AMD

GPGPU, UNLOCKED COMPUTE POWER

TZACHI COHEN

2/9/14

AGENDA

GPU evolution

HSA

CodeXL

GPU evolution

GPU EVOLUTION

AMD

1ST ERA: Fixed Function

3D Geometry Transformation

$$V_{eye}\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = MVP \begin{bmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{bmatrix} \bullet V_{obj} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$V_{tex}\begin{bmatrix} s \\ t \\ r \\ q \end{bmatrix} = M_{proj} \bullet V_{in}$$

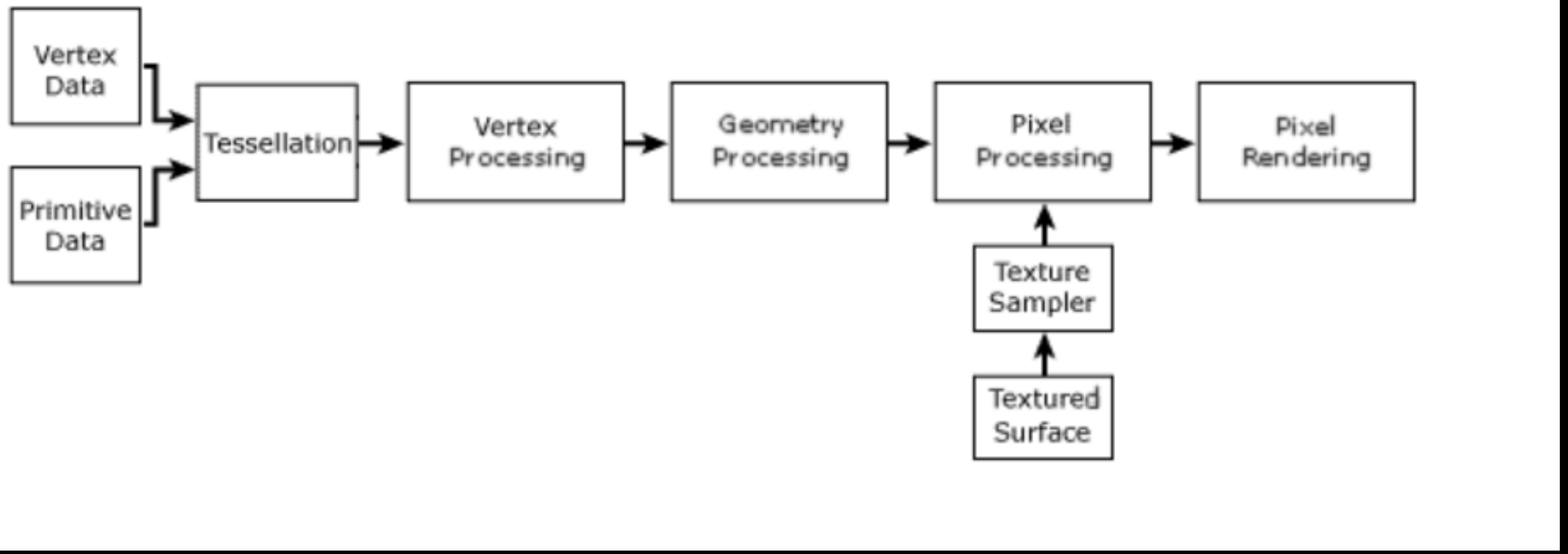
Lighting

$$C_p = k_a L_a + \sum_{n-lights} Att_n (k_d (\hat{L}_n \bullet \hat{N}) + k_s (\hat{R}_n \bullet \hat{V})^{\alpha})$$

Prior to 2002

- Graphics-specific hardware
  - Texture mapping/filtering
    - Multi-texturing
  - Transform & Lighting (T&L) Engines
    - Geometry processing
    - Rasterization
    - Fixed function lighting equations
  - Dedicated texture and pixel caches
- Dot product and scalar multiply-add
  - Sufficient for basic graphics tasks
  - No general purpose compute capability

