A Low-Power Handheld GPU using Logarithmic Arithmetic and Triple DVFS Power Domains

Byeong-Gyu Nam, Jeabin Lee, Kwanho Kim, Seung Jin Lee, and Hoi-Jun Yoo

KAIST

Outline

- Backgrounds
- Proposed Handheld GPU
  - Low-Power Vertex Shader
  - Low-Power Rendering Engine
  - Triple-Domain DVFS
- Evaluation Results
- Conclusion

Motivations

- Handheld 3D Graphics Systems
  - Low-Power → long-battery lifetime
  - Small-Area → limited-footprint
  - High-Performance → more realistic scenes
- Previous Works
  - [Lindholm et al., SIGGRAPH01] Conventional Vertex Shader Architecture
  - [Kameyama et al., GH03] Low-cost, limited performance
  - [Sohn et al., GH04] Fixed-Point Vertex Shader
  - [Yu et al., ISSCC06] High-Performance Floating-Point Vertex Shader

Contributions

- Handheld GPU using Logarithmic Arithmetic
  - Power-, area-efficient, and high-throughput graphics pipeline
  - Vertex shader using multi-function unit
    - Merges matrix, vector, and elementary functions in a single-arithmetic 4-way unit
- Handheld GPU with Triple-domain DVFS
  - Separate control of each power domain according to its workload
  - Minimize power consumption for given workload

Logarithmic Arithmetic

- Various Math Operations for 3DG Pipeline
  - Matrix-vector multiplication
  - Vector operations
  - Elementary functions

MAT,
VMUL, VMAD, VDOT, VDIV, VDSQ,
POW, LOG, SIN, COS, ATAN

→Too complicated for resource limited handsets

- Logarithmic Number System (LNS)
  - Reduces arithmetic complexity
  - Conversion errors

- Handheld 3D Graphics
  - Small screen tolerates a little computation error

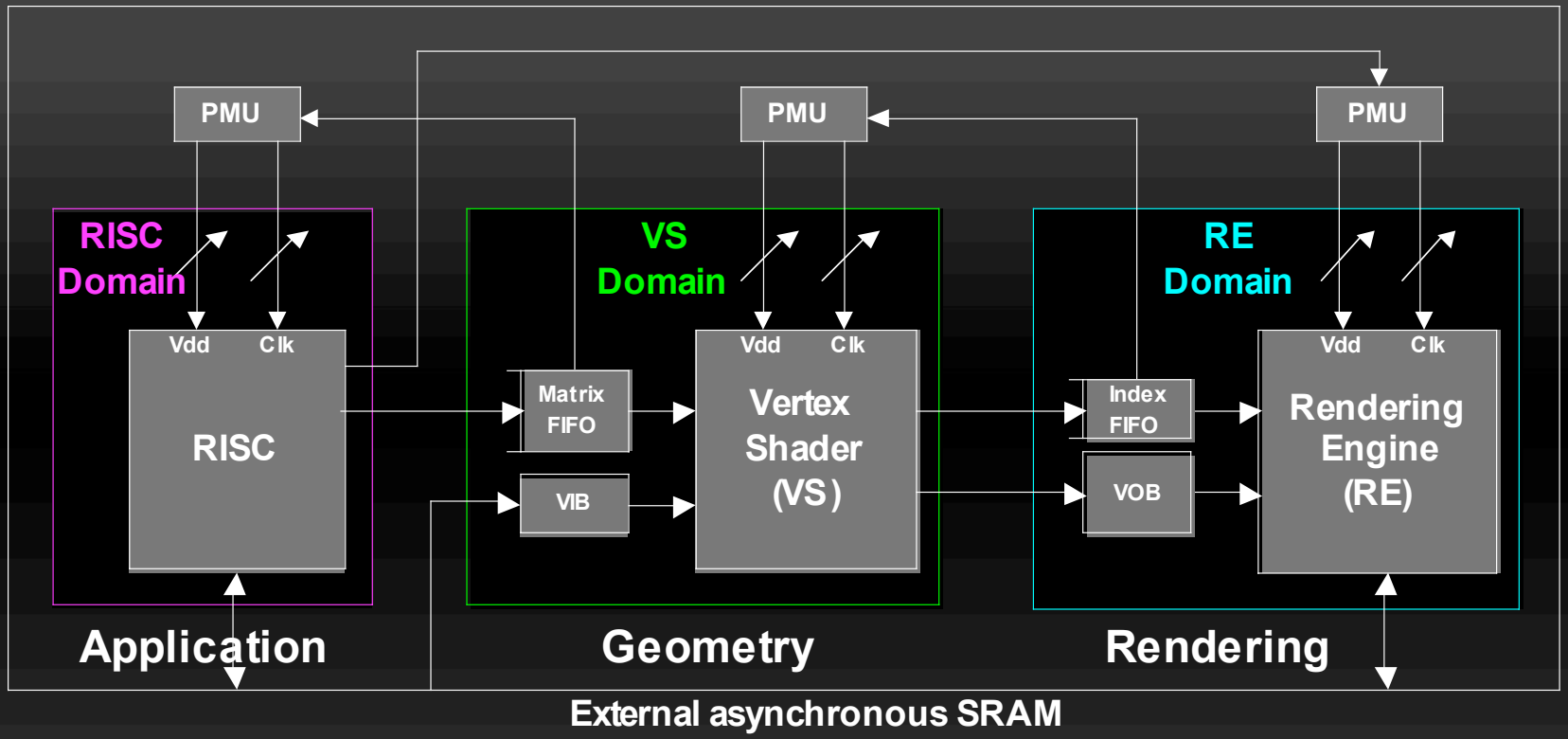
→ Logarithmic arithmetic can be used

Dynamic Voltage Frequency Scaling

- Power Consumption is Proportional to
  - Square of supply voltage
  - Operating frequency

