Graphics Hardware Workshop 2005

Cell Technology for Graphics and Visualization

Bruce D'Amora
Emerging Systems Software
IBM T.J. Watson Research Center
http://www.research.ibm.com

August 11, 2005

Agenda

- Introduction
- Architecture
- Programming
- Graphics & Visualization
- Demos
- Questions

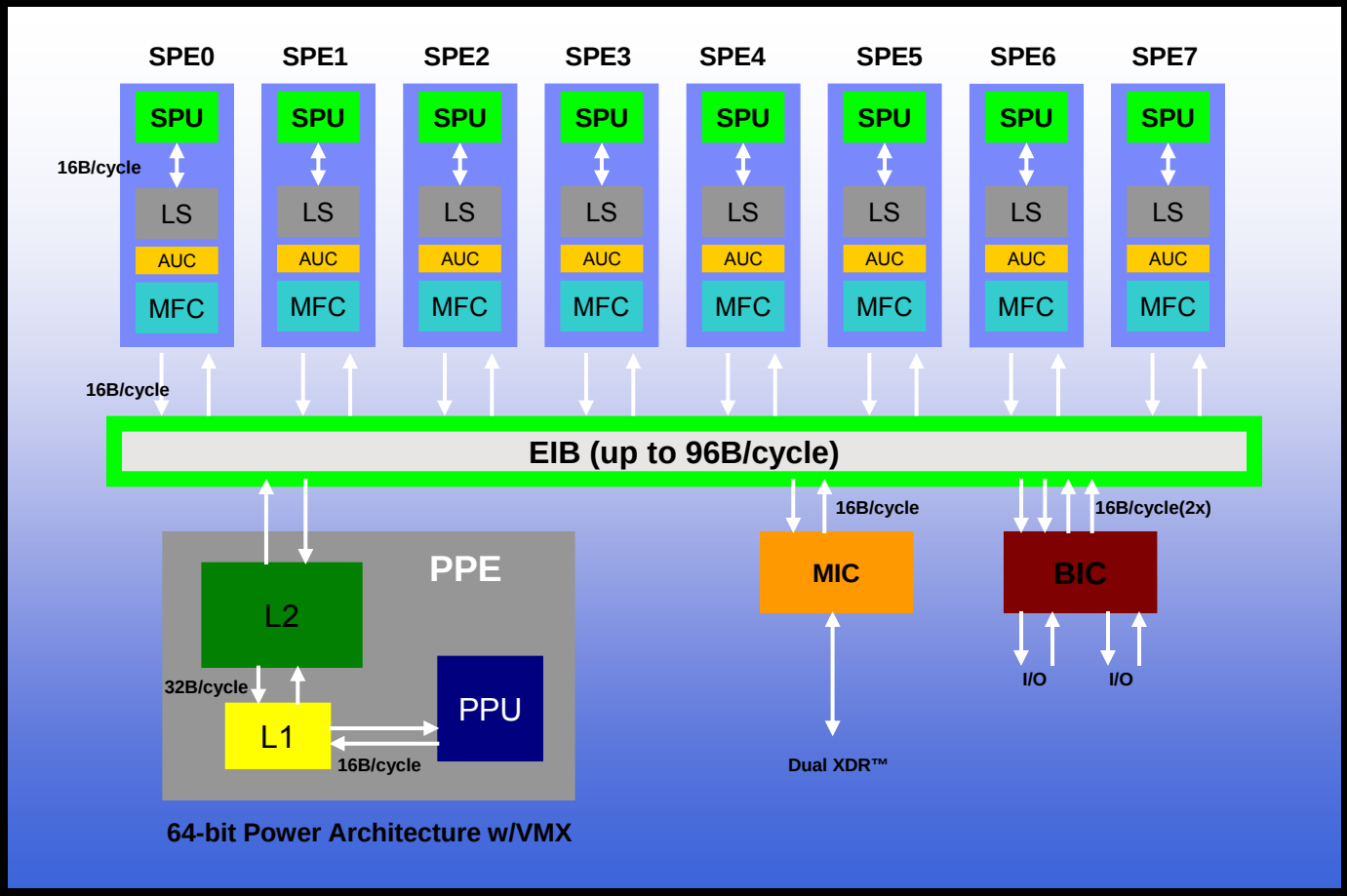
Introduction

What is Cell?

- Sony, Toshiba, IBM alliance established in March, 2001
  - "STI" Design Center is located in Austin, Texas
- Goals
  - Outstanding performance, especially for game/multimedia
  - Real time responsiveness to the user and the network
  - Applicable to a wide range of platforms and applications
- Design Concepts
  - Compatibility with 64-bit Power Architecture™
  - Increased efficiency and performance
  - Non-homogeneous architecture

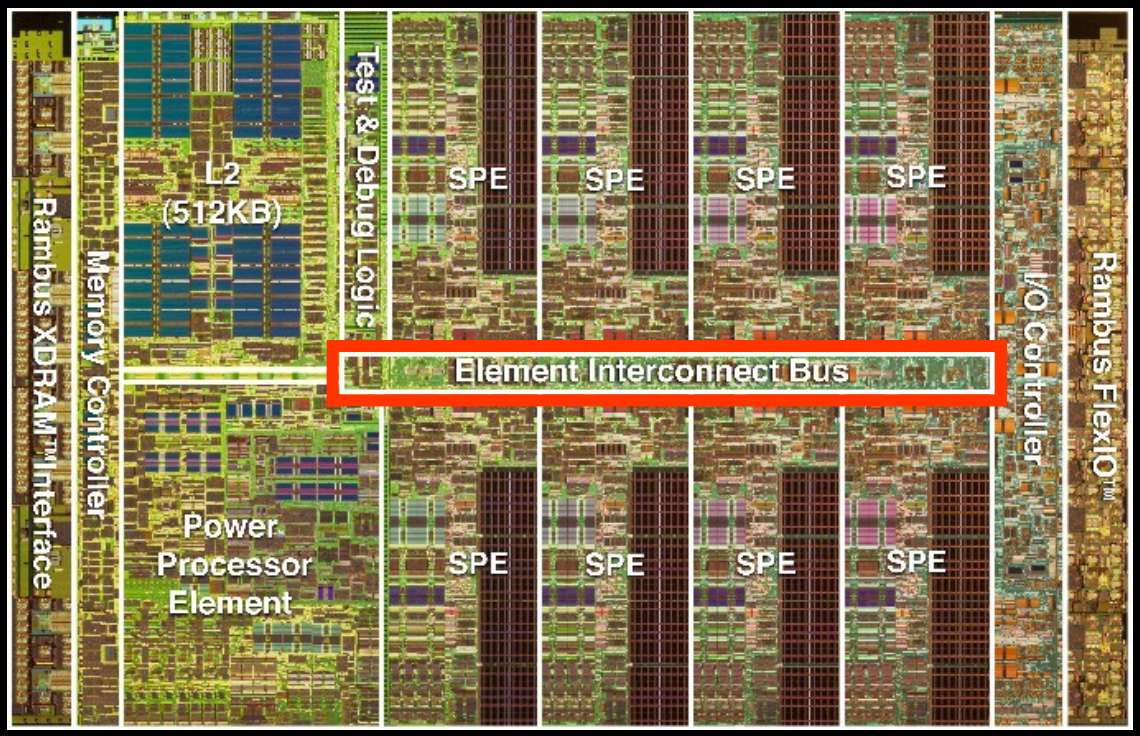
Architecture

Cell Processor Architecture



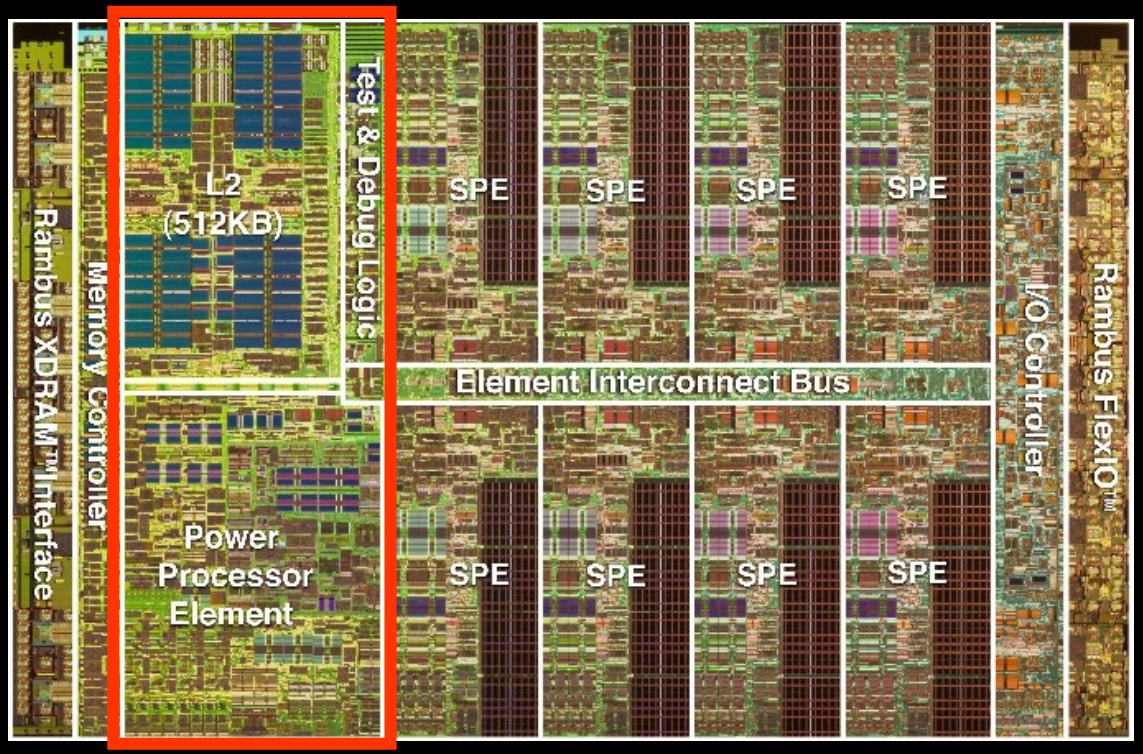
Element Interconnect Bus

- EIB data ring for internal communication
  - Four 16 byte data rings, supporting multiple transfers
  - 96B/cycle peak bandwidth
  - Over 100 outstanding requests