DX9 3D PIPELINE

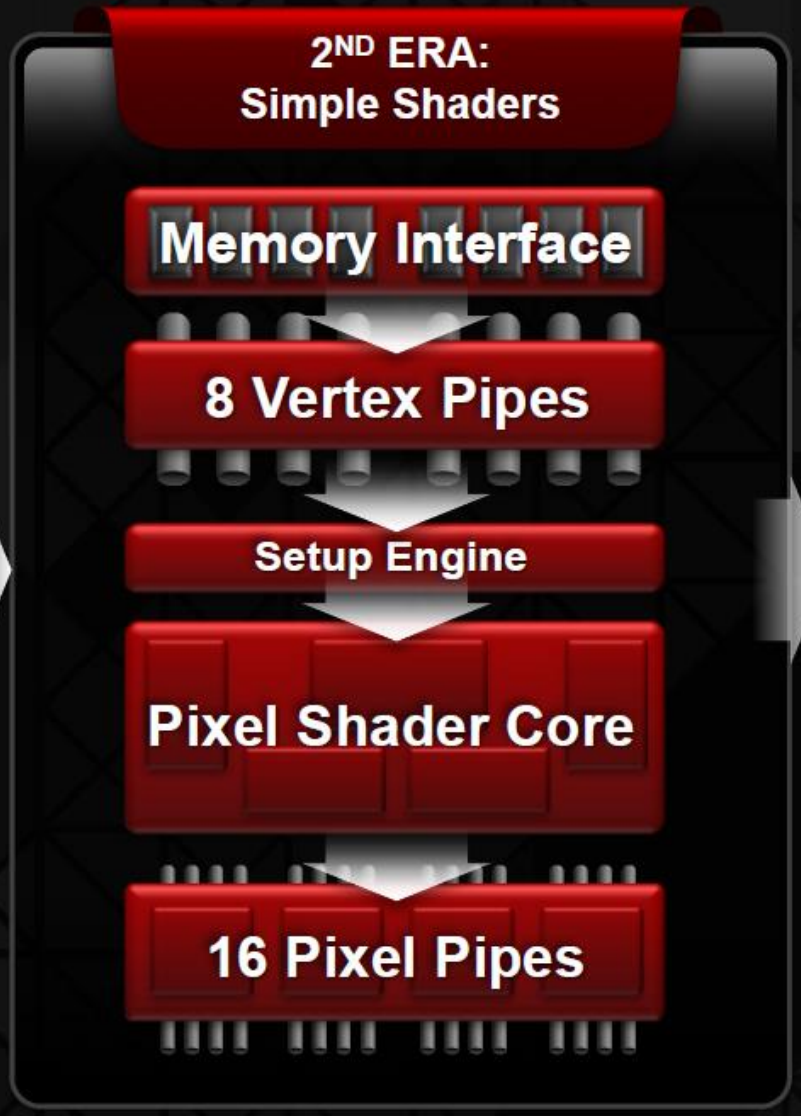


GPU EVOLUTION

AMD

2002-2006

- **Graphics Programmability**
  - **Direct3D 8/9, OpenGL 2.0**
  - Floating point processing
    - **IEEE** *not required*
    - Different precision per IHV
      - **ATI 24-bit** full-speed
      - **NV 16-bit** full-speed
      - **NV 32-bit** half-speed
  - Specialized shader units for vertex & pixel processing
- Added dedicated caches



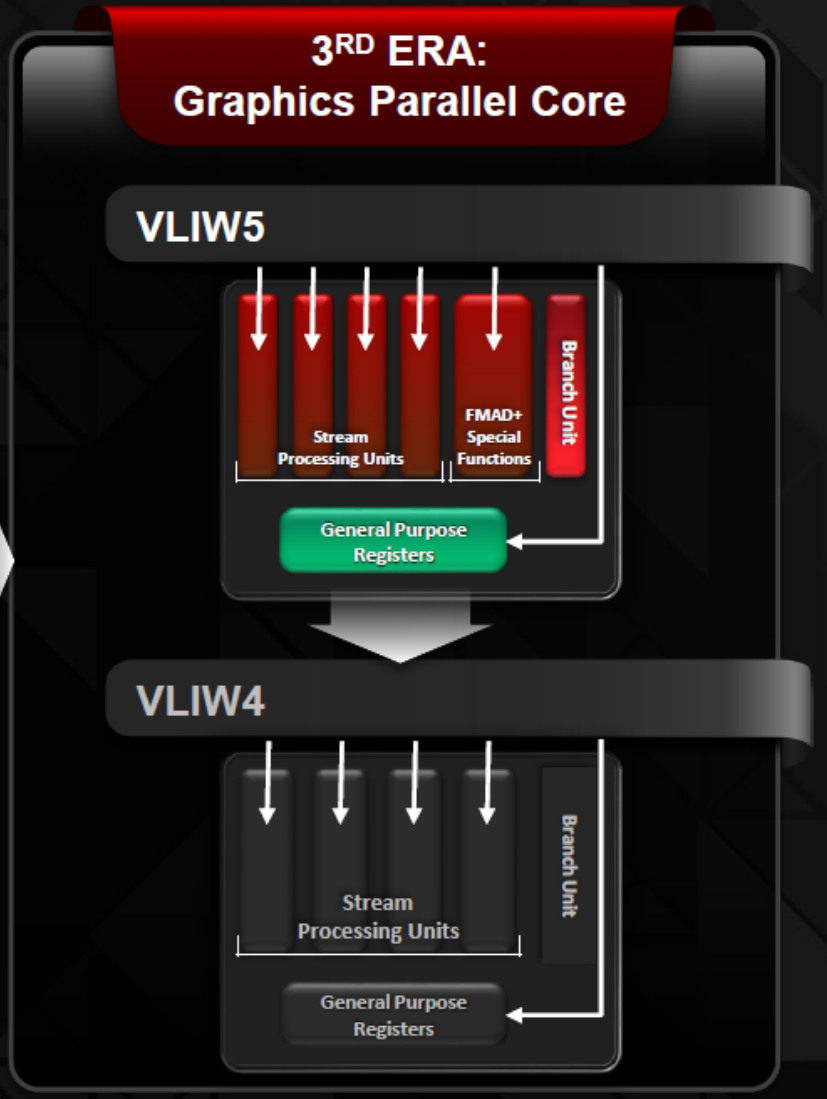
The Rise of Shaders

- **Shader Models 1.0 - 2.0**
  - **VS** and **PS** are *distinct*
    - ✓ Minimal Instruction Sets
    - ✓ Limited Instruction Slots
    - ✓ Limited Shader Lengths
    - ✗ No Dynamic Flow Control
    - ✗ No Looping Constructs
    - ✗ No Vertex Texture Fetch
    - ✗ No Bitwise Operators
    - ✗ No Native Integer ALU
  - […]

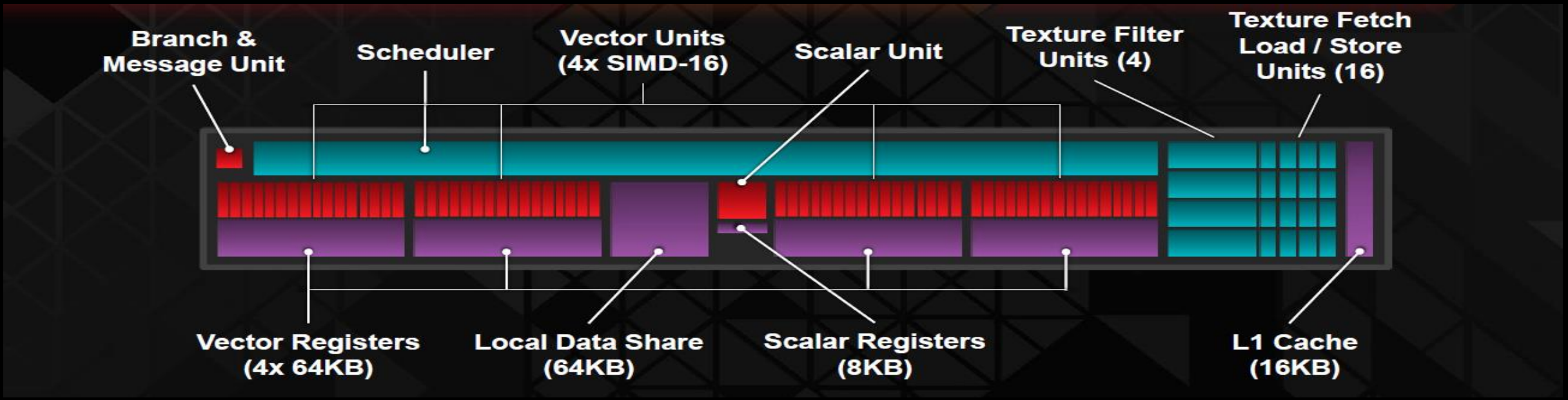
AMD

The Rise of The Unified Shader (VLIW-5)

