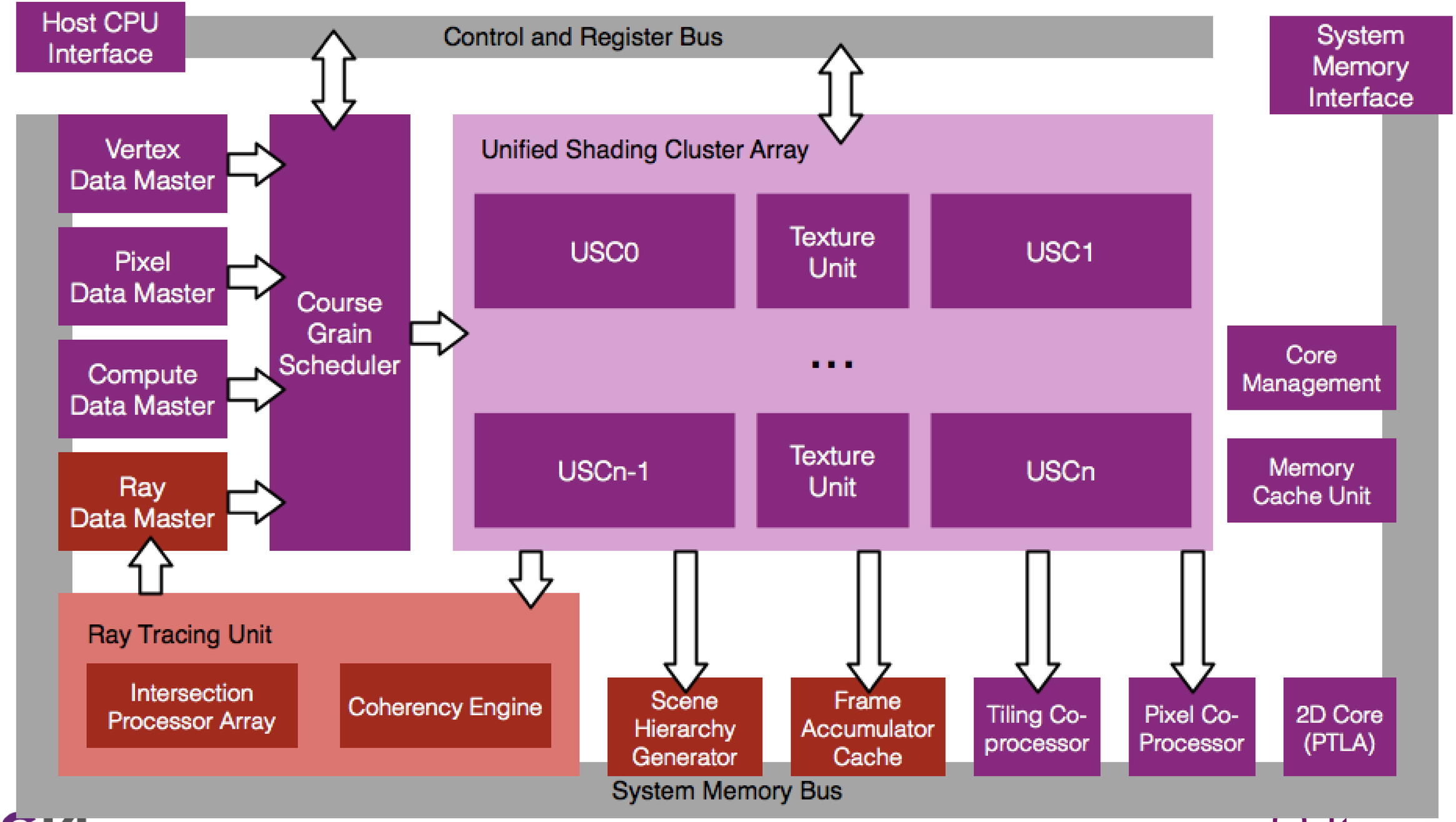
Introduction to PowerVR Ray Tracing

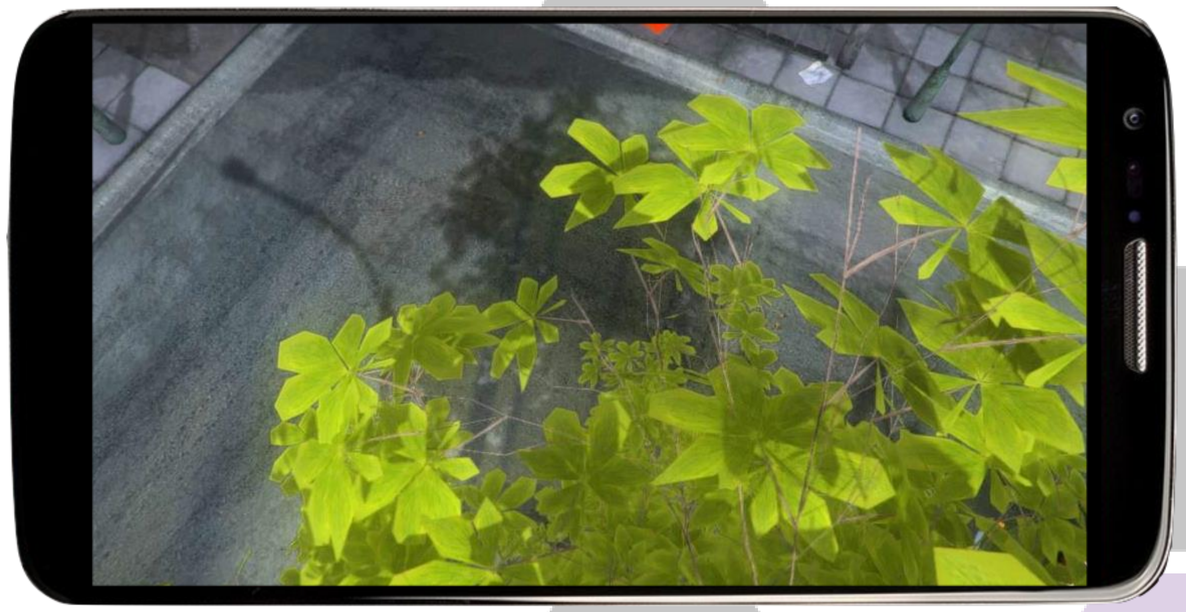
Tuesday 18th March, 2014 @ GDC

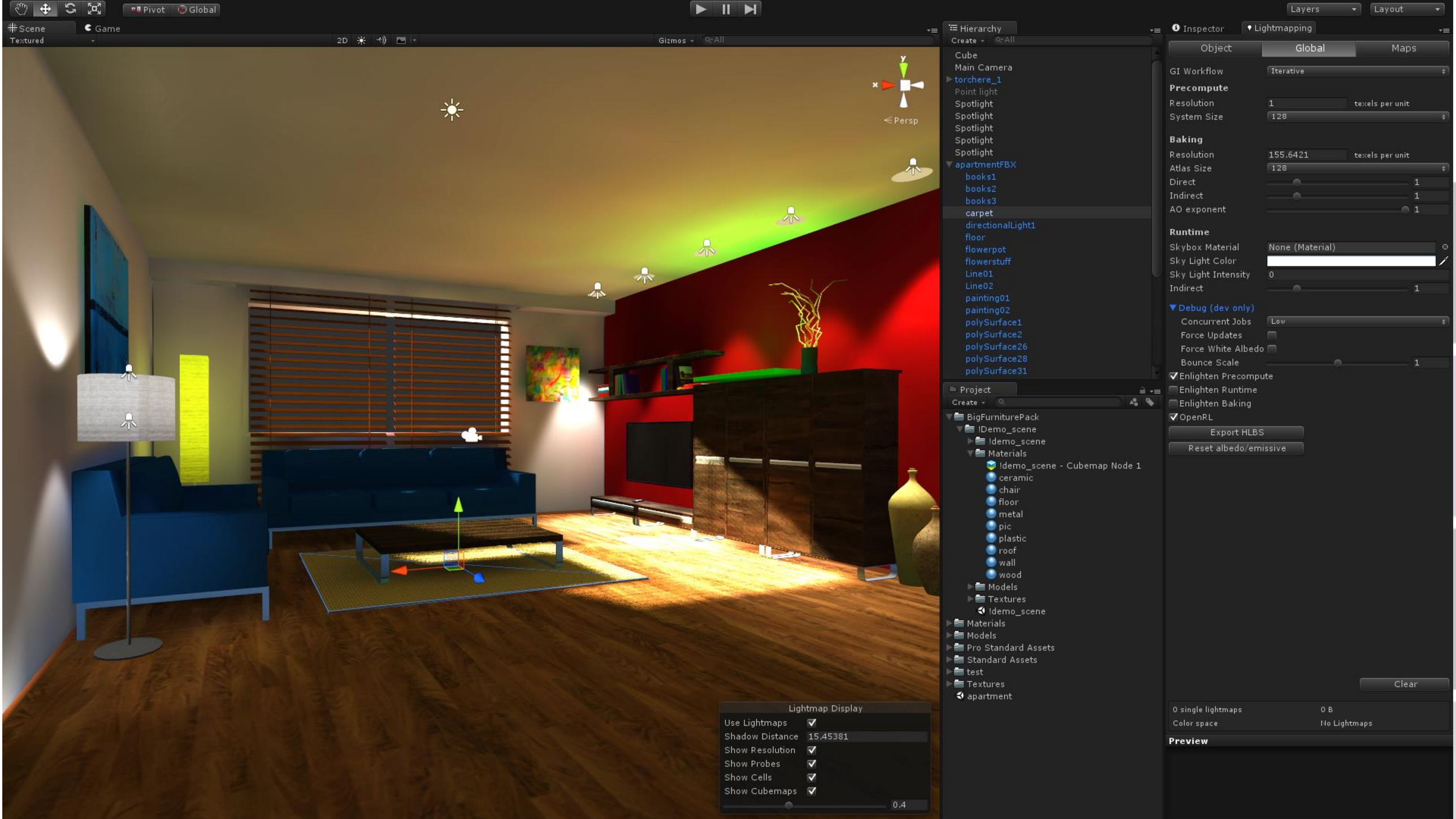
James A. McCombe

What are we launching today?


Host CPU Interface
Control and Register Bus
System Memory Interface
Vertex Data Master
Pixel Data Master
Compute Data Master
Ray Data Master
Course Grain Scheduler
Unified Shading Cluster Array
USC0
Texture Unit
USC1
...
USCn-1
Texture Unit
USCn
Core Management
Memory Cache Unit
Ray Tracing Unit
Intersection Processor Array
Coherency Engine
Scene Hierarchy Generator
Frame Accumulator Cache
Tiling Co-processor
Pixel Co-Processor
2D Core (PTLA)
System Memory Bus

OPEN




Pivot
Global
Layers
Layout
Scene
Game
Textured
2D
Gizmos
Persp
Hierarchy
Create
Cube
Main Camera
torchere_1
Point light
Spotlight
Spotlight
Spotlight
Spotlight
Spotlight
apartmentFBX
books1
books2
books3
carpet
directionalLight1
floor
flowerpot
flowerstuff
Line01
Line02
painting01
painting02
polySurface1
polySurface2
polySurface26
polySurface28
polySurface31
Project
Create
BigFurniturePack
!Demo_scene
!demo_scene
Materials
!demo_scene - Cubemap Node 1
ceramic
chair
floor
metal
pic
plastic
roof
wall
wood
Models
Textures
!demo_scene
Materials
Models
Pro Standard Assets
Standard Assets
test
Textures
apartment
Inspector
Lightmapping
Object
Global
Maps
GI Workflow
Iterative
Precompute
Resolution
1
texels per unit
System Size
128
Baking
Resolution
155.6421
texels per unit
Atlas Size
128
Direct
1
Indirect
1
AO exponent
1
Runtime
Skybox Material
None (Material)
Sky Light Color
Sky Light Intensity
0
Indirect
1
Debug (dev only)
Concurrent Jobs
Low
Force Updates
Force White Albedo
Bounce Scale
1
Enlighten Precompute
Enlighten Runtime
Enlighten Baking
OpenRL
Export HLBS
Reset albedo/emissive
Clear
0 single lightmaps
0 B
Color space
No Lightmaps
Preview
Lightmap Display
Use Lightmaps
Shadow Distance
15.45381
Show Resolution
Show Probes
Show Cells
Show Cubemaps
0.4

What are the applications for this?

Imagination

Augmented Reality

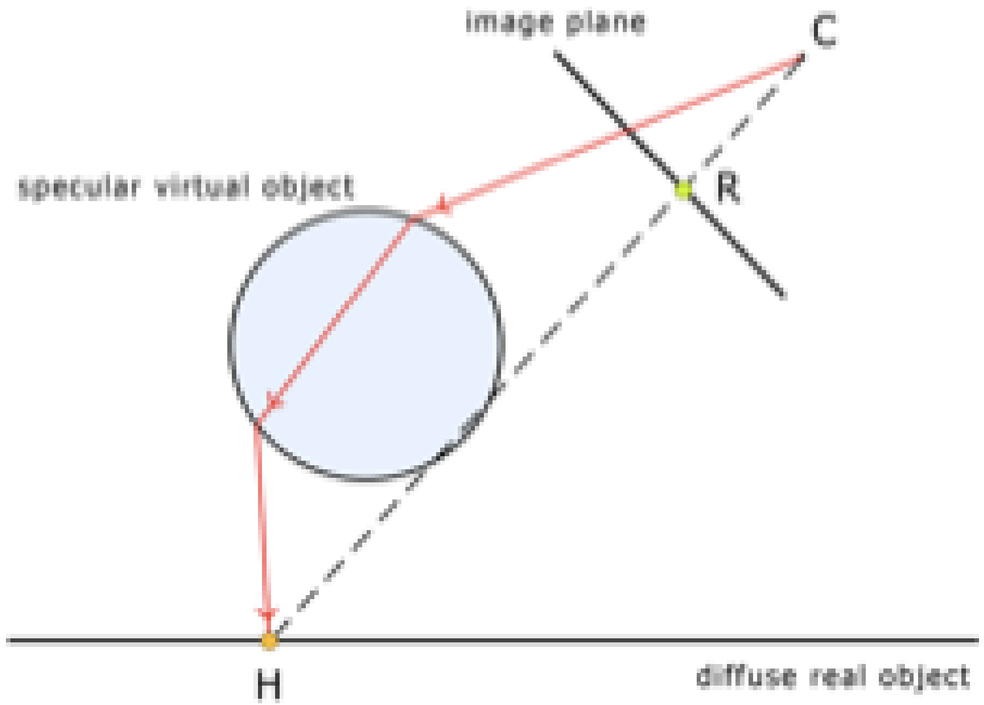


Image and Diagram Copyright
Peter Kán 2012

Imagination

Hybrid rendering in games

- Shadows
- Reflections
- Transparency
- Better scaling with multiple dynamic lights
- Real-time light map updates for the rasteriser
- Easy to bolt on

Imagination

Fully ray traced graphics

- Brute force path tracing
- Produces photo realism easily
- Pretty much requires all the 3D content to be ray traced
- Possible in todays technology in console / desktop using Wizard
- Probably not practical for fully real-time use in mobile for a couple more generations. Fine for sub-realtime use.

