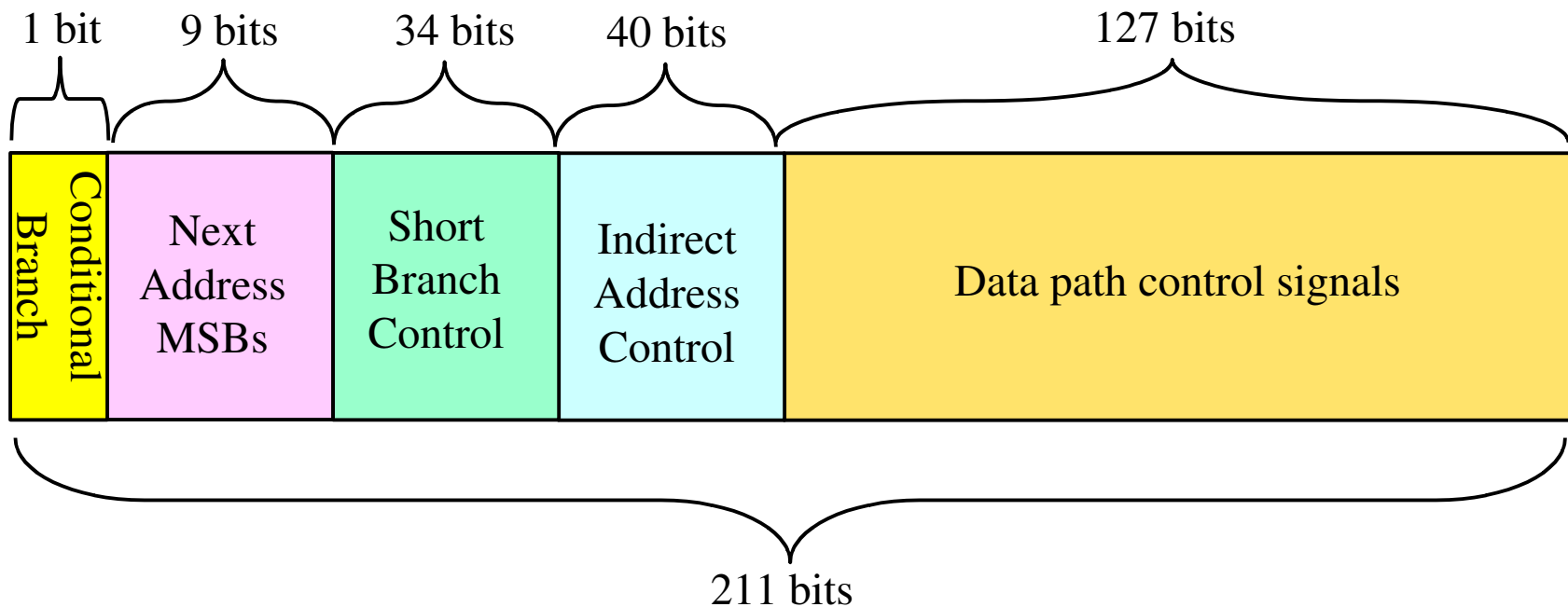
The Blitzen “CPU” Design

- ❑ Leveraged from HP-PA Design
- ❑ Modified ALU Instructions for Enhanced Graphics Support
 - ✓ Add and Subtract with Clamp
 - ✓ Special Truncate/Round/Integer/Fraction Instructions for Sub-pixel Adjustment
- ❑ Separated Bussing
 - ✓ More Instructions Per Clock Cycle
 - ✓ Increased Data Sources/Destinations

The Blitzen “CPU” Design

- ❑ Customized Register File Size/Orientation
- ❑ Both Complex and Predictive Branch Logic Implemented in Separate Control Blocks and in ROM Code for Optimum Flexibility and Performance.
- ❑ Fully Scalable With No Software Impact--threading, load balancing, etc.
- ❑ Single Slice Faster than 200MHz CPU With Tuned Code

Blitzen Control Instruction



Blitzen Time-to-market Enhancements

- ❑ Leverage of Floating Point Core
- ❑ ROM Design Reduced Synthesis/Timing Loop--Small, fixed areas of concentration
- ❑ ROM Design Allows Low-risk Chip Turns
- ❑ Design Easily Partitioned As Micro-code
- ❑ Testability Maximized Throughout

Blitzen Products

☐ HP Unix Workstations

- ✓ Visualize FX-2, FX-4, and FX-6 products

☐ HP PC Workstations

- ✓ HP Kayak XW Ultimate w/Visualize FX-6 and Pentium II Xeon
- ✓ HP Kayak XW Ultimate w/Visualize FX-4 and Pentium II

Blitzen Performance

- ❑ 624 MFLOPS/chip
- ❑ 3.3 Million Triangles/sec/chip
- ❑ 2 Million Lit Triangles/sec/chip
- ❑ 6.2 Million Vectors/sec/chip

Product Performance

	HP Kayak XW/ Visualize fx6	HP Kayak XW/ Visualize fx4	HP C240/ Visualize fx6
CDRS-03	181	138.6	200
DX-03	26.24	20.1	28.98
DRV-04	18.4	12.7	16.64
Light-01	2.56	1.68	2.17
Awadvs-01	52.02	31.5	52.02

Blitzen Chip Facts

- ❑ 1.6 Million Transistors
- ❑ I/O Frequency 200MHz (HSTL)
- ❑ Internal Clock Frequency 156MHz
- ❑ 300 pin CPGA package
- ❑ 10% RAM

Blitzen Summary

- ❑ Need Scalable Designs--decrease R&D cost
- ❑ Leverage Where Possible--good floating point
- ❑ Customize For Significant, Measurable Performance Gains--instructions, bussing
- ❑ Simplify Without Sacrifice--Micro-code ROM for easy coding and timing closure with branch/flag logic for performance.