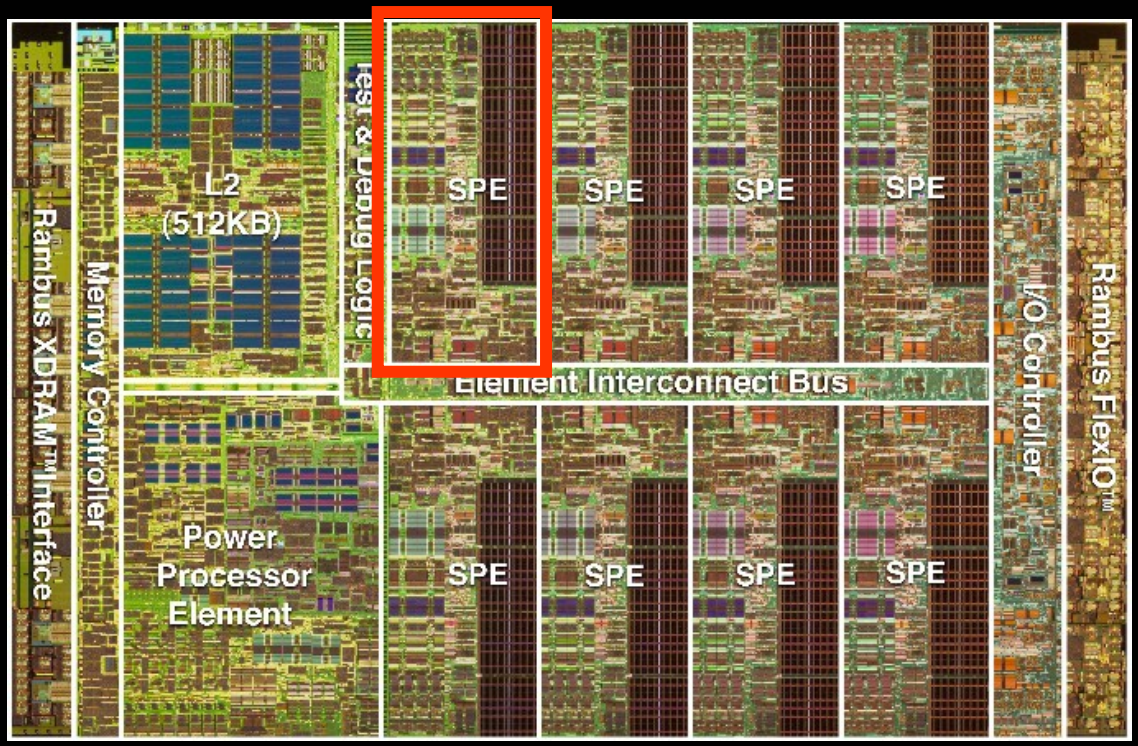
Power Processor Element

- PPE handles operating system and control tasks
  - 64-bit Power Architecture™ with VMX
  - In-order, 2-way hardware symmetrical multi-threading (SMT)
  - Load/Store with 32KB I & D L1 and 512KB L2

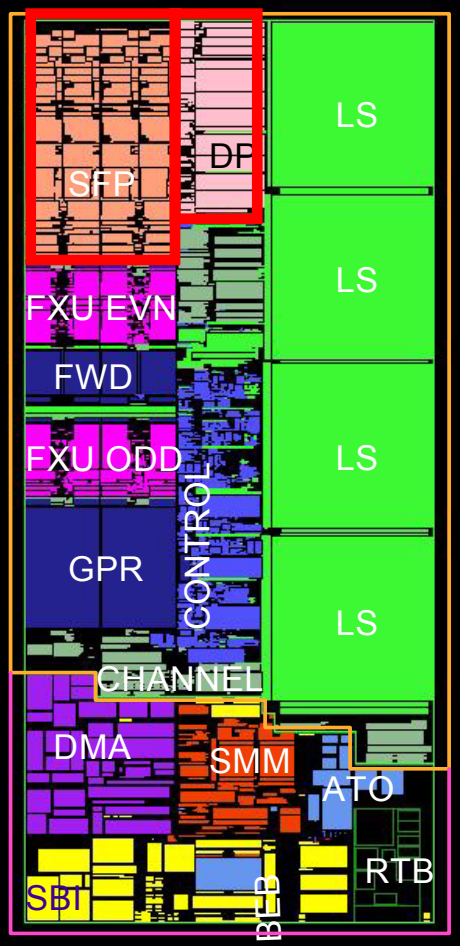


Synergistic Processor Element

- SPE provides computational performance
  - Dual issue, up to 16-way 128-bit SIMD
  - Dedicated resources: 128 128-bit register file, 256KB Local Store
  - Each can be dynamically configured to protect resources
  - Dedicated DMA engine: Up to 16 outstanding requests per SPE



SPE Highlights

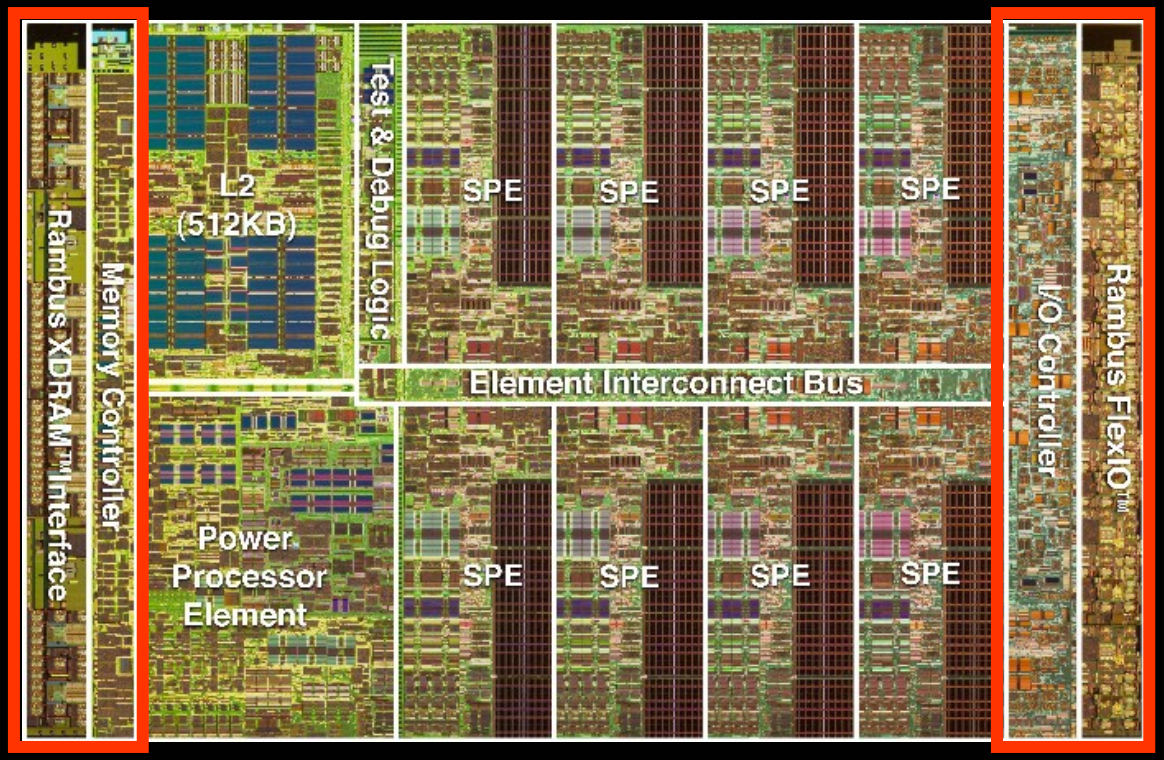


14.5mm² (90nm SOI)

- User-mode architecture
  - No translation/protection within SPU
  - DMA is full Power Architecture protect/x-late
- RISC like structures
  - 32 bit fixed instructions
  - Clean design – unified Register file
- VMX-like SIMD dataflow
  - Broad set of operations (8 / 16 / 32 Byte)
  - Graphics SP-Float
  - IEEE DP-Float
- Unified register file
  - 128 entry x 128 bit
- 256KB Local Store
  - Combined I & D
  - 16B/cycle L/S bandwidth
  - 128B/cycle DMA bandwidth

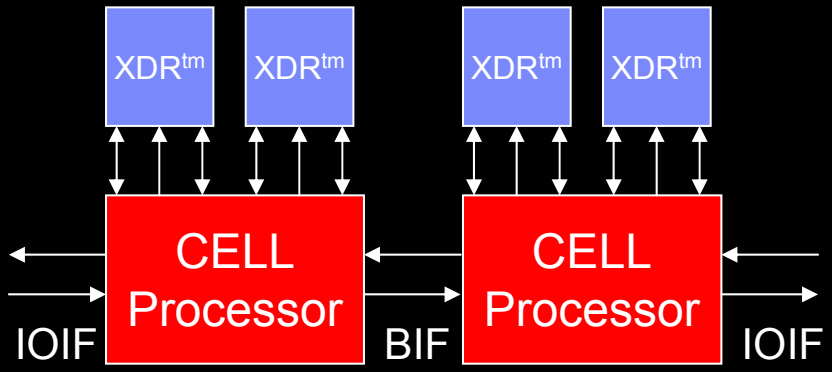
I/O and Memory Interfaces

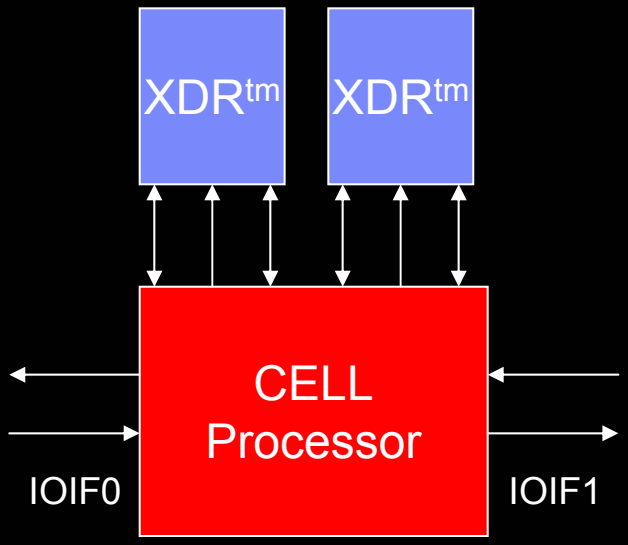
- I/O Provides wide bandwidth
  - Two configurable interfaces
  - Up to 25.6 GB/s memory B/W
  - Up to 70+ GB/s I/O B/W
    - Practical ~ 50GB/s

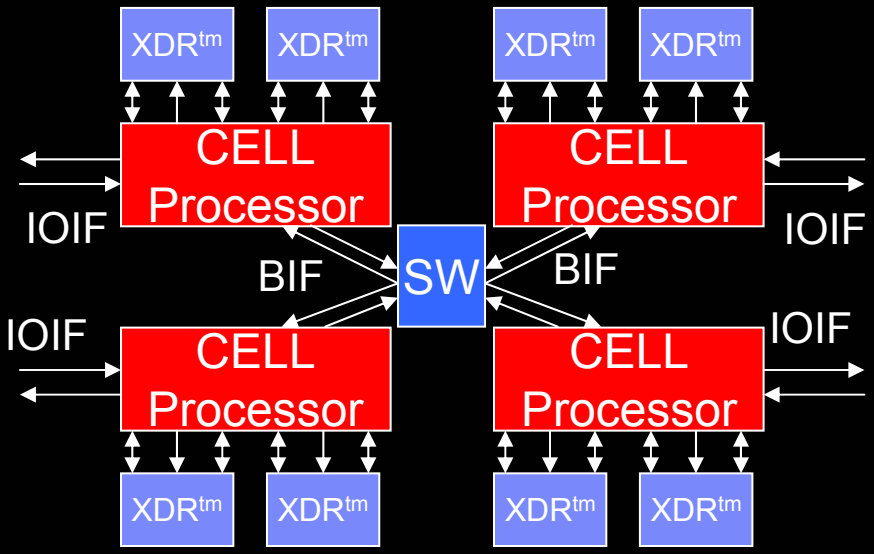


Configurability

- **Direct Attach XDR**
- **Two I/O interfaces**
  - Configurable number of Bytes
  - Coherent or I/O Protection