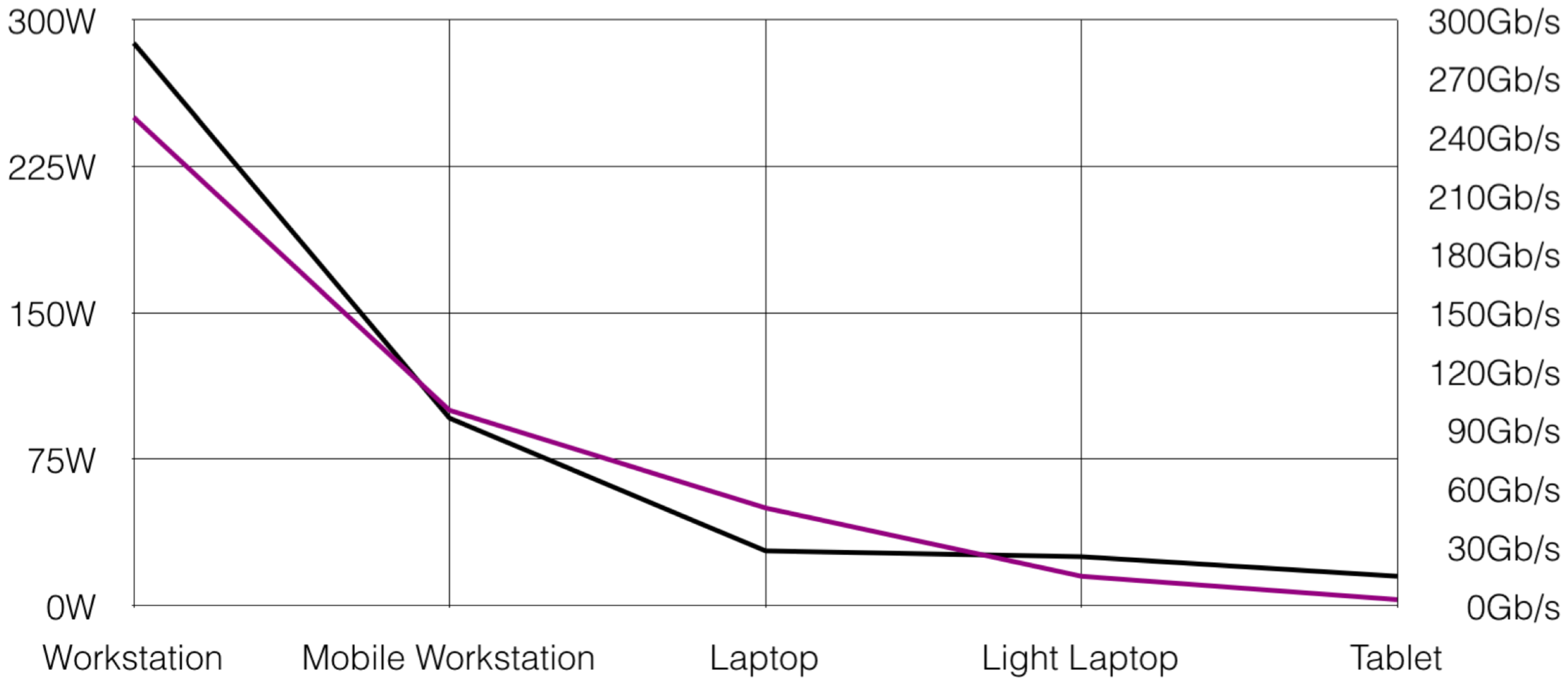
Non-graphics

- Better in-game AI
- Collision Detection
- 3D spatial search

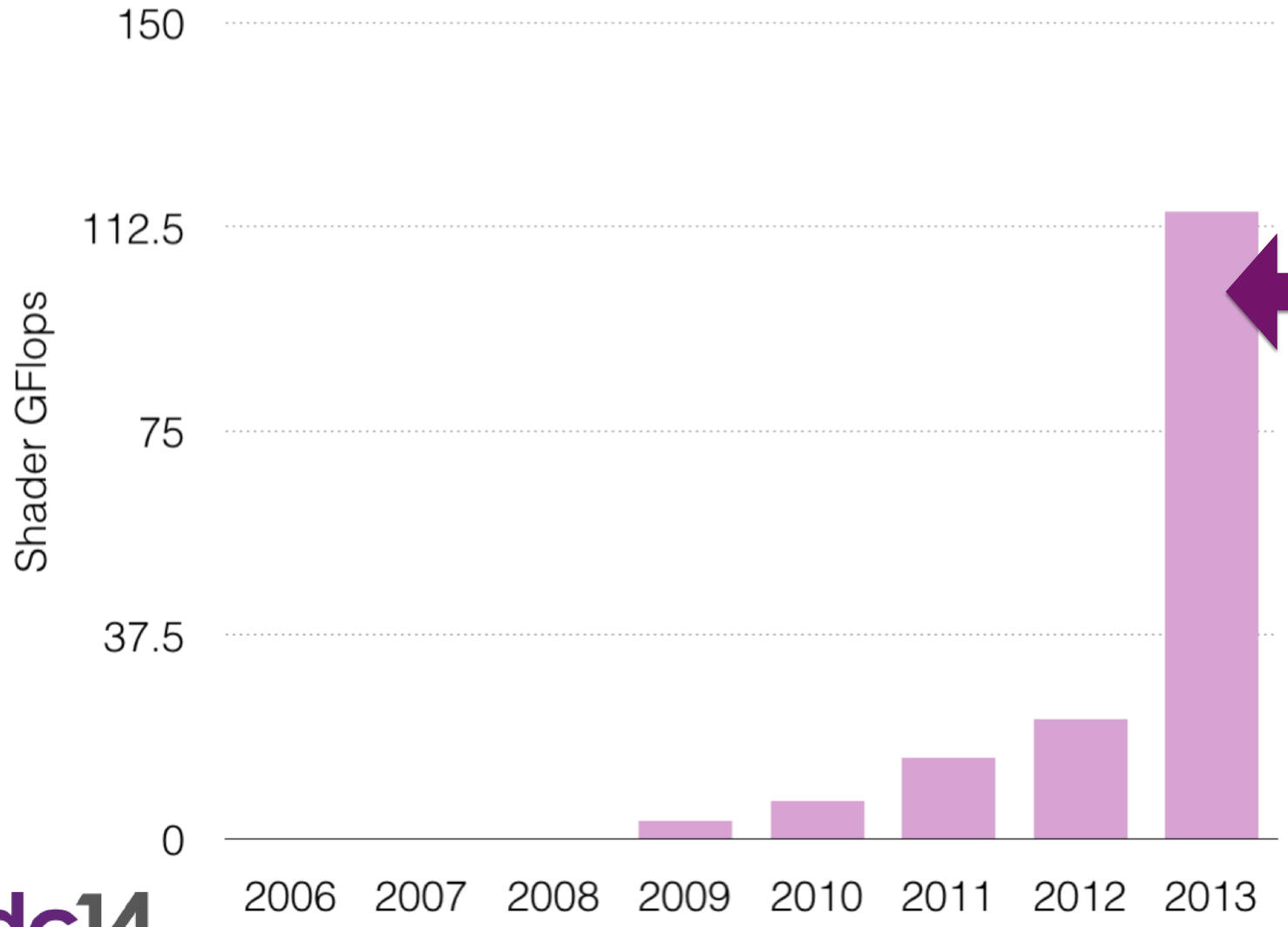
Imagination

Overview of PowerVR

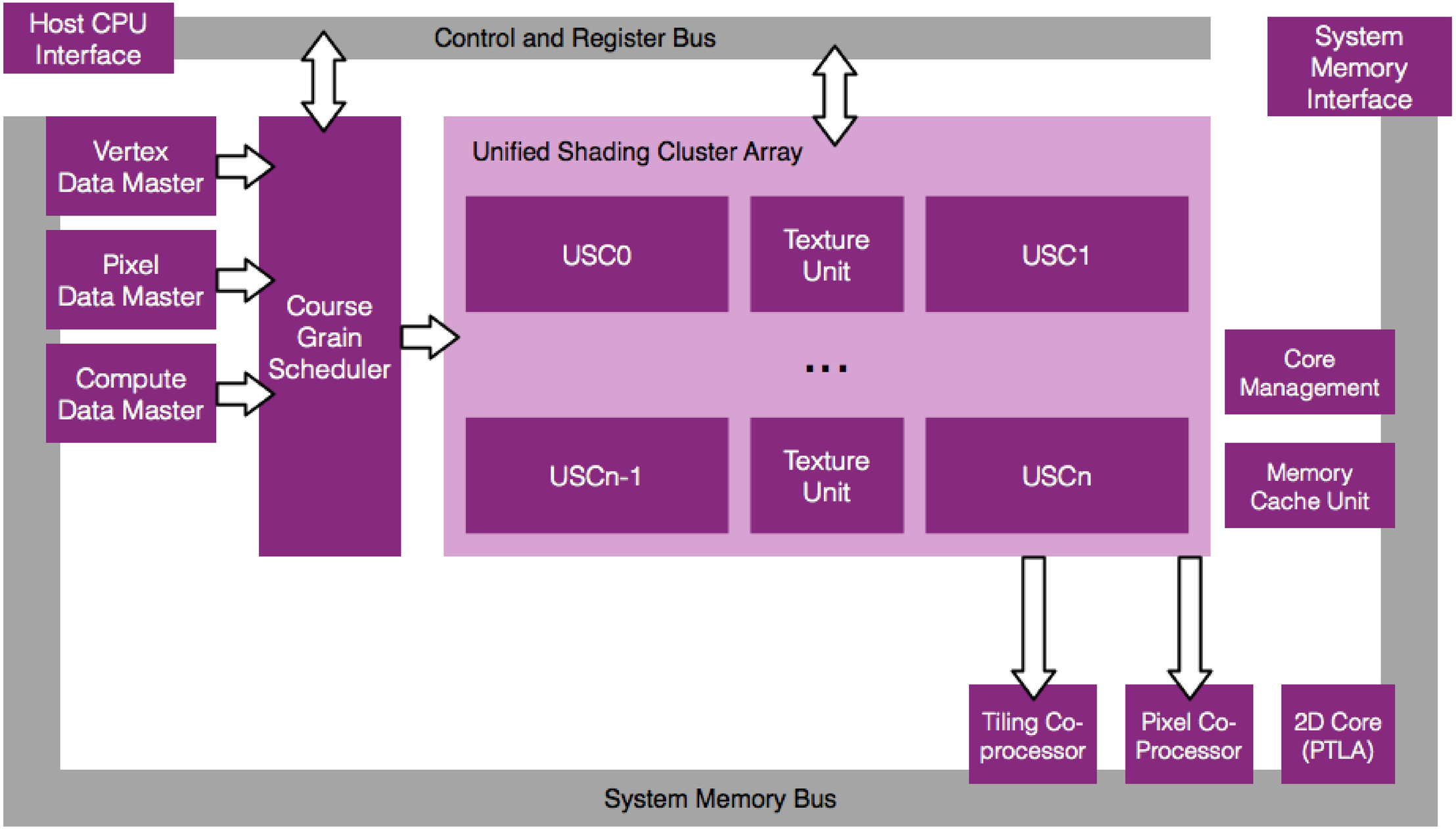
Power and Bandwidth

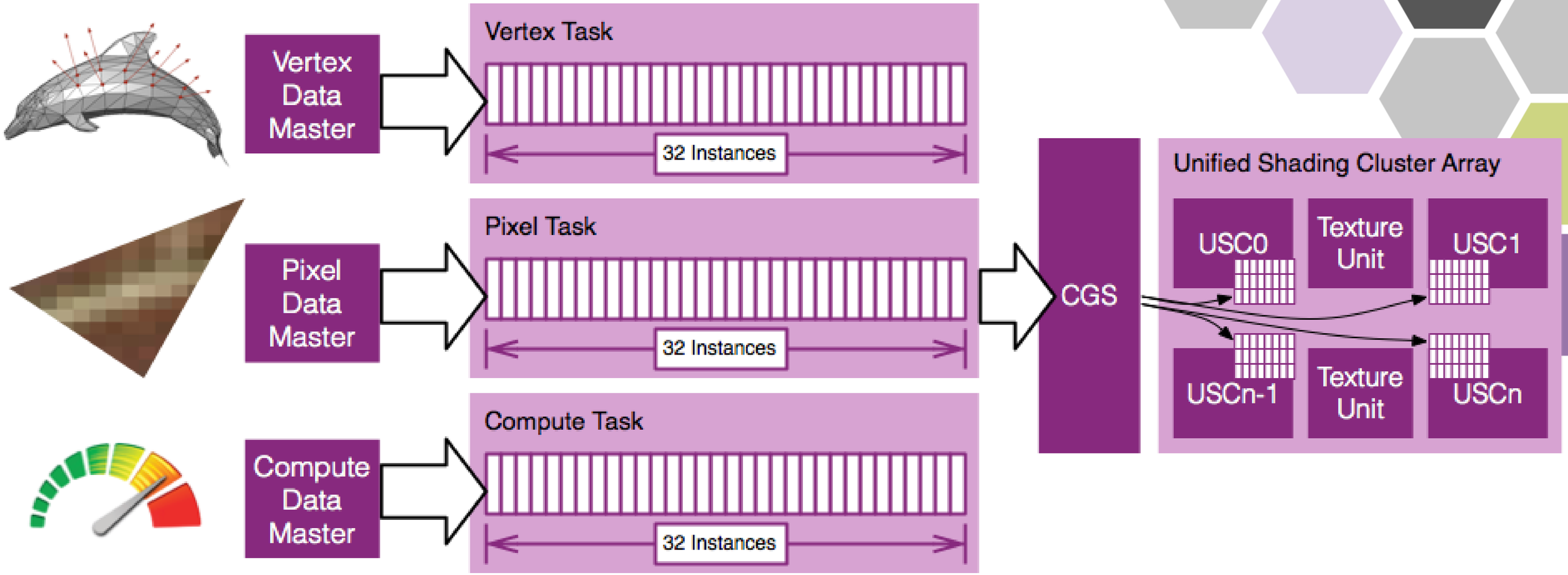


Capability of our mobile graphics



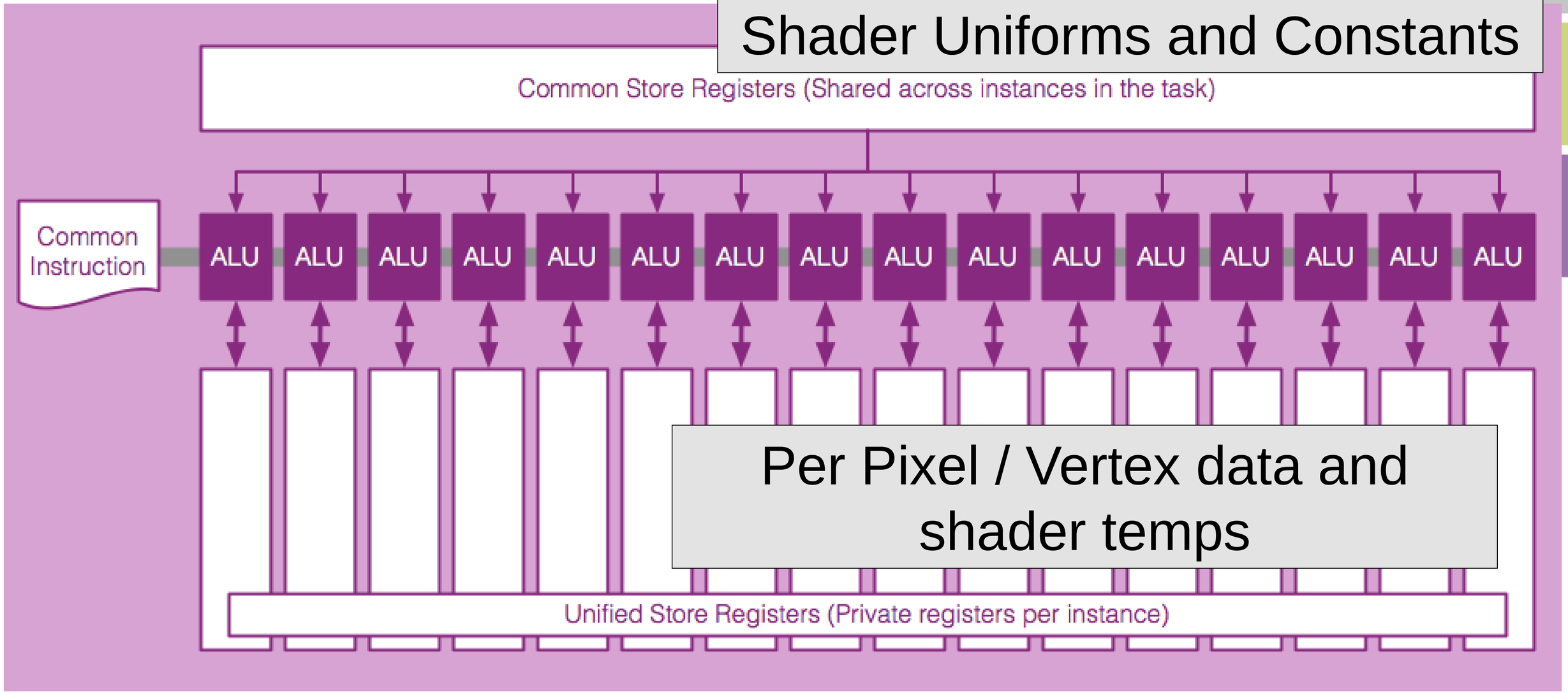
- Almost 5x compute capability
- 4x vec4 MAD → 16x Scalar 2xMAD
- 1.5 → 3.6 Gtexels/s
- OpenGLES 3.0 Support


