July 2019

Qualcomm

Mobile GPU approaches to power efficiency

Andrew Gruber

VP, Technology
Qualcomm Technologies, Inc.

Overview

- What is the typical mobile environment?
- Rendering algorithm differences with desktop
- Performance and perf/watt comparison with desktop
- Physical design and power management

Mobile Memory Systems

Architecture

- Desktop GPUs have dedicated DDR
  - Typically GDDR used for higher end GPUs
- Mobile GPUs share DDR with other IPs
  - LPDDR used for Mobile SOCs
  - Mobile GPU typically given low priority (higher latency) compared to other real-time IP's (Camera/Modem) on the SOC

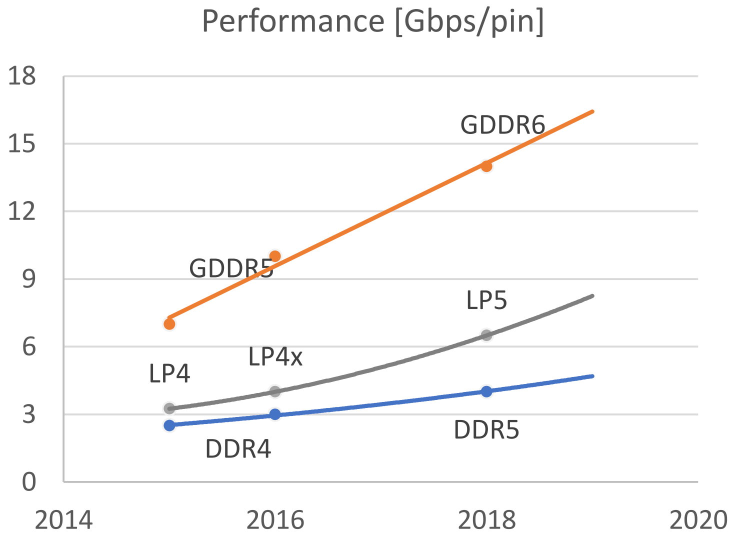
Performance/Power

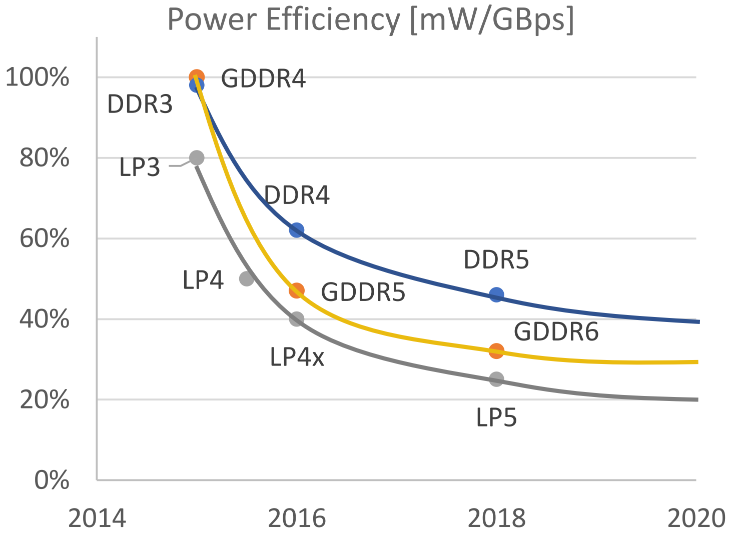
- LPDDR has less than half BW per pin compared to GDDR
- LPDDR ~10% more power efficient than GDDR at peak freq
  - Larger benefit for LPDDR exists at lower frequencies

Product	Class	Tier	BW (GB/s)	Capacity (GB)
Nvidia 2080 Ti	Desktop	High	616	11
Nvidia 1050	Desktop	Low	112	2
Galaxy S10	Mobile	High	33	8
Nokia 8810 4g	Mobile	Low	4	0.5

Memory technology trend

- GDDR6 with over 14Gbps, beyond 10Gbps GDDR5
- LP5, 20% more power-efficient than LP4X