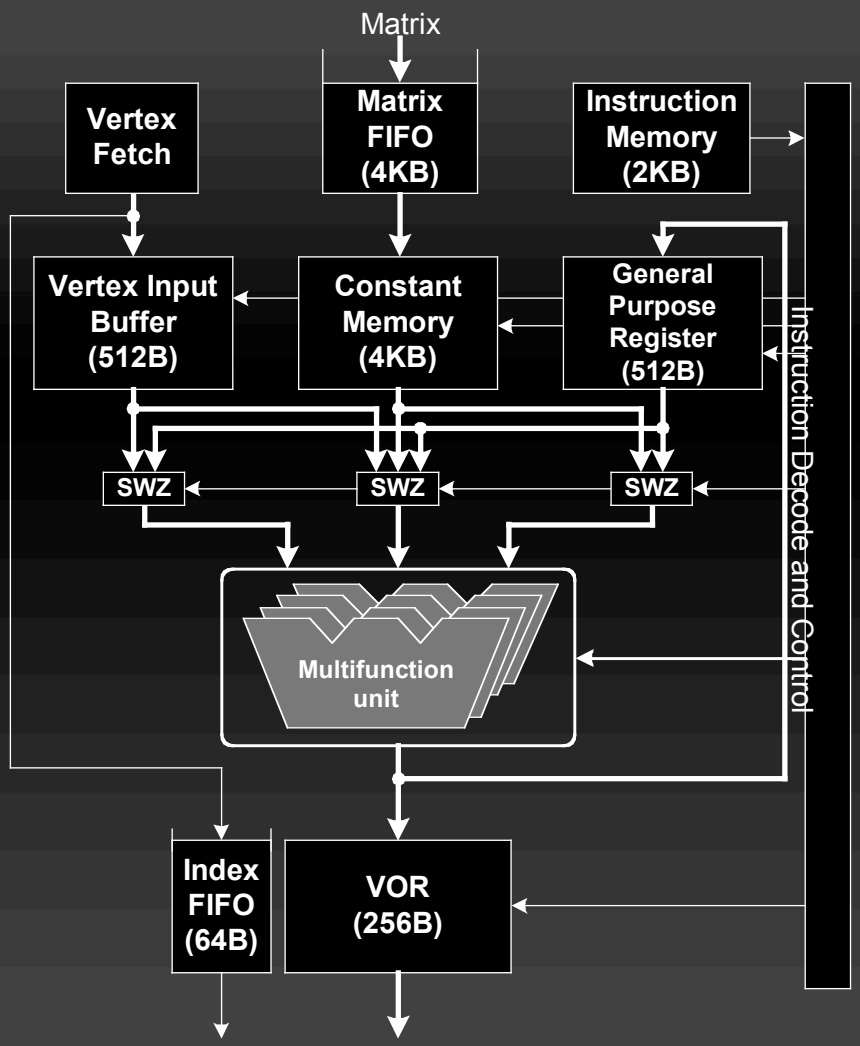
  → Dynamic scaling supply voltage and frequency is important for low-power design
- DVFS in module-wise manner
  - Can assign the optimal frequency and voltage to each module [Sheaffer et al. GH04]
    - cf) Chip-wise DVFS cannot exploit dynamic workload characteristic of each module

Overall Architecture



- Integrates RISC, VS, and RE
- Logarithmic arithmetic based pipeline
- Triple power domains with DVFS
- 3 PMUs for management and FIFO for communication

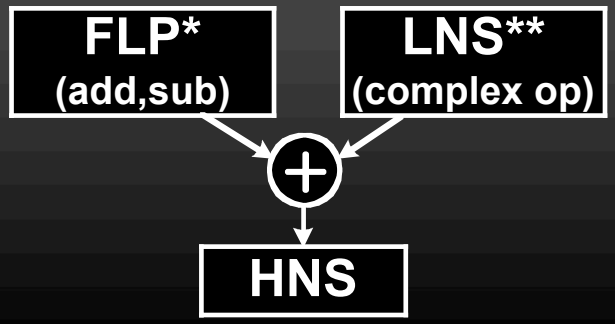
Vertex Shader



- Multi-function Unit
  - Unifies matrix, vector, and elementary ftn.
  - Single-cycle throughput for vector and elem. ftn.
- 2-cycle Transformation
  - Matrix-vector multiplication using LNS
  - 4-cycle in conventional way
- Operand Forwarding
  - Log-domain forwarding

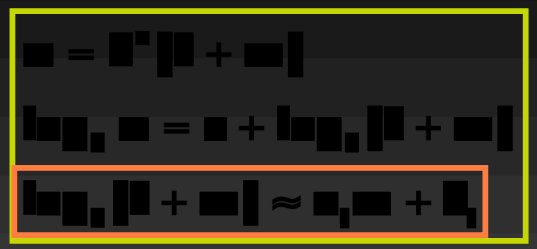
Number System

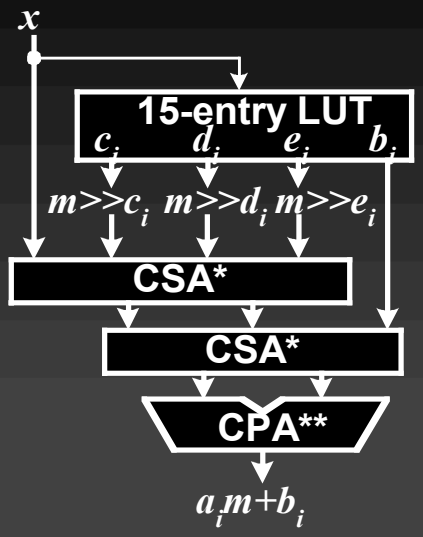
- Hybrid Number System (HNS)
  - Complex op. in LNS
  - Linear add, sub in FLP
  - Number Converters based on piecewise linear approximation