Host CPU Interface
Control and Register Bus
System Memory Interface
Vertex Data Master
Pixel Data Master
Compute Data Master
Course Grain Scheduler
Unified Shading Cluster Array
USC0
Texture Unit
USC1
...
USCn-1
Texture Unit
USCn
Core Management
Memory Cache Unit
Tiling Co-processor
Pixel Co-Processor
2D Core (PTLA)
System Memory Bus

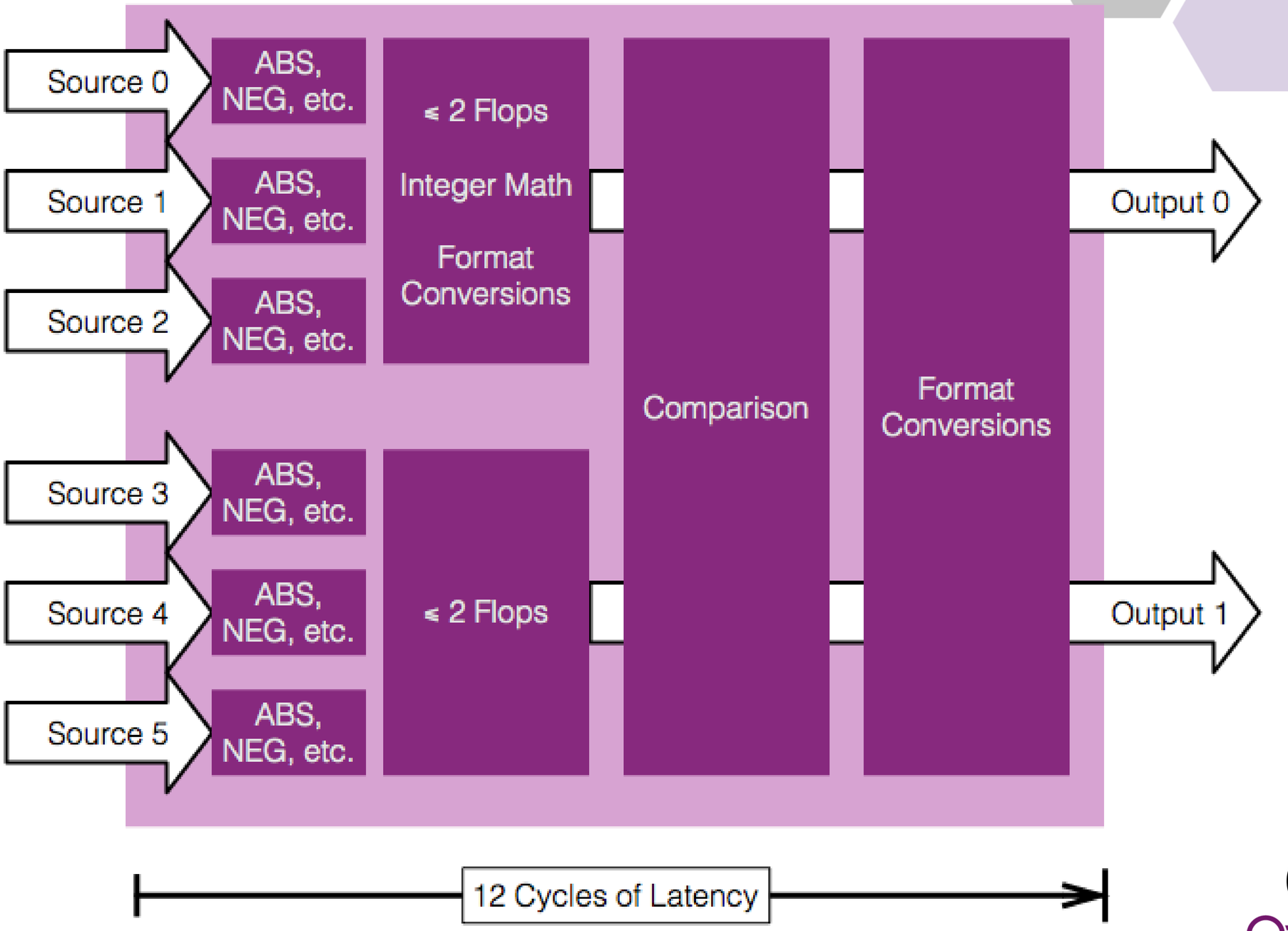


- An instance is a vertex, pixel or OpenCL thread
- All instances in the task share Program, Uniforms, Parameters, etc.

Shader Uniforms and Constants

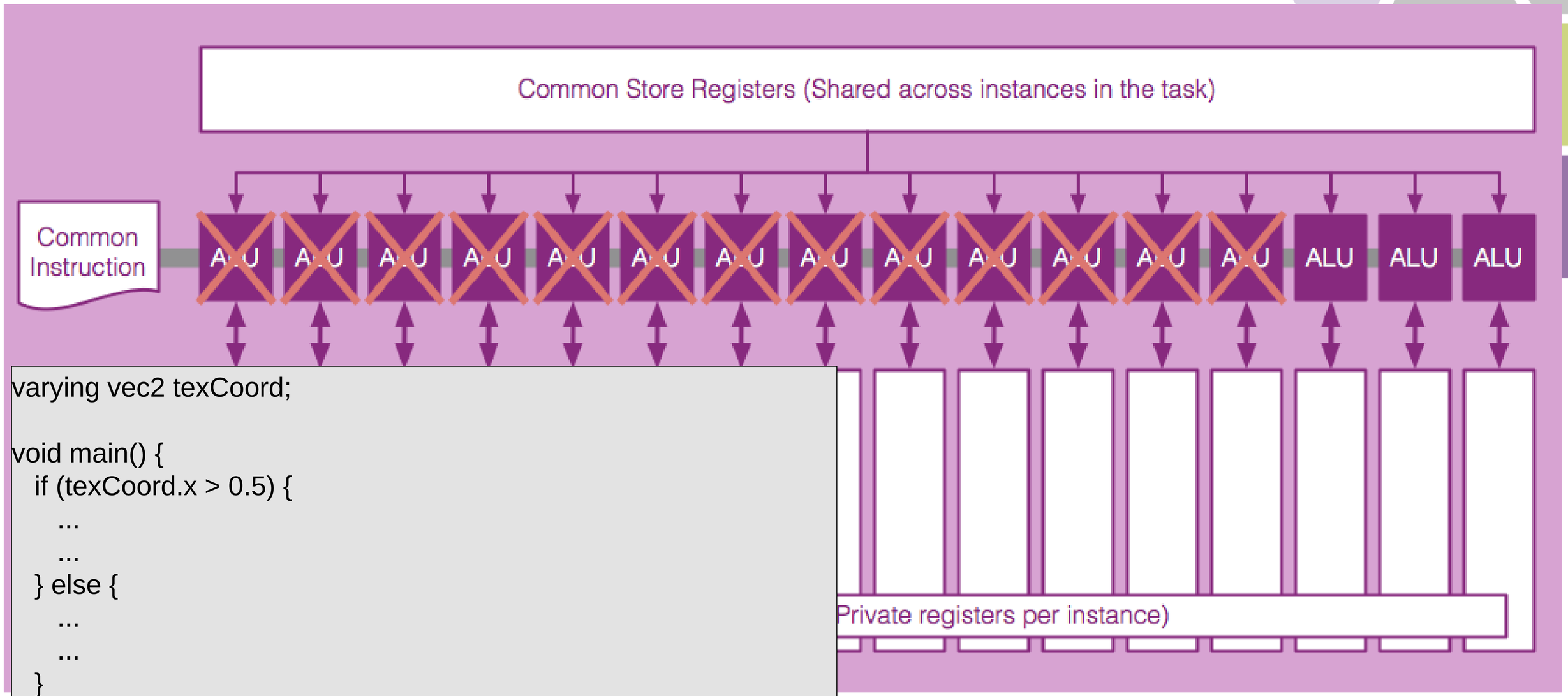


(not exact)



(not exact)

Imagination



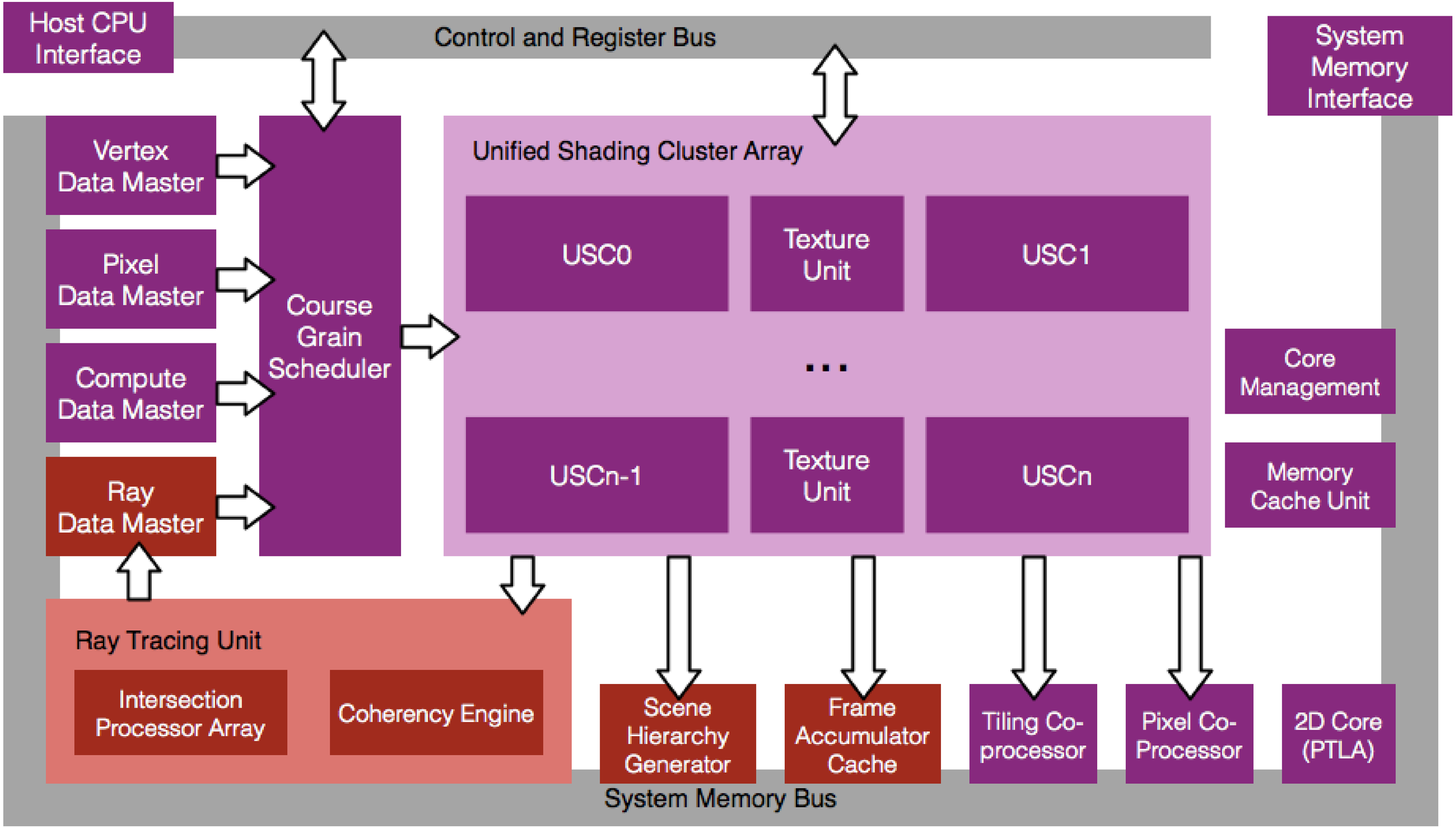

```
varying vec2 texCoord;

void main() {
   if (texCoord.x > 0.5) {
      ...
      ...
   } else {
      ...
      ...
   }
```