Mobile GPU, APIs and other trends

- In terms of APIs, recent mobile GPUs are on parity with Desktop parts
  - DX12 and Vulkan are widely supported including shaders required for Tesselation
- New features such as Variable Rate Shading and WaveMath will rapidly migrate to Mobile
- As new rendering techniques (Ray Tracing/Mesh Shading) gain traction in the Desktop space, they may migrate to mobile.
- Power saving features (Render Target Compression, FP16 math ops, ASTC, Vulkan Subpasses) show up first – even prior to Desktops.
- Gaming capability is rapidly approaching previous Generation consoles. The Qualcomm® Snapdragon™ 855 mobile platform in Galaxy S10 has 954 GFLOPS of Shader performance vs. 1300 GFLOPS in the Xbox-One.
- Mobile SoCs are widely used in VR applications and have support for View Instancing
- Mobile GPUs do support OpenCL compute and machine learning, often with some specialized 8 bit integer instructions. Within the mobile SOC it is common to have dedicated 'AI' cores as well.

Specs	Qualcomm® Adreno™ 640 GPU	Xbox One
Shader ALU (FP32)	954.7 GFLOPs	1300 GFLOPs
Shader ALU (FP16)	1853.3 GFLOPs	1300 GFLOPs
Texture (Bilinear)	28.1 Gtex/sec	40.9 Gtex/sec
ROPs	9.4 Gpix/sec	13.6 Gpix/sec
System/Tile Memory Bandwidth	34.1 GB/sec 300 GB/sec	68 GB/sec 200 GB/sec