* FLP: Floating-Point Number System
** LNS: Logarithmic Number System

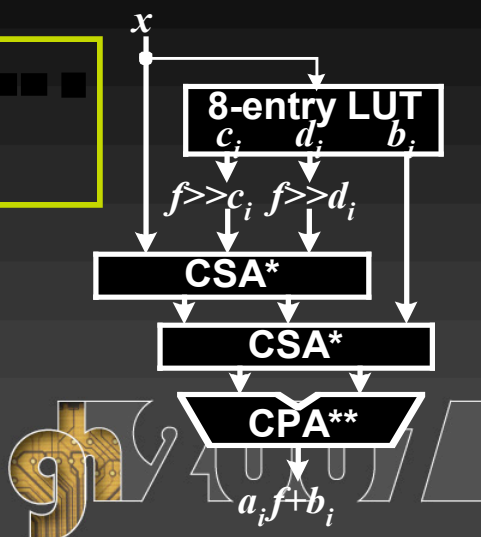
- Logarithmic converter





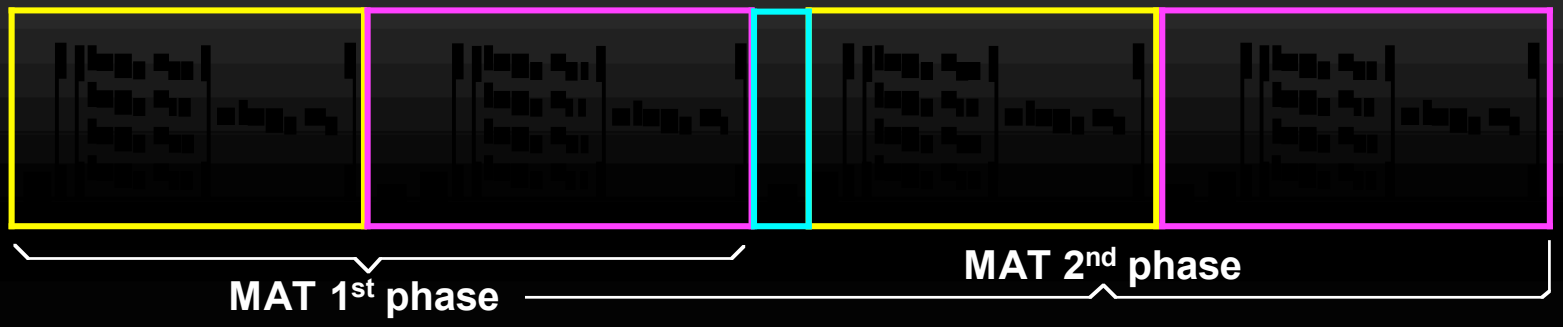
* CSA: Carry Save Adder
** CPA: Carry Propagate Adder

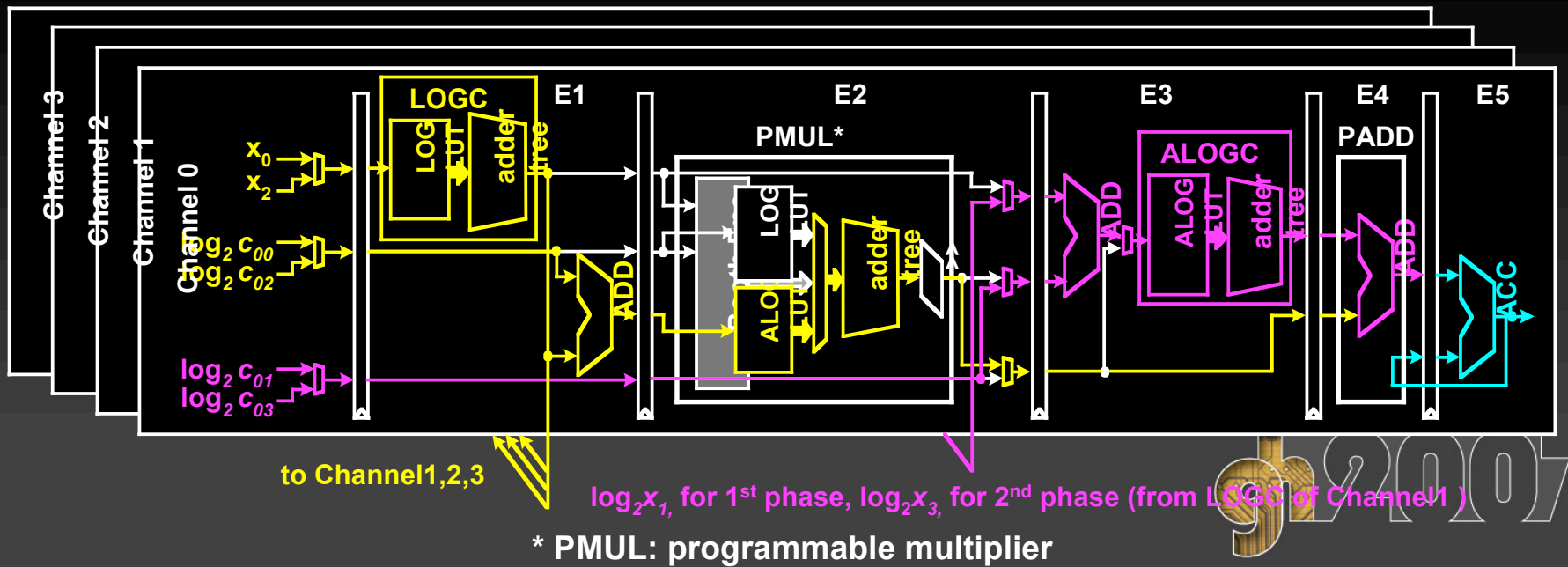
- Antilogarithmic Converter



Matrix-Vector Multiplication

- MAT requires 16 multiplications
- MAT in HNS requires 20 LOGCs, 16 ADDs, 16 ALOGCs
- Fixed matrix coeffis. for an object → 4 LOGCs, 16 ADDs, 16 ALOGCs
- 2-phase implementation requires 2 LOGCs, 8 ADDs, 8 ALOGCs / phase