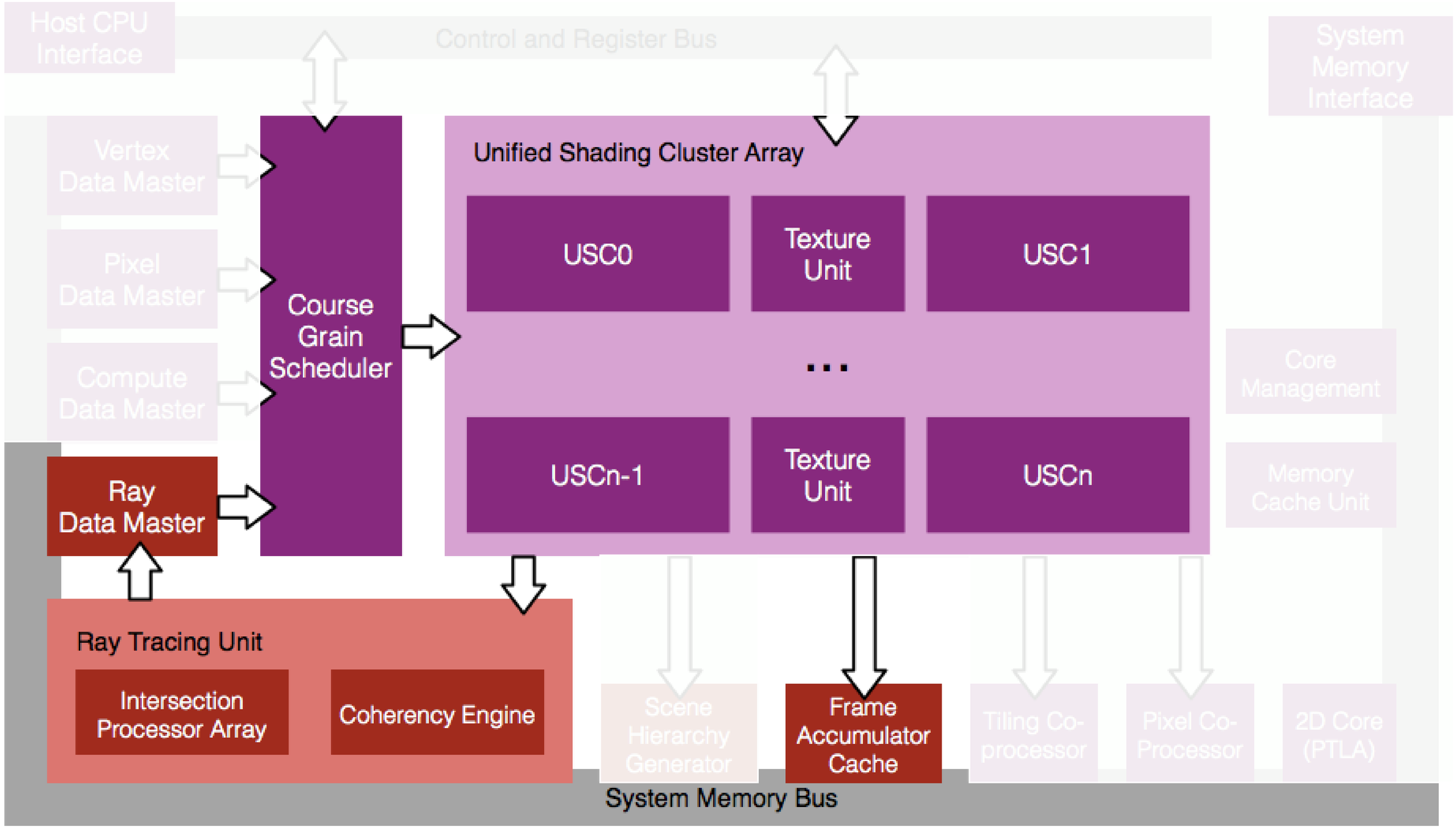

Imagination

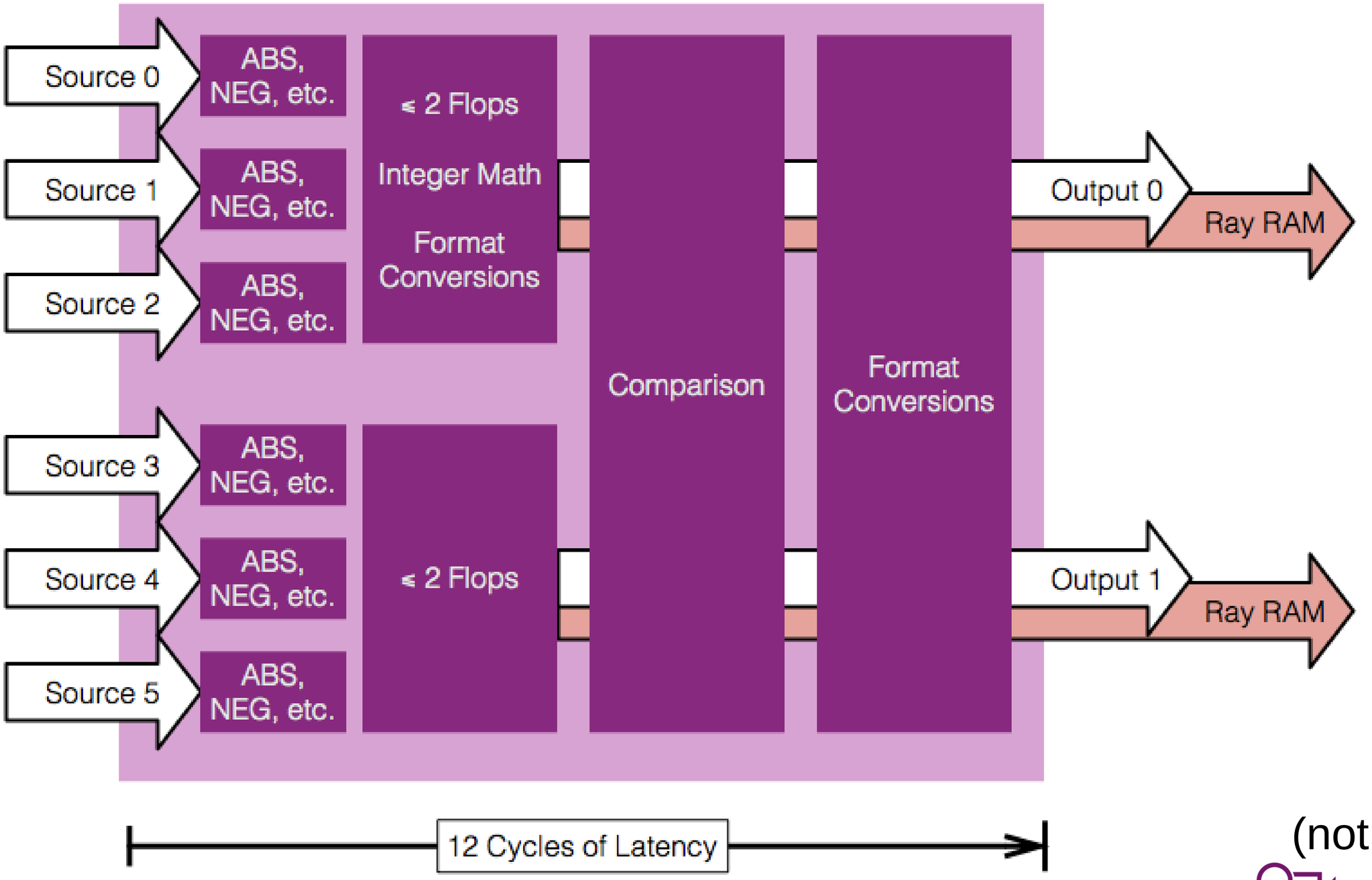
Overall capability

- Series6 gives you over 100Gflops for shading on latest mobile devices
- OpenGL ES 3.0 support
- Massive capability gain allowing math heavy shaders
- Mobile graphics is where the actual innovation is happening
- Try it!

A Technical introduction to Wizard


Host CPU Interface
Control and Register Bus
System Memory Interface
Vertex Data Master
Pixel Data Master
Compute Data Master
Ray Data Master
Course Grain Scheduler
Unified Shading Cluster Array
USC0
Texture Unit
USC1
...
USCn-1
Texture Unit
USCn
Core Management
Memory Cache Unit
Ray Tracing Unit
Intersection Processor Array
Coherency Engine
Scene Hierarchy Generator
Frame Accumulator Cache
Tiling Co-processor
Pixel Co-Processor
2D Core (PTLA)
System Memory Bus


Host CPU Interface
Control and Register Bus
System Memory Interface
Vertex Data Master
Pixel Data Master
Compute Data Master
Ray Data Master
Course Grain Scheduler
Unified Shading Cluster Array
USC0
Texture Unit
USC1
...
USCn-1
Texture Unit
USCn
Core Management
Memory Cache Unit
Ray Tracing Unit
Intersection Processor Array
Coherency Engine
Scene Hierarchy Generator
Frame Accumulator Cache
Tiling Co-processor
Pixel Co-Processor
2D Core (PTLA)
System Memory Bus



(not exact)

Imagination

New shader type

- Ray Shaders are invoked when a ray intersects a triangle
- Shaders types can emit any number of rays
- OpenRL has a frame shader to emit primary rays
- Existing fragment / pixel shaders can emit rays also!

- GLSL Programming model is the same
- Built-in functions, parameters, etc.

Imagination

Primitive Objects

- Encapsulate the rendering state for one mesh
- Includes VBOs, uniforms, shader programs, texture bindings, etc.
- Ray tracing unit sorts rays into tasks with common primitive objects

- Persistent between frames
- Mutable objects by the client

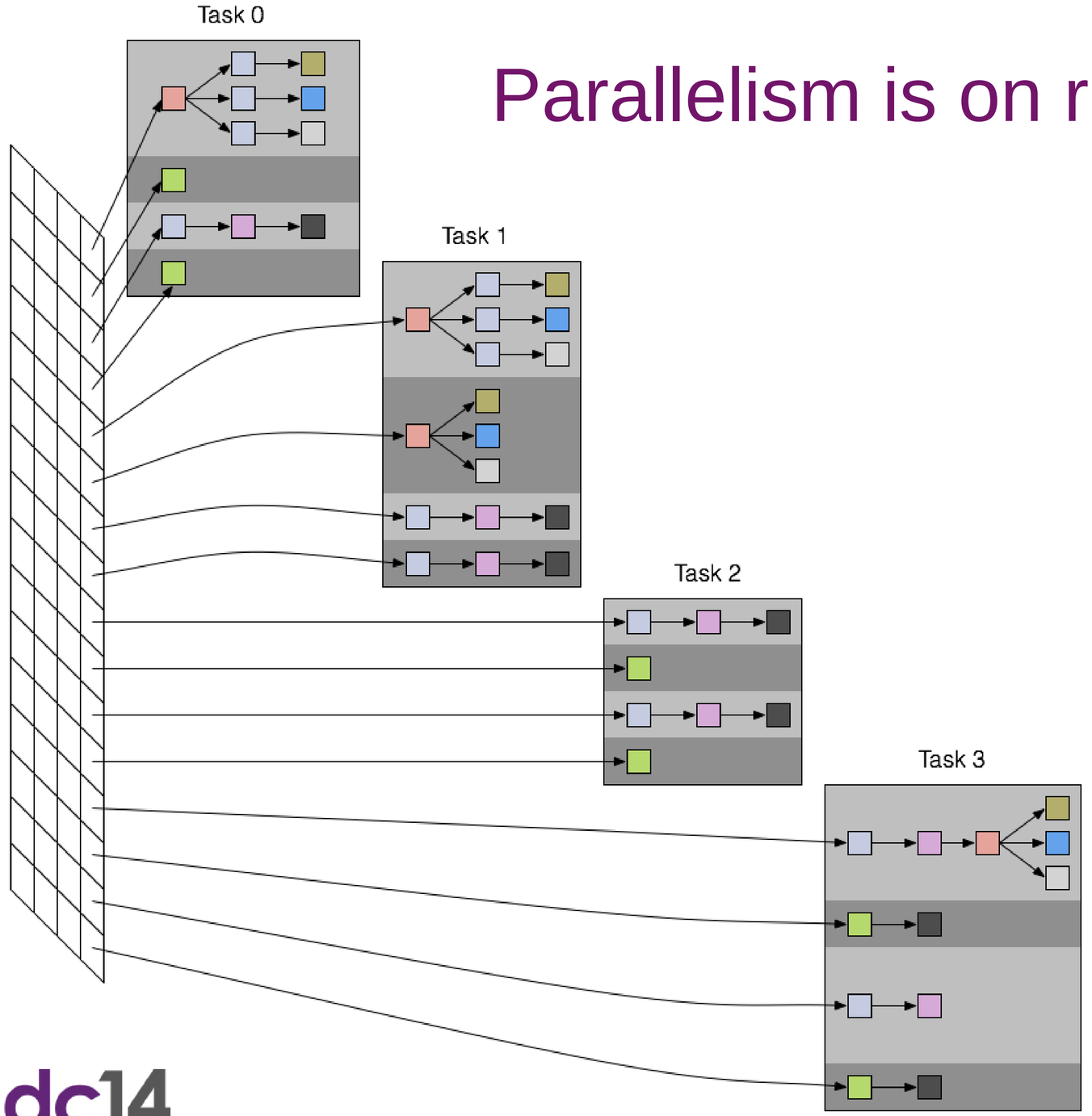
Imagination

Constraints / Limitations

- Shaders cannot wait for results of individual ray trace operations
- Shaders must provide a worst case estimate on number of child rays
- Per-ray user data payload must be carefully managed

Imagination

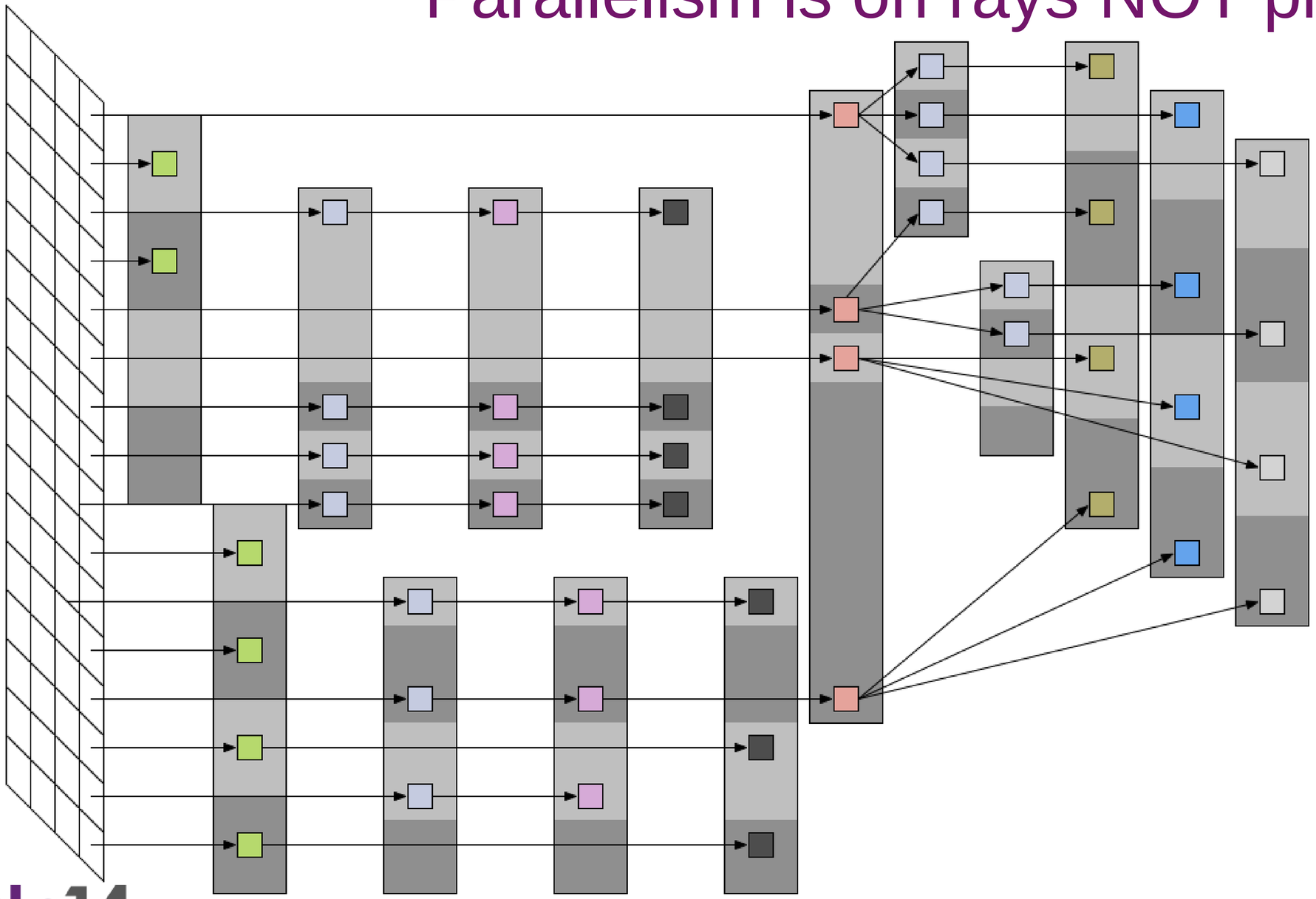
Parallelism is on rays NOT pixels



(In this example, ALU width is 4 to make the diagram smaller)

