



# Low-Power Parallel Computing on GPUs



Ben Juurlink



Technische Universität Berlin



## Critical Questions We Seek to Ask

---

- Power consumption has become the critical limiting factor in performance of processors (both CPUs and GPUs)
- GPUs are becoming the vanguard of parallel programming, delivering increasingly greater performance and programmability
- But the critical issue for power consumption is about bandwidth and hierarchical memory architectures, about which we have very little reliable information
- Questions we seek to obtain answers to:
  - How do we compare the huge range of memory architecture choices?
  - What are the bandwidth requirements for performance-critical software on hierarchical memory architectures?
  - How can we optimize software for new memory architectures?
  - What tools do we need to bring performance-critical software onto GPUs?

- To answer these questions we have brought together a group of complementary groups
- To analyse the software on different architectures, we have:
  - A commercial tools provider: Codeplay  **codeplay**<sup>®</sup>
  - And an academic tools and architecture research group at TU Berlin 

- To produce GPU designs and memory architectures, we have:
  - Think Silicon: a GPU architecture designer 
  - And an academic architecture research group at Uppsala 

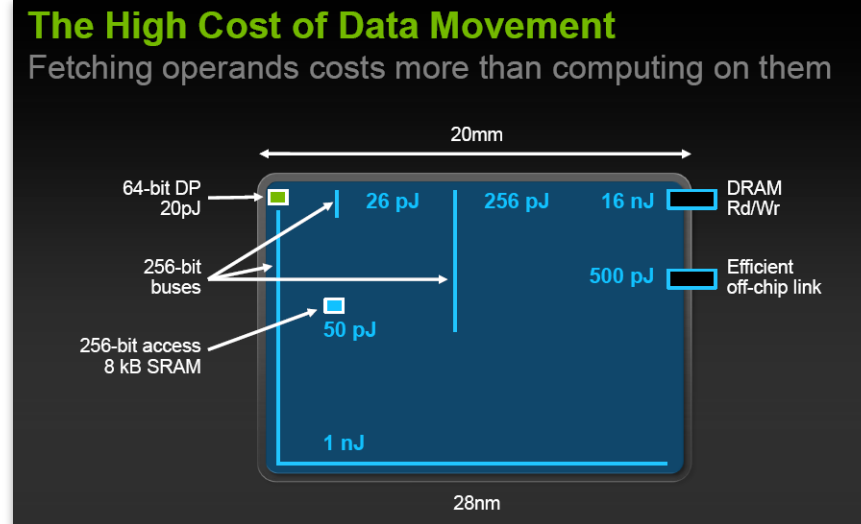
- To produce relevant benchmark software, we have:
  - Geomerics: a producer of new real-time lighting software for games 
  - AiGameDev.com: a company that researches and teaches about commercial game AI techniques 

## Project Objectives

---

- To develop applications for and port applications to massively parallel, low-power GPUs
  - lighting, game AI, video coding
- To develop a set of tools that will allow analyzing and reducing power consumption
- To propose and evaluate architectural enhancements that enable the efficient execution of applications that contain a lot of conditionally executed code
  - To evaluate the trade-off of SIMD versus MIMD
- To propose and evaluate architectural techniques to reduce the power consumption of GPUs
- To develop a hardware demonstrator for the most promising architecture techniques

# Power: Where is it being used?



## Energy Cost by Operation Type

Operation	Approximate energy consumed today
64-bit multiply-add	200 pJ
Read 64 bits from cache	800 pJ
Move 64 bits across chip	2000 pJ
Execute an instruction	7500 pJ
Read 64 bits from DRAM	12000 pJ

Notice that 12000 pJ @ 3 GHz = 36 watts!  
SiCortex's solution: drop the memory speed, but the performance dropped proportionately.  
Larger caches actually reduce power consumption.