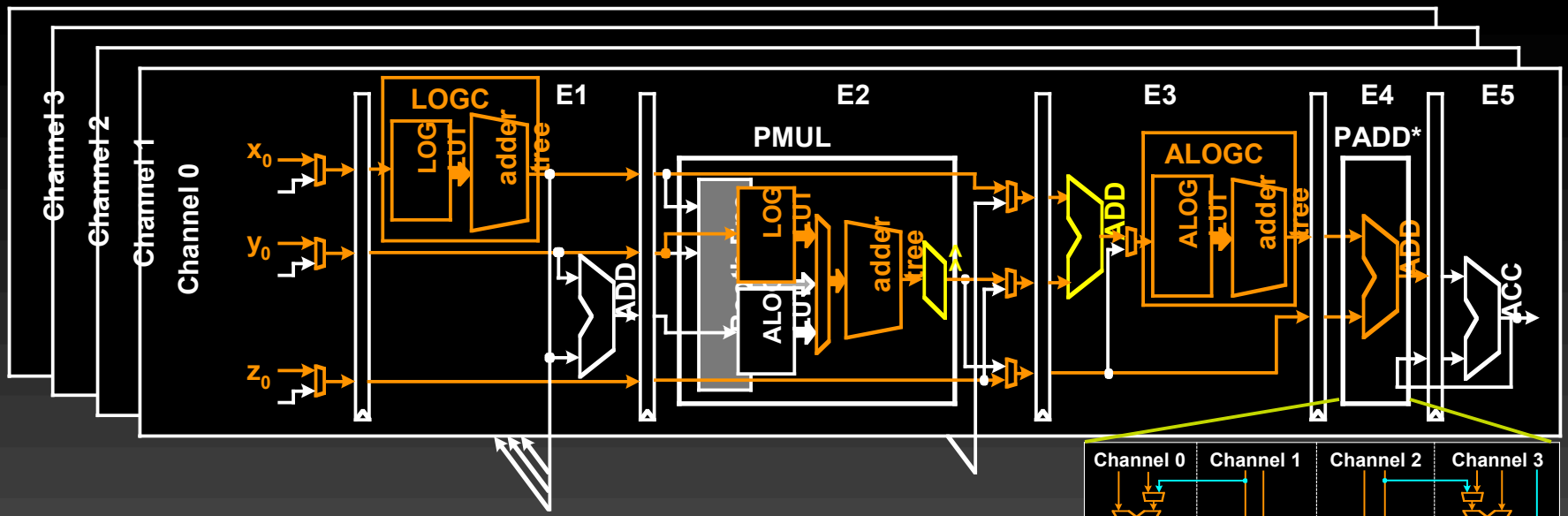


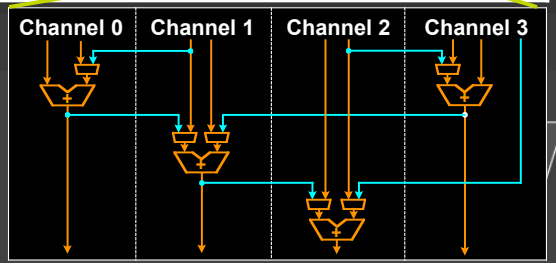


* PMUL: programmable multiplier

Vector Operations

- Defined as a Generic Operation
  - Vector MUL,DIV,DSQ,MAD
- Converted into HNS
  - 2 LOGCs are required / Channel