Imagination

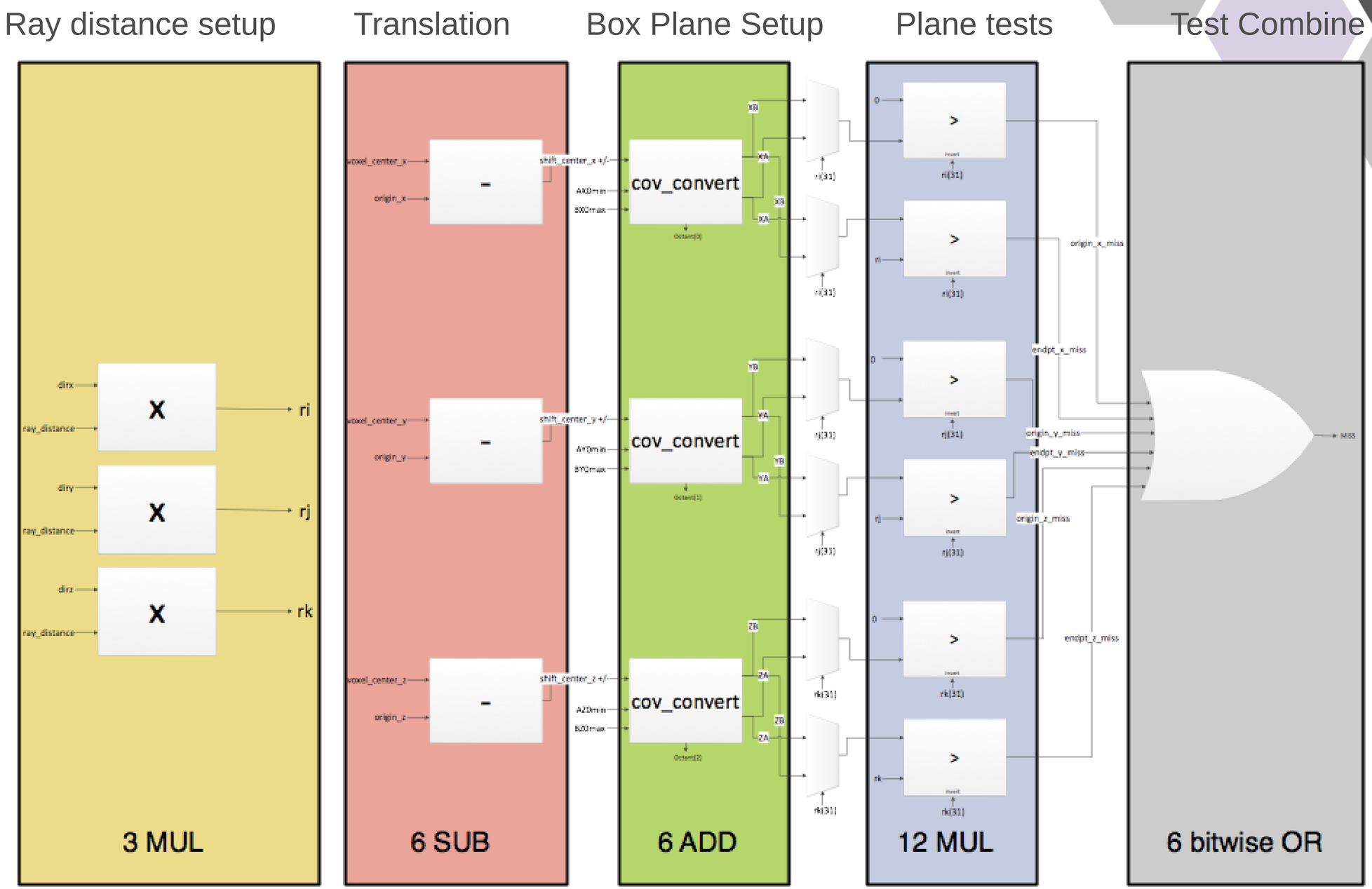
Parallelism is on rays NOT pixels



Relief: Quick ray emission demo

Intersection Processor Array
or
“Who needs Fixed Function?”

Imagination


Ray distance setup
Translation
Box Plane Setup
Plane tests
Test Combine
dirx
ray_distance
X
ri
diry
ray_distance
X
rj
dirz
ray_distance
X
rk
3 MUL
voxel_center_x
origin_x
-
shift_center_x +/-
voxel_center_y
origin_y
-
shift_center_y +/-
voxel_center_z
origin_z
-
shift_center_z +/-
6 SUB
AX0min
BX0max
AY0min
BY0max
AZ0min
BZ0max
cov_convert
cov_convert
cov_convert
6 ADD
>
ri(31)
rj(31)
rk(31)
12 MUL
origin_x_miss
endpt_x_miss
origin_y_miss
endpt_y_miss
origin_z_miss
endpt_z_miss
6 bitwise OR
Miss

Hit or Miss

USC instruction group packing

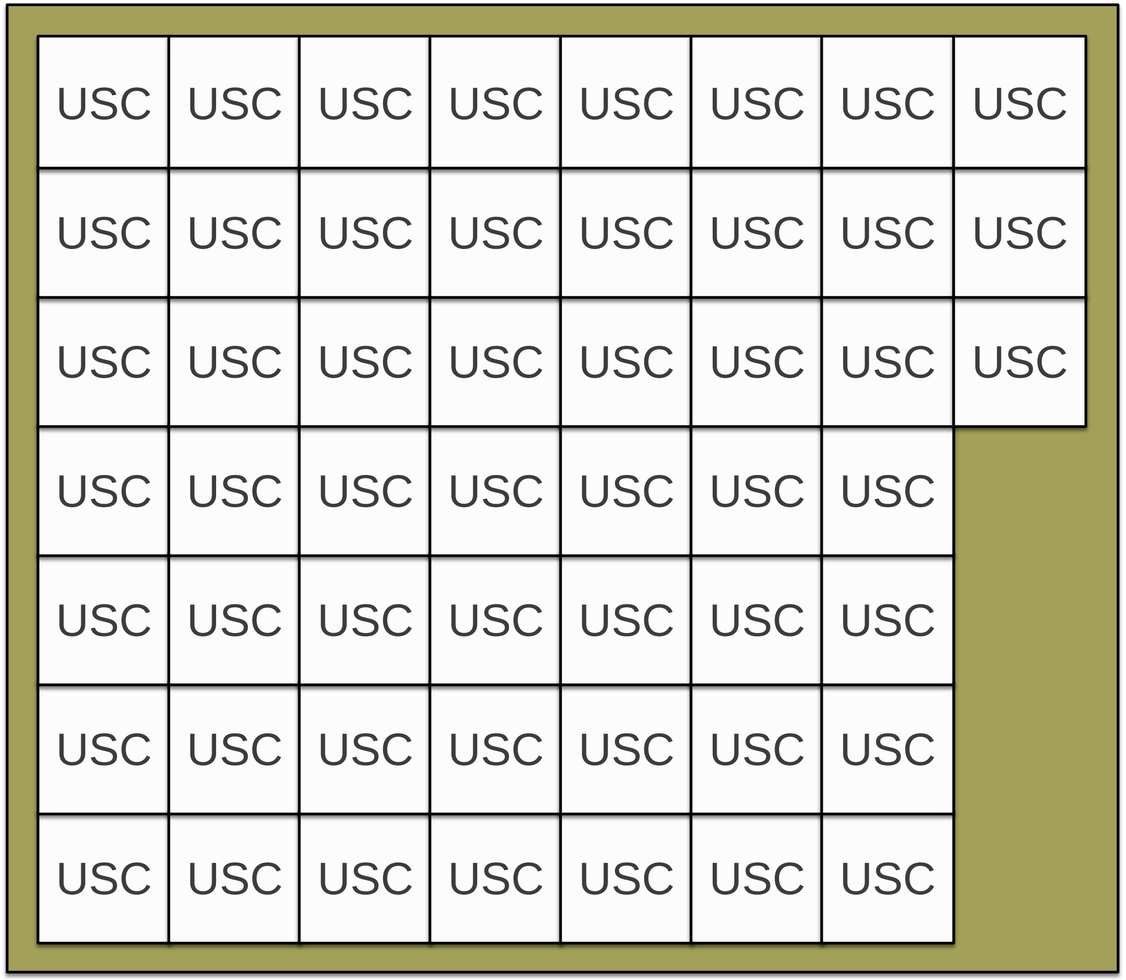
LD	MUL	ADD	MUL	OR
LD	MUL	ADD	MUL	–
LD	MUL	ADD	MUL	OR
LD	SUB	ADD	MUL	–
LD	SUB	ADD	MUL	OR
LD	SUB	ADD	MUL	–
	SUB		MUL	OR
	SUB		MUL	–
	SUB		MUL	OR
	–		MUL	–
			MUL	OR
			MUL	–

26 Instructions

(32 if using compressed data formats)

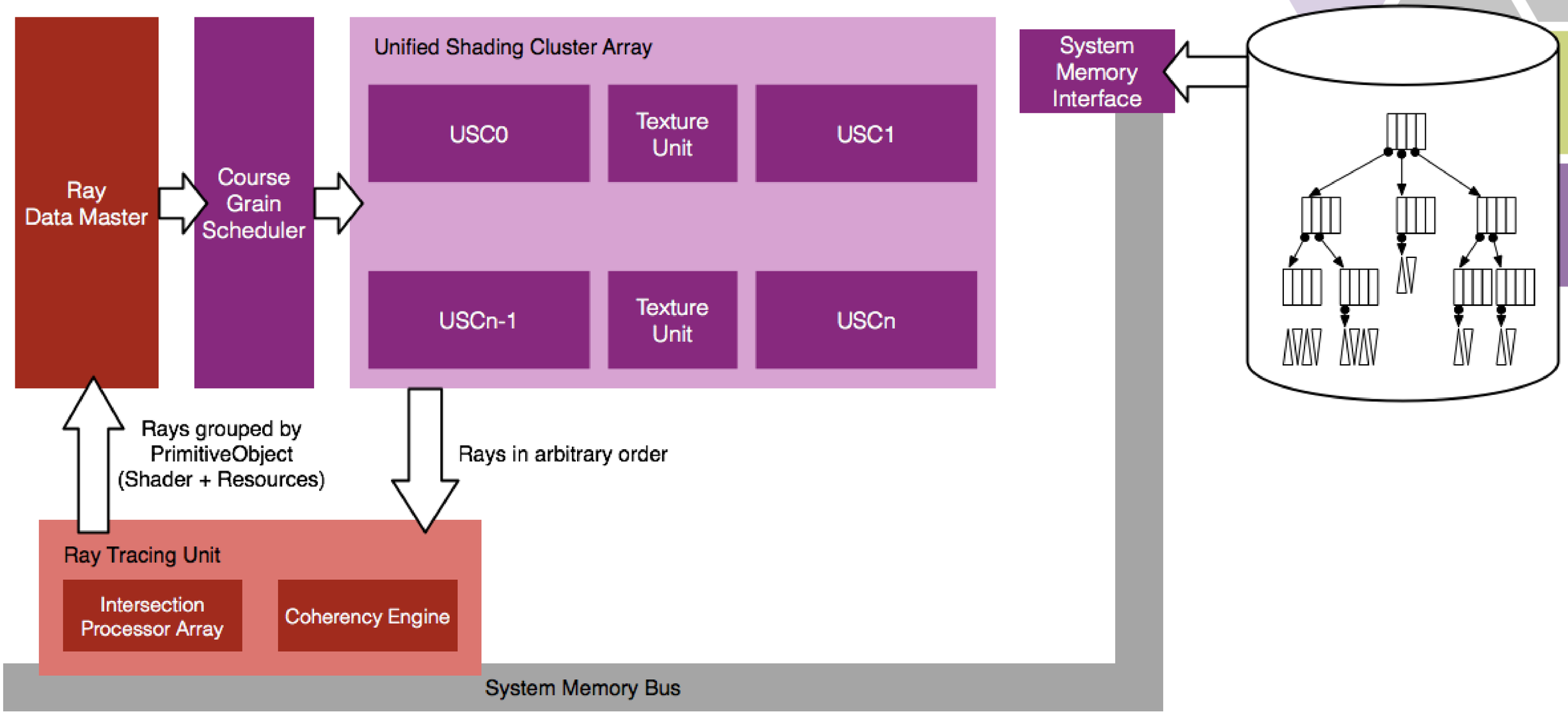
Imagination

44x less area for this function



Imagination

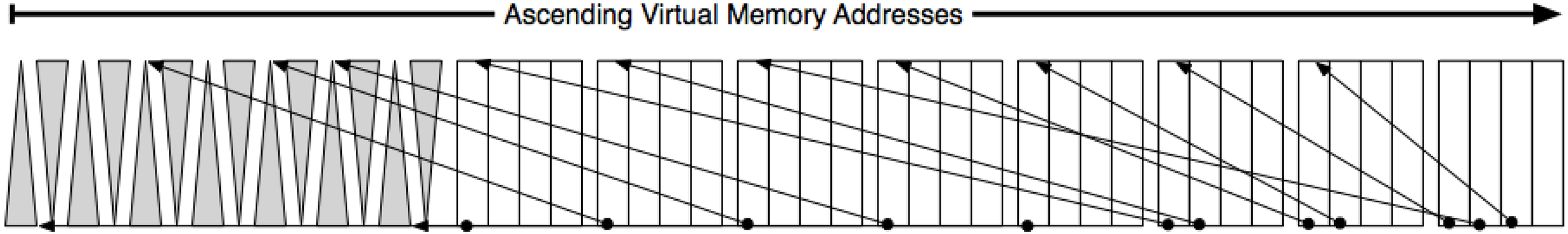
The Ray Tracing Unit
and
Coherence Engine


Ray
Data Master
Course
Grain
Scheduler
Unified Shading Cluster Array
USC0
Texture
Unit
USC1
USCn-1
Texture
Unit
USCn
System
Memory
Interface
Rays grouped by
PrimitiveObject
(Shader + Resources)
Rays in arbitrary order
Ray Tracing Unit
Intersection
Processor Array
Coherency Engine
System Memory Bus