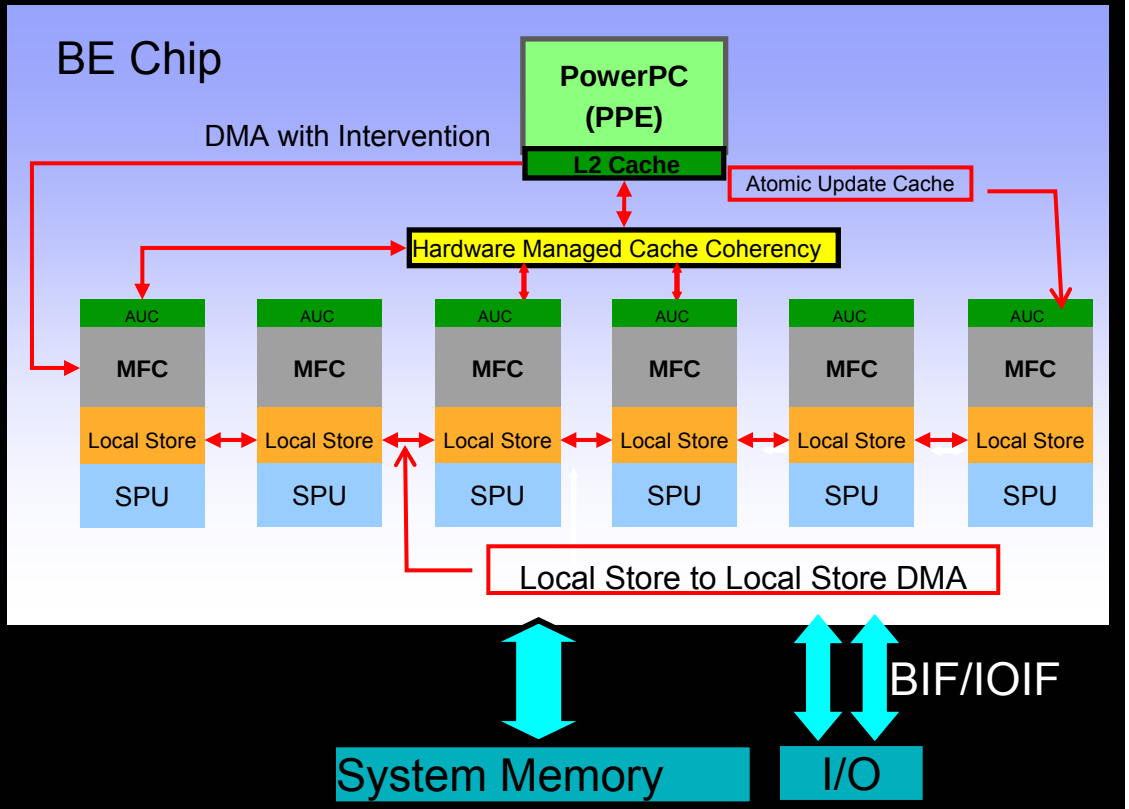


Programming

BE Features Exploited by Software

- Keeping Intermediate/Control Data on-Chip
  - MMU – SLBs, TLBs
  - DMA from L2 cache-> LS
  - LS to LS DMA
  - Cache <-> Cache transfers (atomic update)
  - SPE Signalling Registers
  - SPE <-> PPE Mailboxes

- Resource Reservation and Allocation
  - PPE can be shared across logical partitions
  - SPEs can be assigned to logical partitions
  - SPEs independently or Group Allocated

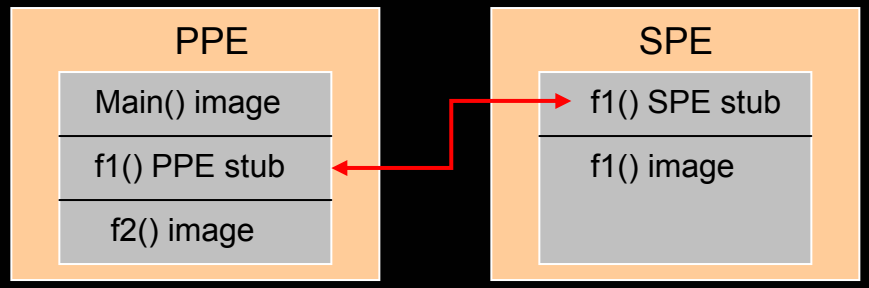


Models for Programming the Cell

- Function Offload
- Application Specific Accelerators
- Computation-Acceleration
- Streaming
- Shared Memory Multi-processor
- Heterogeneous Thread Runtime Model
- Single Source Compiler

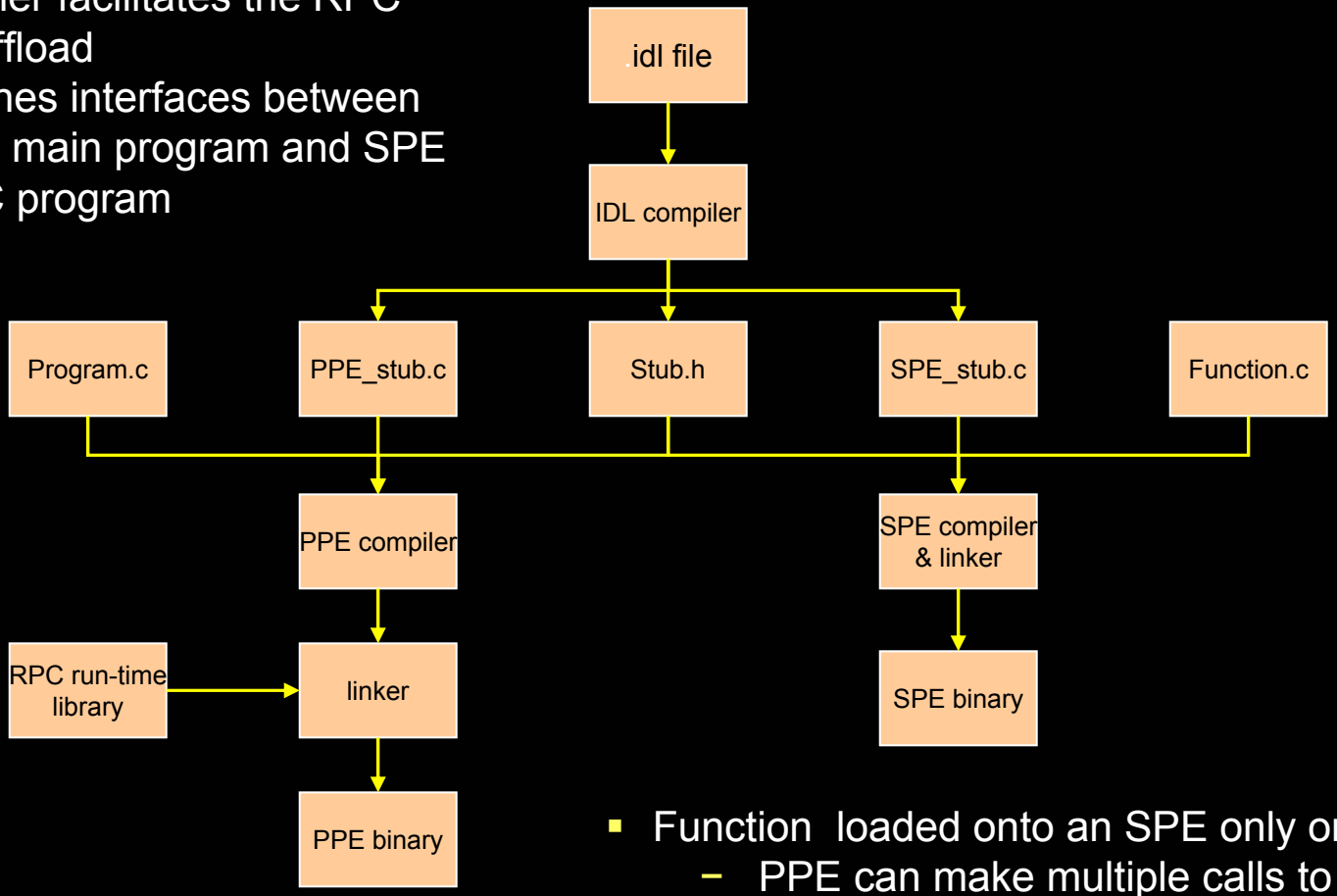
Function Offload

- Easiest way to port applications to Cell
- PPE can call procedure located on the SPE as if it were a local PPE function
- PPE and SPE use “stubs” as placeholders for local and remote procedures