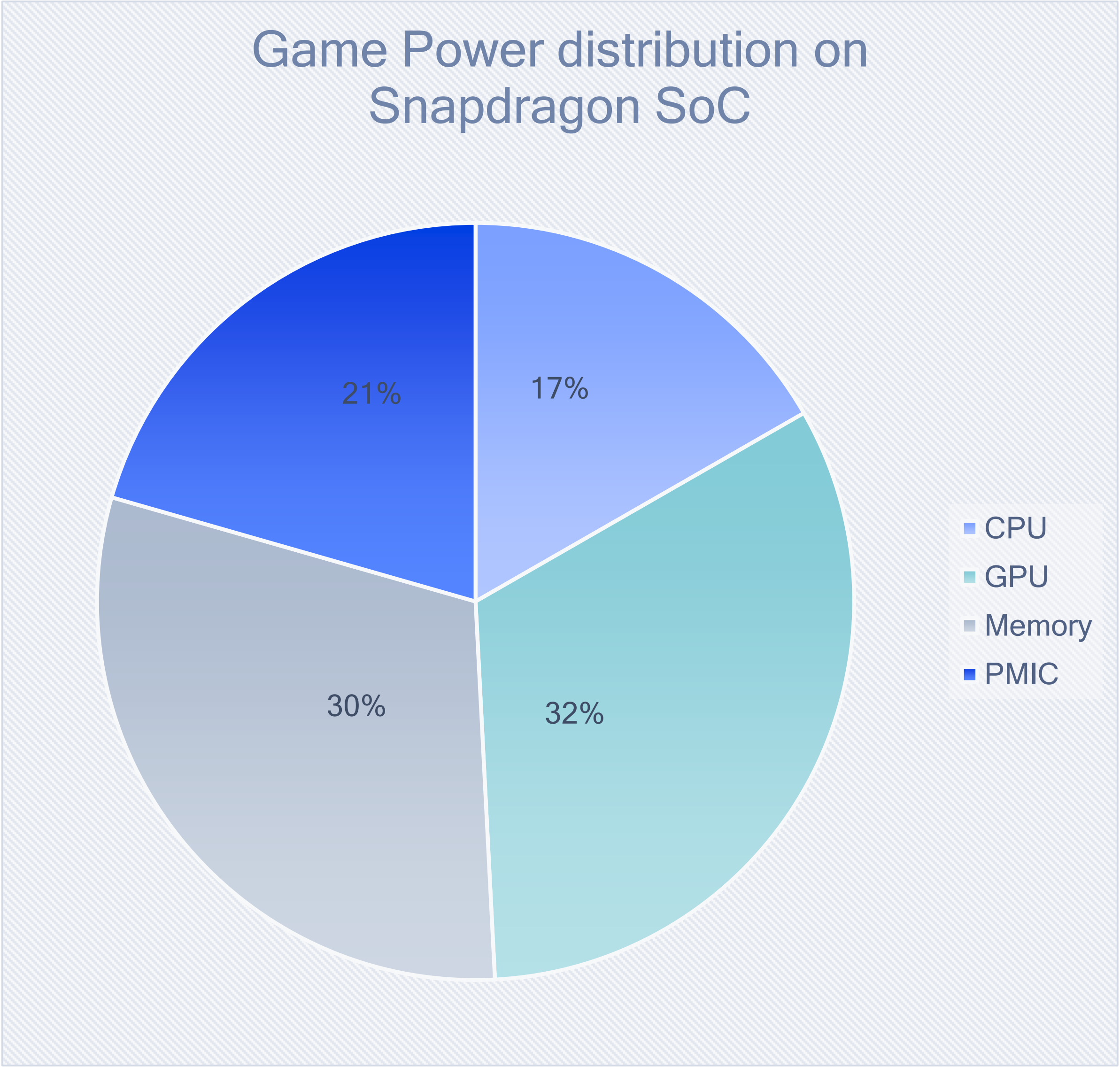
Power Consumption within a Mobile GPU

GPU and Memory power are about as dominant power consumers

- CPU power consumption is less, but still significant
- PMIC (Power Management IC [voltage regulator]) can also burn significant power

API chosen and driver maturity can have a large effect on overall power consumption

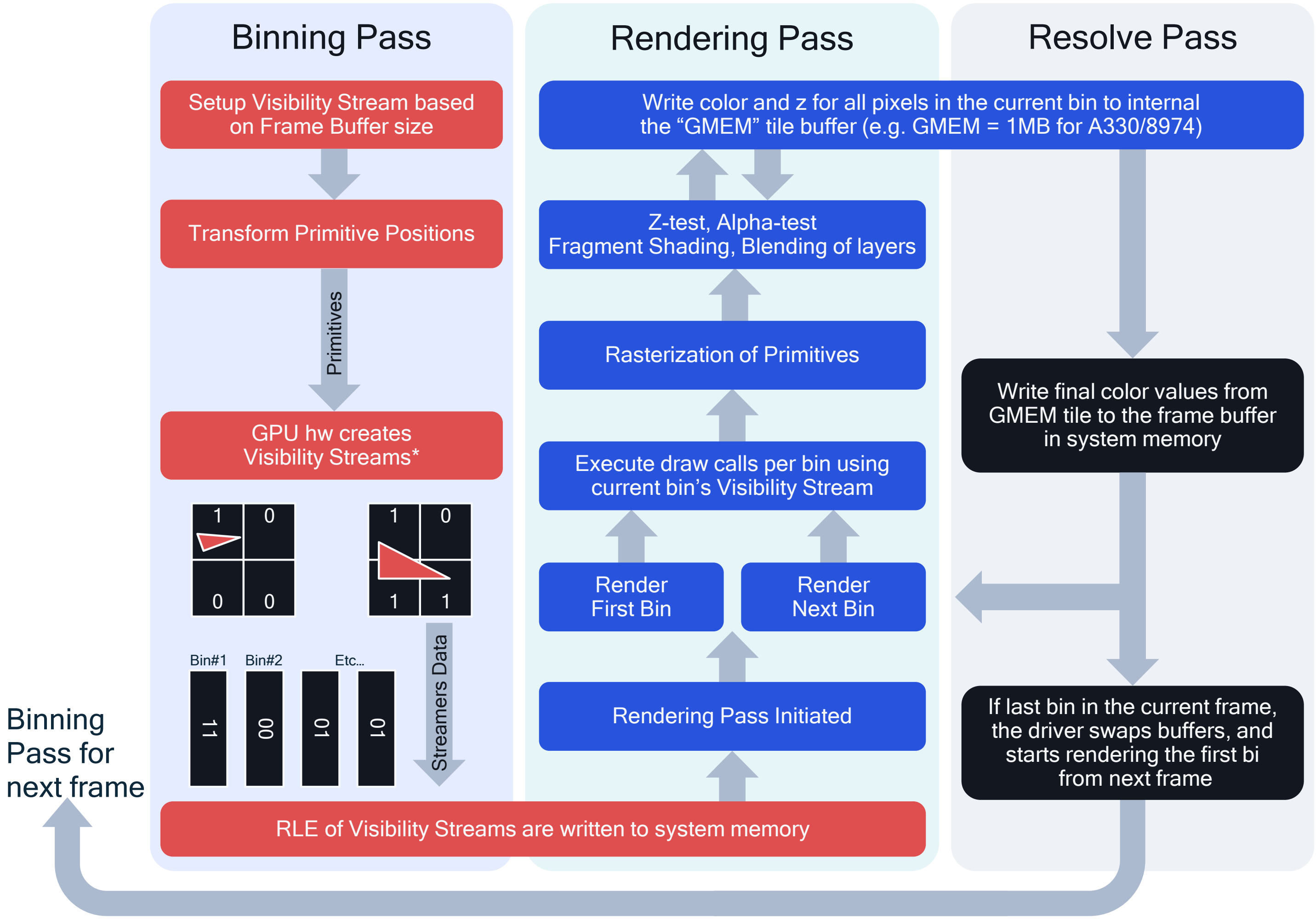
- Vulkan based application's power consumption is lower on the CPU than with OpenGL ES