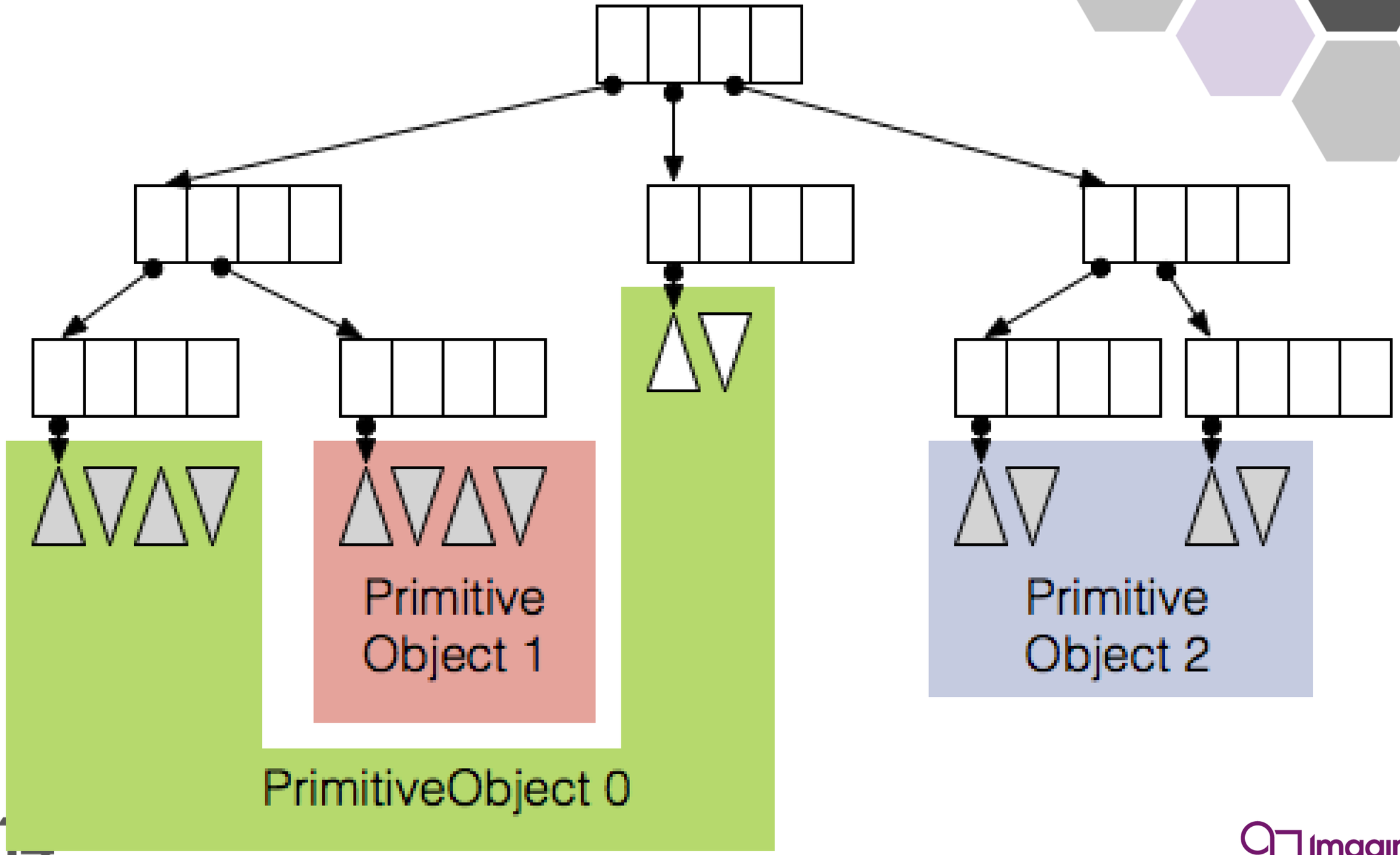
AABB Blocks

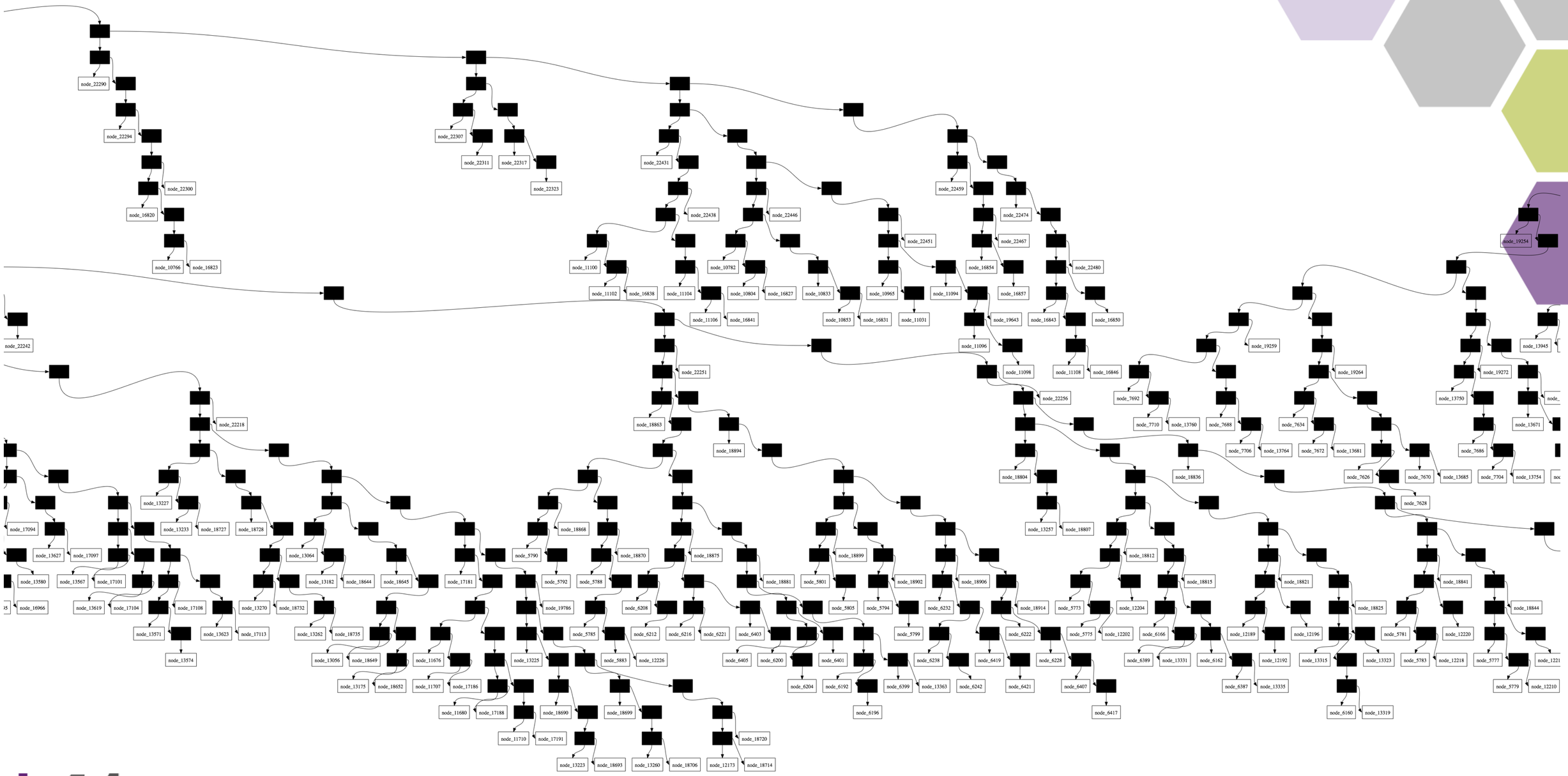
Vertex Blocks

Varyings



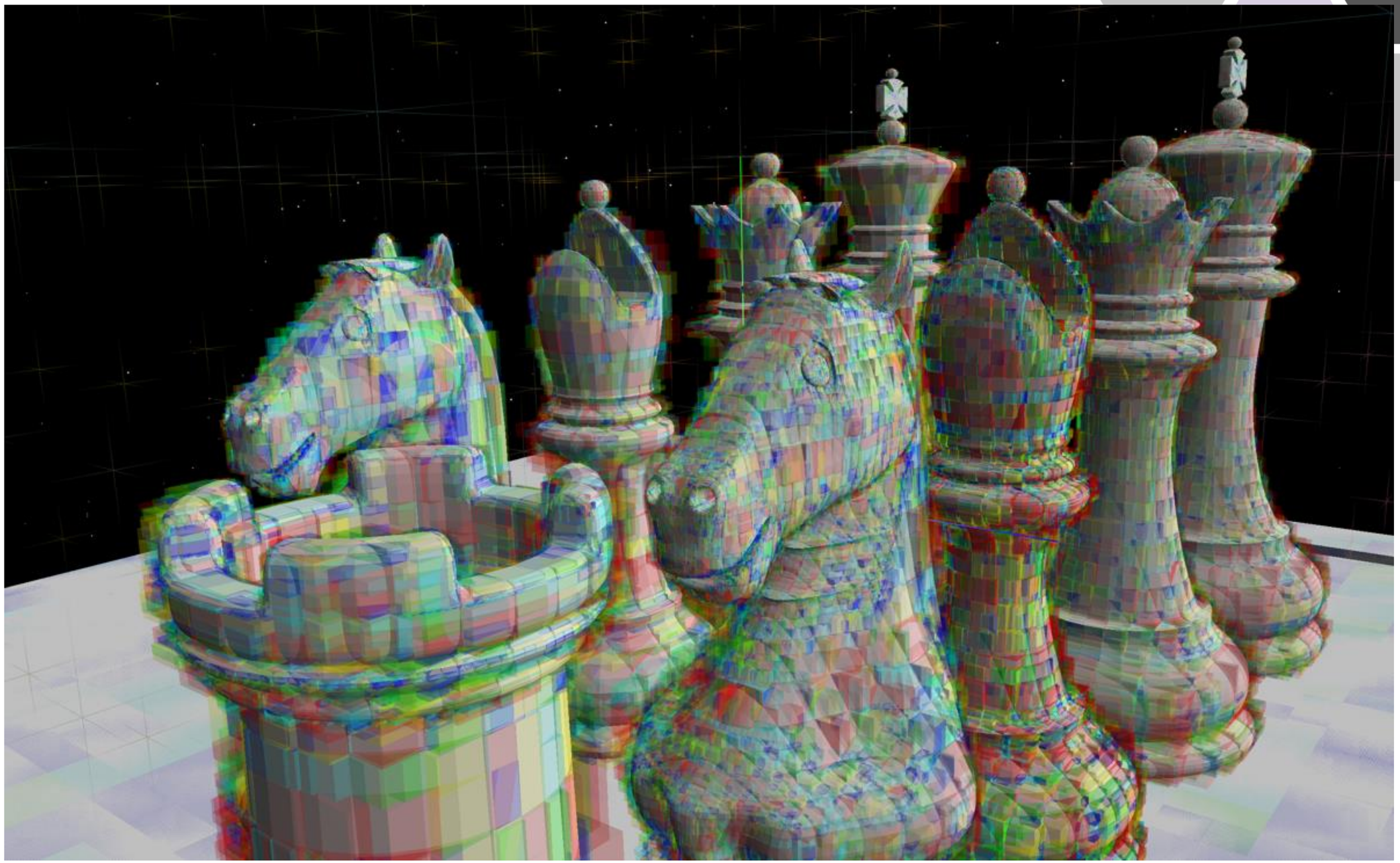
(around 100Mbytes for 1M triangles including varyings)