Function Offload via IDL

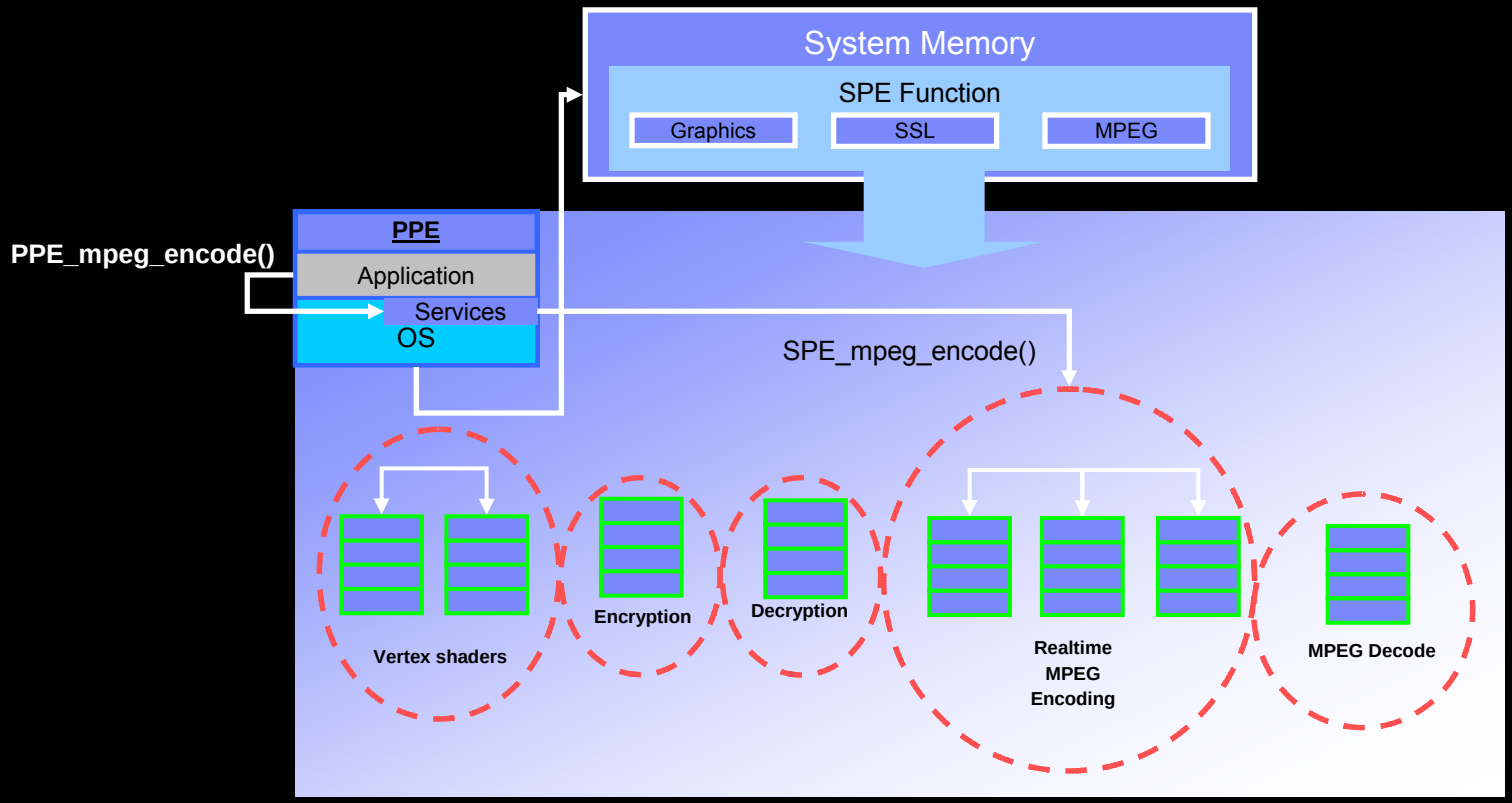
- IDL compiler facilitates the RPC function offload
  - Defines interfaces between PPE main program and SPE RPC program



- Function loaded onto an SPE only once
  - PPE can make multiple calls to that procedure without having to reload it.

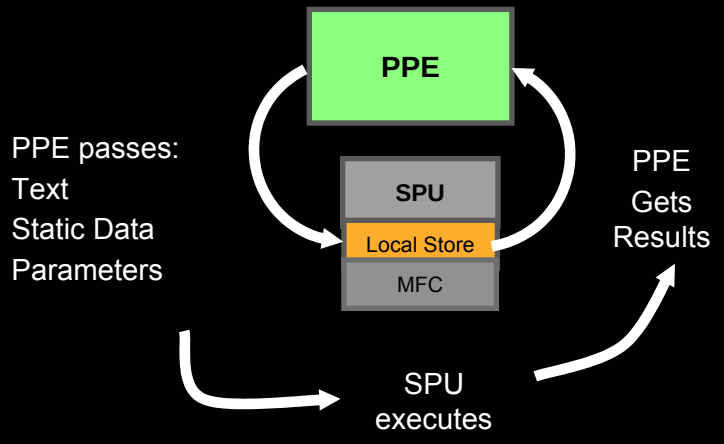
Special Case: Application Specific Accelerators

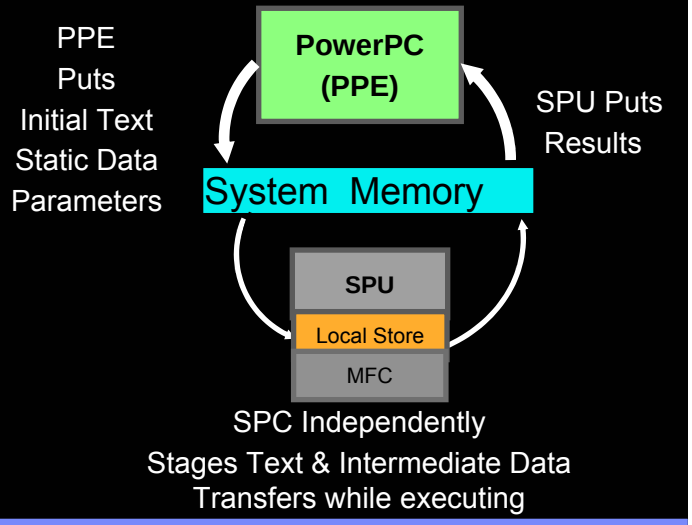
- OS allocates and initializes SPE resources
- OS services invokes SPE function



Computational Acceleration

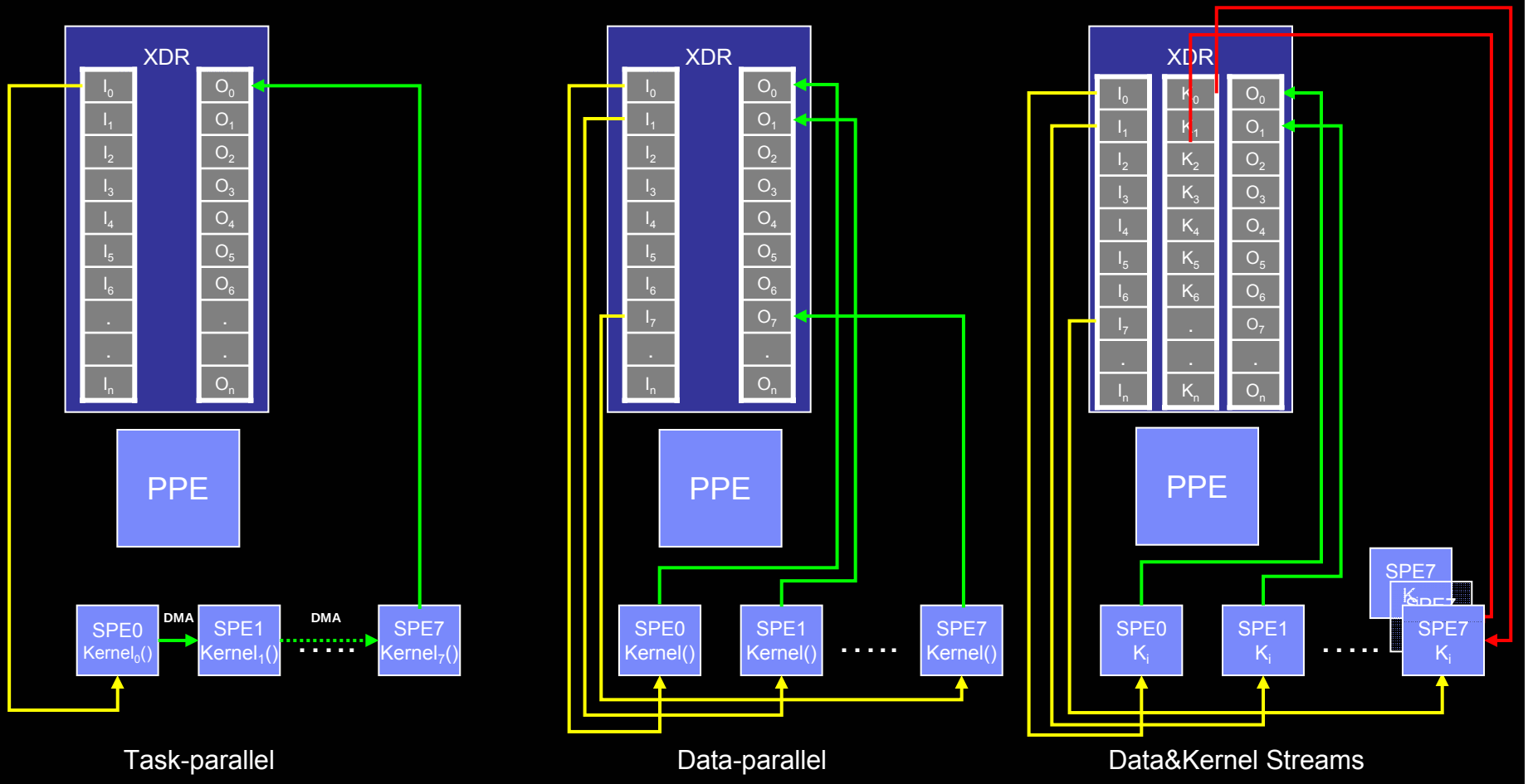
- User created RPC libraries
  - User acceleration routines
  - User compiles SPE code
- Local Data
  - Data and Parameters passed in call
- Global Data
  - Data and Parameters passed in call
  - Code manages global data





Streaming Model

- SPE initiated DMA
- Input/output/kernel streams
  - –DMA'd from system memory->LS or LS->LS

