AN EVOLUTION OF MOBILE GRAPHICS

Michael C. Shebanow
Vice President, Advanced Processor Lab
Samsung Electronics

July 20, 2013
DISCLAIMER

• The views herein are my own
• They do not represent Samsung’s vision nor product plans
- The Mobile Market
- Review of GPU Tech
- GPU Efficiency
- User Experience
- Tech Challenges
- Summary
The Rise of the Mobile GPU & Connectivity

A NEW WORLD COMING?
DISCRETE GPU MARKET

Graphics shipment 1981 to present (M units)

- Desktop GPUs
- Notebook GPUs
- Desktop PCs
- Notebook PCs
- TOTAL Graphics (M units) 30 Years CAGR 22.2%
- TOTAL PCs (M units) 30 yr. CAGR 21%

Flattening
MOBILE GPU MARKET

- In 2012, an estimated 800+ million mobile GPUs shipped
  - ~123M tablets
  - ~712M smart phones
- Will easily exceed 1B in the coming years

- **Trend:**
  - Discrete GPU relatively flat
  - Mobile is growing rapidly

![Chart showing Tablets and PCs (M Units)]
The internet traffic growth rate is staggering.

- **2012** total traffic is 13.7 GB per person per month.
- **2012** smart phone traffic at 0.342 GB per person per month.
- **2017** smart phone traffic expected at 2.7 GB per person per month.

### WW INTERNET TRAFFIC

**Source**: Cisco VNI

<table>
<thead>
<tr>
<th>Year</th>
<th>IP Traffic (TB/sec)</th>
<th>Growth Rate</th>
<th>Mobile INET Traffic (TB/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>0.9</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>2006</td>
<td>1.5</td>
<td>65%</td>
<td>0.00</td>
</tr>
<tr>
<td>2007</td>
<td>2.5</td>
<td>61%</td>
<td>0.01</td>
</tr>
<tr>
<td>2008</td>
<td>3.8</td>
<td>54%</td>
<td>0.01</td>
</tr>
<tr>
<td>2009</td>
<td>5.6</td>
<td>45%</td>
<td>0.04</td>
</tr>
<tr>
<td>2010</td>
<td>7.8</td>
<td>40%</td>
<td>0.10</td>
</tr>
<tr>
<td>2011</td>
<td>10.6</td>
<td>36%</td>
<td>0.23</td>
</tr>
<tr>
<td>2012</td>
<td>12.4</td>
<td>17%</td>
<td>0.34</td>
</tr>
</tbody>
</table>
WHERE ARE WE HEADED?...

- Enormous quantity of GPUs
- Large amount of interconnectivity
- Better I/O
The OpenGL ES 2.0 pipeline

GPU Pipelines

A BRIEF REVIEW OF GPU TECH
MOBILE GPU PIPELINE ARCHITECTURES

Tile-based immediate mode rendering (TBIMR)

Tile-based deferred rendering (TBDR)

IA = input assembler
VS = vertex shader
CCV = cull, clip, viewport transform
RS = rasterization, setup
PS = pixel shader
ROP = raster operations (blend)
TBDR W/ HSR

- **HSR = hidden surface removal**
  - Sort all objects across each projection ray
    - Use tiling to reduce data set size
  - Only nearest opaque and closer transparent objects need to be drawn
  - Remaining fragments can be killed => not drawn
# Mobile GPU Landscape

<table>
<thead>
<tr>
<th>Company</th>
<th>Product</th>
<th>Pipeline</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM</td>
<td>Mali</td>
<td>TBIMR</td>
<td>Unified shader, 2-4 math pipes per core</td>
</tr>
<tr>
<td>Imagination</td>
<td>PowerVR</td>
<td>TBDR/HSR</td>
<td>Latest is Rogue (S6). Unified shader. DX11 support</td>
</tr>
<tr>
<td>Qualcomm</td>
<td>Adreno</td>
<td>FlexRender</td>
<td>Unified shader. &quot;FlexRender&quot; = automatic switching between direct render (IMR) and tile-based deferred rendering (TBDR).</td>
</tr>
<tr>
<td>NVIDIA</td>
<td>Tegra</td>
<td>TBDR &amp; TBIMR</td>
<td>Evolution:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Tegra 1/2/3/4: non-unified TBDR architecture</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Logan: Kepler-based GPU, TBIMR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Parker: Maxwell-based GPU, TBIMR</td>
</tr>
<tr>
<td>Vivante</td>
<td>ScalarMorphic</td>
<td>IMR</td>
<td>Unified Shader.</td>
</tr>
<tr>
<td>Intel</td>
<td>Gen / Atom</td>
<td>IMR / PowerVR</td>
<td>Market leader in integrated graphics. Atom-based devices using Imagination PowerVR</td>
</tr>
<tr>
<td>AMD</td>
<td>Radeon</td>
<td>IMR</td>
<td>Hondo/Temash pipes.</td>
</tr>
</tbody>
</table>
A PATH TO A BETTER MOBILE GPU? [PART 1]
WHAT IS IMPORTANT?

- More with less
- Better user experience
POWER EFFICIENCY

• Performance = power efficiency

• Two types of efficiency:
  • “perf@watts”:
    
    The ability to deliver maximum performance
  
  • “watts@perf”:
    
    The ability to deliver maximum battery life at a minimum required performance
WHAT IS EFFICIENCY?