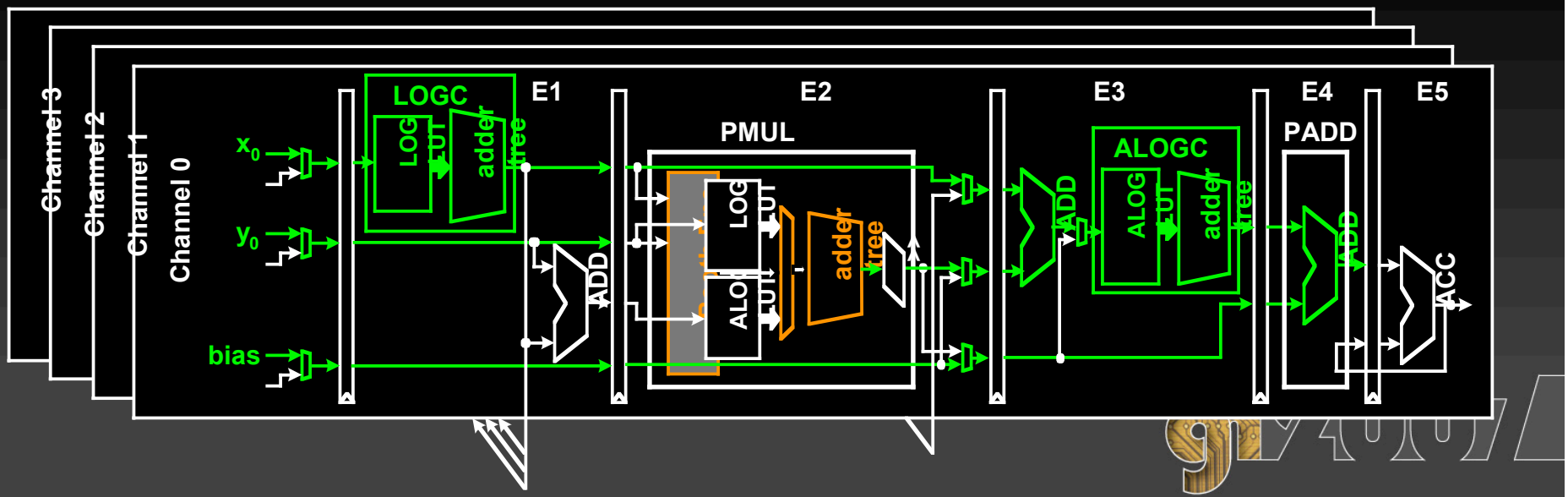




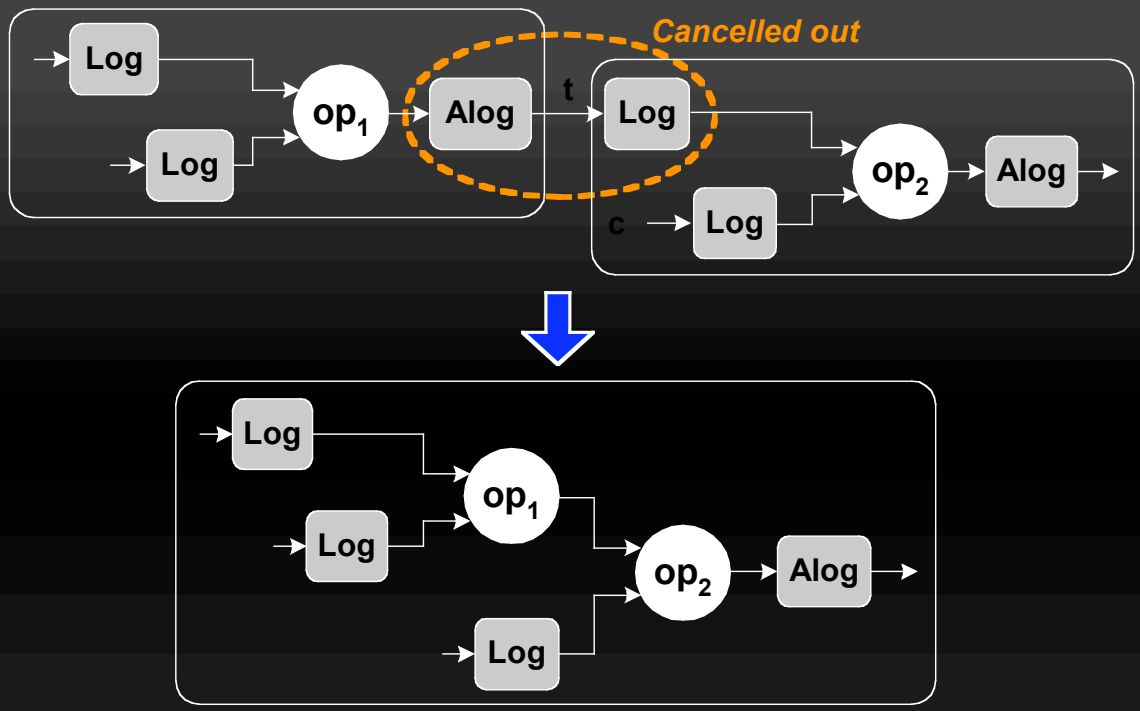
* PADD: programmable adder

Elementary Functions

- Requires Power Ftns. Evaluations
  - POW, Taylor Series for Trigonometric (SIN,COS,ATAN...)
- Converted into HNS
  - Booth multiplier is required

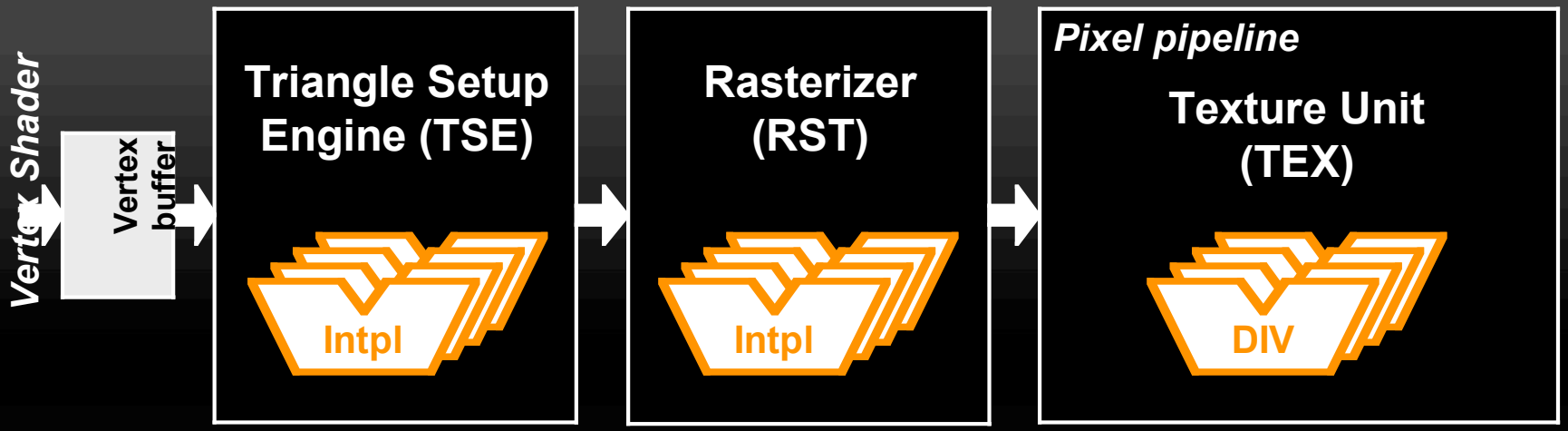


Operand Forwarding in Log-domain

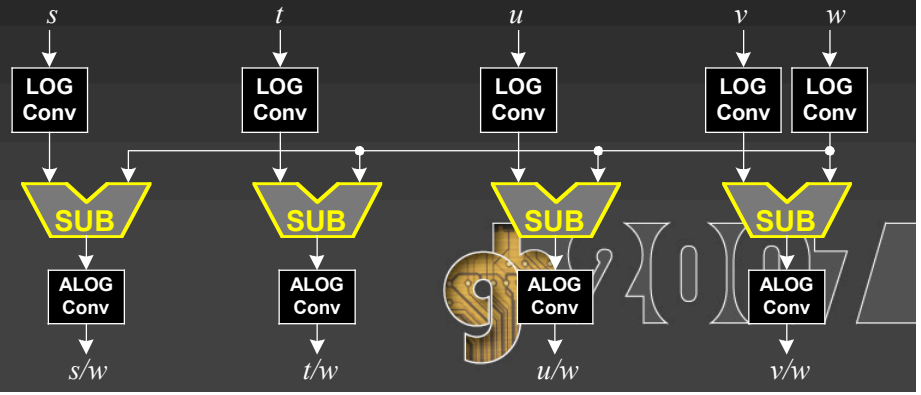


- Avoids back-to-back antilog and log converters
  - Forward intermediate log value to next instruction
  - Pipeline latency reduction
  - Computation error reduction