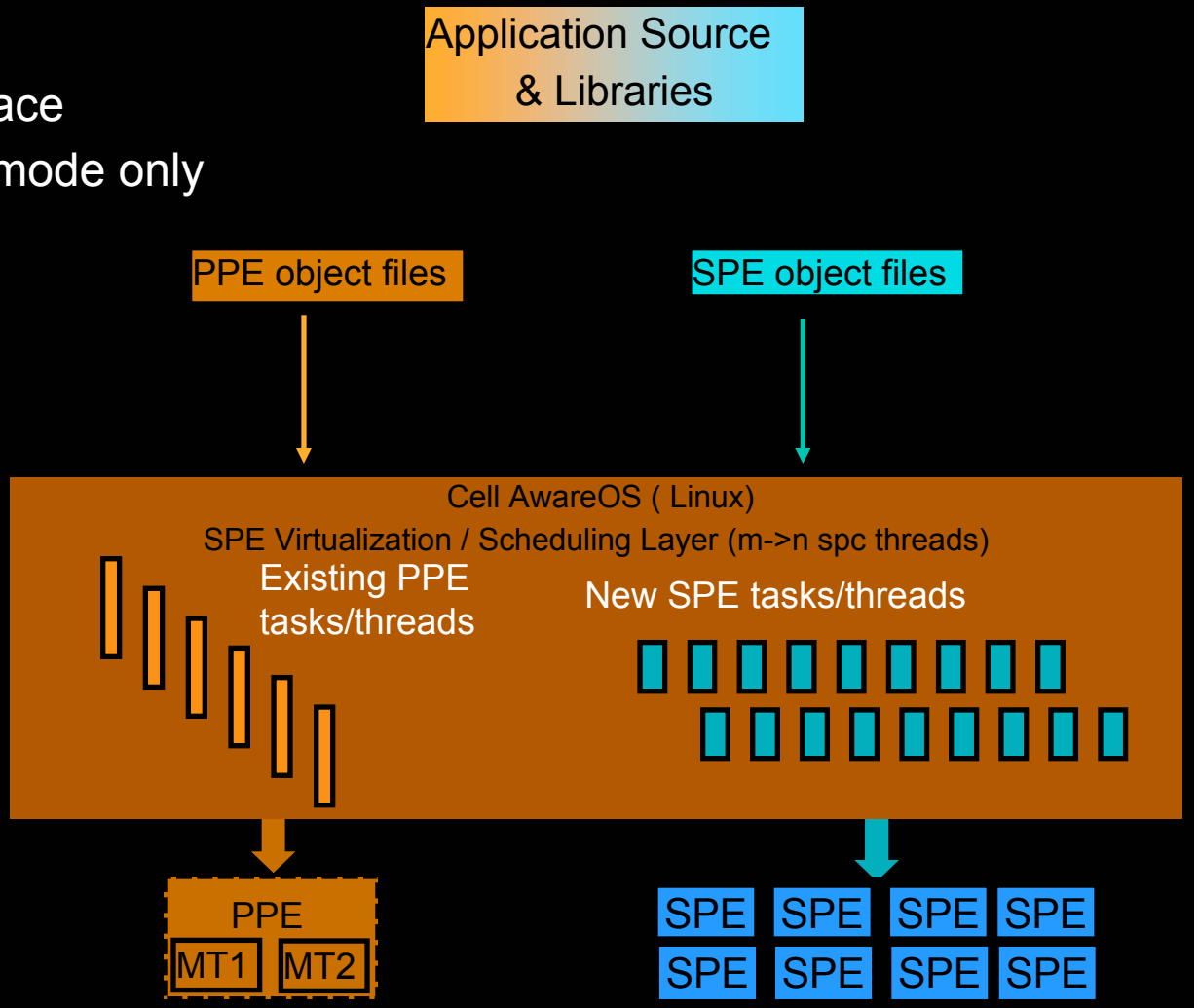


Shared-memory Multiprocessor

- The Cell Processor can be programmed as a shared-memory multiprocessor, using two different instruction sets
- The SPEs and the PPE fully inter-operate in a cache-coherent Shared-Memory Multiprocessor Model
  - All DMA operations in the SPEs are cache-coherent
  - The DMA operations use an effective address that is common to all PPE and SPEs
  - Shared-memory store instructions are replaced by a store from the register file to the LS, followed by a DMA operation from LS to shared memory.
- A compiler or interpreter could manage part of the LS as a local cache for instructions and data obtained from shared memory.

Heterogeneous Multi-Threading Model

- PPE Threads, SPE Threads
- Shared effective address space
- SPE Virtualization in debug mode only



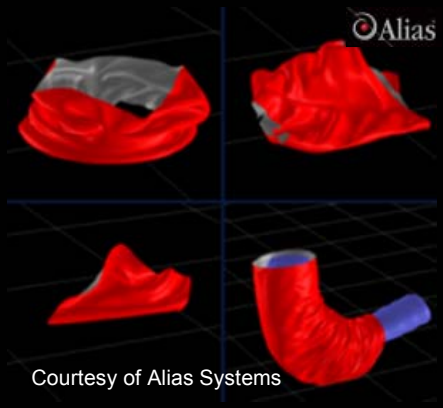
Single source approach to programming Cell

- Single Source Compiler
  - Auto parallelization (treat target Cell as an SMP)
  - Auto SIMD-ization (SIMD-vectorization)
  - Compiler management of Local Store as 2nd level register file / SW managed cache (I&D)
    - Most Cell unique piece
- Optimization
  - OpenMP pragmas
  - Vector.org SIMD intrinsics
  - Data/Code partitioning
  - Streaming / pre-specifying code/data use
- Prototype Single Source Compiler Developed in IBM Research

Graphics & Visualization

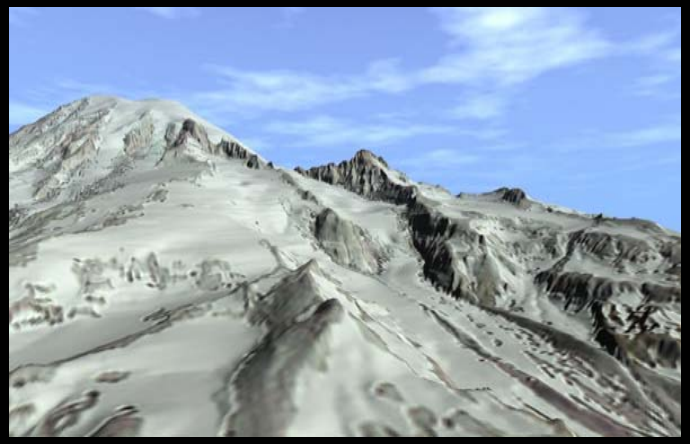
Graphics and Visualization on Cell

Geometry Processing



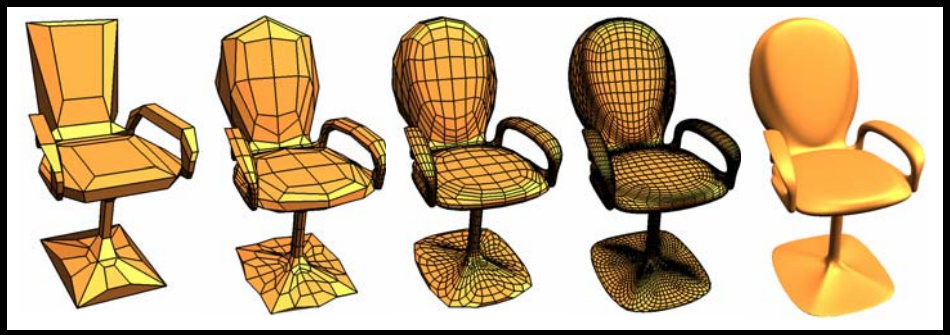
Courtesy of Alias Systems

Cloth Simulation



Terrain Rendering Engine

Multi-resolution Subdivision Surfaces



The Cell and GPUs

- Extension to shader pipeline on GPUs
  - Vertex shaders for geometric modeling
    - NURBS
    - Subdivision surfaces
    - Continuous level of detail
  - Vertex shaders for a additional lighting models
- Physically Based Modeling
  - Soft-body dynamics
  - Rigid-body dynamics
- Image Processing
  - Background Subtraction for Video Surveillance
- Global Illumination
  - Raytracing
  - Raycasting Terrains
    - Rendering of DEMs
    - Surface shaders for lighting, shadows

The Cell and GPUs

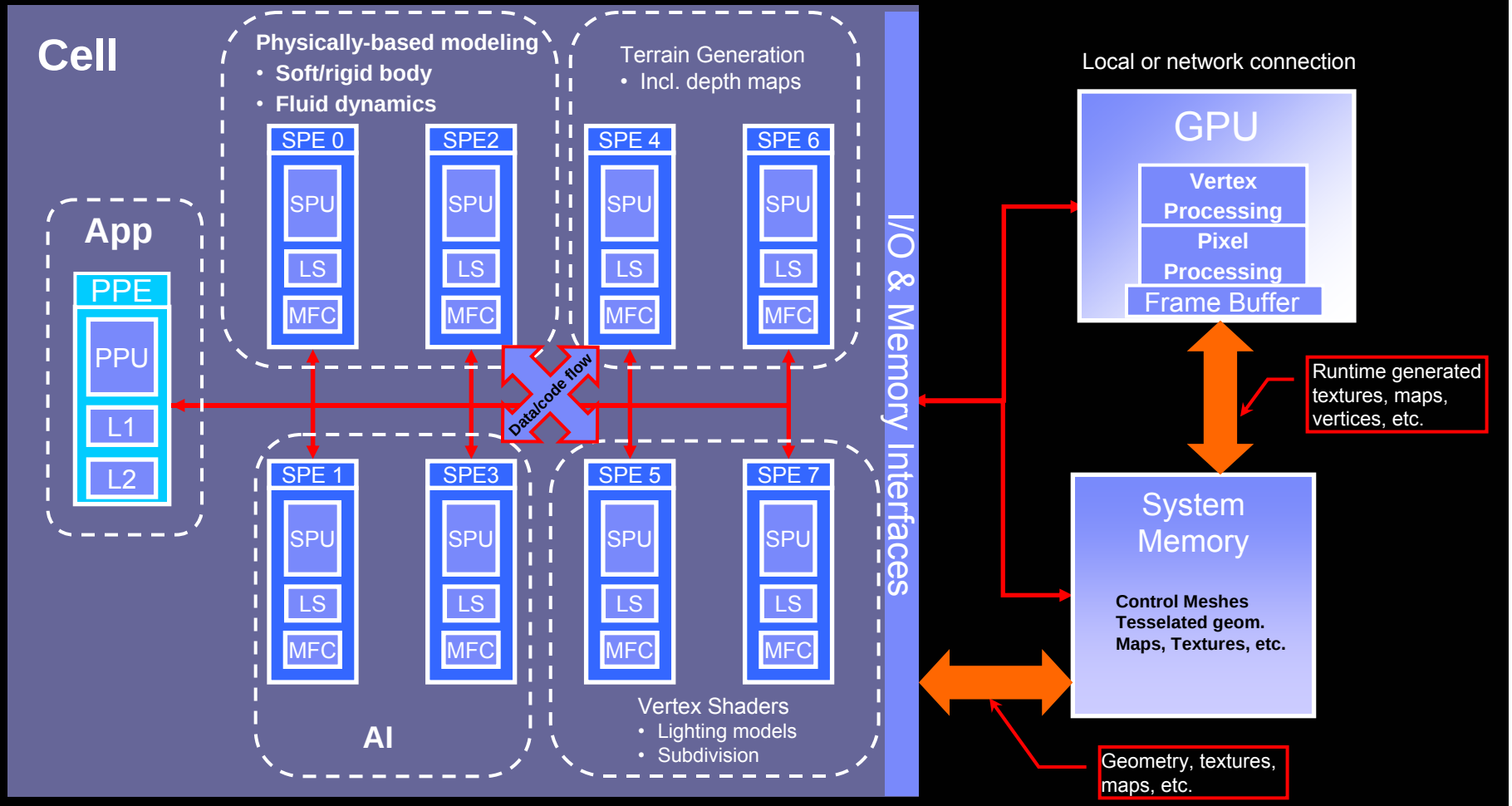
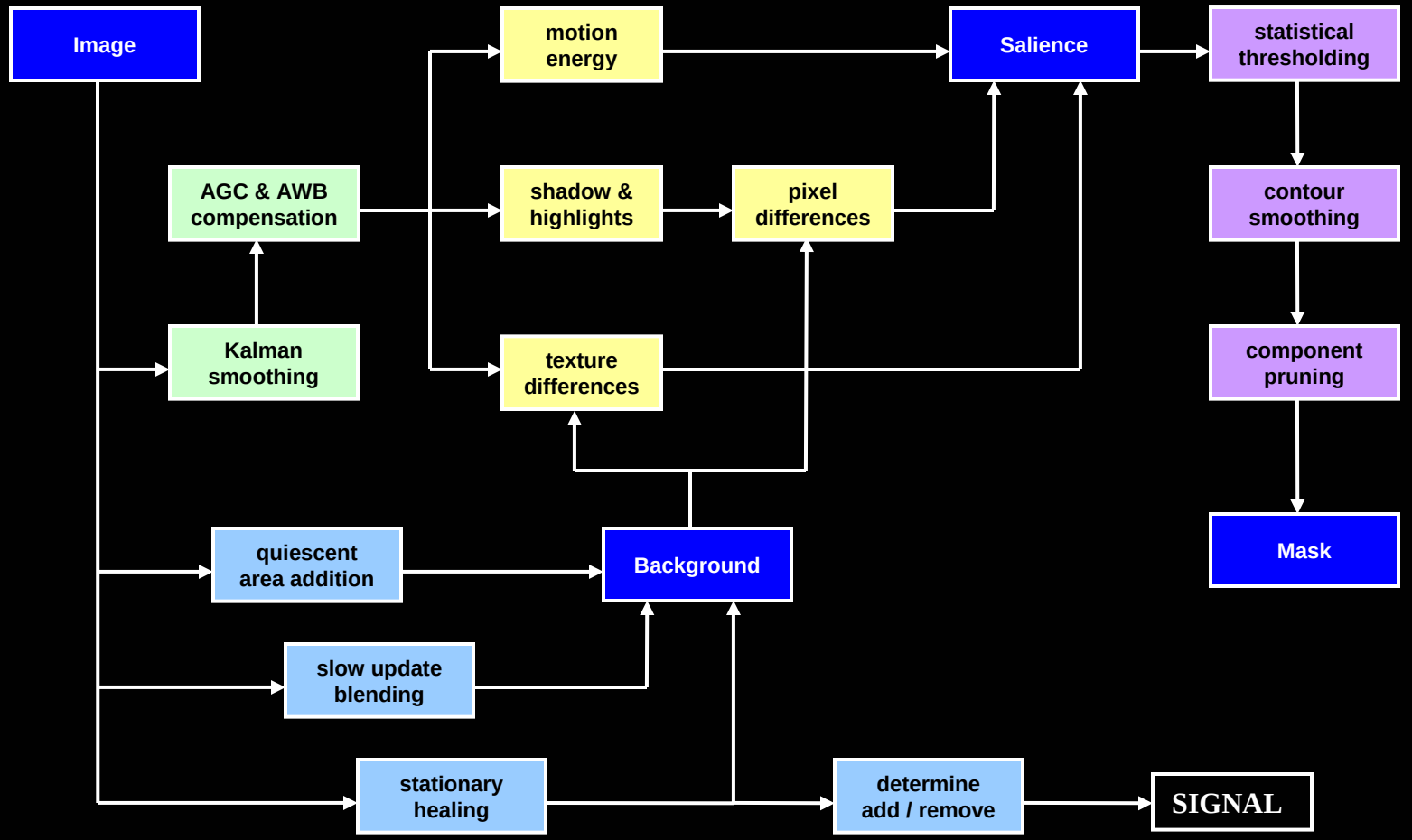