Note: Silicon based measurement of Fortnite

Rendering algorithm differences with desktop GPUs

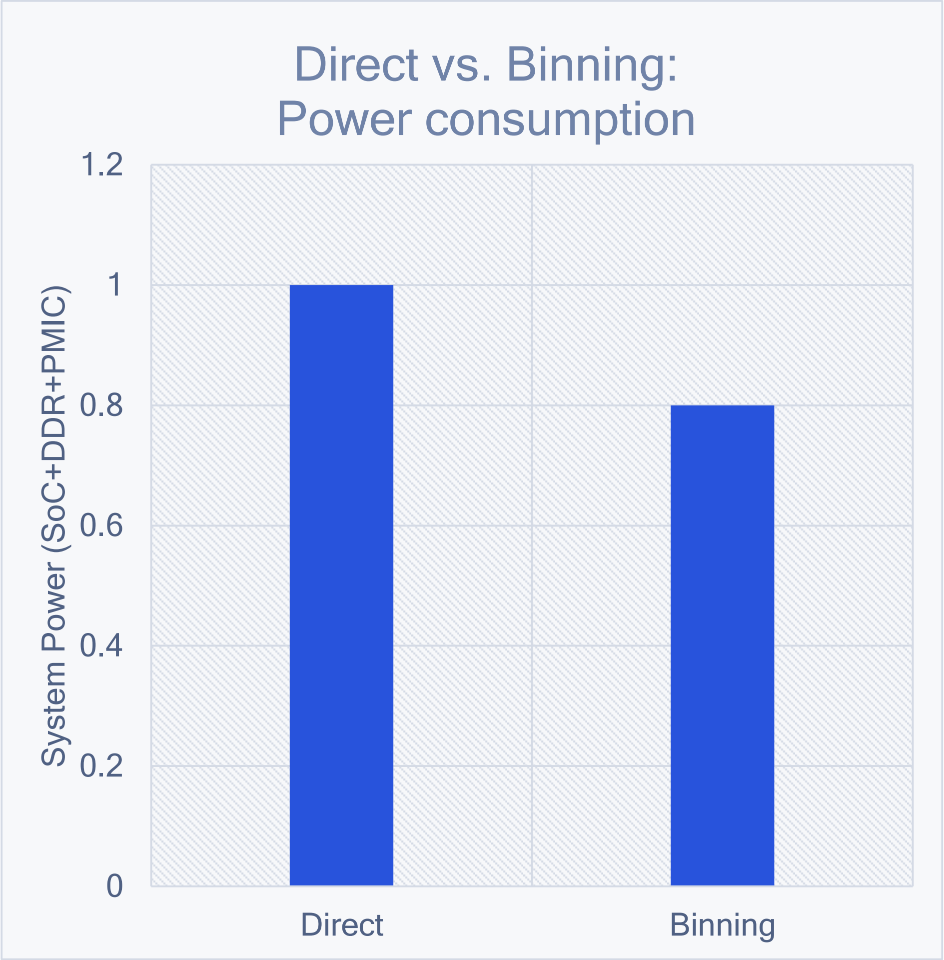
- Mobile chips typically use some form of 'Binned' rendering into an on-chip tile buffer
- There is commonly a separate 'Binning Pass' that generates visibility information that is used for later 'Rendering Pass'. Intel calls this pass 'POSH' – position only shading – as only position information is required.
- Finally the results are copied to the system memory surface in a 'resolve pass'. Note that surfaces that are no longer needed (such as depth) are not resolved and any traffic associated with them stays on-chip.
- Adreno™ GPU supports Direct Rendering as well (we call this 'Qualcomm® FlexRender™ Technology') for situations in which the depth complexity is low and not worth the binning and resolve overhead. In these cases, some of the tile buffer is used as system memory cache.