John Gustafson, HPC User Forum, Seattle, September 2010

From Bill Dally's presentation at SC10

To deal with power, we need to control how far data has to move, right down to tiny distances on a chip. Even different kinds of registers have massively different power consumptions

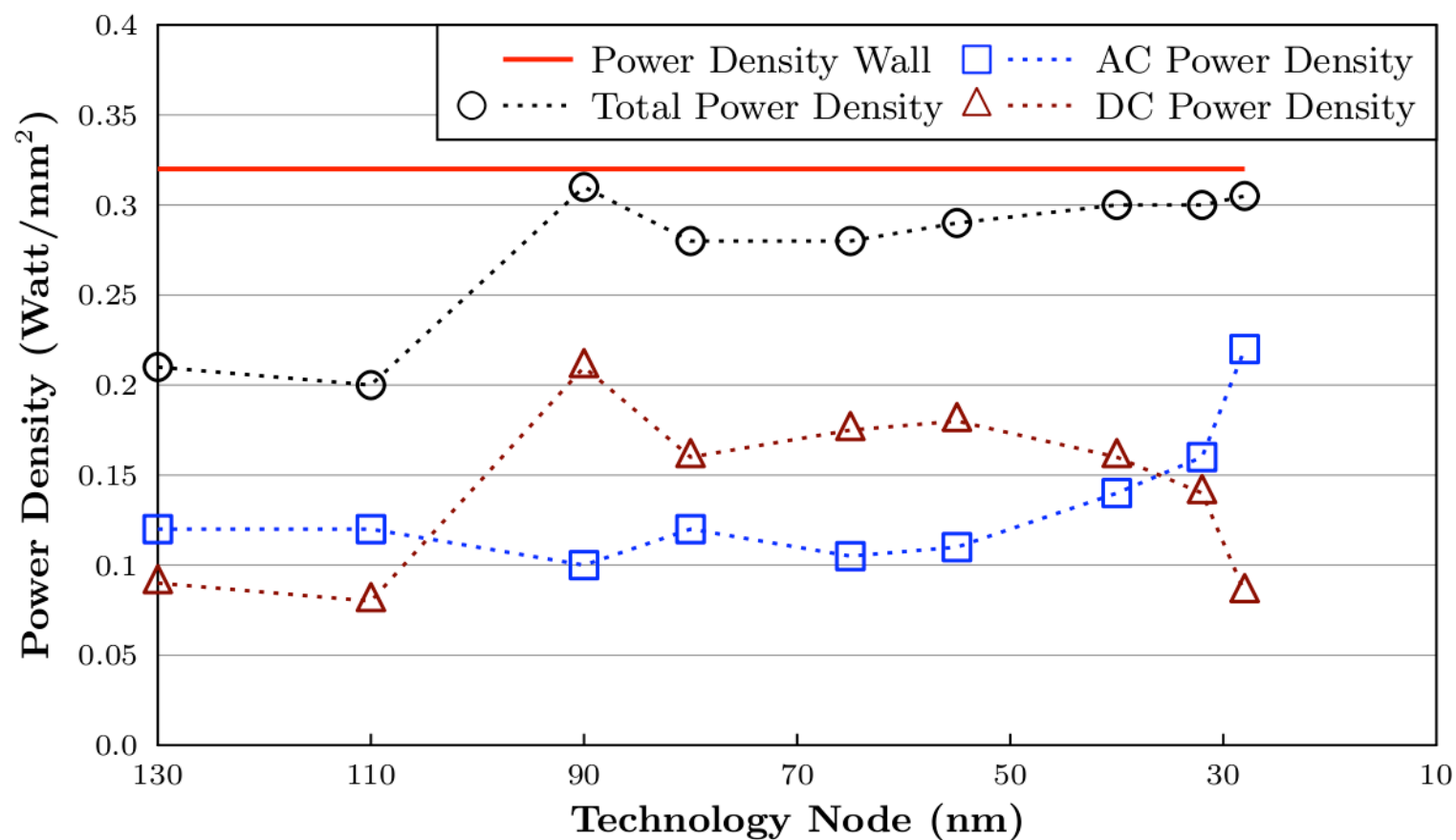
We want to measure and investigate this

**Scaling makes locality even more important**

	2010	2017	
		Processor (GPU) Scaling Targets	
		10 nm	
Process technology	40 nm	High Frequency	Low Voltage
Vdd (nominal)	0.9 V	0.75 V	0.65 V
Frequency Target	1.6 GHz	2.5 GHz	2 GHz
DFMA energy	50 pJ	8.7 pJ	6.5 pJ
64-bit read from 8KB SRAM	14 pJ	2.4 pJ	1.8 pJ
Wire energy (per transition)	240 fJ/bit/mm	150 fJ/bit/mm	115 fJ/bit/mm
Wire energy (256 bits, 10mm)	310 pJ	200 pJ	150 pJ
		DRAM Scaling Targets	
DRAM process technology	45 nm	16 nm	
DRAM Interface Pin Bandwidth	4 Gbps	50 Gbps	
DRAM Interface Energy	20-30 pJ/bit	2 pJ/bit	
DRAM access energy [10]	8-15 pJ/bit	2.5 pJ/bit	