Image Processing on Cell

- Video Surveillance

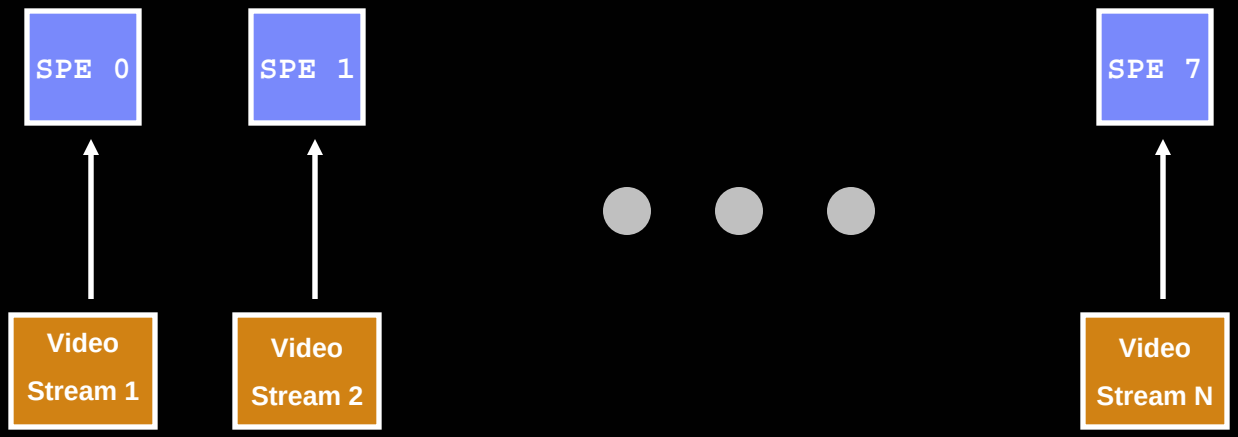
- Background subtraction compares the current image (left) with a reference image (middle) to find the changed regions (right) corresponding to objects of interest.

BGS Application Structure



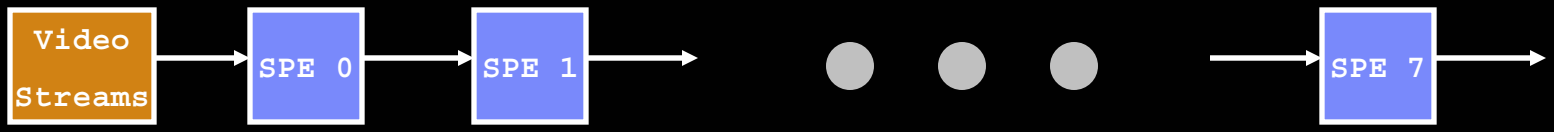
- per Jonathan Connell

Alternative 1 – Single SPE for Video Stream



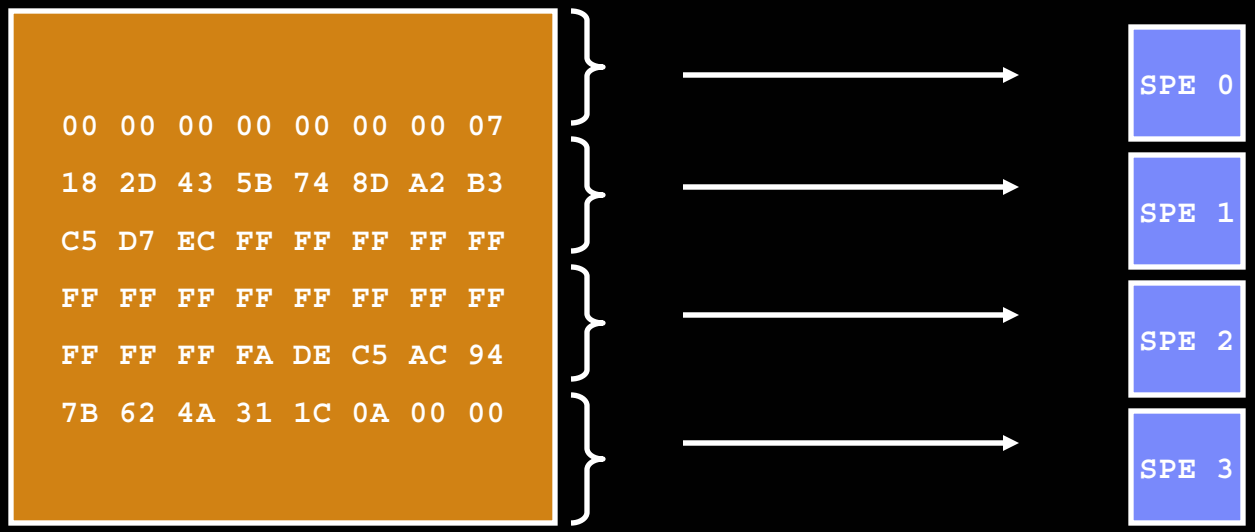
- Each SPE dedicated to one or more video streams
- Code overlaying may be needed to overcome the local memory limitation
- Seems most compatible with existing application structure
- No code or data partitioning across SPEs

Alternative 2 – Image Pipeline for Task-level Parallelism



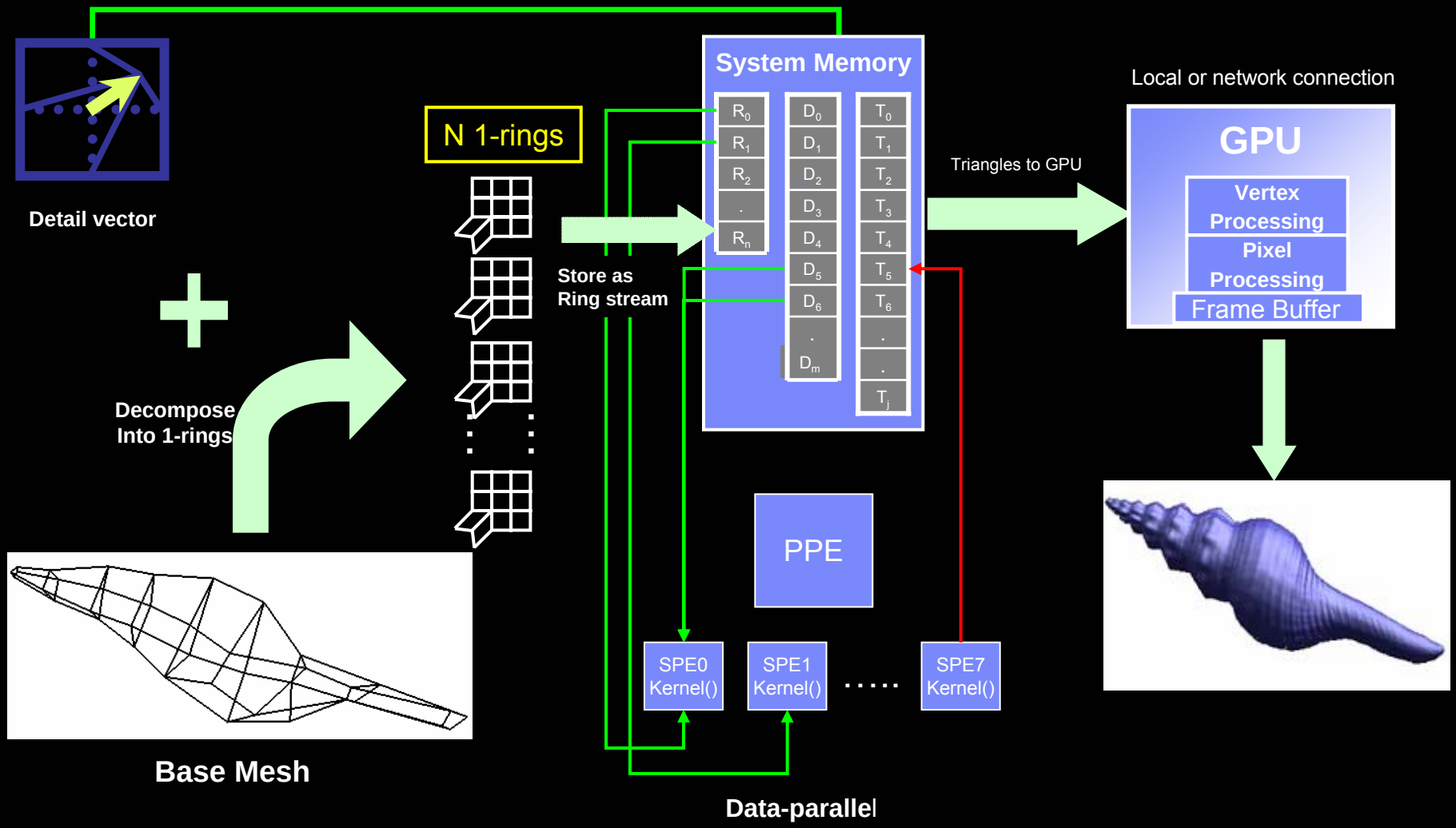
- Functions are divided into groups
- Groups set up in pipelined fashion
- Each SPE dedicated to only one group
- Outputs from one group passed SPE-to-SPE as inputs to the next
- Code is partitioned – Data is not partitioned
- Efficient utilization of SPEs