Qualcomm Adreno and FlexRender is a product of Qualcomm Technologies, Inc. and/or its subsidiaries

Power advantages of tiled rendering

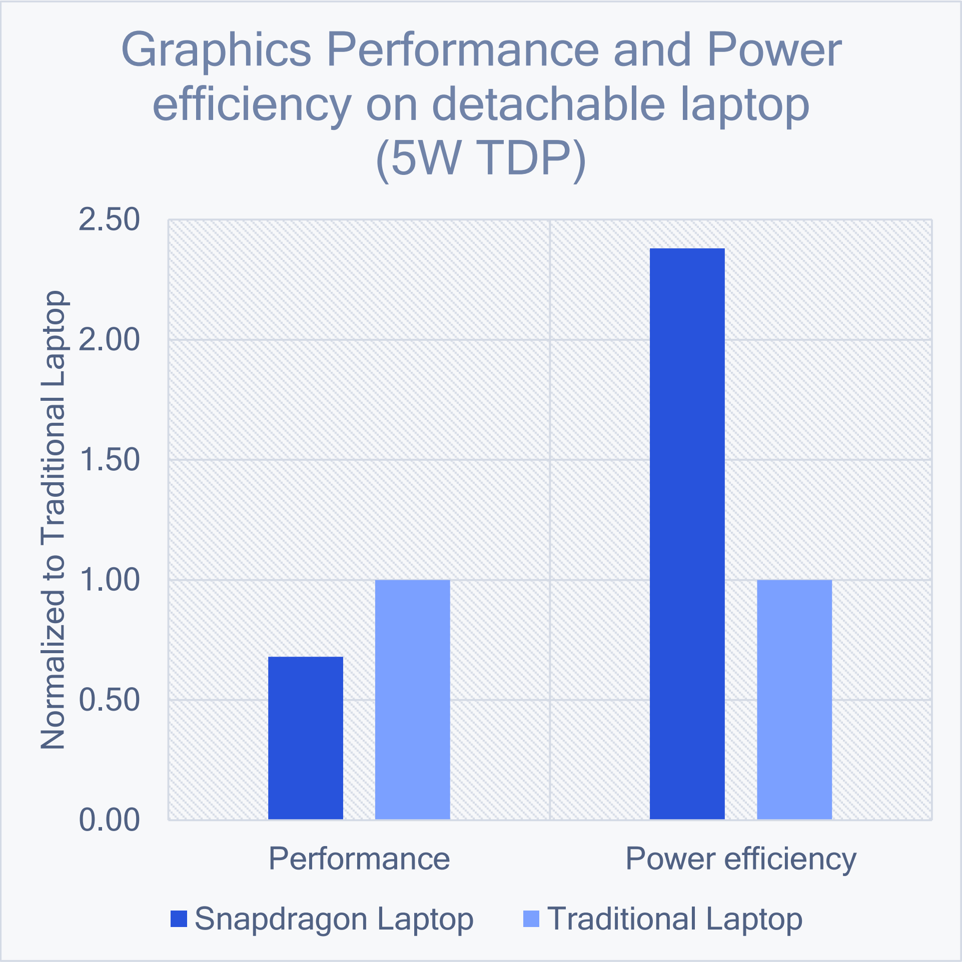
- The binning pass can generate an low-resolution Z buffer, which is then used in later passes. Similar to a depth pre pass, this provides a level of Hidden-Surface-Removal even for late occludes. This is typically kept in system semory as it is very low bandwidth
- Memory bandwidth is obviously saved during fragment shading as only a single write per pixel is done. For MSAA, any sample filtering is done purely on-chip.
- While vertex bandwidth and processing may appear to increase relative to direct rendering, this is not really so as:
  - Only the position information is required during the binning pass. The visibility information produced requires minimal bandwidth.
  - During the render passes – any back-facing or LRZ occlude vertex is not fetched. So in many cases, the non-position bandwidth associated with a vertex is totally saved.
  - Qualcomm, at least, uses fairly large tile buffers (512KB or more). Most primitives hit in only a single tile, limiting any vertex over fetch.
- 'Vulkan Subpass' usage can allow data to stay in the tile buffer for re-use on a later sub-pass.