- 5-Element Very-Long-Instruction-Word (XYZWT)
  - Began with XENOS and utilized from R600 until "Cayman"
  - Flexible and optimized for **Graphics** workloads
    - Ideal for 4-element Vector and 4x4 Matrix Operations
      - Vector/Vector math in a single instruction
    - Plus One Transcendental-Unit function per Instruction
- More advanced caching
  - Instruction, constant, multi-level texture/data, & later: LDS/GDS
- Single Precision 32-bit IEEE-Compliant Floating Point ALUs
- More flexible: Unified ALU, Branch Unit, Dynamic Flow Control, Vertex Texture, Geometry Shader, Tessellation Engines, etc.



• COMPUTE UNIT STRUCTURE



- 4 SIMDs , each SIMD has 16 compute elements. Total of 64 compute elements per CU.
- Each 64 threads within SIMD execute in lock-step fashion.
- On ‘if /else’ clause the GPU may take both ‘if’ and ‘else’ clauses if there is at least one divergent thread.

THE GPU

HAWAII

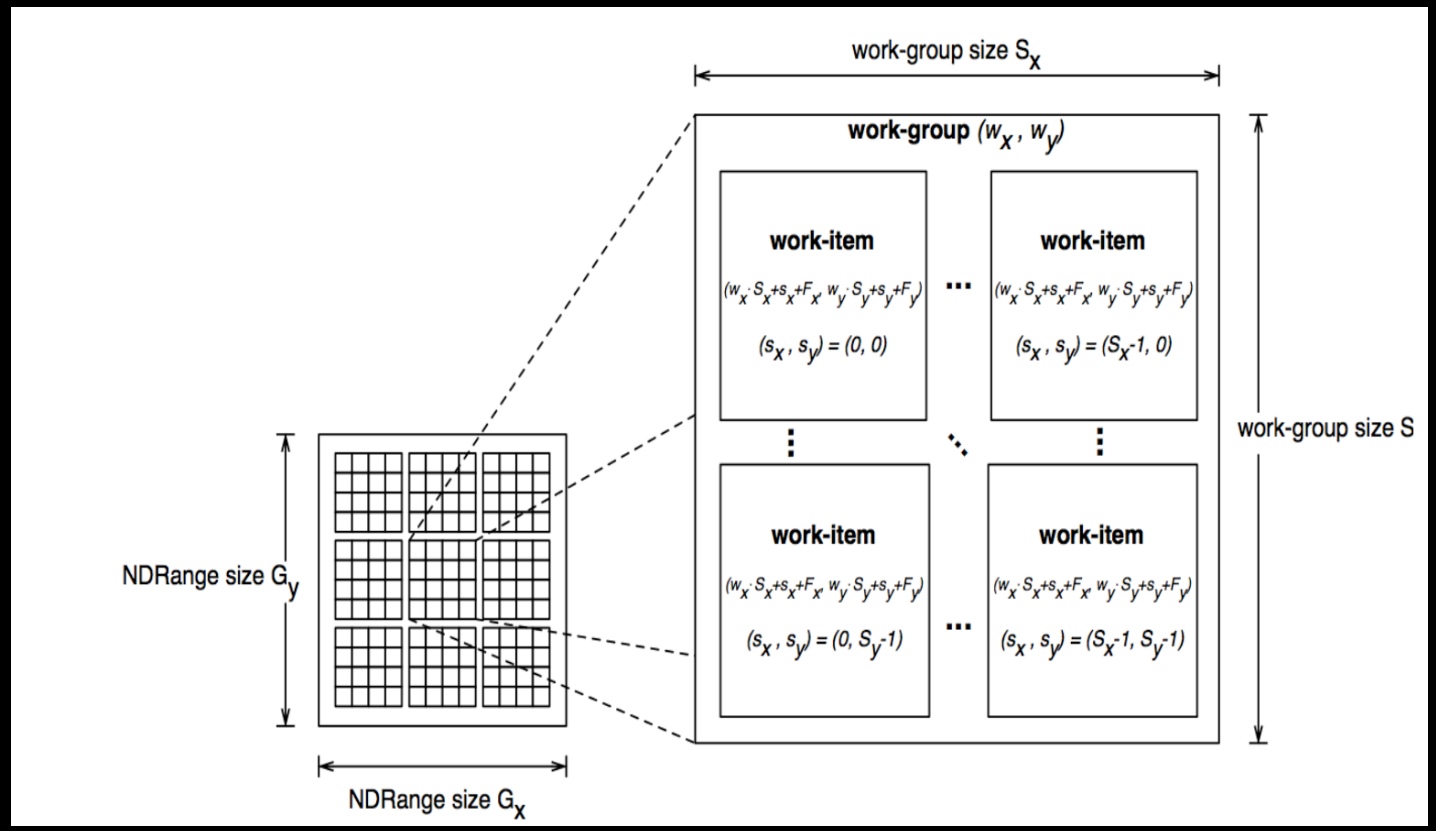
WHY GPUs?

- CPU
  - 3 GHz * 4 Cores * 8 (flops per AVX instruction) * 2 (FMA) = 192 Giga flop per second.
- GPU (Hawaii)
  - 2816 compute elements * 1 GHz * 2 (FMA) = 5.6 Tera-flop.
  - (29 times CPU theoretical peak.)

OPENCL ™

AMD

- Open Standard incepted by Apple.
- Supported on all recent AMD GPUs.
- Write a single threaded c/c++ to be executed by a multitude of threads.
- Each thread has x,z,y identification.