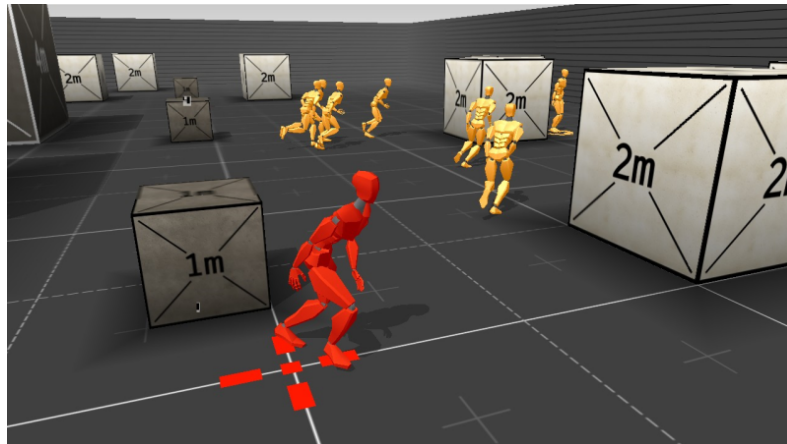
Table 1: Technology and circuit projections.

# GPU Power Density



original figure due to John Y. Chen, NVIDIA

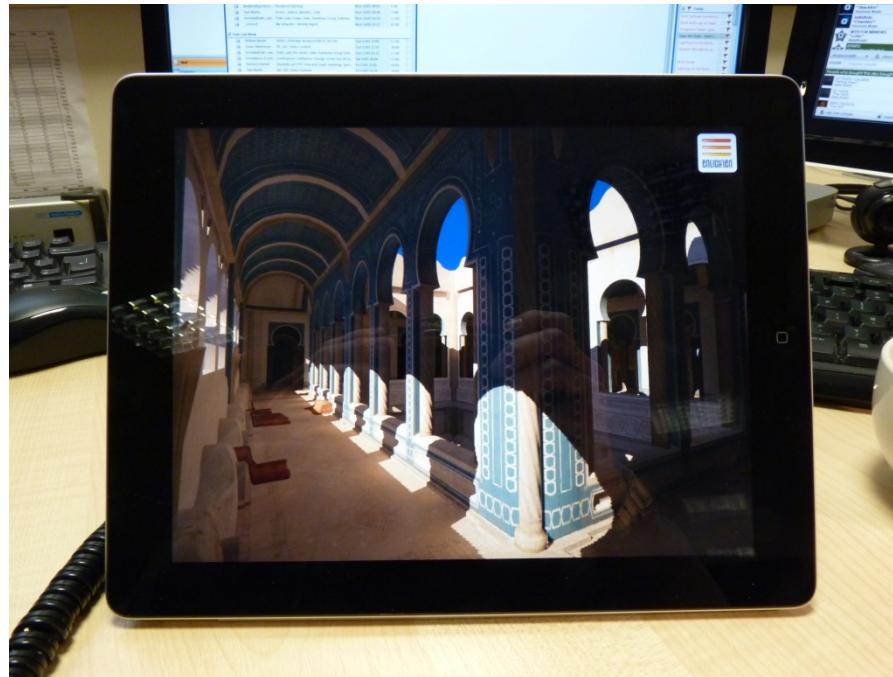
- SIMD GPUs most suited for **data-parallel workloads**
- But many important application domains (e.g. advanced lighting, game AI) are **control-intensive**
- According to game developers increased GPU performance is not leading to improvements in visual quality because the way GPUs render the graphics fundamentally restricts their flexibility
- Need to investigate new graphics techniques and how they impact GPU design