Note:
1. Application - GFXBench Manhattan 3.0
2. Measured on Snapdragon silicon

Comparison of mobile vs traditional laptop GPUs

- Performance:
  - Traditional Laptop GPUs still show a significant performance over mobile parts in the laptop environment.
  - The delta is larger when comparing peak performance in traditional laptops with higher end SoCs and discrete graphics cards.
- Power efficiency (Perf/Watt), is just the opposite with mobile based parts showing >2x advantage.



Note:
1. Benchmark: DX11 based GFXbench Manhattan 3.0
2. Traditional laptop: HP Enxy X2 (Kaby Lake - core i5 7Y54)
3. Snapdragon laptop: HP Envy X2 (SDM835)

Physical design and power management approaches for mobile

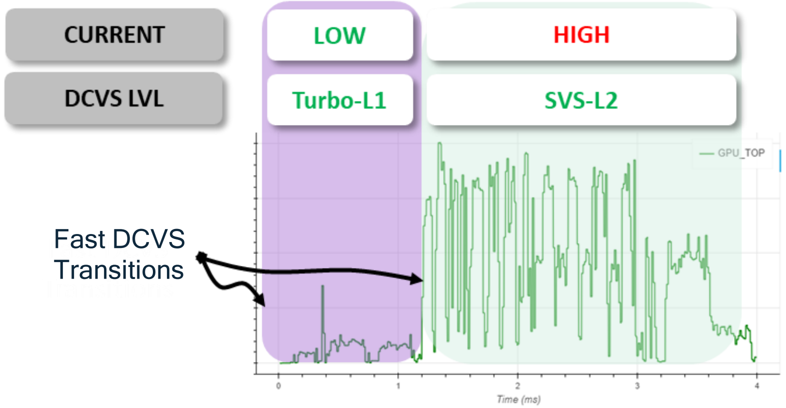
- Aggressive Dynamic Clock and Voltage Scaling (DCVS) managed by local power management processor
- Extensive clock tree gating with analysis tools to point out ungated clock trees.
- Dedicated data paths and bypasses for different instructions.
  - Despite a MUL-ADD pipeline, an A+B is not executed as 1.0*A+B
  - Often data values of 0 or 1 are detected to trigger a ‘bypass’ path to avoid lighting up a multiplier or adder.
- A key solution to achieve lower power consumption and yet fulfil the need for higher performance is by going ‘wide and slow’ with lower clocks and voltages
- Leakage is constrained by trading off frequency and avoiding low Vt device selection