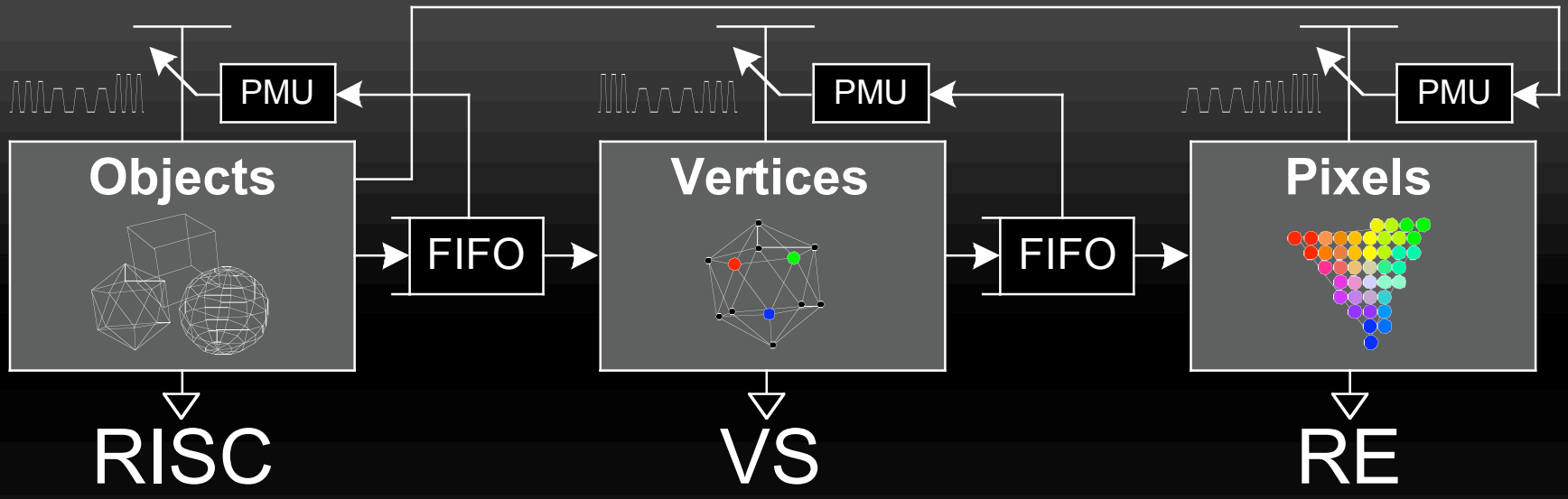
Rendering Engine



- Divisions required in RE are implemented as subtractors
  - SIMD Interpolators in TSE and RST
  - SIMD Divider in Texture Unit



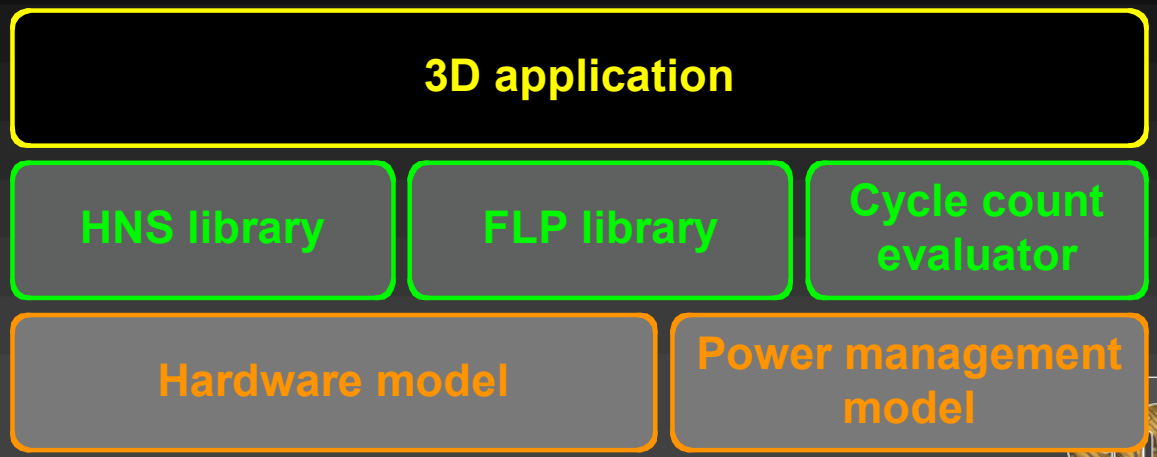
Triple-Domain DVFS



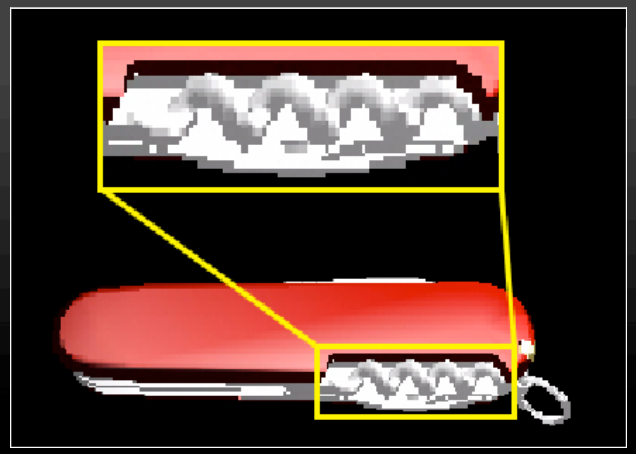
- Workloads for each module in 3D pipeline are different
  - Objects for RISC, vertices for VS, pixels for RE
- Separate power-domains with DVFS
- Managed separately according to its workload

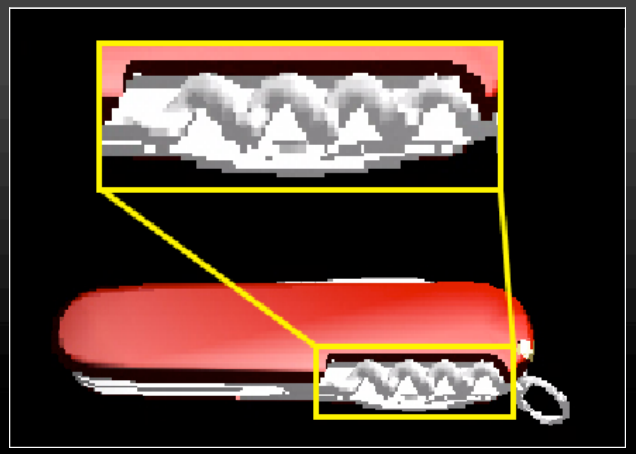
Performance Evaluations

- Environment Setup
  - Accuracy simulation
    - C libraries for HNS & FLP ops.
  - Performance measurement
    - Cycle count evaluator
  - Hardware model for graphics processor core
    - Fully synthesizable structural HDL model