VECTOR ADD EXAMPLE

```
__kernel void vecAdd(  __global double *a,
                       __global double *b,
                       __global double *c,
                       const unsigned int n)
{
    //Get our global thread ID
    int id = get_global_id(0);

    //Make sure we do not go out of bounds
    if (id < n)
        c[id] = a[id] + b[id];
}
```

AMD

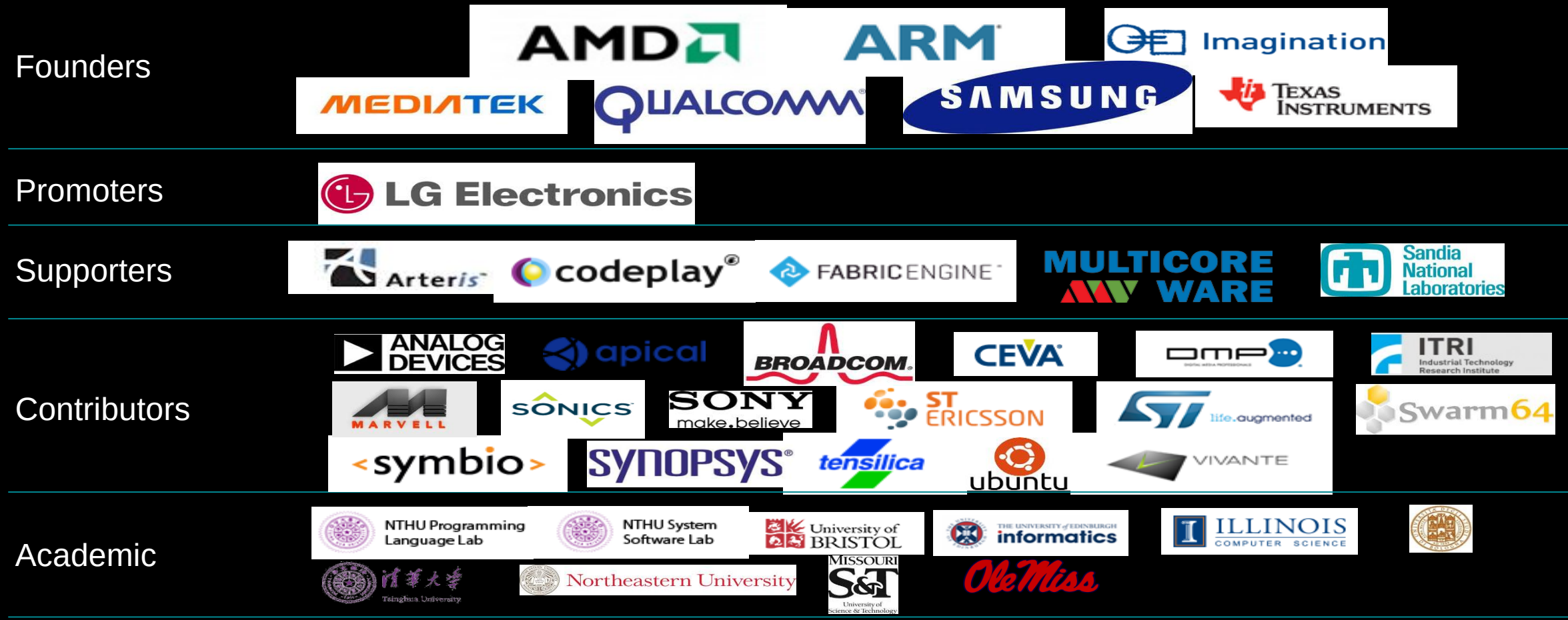
HSA

HSA HIGHLIGHTS

- Implemented above AMD APU .
- Industry wide standard.
- Unified memory architecture.
- User Mode Queues.
- Cross vendor IL (HSA- Intermmediate Language)
- Gains
  - Easier to program
  - Easier to load balance
  - Higher performance
  - Lower power

- **Available with AMD Kaveri APU**.
- Project Sumatra.

HSA FOUNDATION MEMBERSHIP — AUGUST 2013



HSA UNIFIED MEMORY

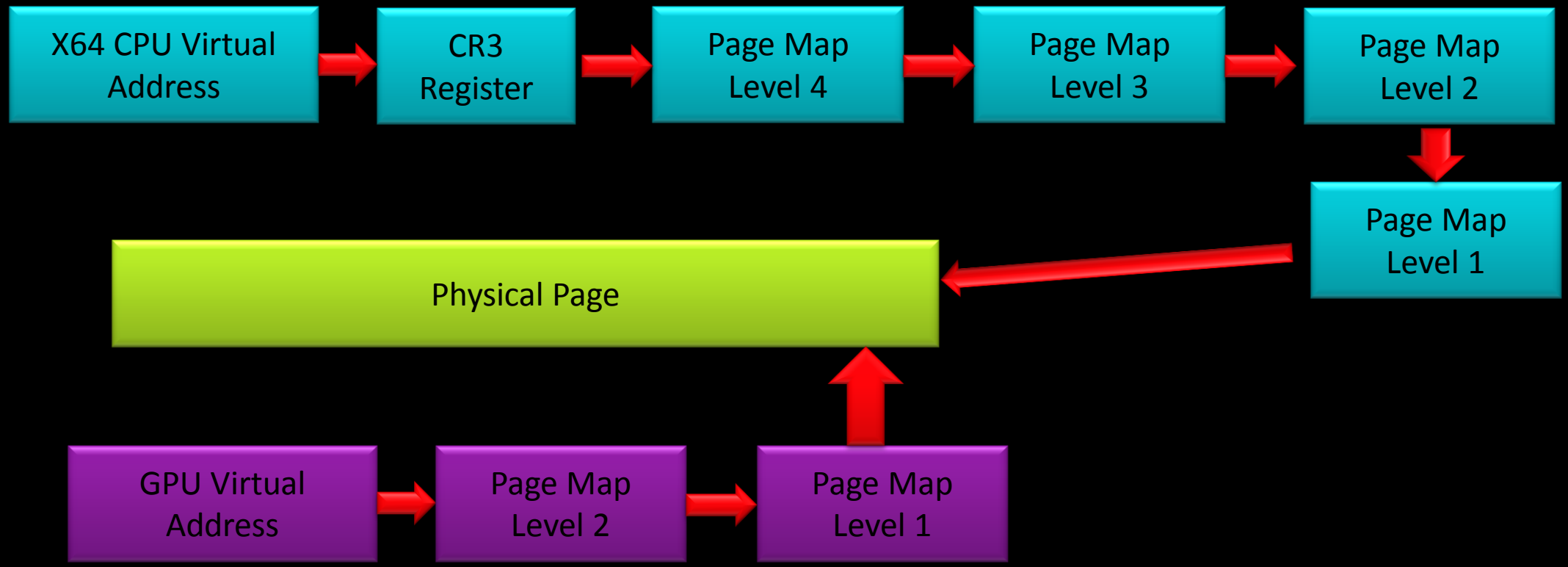
WHAT ARE THE BENEFITS?