Exploiting fast transitions

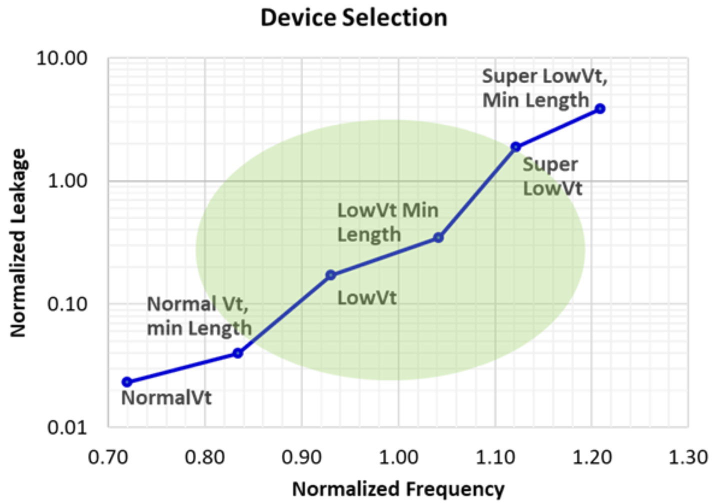
Exploiting fast DCVS transitions for sub-frame Clk/Voltage selection with workload awareness

DCVS during GPU binning phase

Low peak current region of frame run at higher FMAX



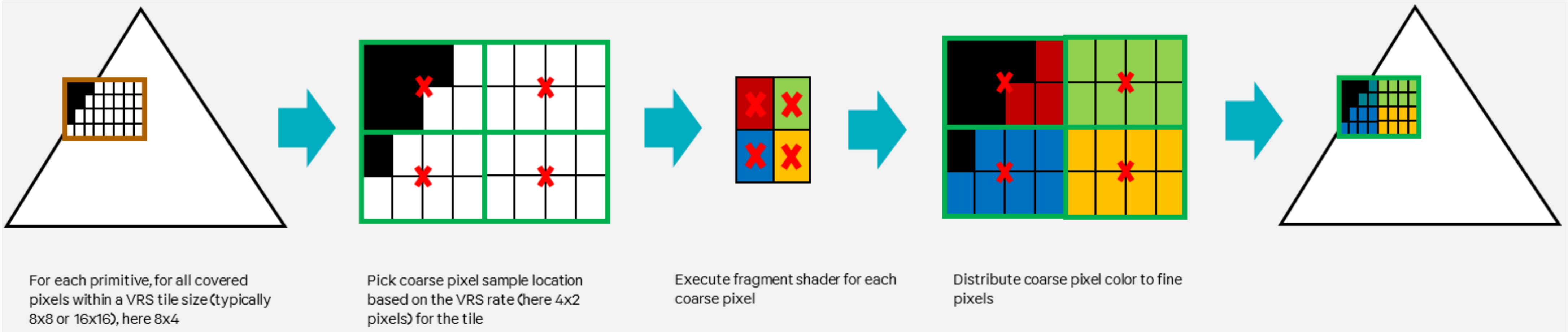
2-5% perf gain across use cases



Future Challenges

- While aggressive cooling technologies could help, the overall heat dissipation envelope needed for a handheld device is unlikely to change. Hence 6W to 8W is still the absolute mobile limit.
- Mobile GPUs are nearing the limitations of the 'wide and slow' approach as we hit the minimum voltage levels for a given process.
- Moore's Law is slowing down. Process improvements are minimal moving forward.

- Some possible solutions:
  - More Power Efficient Memory Systems. 'HBM' (High Bandwidth Memory) where memory and GPU are interconnected via silicon and thru-silicon-via.
  - New rendering techniques that inherently render less (such as VRS).
  - Continued Compression improvements`



Qualcomm

Thank you

Follow us on:

For more information, visit us at:

www.qualcomm.com & www.qualcomm.com/blog