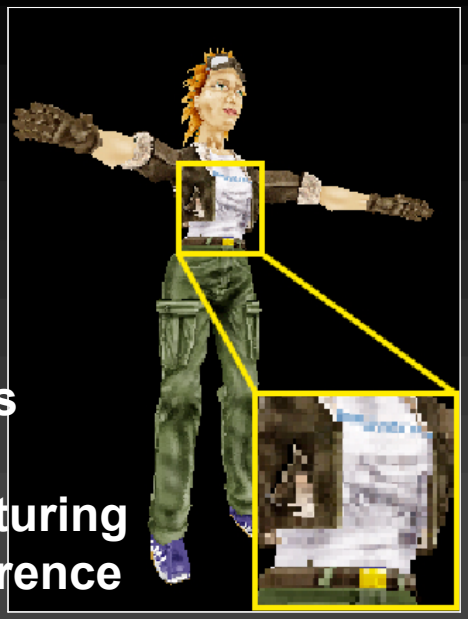
Accuracy Comparison

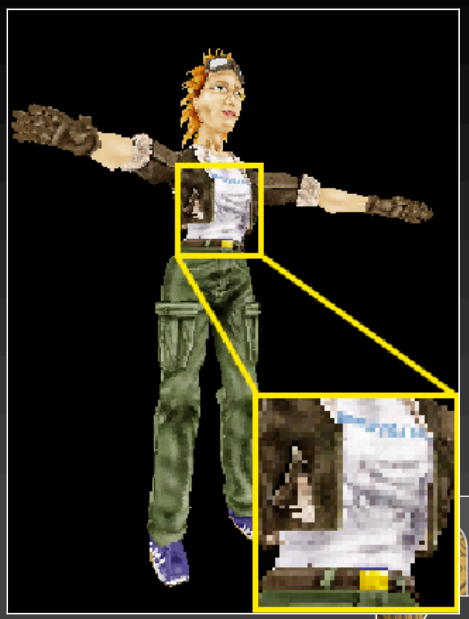




from FLP

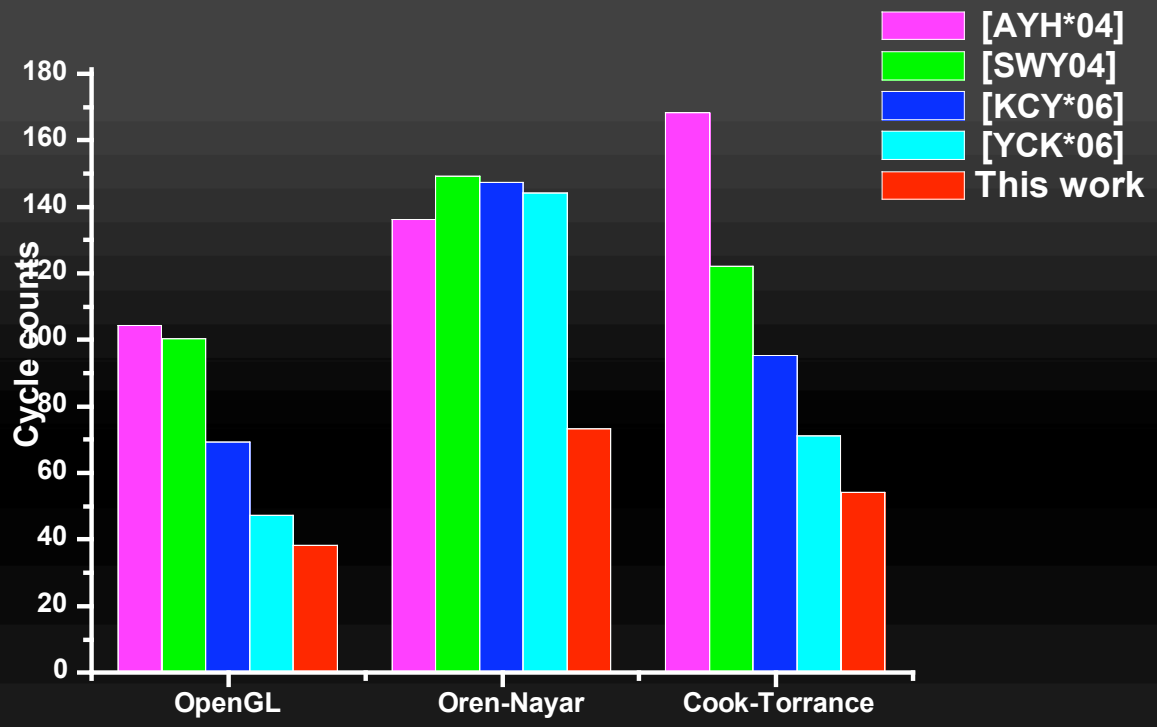
from HNS





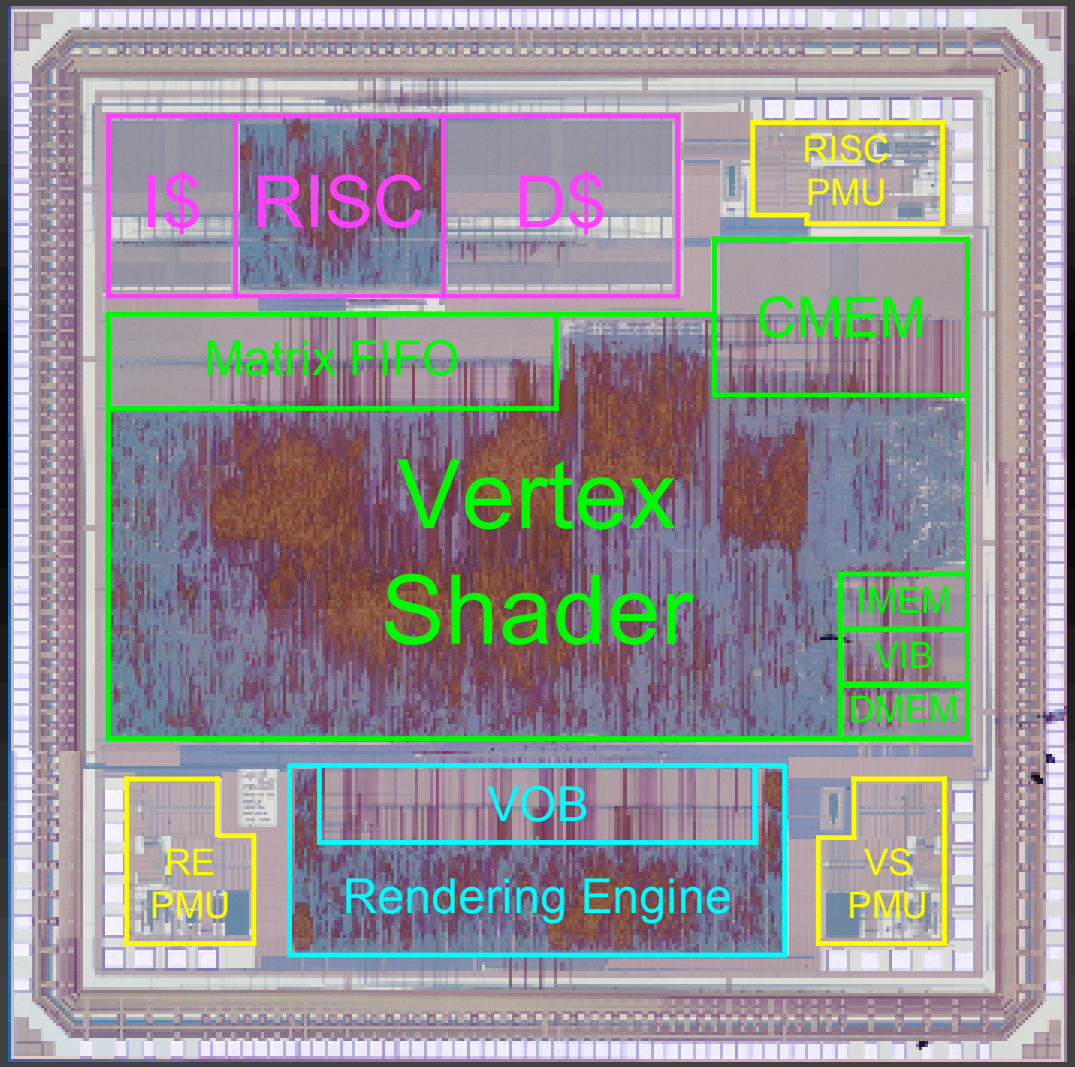
- FLP and HNS libraries
- QVGA Test Screens
- Tr&Lighting , Tr&Texturing
- Tolerable image difference