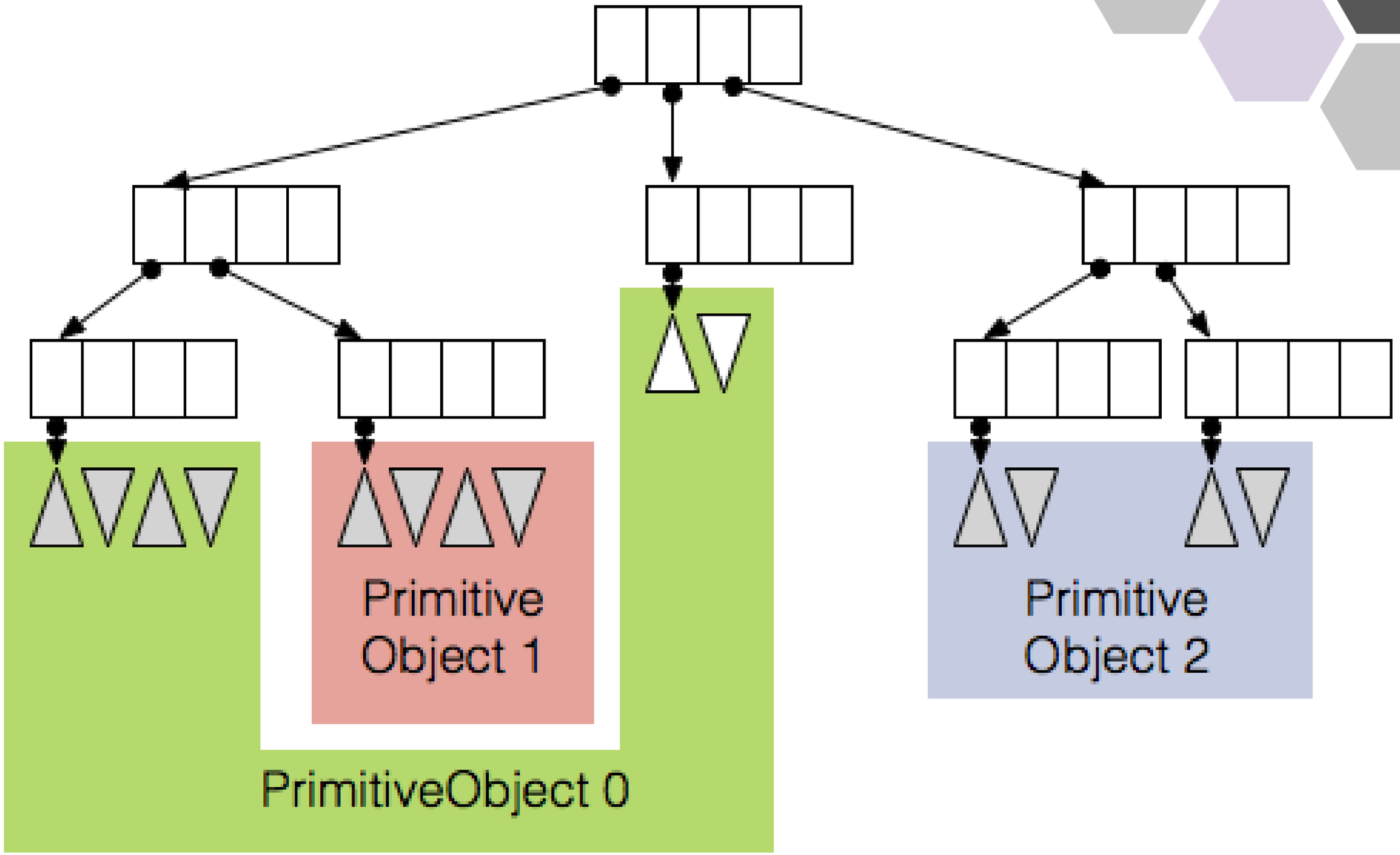

Primitive
Object 1
Primitive
Object 2
PrimitiveObject 0



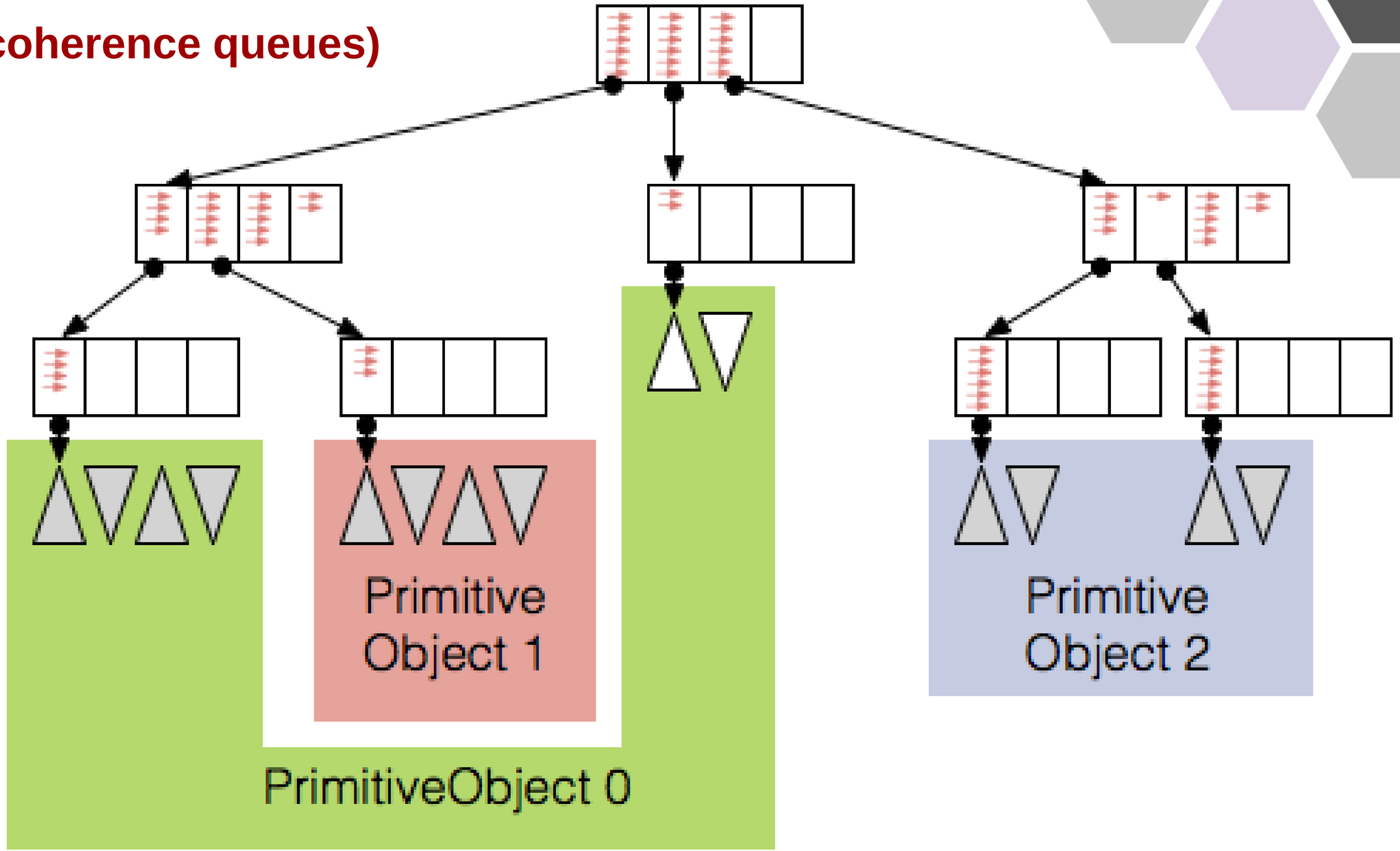
idc14

Imagination



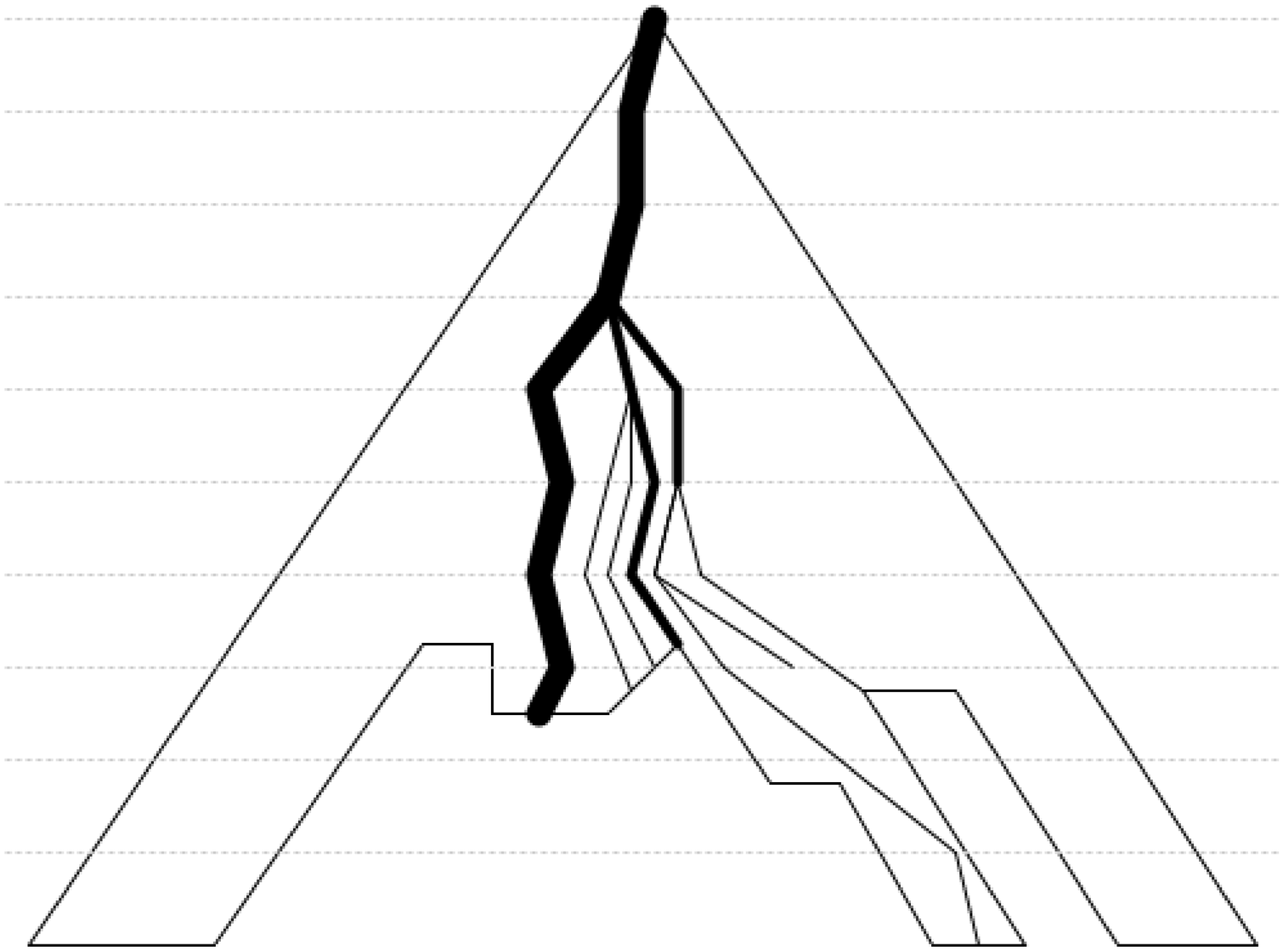

Primitive
Object 1
Primitive
Object 2
PrimitiveObject 0

(coherence queues)

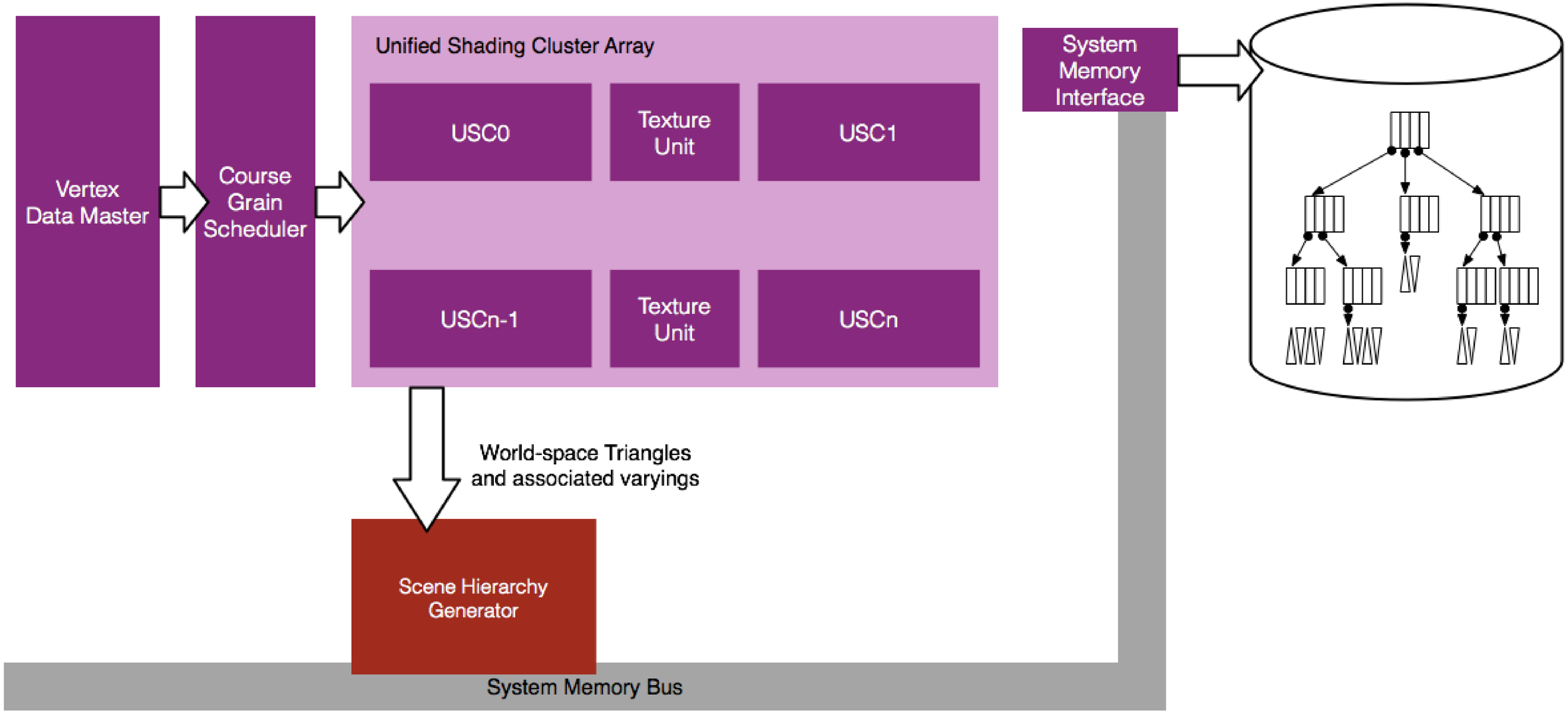


Imagination

Automatically finds coherence paths



The Scene Hierarchy Generator


Vertex Data Master
Course Grain Scheduler
Unified Shading Cluster Array
USC0
Texture Unit
USC1
USCn-1
Texture Unit
USCn
System Memory Interface
World-space Triangles and associated varyings
Scene Hierarchy Generator
System Memory Bus

Limitations

- Scene is represented by triangles – same as today
- BVH is in a defined format optimised for construction and traversal
- Triangle order must generally follow a spatially coherent flow
- An approximate scene scale estimation is needed
- Geometry shaders are not inline with the ray tracing pipeline

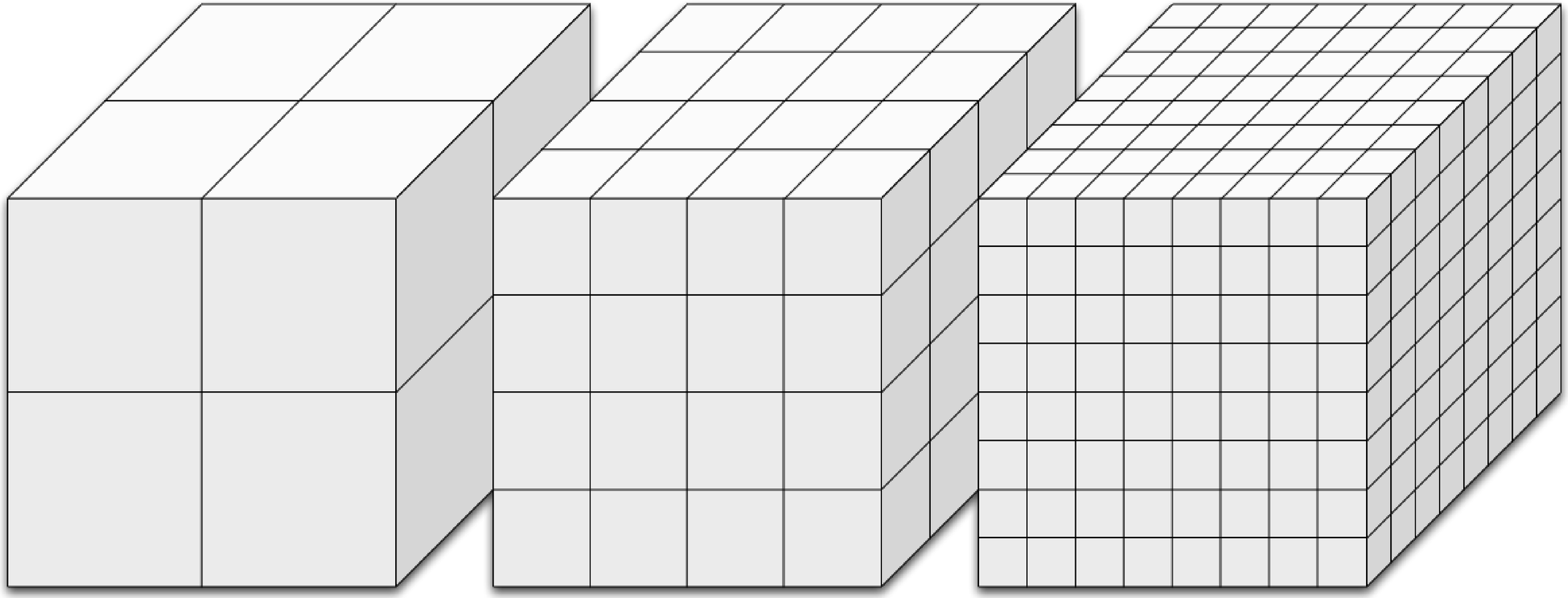
Imagination

Strengths

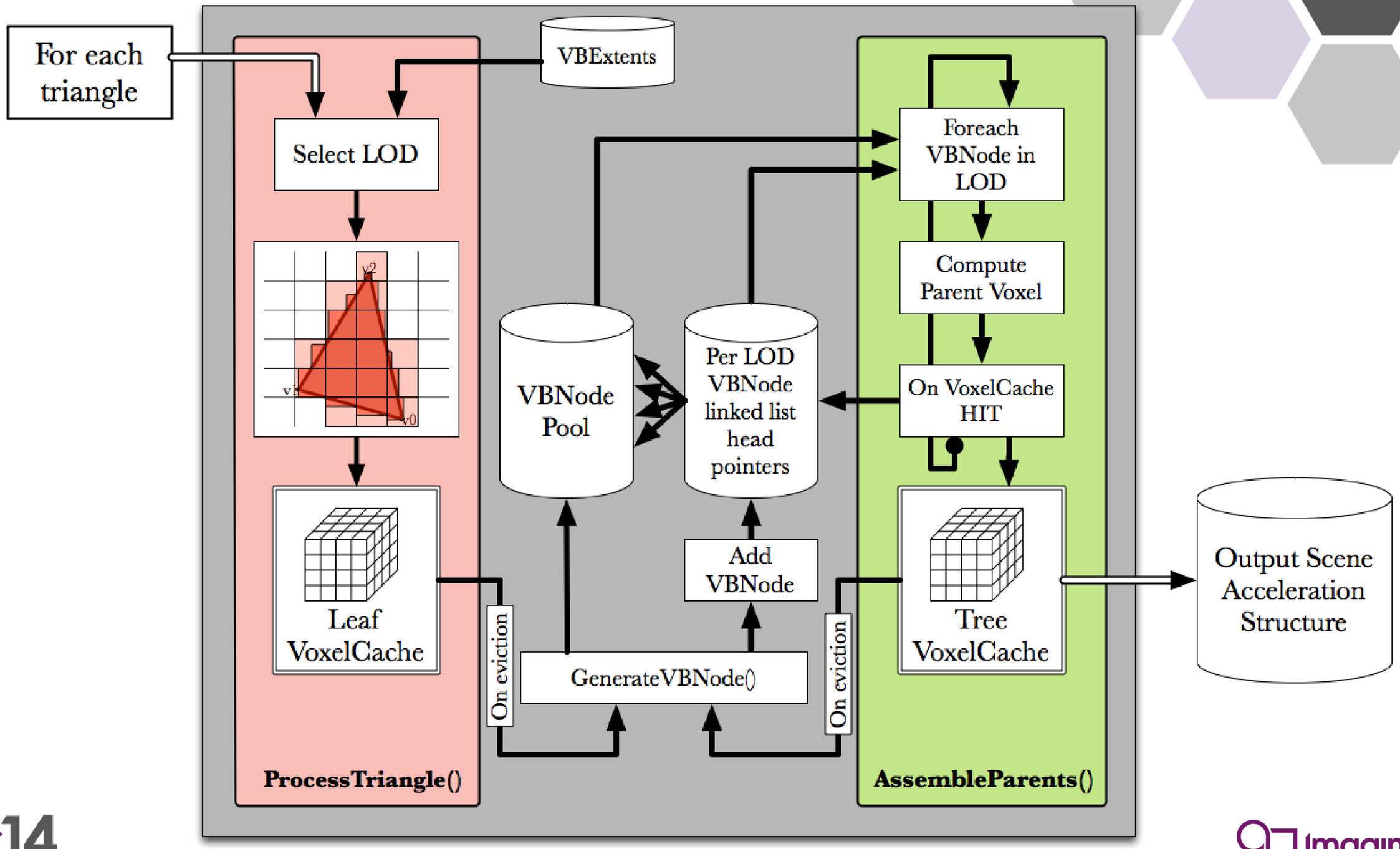
- Shading cluster workload is no higher than a vertex shader
- Only needs to process geometry that actually moved in world space
- Unique algorithm constrains working set to internal registers only
- Single pass operation: in-line with vertex shader execution
- Handles the “long skinny triangles” problem well
- Streaming writes to external memory
- Losslessly compressed output formats due to build algorithm
- Compact logic