- Abolishes the need for memory transfers leading to improved power consumption.
- Allows the GPU to access non-linear data structure. (Tree, sparse matrices.)
- Makes it easier to port existing CPU code to GPU.
- Reduces memory overhead.
- Platform atomics. Enables implementation of fine grained CPU-GPU producer consumer patterns.
- Enables implementation of many concurrency models above HSA (e.g. OpenMP, C++ AMP).

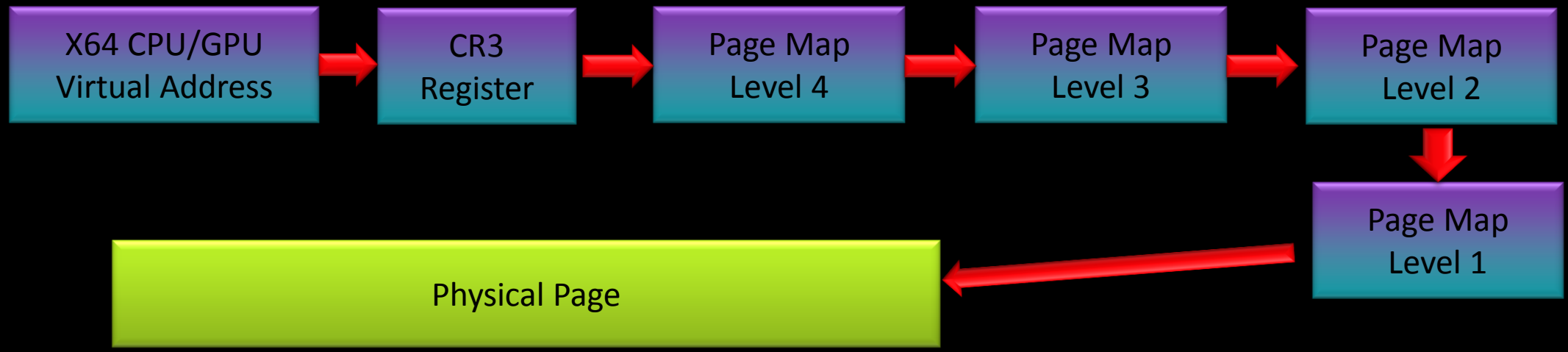
HSA UNIFIED MEMORY

BACK TO OPERATING SYSTEMS COURSE

- How virtual address is translated to physical address?