Performance Comparison



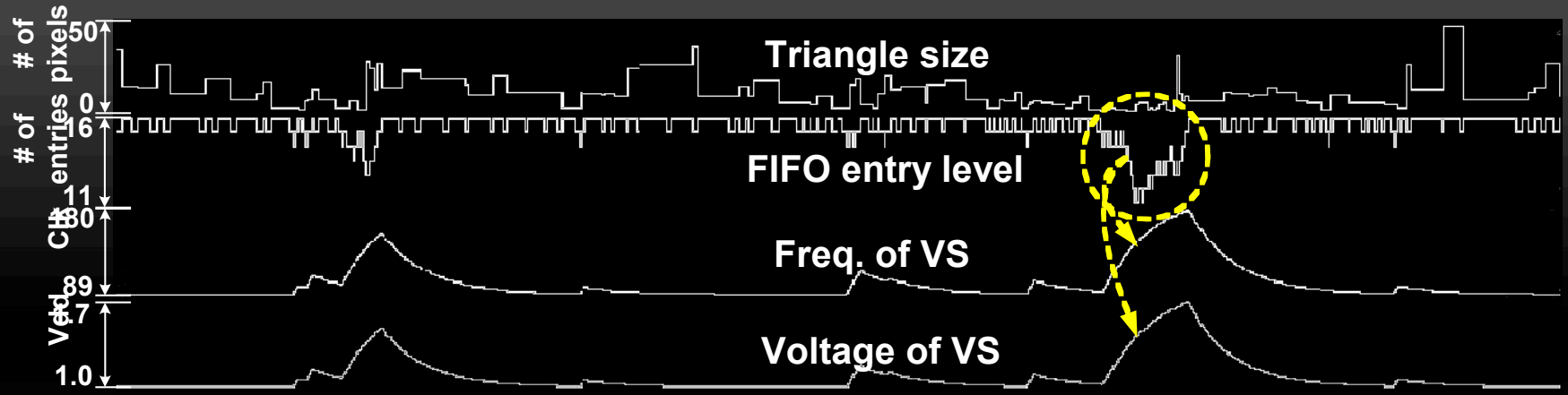
- Test vectors
  - Standard OpenGL TnL
  - O-N and C-T for advanced diffuse and specular lighting
- Comparison results
  - 19.1%, 49.3%, and 23.9% cycle count reduction for OpenGL, O-N, and C-T models from latest work [YCK*06]

Implementation



- Process Technology
  - TSMC 0.18um CMOS 1P 6M
- Area
  - 17.2mm^2
  - 393K gates, 29KB SRAM
- Processing Speed
  - 5.26 Mvertices/s
  - 50Mpixels/s
- Power Consumption
  - 52.4mW @ 60fps
  - 153mW @ full speed
- PMU Specifications
  - 1.0 – 1.8V, 89 – 200MHz
  - 0.45mm^2, 5mW

Power Consumption



- DVFS for VS according to the workload
  - Higher workload makes a droop in FIFO entry level
  - FIFO droop increases Freq and Vdd of VS
  - Increased VS speed restores FIFO entry level
- For test model with 5,878 triangles, 5 DC, and OpenGL TnL
  - GPU consumes 52.4mW @ 60fps

Handheld GPUs

	Performance (Mvertices/s)	mW @ full speed	Area (Kgates)	Functions	Process / Freq.	Kvertices/s/ MHz
[KKF*03]	0.185	38	360	GE+RE	0.18um / 30MHz	6.17
[SWY04]	7.2	115	230	GE	0.13um / 400MHz	18
[YCK*06]	2.13	157	375	GE	0.18um / 100MHz	21.3
This work	5.26	153 (87 for GE)	393 (242 for GE)	GE+RE	0.18um / 200MHz	26.3

- Compared with the latest work [YCK*06]
  - 2.47x performance improvement
  - 50.5% power reduction
  - 38.5% area reduction
  - 23.5% Kvertices/MHz improvement

Conclusion

- A Handheld GPU for Low-Power Wireless Applications
  - 2.47x performance improvement
  - 50.5% power reduction
  - 38.5% area reduction
- 3D Graphics Pipeline based on Logarithmic Arithmetic
  - 5.26Mvertices/s for OpenGL TnL
- Triple power domains with DVFS
  - 52.4mW @ 60fps

Thank You

- For more information
  - e-mail: byeonggyu.nam@gmail.com