- **Perf @ Watts**
  - *Maximum performance at some power limit*
  - Limits:
    - electrical (Pidd)
    - die temp (Tj)
    - skin temp (Tcase)
    - battery life (Pbat)

- **Watts @ Perf**
  - *Minimum power at constant performance*
  - Example: deliver 60 frames/sec at lowest power
PARALLELISM

- Parallel vs. Sequential
  - Parallel $\Rightarrow$ independence
  - Sequential $\Rightarrow$ dependence
- Three fundamental forms of parallelism
  - Spatial: executing operations between threads at the same time
  - Temporal: executing operations between threads at the same place
  - ILP: executing operations from within the same thread in parallel
- Fundamental differences between ILP-only machines and massive TLP-ILP machines
  - CPUs vs. GPUs
THROUGHPUT VS. LATENCY

- Throughput = rate at which operations complete
- Latency = time it takes to complete an operation or set of operations
- CPUs versus GPUs
  - In CPUs, the primary objective is low latency
  - In GPUs, the primary objective is high throughput
- CPUs versus GPUs
  - In an application suitable for CPUs, we assume a low degree of TLP
  - In an application suitable for GPUs, we assume a high degree of TLP
GPU PERFORMANCE

- Supply and demand:

\[ \vec{S} \geq \lambda \vec{D} \]

(“limiter equation”)

- Lambda (\(\lambda\)) is throughput
- Supply examples:
  - FP BW (flops/clock)
  - Texture BW (quads/clock)
  - Memory BW (bytes/clock)
- Demand density examples:
  - FP ops per shader
  - Sample ops per shader
ENERGY REDUCTION TECHNIQUES

- Work Reduction
- Memory Avoidance
- Memory BW Reduction
- Memory Access Management
WORK REDUCTION

• Pixel shaders in ES games ~95% of the shader load
  • A pixel shader killed is raw power savings
  • HSR can kill 30-50% of the shader threads

• Geometry in DX11 a problem
  • Unigine Heaven ~10M Tri/frame

• Inter-frame work reduction?
RELATIVE ACCESS ENERGY COSTS

- LPDDR
- WIO1
- SRAM
- Small RF
- SP FMA

Energy/byte
Energy/op
MEMORY AVOIDANCE

• Memory power a problem
  • LPDDR ~150 pJ/byte
    (150 mW @ 1 GB/sec)
  • WIO1 ~24 pJ/byte
    (24 mW @ 1 GB/sec)
  • On-chip SRAM ~0.6 pJ/byte
    (0.6 mW @ 1 GB/sec)

• Reduction in working set for non-essential traffic (i.e., not texture, attribute, command, or render target)
  • Rematerialize? (computation vs. BW)
  • Scheduling to reduce lifetimes?
MEMORY BW REDUCTION

• Texture compression (RD)
  • Better compression?
  • Tessellation use of textures?

• Tile compression (WT)
  • TB-based signature checking
  • Lossless compression

• Attribute compression (RD)
  • Reduce stream BW
MEMORY ACCESS MANAGEMENT

• SOC memory architecture
  • Blood rivals (antagonists)
  • Effect of CPU/GPU traffic on Memory Controller (MC)
    • Intelligent page open/close management
    • Balance latency vs. BW

• Mismanaging DRAM results in both performance loss AND extra energy – double whammy
A better user experience…

A PATH TO A BETTER MOBILE GPU? [PART 2]
ISO 9241-210[1] defines user experience as "a person’s perceptions and responses that result from the use or anticipated use of a product, system or service". - Wikipedia
APPLICATION: NAVIGATION

- Hercules
  - Rises: 6:03 a.m.
  - Sets: 9:25 p.m.
  - A large constellation representing the mythological hero

- U.S. Capitol complex
  - 0.7 miles
  - Construction of the Capitol began in 1793.
  - When built, it was...

- CoroNA borealis

- Sanphat restaurant
  - 41 reviews
  - Thai
  - $$

- Gas station
  - 550 feet
  - Unleaded
  - $3.19

- Car locator

- Eastern Market
  - 580 feet
  - Turn right on 7th St.
APPLICATION: FACE RECOGNITION
APPLICATION: TELEPRESENCE

http://www.youtube.com/watch?feature=player_detailpage&v=NziOs81tP4

APPLICATION: VIRTUAL COMPUTER
The UX Opportunity

- Killer apps will be integration of:
  - AR/MR technology
  - Big Data operations

- Subject to:
  - Real-time constraints
  - Parallelization on a massive scale
Making a better UX

FUTURE MOBILE TECH CHALLENGES?
KEY CHALLENGES

• I/O:
  • AR Headsets
  • Environment Imaging

• Computational:
  • API Improvements
  • Cloud-device integration
AR HEADSETS

• Google Glass is pretty cool, but...

• Better imaging
  • Stereo/Light field
  • HD → UHD
  • Speed

• More sensors

• Wireless power?

• Fashion/ubiquity
• For telepresence, headset camera is insufficient
• Need “environment cameras”
• Lots of privacy concerns
• Localizing environment to a client?
API IMPROVEMENTS

• Today’s APIs are power inefficient

• Needed:
  • Hints
  • State-less rendering
    • API commands supply state with action
  • Frame-less rendering
    • Compositing deferred and on-demand
  • Hierarchical geometry
    • Deferred detail
CLOUD-DEVICE INTEGRATION

• SW Challenge:
  • Making cloud queries easier
  • Utilizing the parallelism of the cloud

• Ultimate challenge:
  • The “network GPU”
  • Analogously extend the GPU model to network scale
  • $10^9$ GPUs $\rightarrow 10^{21}$ FLOPs?
SUMMARY

- Mobile computing, in particular graphics, is growing rapidly and becoming ubiquitous
- Tomorrow’s machines:
  - Ever improving efficiency
  - Integrated visual UX
  - Tied to the cloud
- Challenges remain to make this a reality
- Exciting prospects…
The End