UNIFIED MEMORY WITH HSA



PROJECT SUMATRA

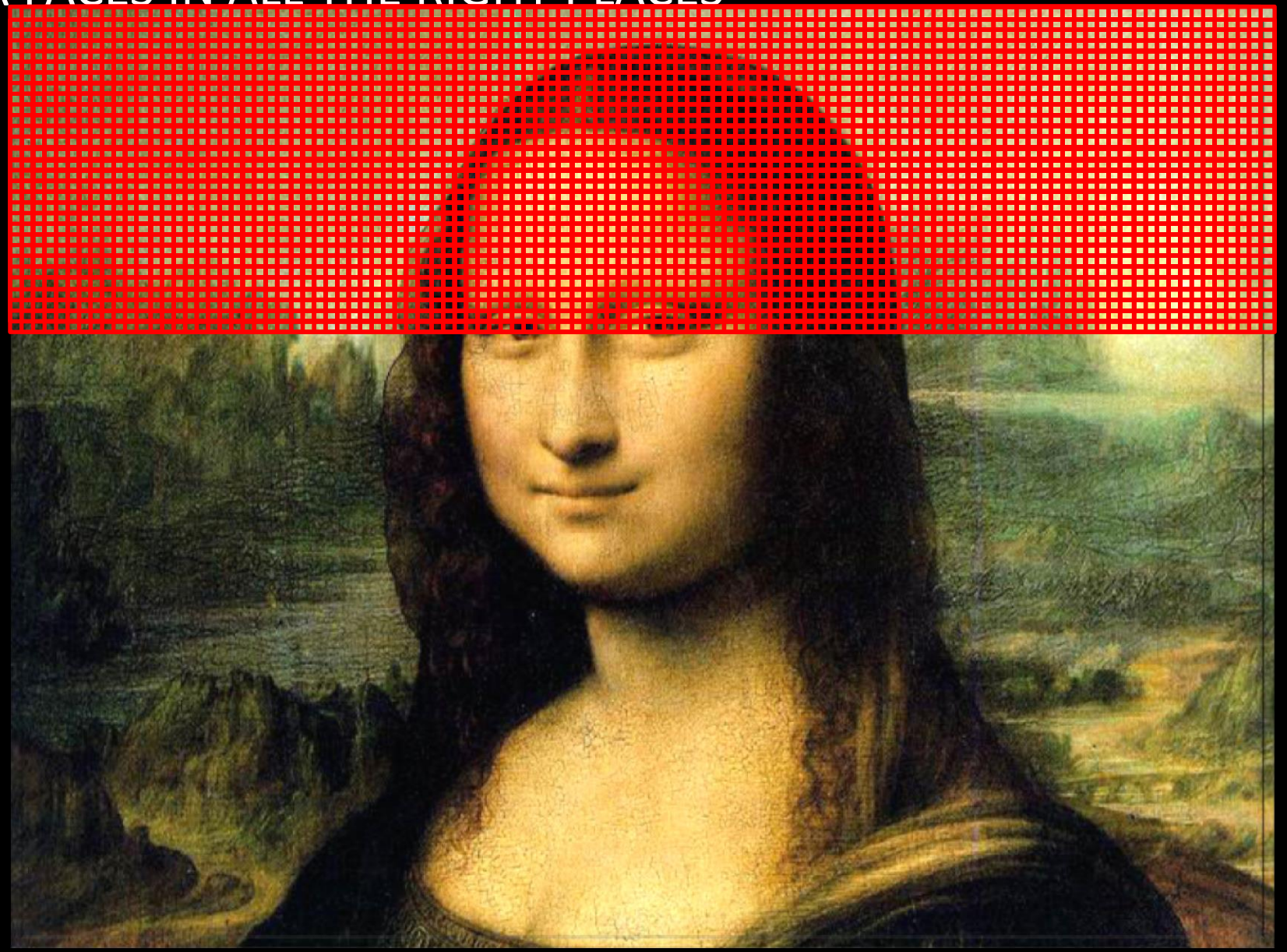
AMD

- GPU access from native Java code.

```
IntStream.range(0, in.length).forEach( p -> {
        out[p] = in[p] * in[p];
        });
```

- Available in Linux .

LOOKING FOR FACES IN ALL THE RIGHT PLACES



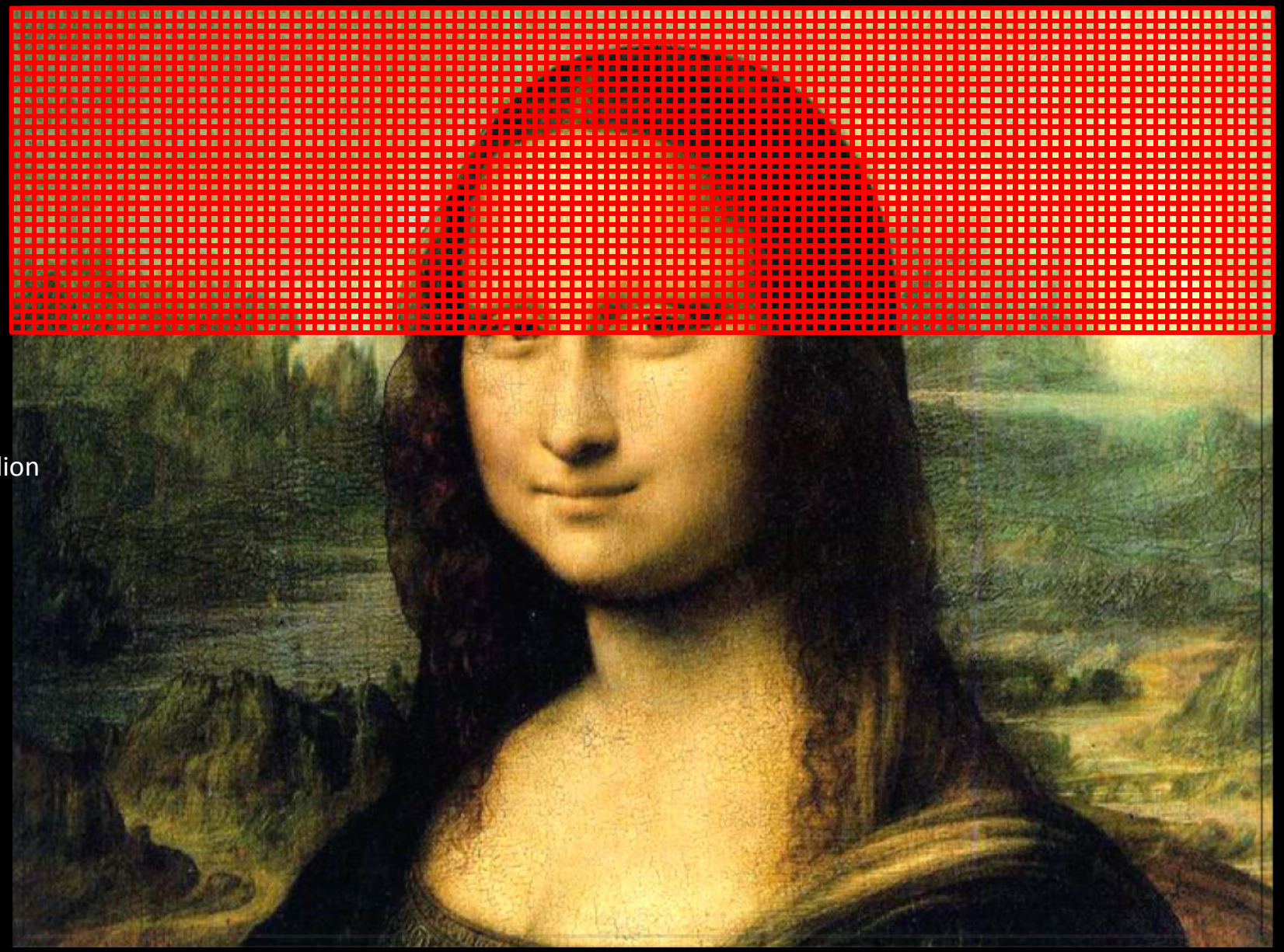
LOOKING FOR FACES IN ALL THE RIGHT PLACES