Alternative 3 – Data-level Parallelism



For a given function, each SPE processes part of a frame
Not easily applied to all BGS functions

Multi-resolution Subdivision

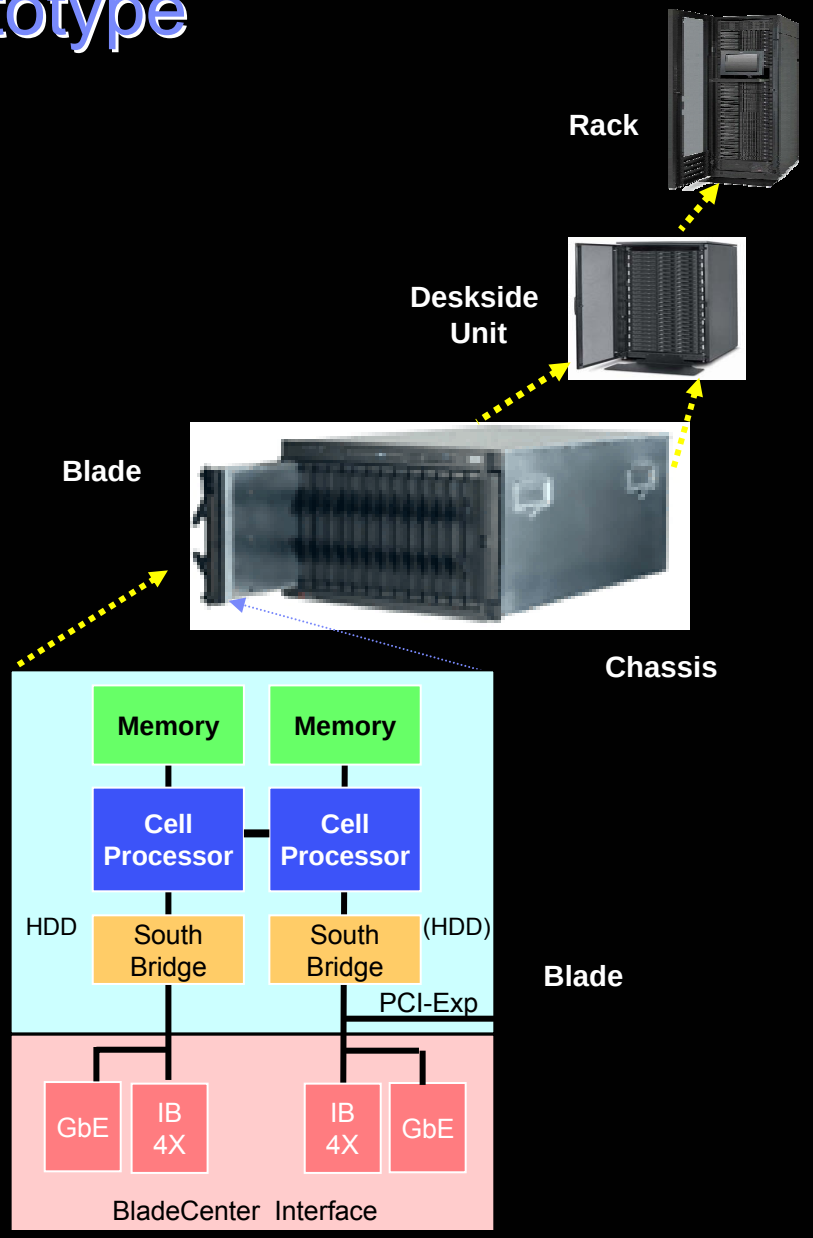


Demonstrations

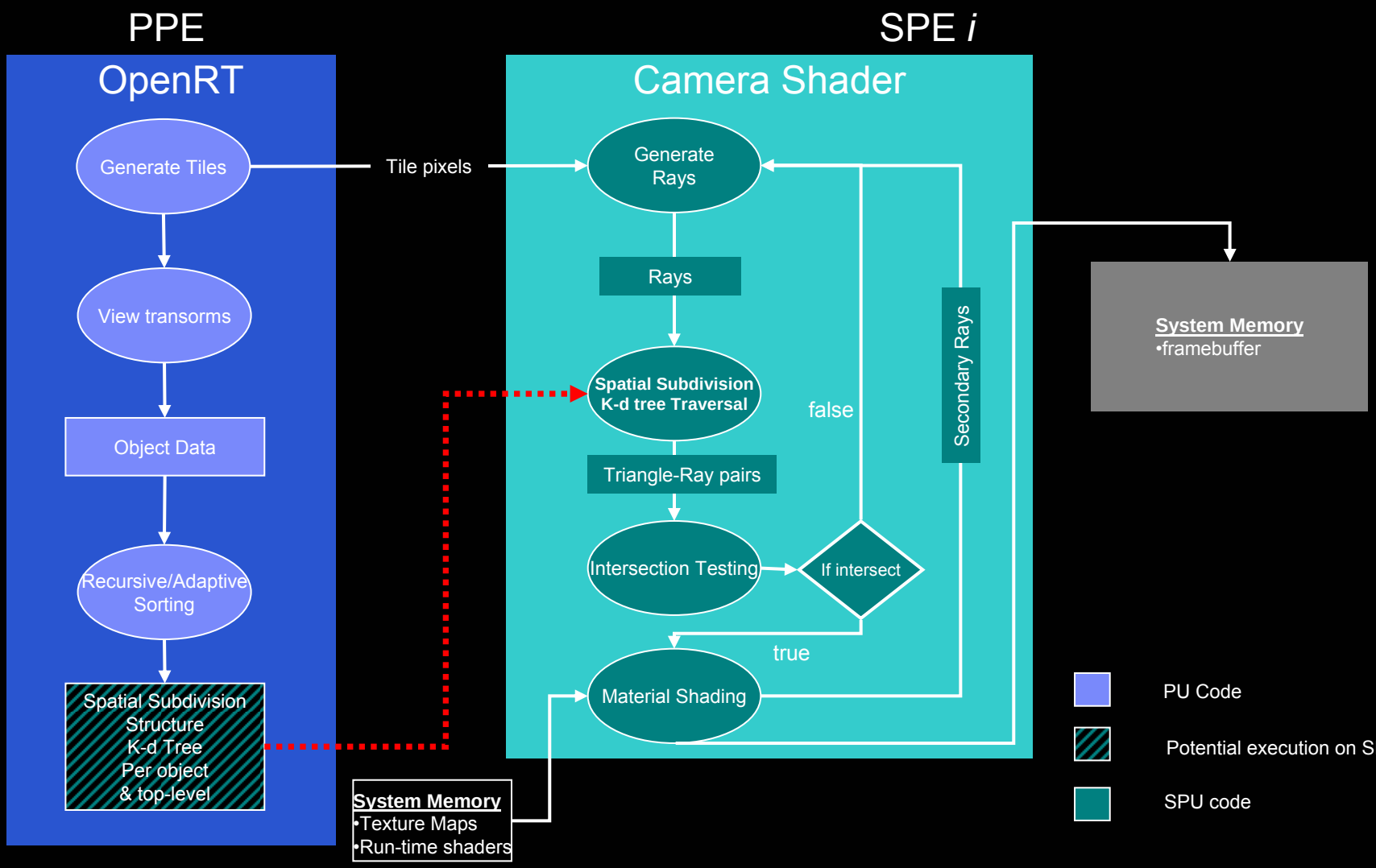
- Raytracing
- Terrain Rendering (TRE)
- Cloth Simulation

Demo Platform: Cell Blade Prototype

- **Blade Server**
  - Dual Cell Processors (SMP), Support Logic, Memory, Storage
  - PCI Express 4X option port
  - BladeCenter Interface (Based on IBM JS20)
  - Infiniband 4x (10Gbps) interconnect
- **Chassis**
  - Standard IBM BladeCenter form factor with:
    - 7 Blades (1 blade/2 slots)
    - 2 internal switches (1Gb Ethernet) with 4 external ports each
  - Separate, external Infiniband Switch with optional FC port
- **Software**
  - Linux OS
  - Tool chain and compilation support
  - Additional IBM Software Development Tools and Cluster Management software
- **Client Systems**
  - **IBM T41 Thinkpad (1.7 Ghz Pentium-M)**
  - **Apple Mac-dual G5@2Ghz**