Imagination

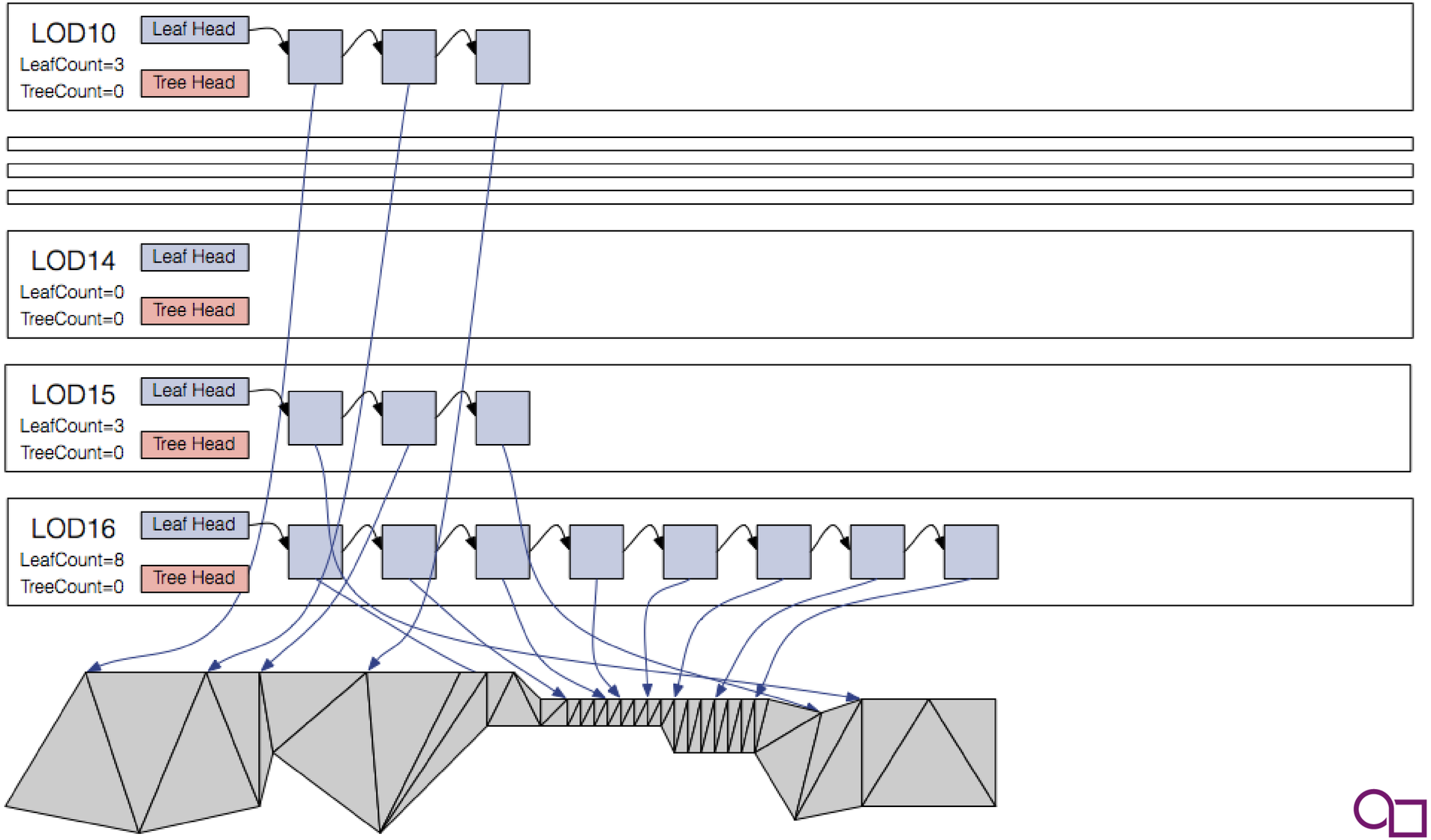
Built on sparse log2 oct-tree scaffolding



Level = 1 Level = 2 Level = 3

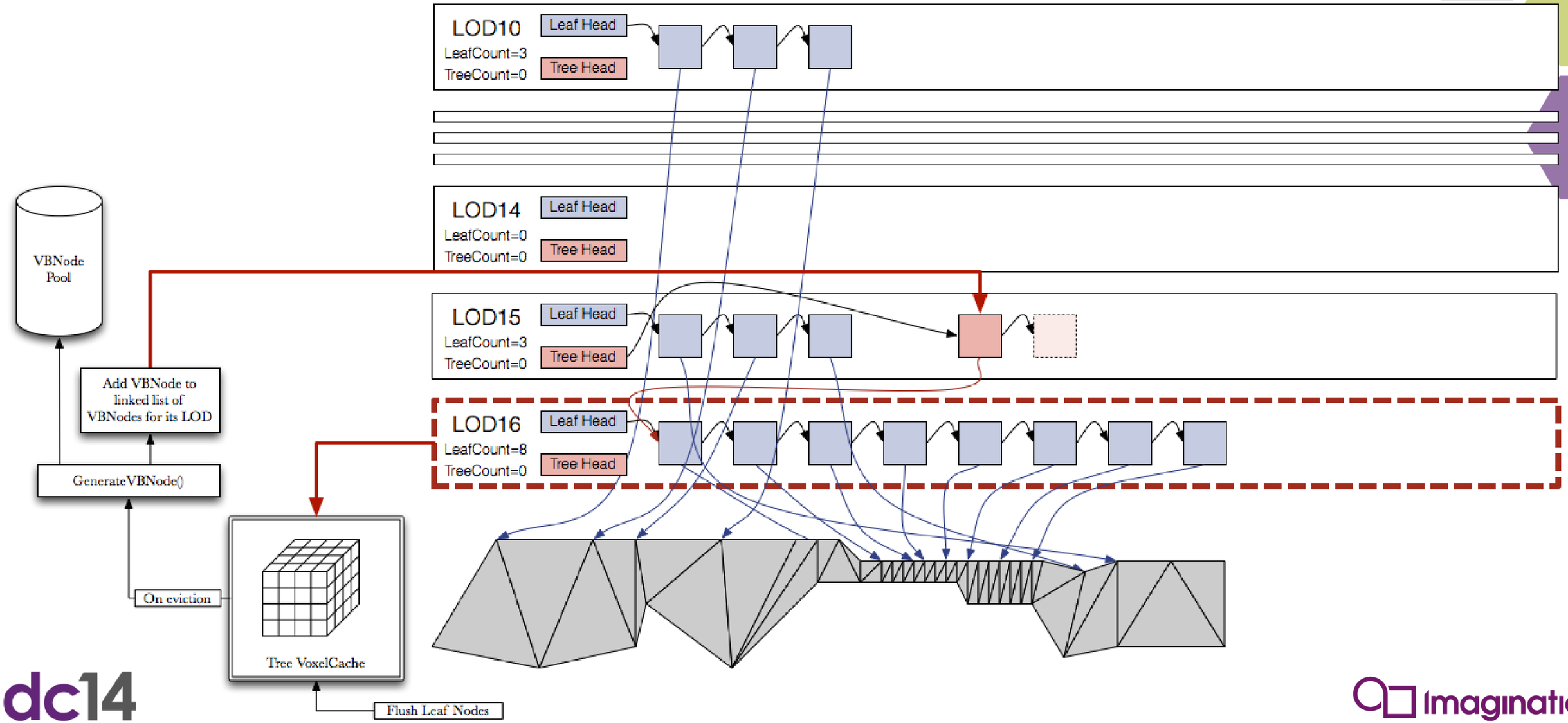

For each triangle
VBExtents
Select LOD
v2
v1
v0
Leaf VoxelCache
ProcessTriangle()
On eviction
GenerateVBNode()
VBNode Pool
Per LOD VBNode linked list head pointers
Add VBNode
On eviction
Foreach VBNode in LOD
Compute Parent Voxel
On VoxelCache HIT
Tree VoxelCache
AssembleParents()
Output Scene Acceleration Structure

After some triangles...



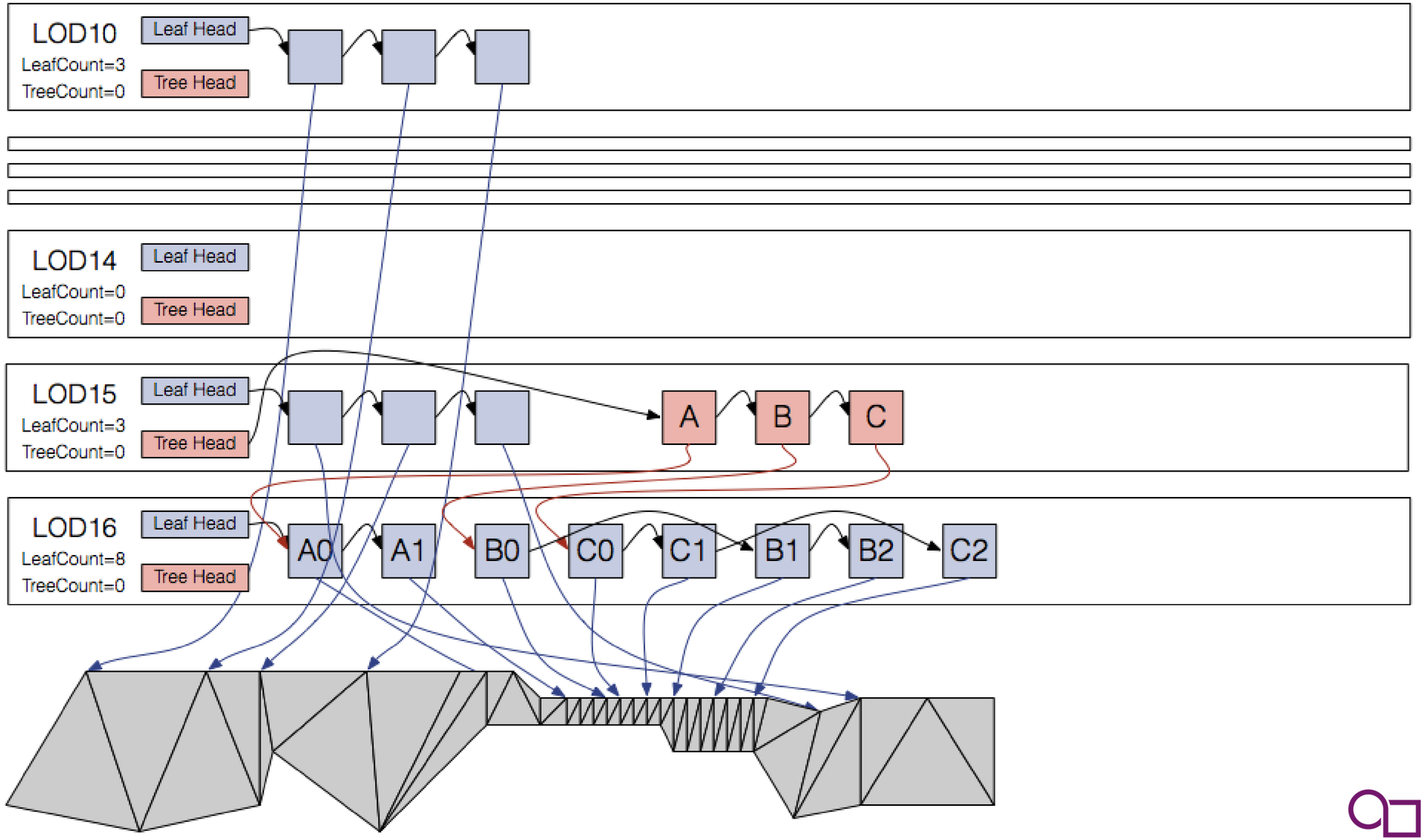
Imagination

Assembling parents...



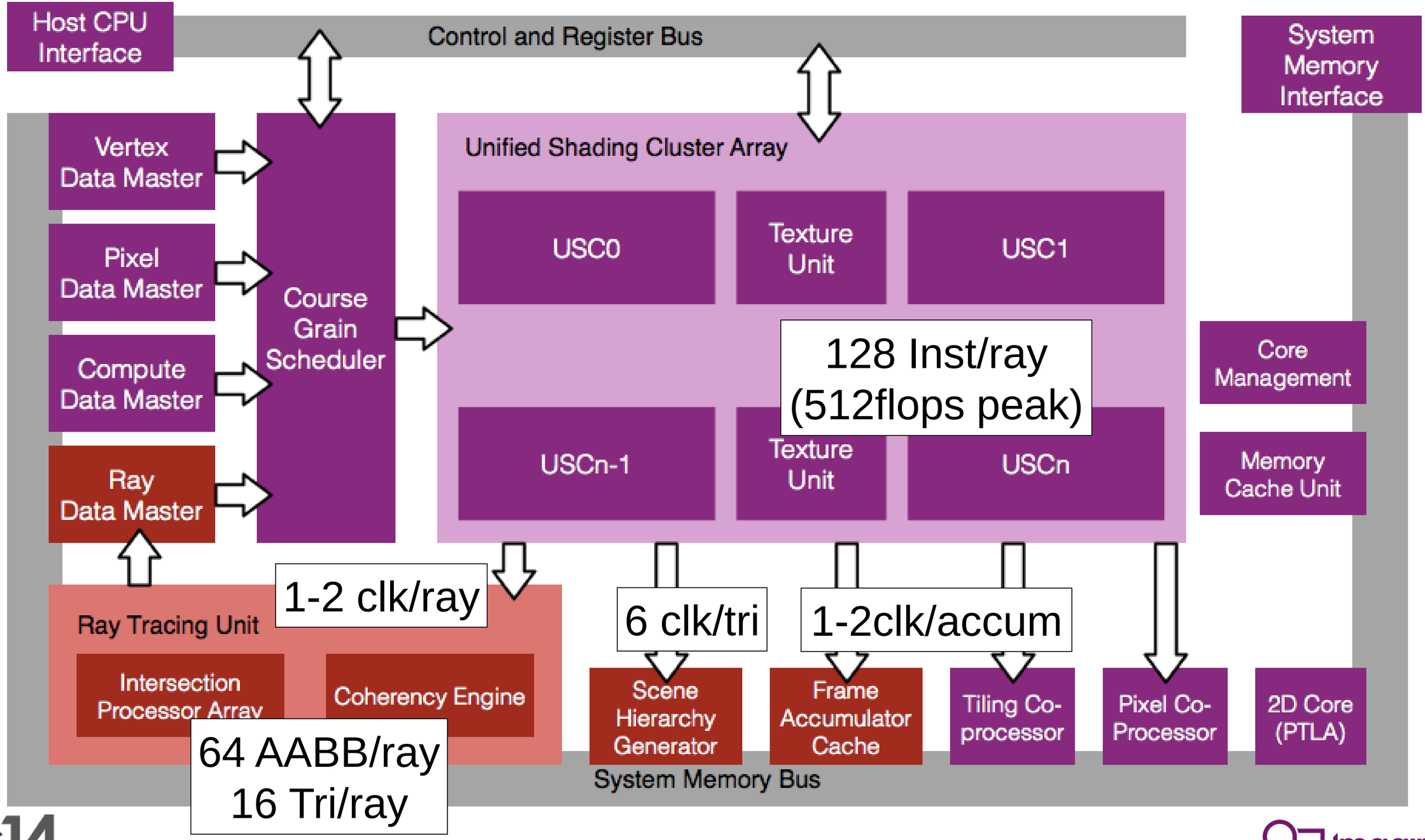
Imagination

After parents for one level have been assembled...



Imagination

Performance Expectations


Host CPU Interface
Control and Register Bus
System Memory Interface
Vertex Data Master
Pixel Data Master
Compute Data Master
Ray Data Master
Course Grain Scheduler
Unified Shading Cluster Array
USC0
Texture Unit
USC1
USCn-1
Texture Unit
USCn
128 Inst/ray (512flops peak)
Core Management
Memory Cache Unit
1-2 clk/ray
Ray Tracing Unit
Intersection Processor Array
Coherency Engine
64 AABB/ray
16 Tri/ray
6 clk/tri
1-2clk/accum
Scene Hierarchy Generator
Frame Accumulator Cache
Tiling Co-processor
Pixel Co-Processor
2D Core (PTLA)
System Memory Bus

Get early access to the programming concepts at:
http://tinyurl.com/9ddlsv2

Talk immediately after this on Hybrid rendering

See the demos at Imagination Booth #402 South Hall

QA

Thank you