Quick HD Calculations

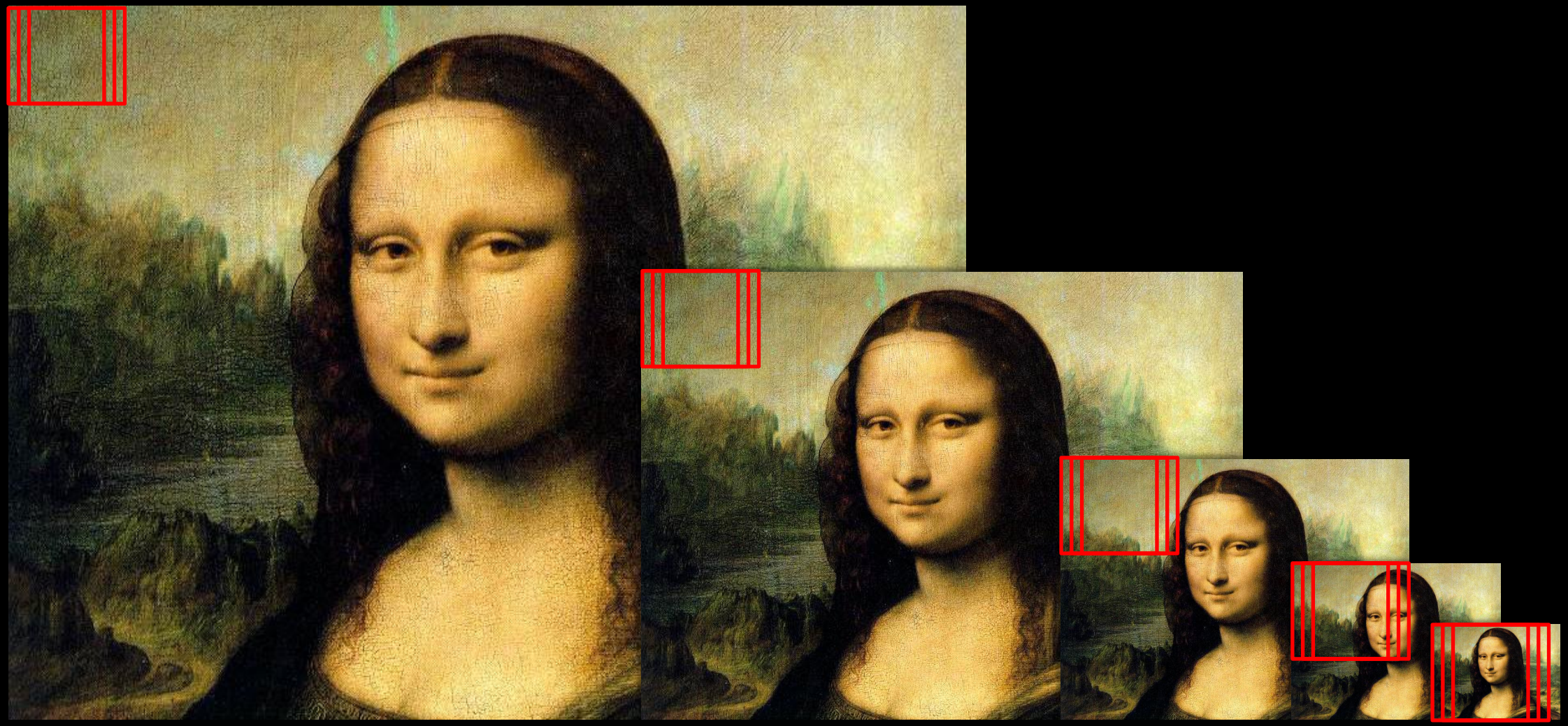
Search square = 21 x 21

Pixels = 1920 x 1080 = 2,073,600

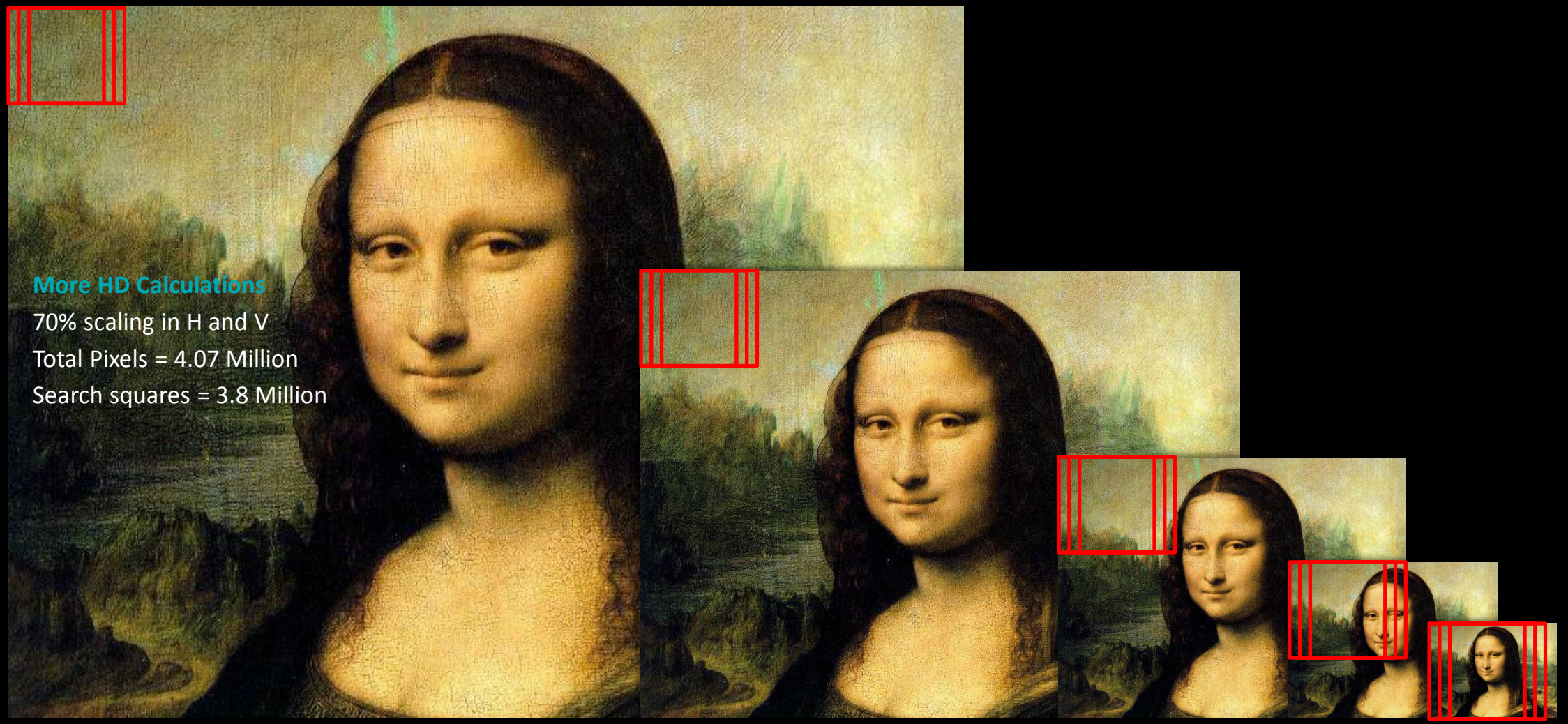
Search squares = 1900 x 1060 = ~2 Million