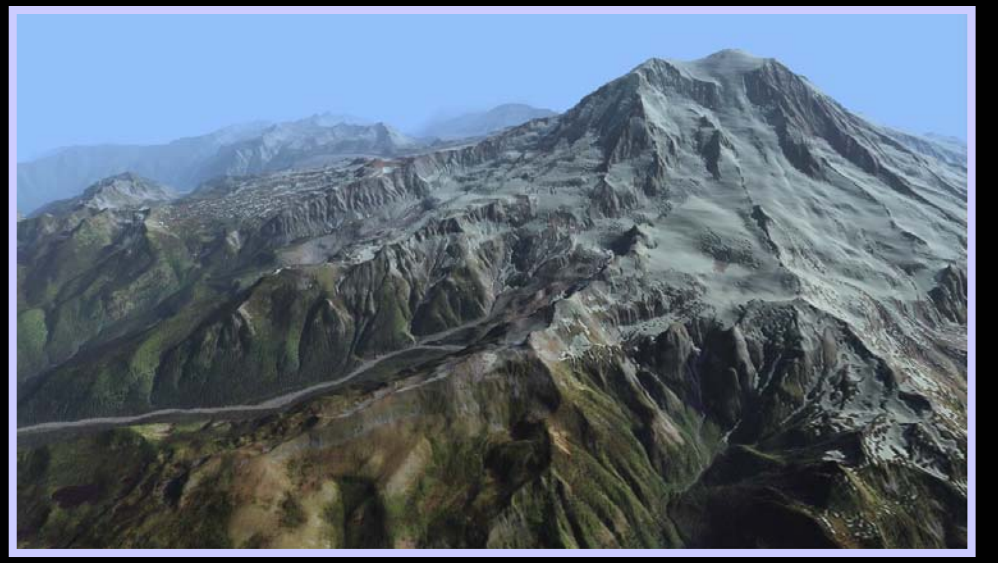


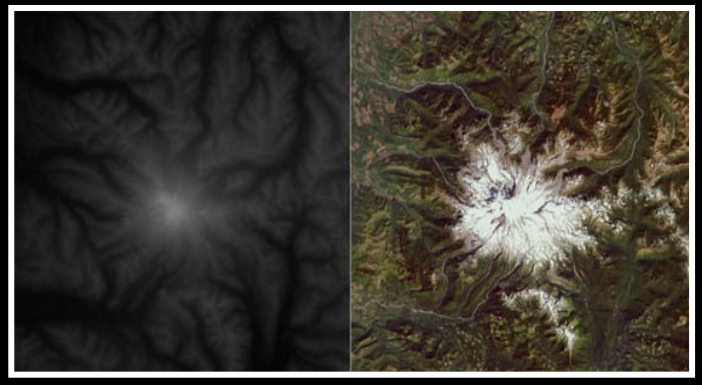
Example: Raytracing Architecture



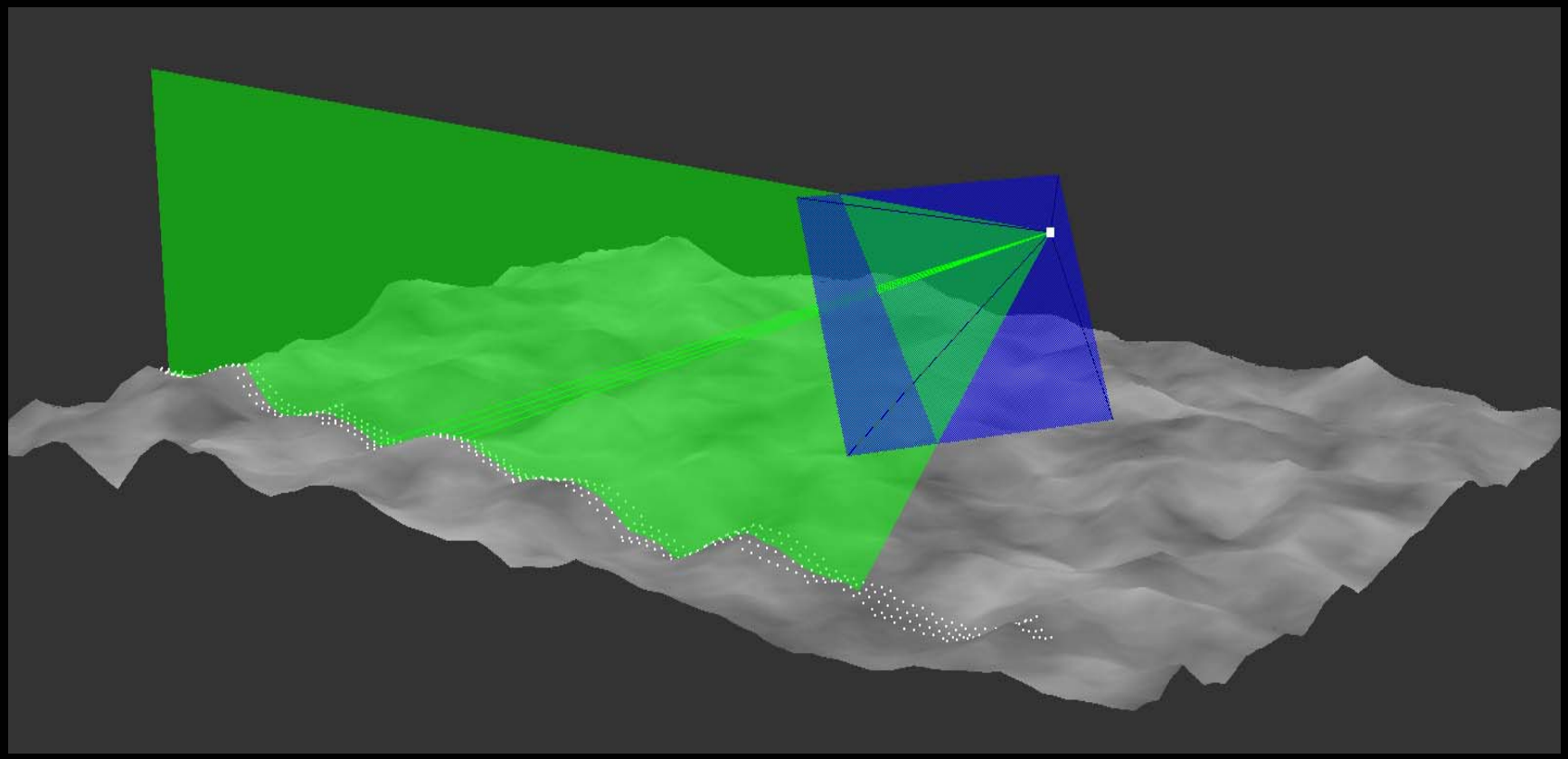
Cell Optimized Ray Caster: Terrain Rendering Engine

- 30+ frames per second with only one Cell processor
  - No graphics adapter assist
  - 1280x720 (HD 720P) resolution
- HD 1080P at 30+ frames per second via 2 way SMP Cell
- Advanced SPE shader function
  - Ray/Terrain intersection computation
  - Texture Filtering
  - Normal computation
  - Bump map computation
  - Diffuse + Ambient lighting model
  - Perlin Noise based clouds
  - Atmosphere computation (haze, sun, halo)
  - Dynamic multi-sampling (4 – 16 samples per pixel)
  - Image based input (16 bit height + 16 bit texture)
  - 29 KB of SPE object code
  - 224 KB of SPE local store data
- M-JPEG compression via SPE
- Performance scales linearly with number of available SPEs
- Written completely in C with intrinsics
- Currently up and running on bring up systems!

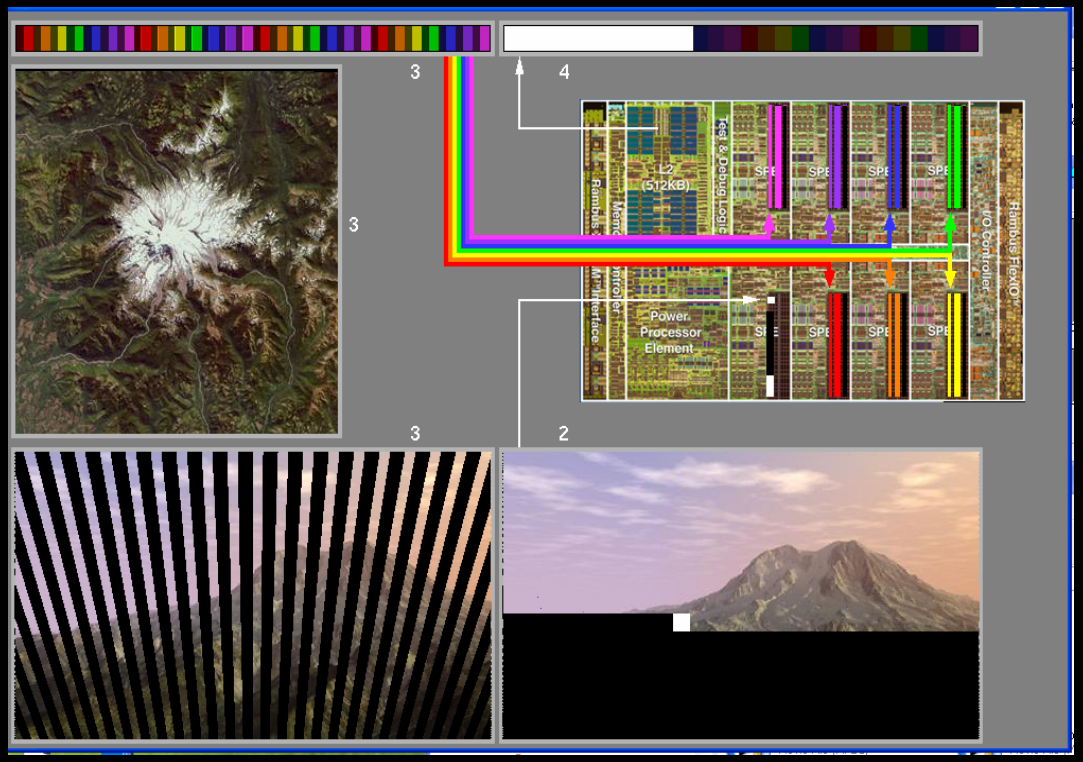




Vertical Ray Coherence

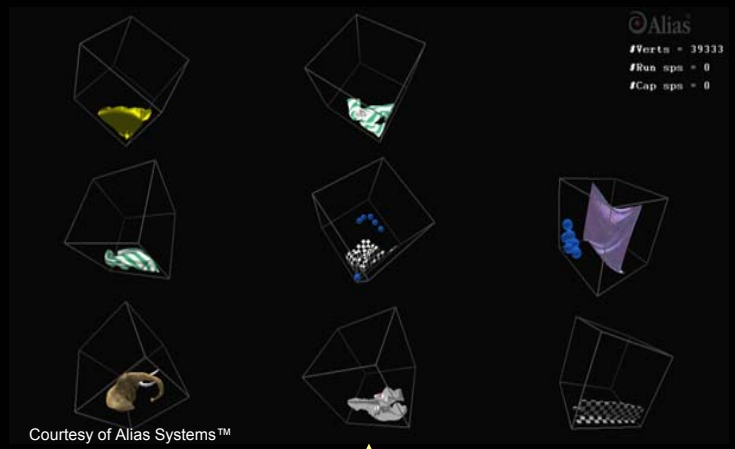


Algorithm Mapped to Cell



Physically Based Modeling: Alias research prototype

- Cell implementation of Alias cloth solver
  - Compute the displacement of 3-D cloth mesh vertices in response to various gravitational, collision and constaints
  - Each simulation can occur on a separate SPE
- Render results on commodity graphics adapter – locally or over network



Courtesy of Alias Systems™

Cell beyond the PS3

- IBM support for custom systems and applications based on Cell
  - Through IBM Engineering and Technology Services
- Planned release (next several months) of architecture, simulator, and compiler to enable Cell software development and evaluation in a large community
  - Linux OS, compilers, debugger, full system simulator
  - Open source and IBM alphaworks
- Prototype system implementations and evaluation of markets beyond console
  - Cell Processor Based Blades (acceleration and other apps.)
  - CE devices (HDTV, home media server)
  - Standard products for embedded applications
  - Prototype reference designs for smaller systems

Summary

- Cell ushers in a new era of leading edge processors optimized for digital media and entertainment
- New levels of performance and power efficiency beyond what is achieved by PC processors
- Step towards HPC / game processor convergence

Acknowledgements

Sony Computer Entertainment
Toshiba Corporation
IBM STI Design Center:
Peter Hofstee
Barry Minor
Mark Nutter
Gerhard Stenzel, IBM Boeblingen Development Labs
Philipp Slusallek, Saarland University
Alias Systems:
Joyce Janczyn
Francesco Iorio
Jos Stam

Thank you