## Applications: Graphics

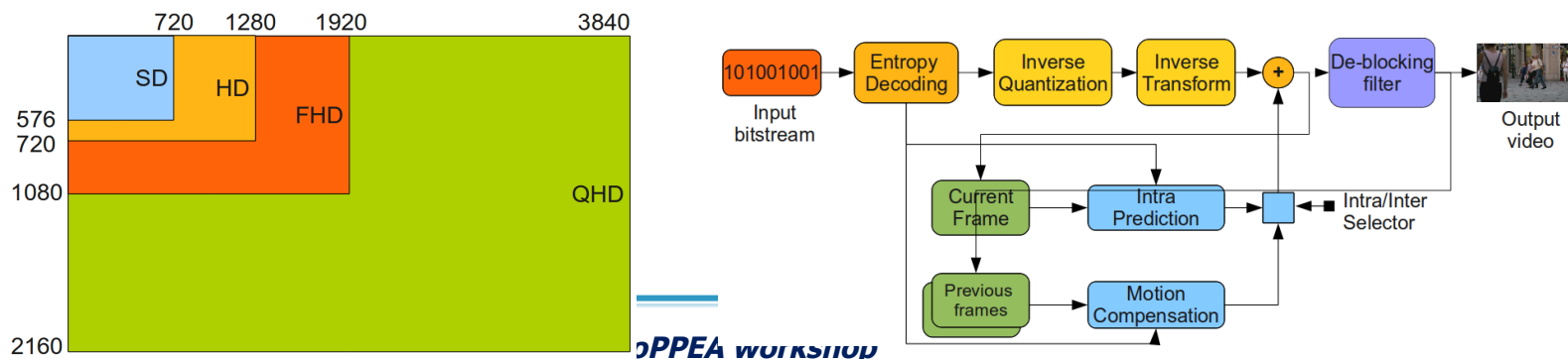
- Port Enlighten real time radiosity to mobile (in progress)
- Mobile graphics radically different to desktop
  - PowerVR architecture – tile-based deferred renderer in hardware
- Investigate new software techniques for mobile graphics





# Applications: Video Codecs

- Video coding applications require more computing power with each generation (e.g.: FHD (1920x1080) → QHD (3840x2160))
- No direct match between video requirements and GPU capabilities:
  - Entropy decoding: Bit-level dependencies → not appropriate for GPU
  - Inverse Transform (IDCT): frame-level parallelism, regular data accesses
  - Motion Compensation (MC): frame-level parallelism, non-regular data accesses, branch divergence due to multiple interpolation modes.
  - Intra-Prediction: wavefront parallelism, branch divergence
  - Deblocking Filter: wavefront parallelism, divergence due to pixel adaptation
- Current work: H.264/AVC IDCT on GPU
- Next steps: High Efficiency Video Coding (HEVC) on GPUs



- GPU applications consist of several *kernels*
- If data set larger than on-chip memory, data must be streamed in and off-chip
- Off-chip memory accesses consume two orders of magnitude more energy than on-chip memory accesses
- Goal is to develop a tool that *fuses* kernels such that kernels are iteratively applied to data subset that can be kept on-chip



- Instrument Codeplay's PS3/GPU Offload C++ compiler
  - Monitor accesses to global versus local data
  - Apply the concepts to unsupported architectures
  - Visualise bandwidth and power consumption of real-world code from AIGameDev and Geomerics
- Apply the tool to the Geomerics Enlighten codebase
  - Accelerate the reference implementation on PS3 and GPU
  - Apply to ThinkSilicon GPU hardware designs
- Modify existing OpenCL tools for power consumption estimates

- To improve GPU power efficiency, we will explore several directions
  - Different memory architectures – GPUs are designed with a variety of hierarchical memory architectures to reduce bandwidth
  - Redundancy – redundant computations and data movement can be omitted by transforming computation into caching
  - Slack - slack originating from unbalanced processing in each graphics pipeline stage is major source for power-inefficiency. Can exploit this slack by applying DVFS to underutilized pipeline stages
  - Accuracy (QoS) - Reducing computational accuracy may not have a significant impact on QoS but at the same time save considerable energy

- We will produce:
  - Prototypes of commercially licensable tools to analyse memory architecture options
  - New commercial graphics techniques for power-efficient lighting
  - Research results showing impact on power of various options available in designing GPUs
  - New ideas for designing more power efficient GPUs
  - Training materials and examples to show how to take complex video game code (such as AI code) and move them onto GPU-accelerated architectures
- By working together we will achieve more than we can achieve individually!

- Bit too early to tell ...
- Research will show where additional research is needed
- DARPA study identifies four challenges for ExaScale Computing
  - Energy and Power challenge
  - Memory and Storage challenge
  - Concurrency and Locality challenge
  - Resiliency challenge