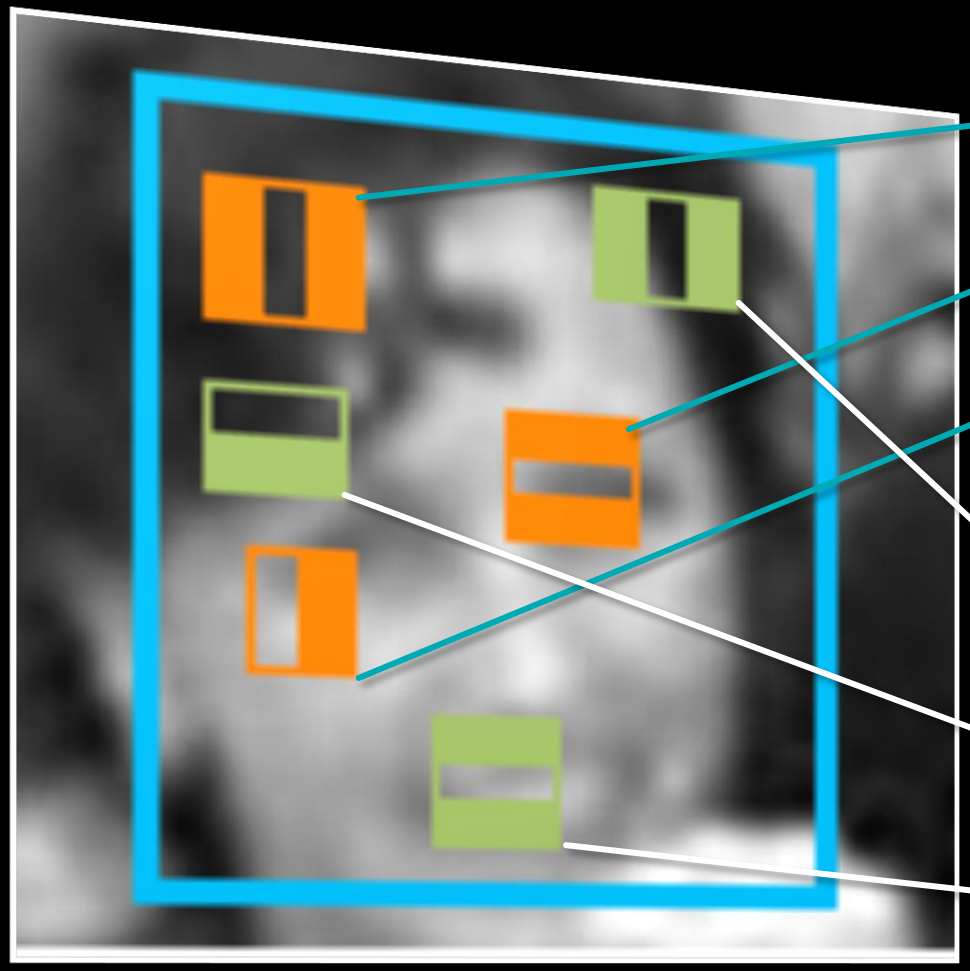
LOOKING FOR DIFFERENT SIZE FACES – BY SCALING THE VIDEO FRAME



LOOKING FOR DIFFERENT SIZE FACES – BY SCALING THE VIDEO FRAME



HAAR CASCADE STAGES



Feature k

Feature l

Feature m

Stage N

Face still possible?

Yes

No

Feature p

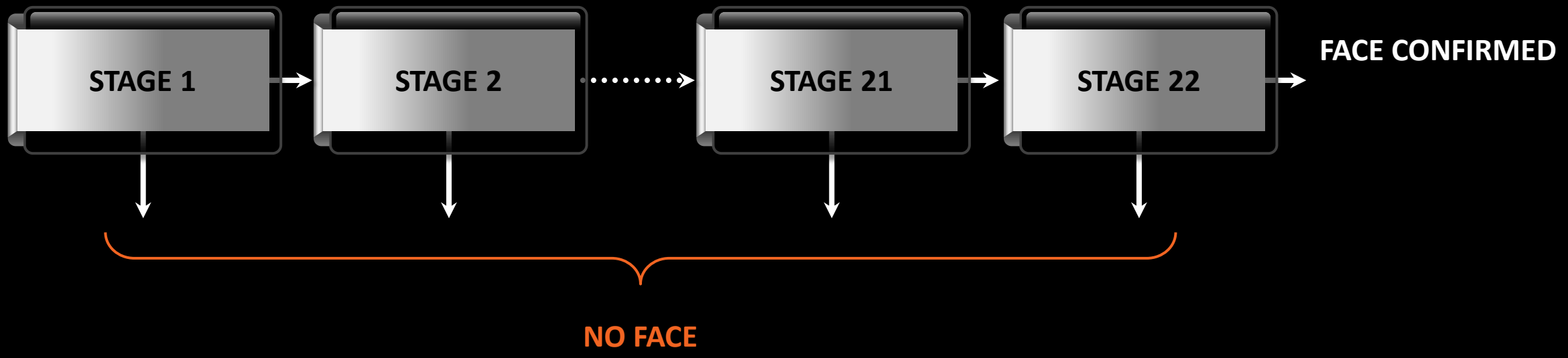
Feature r

Feature q

Stage N+1

REJECT FRAME

22 CASCADE STAGES, EARLY OUT BETWEEN EACH



Final HD Calculations

Search squares = 3.8 million

Average features per square = 124

Calculations per feature = 100

Calculations per frame = 47 GCalcs

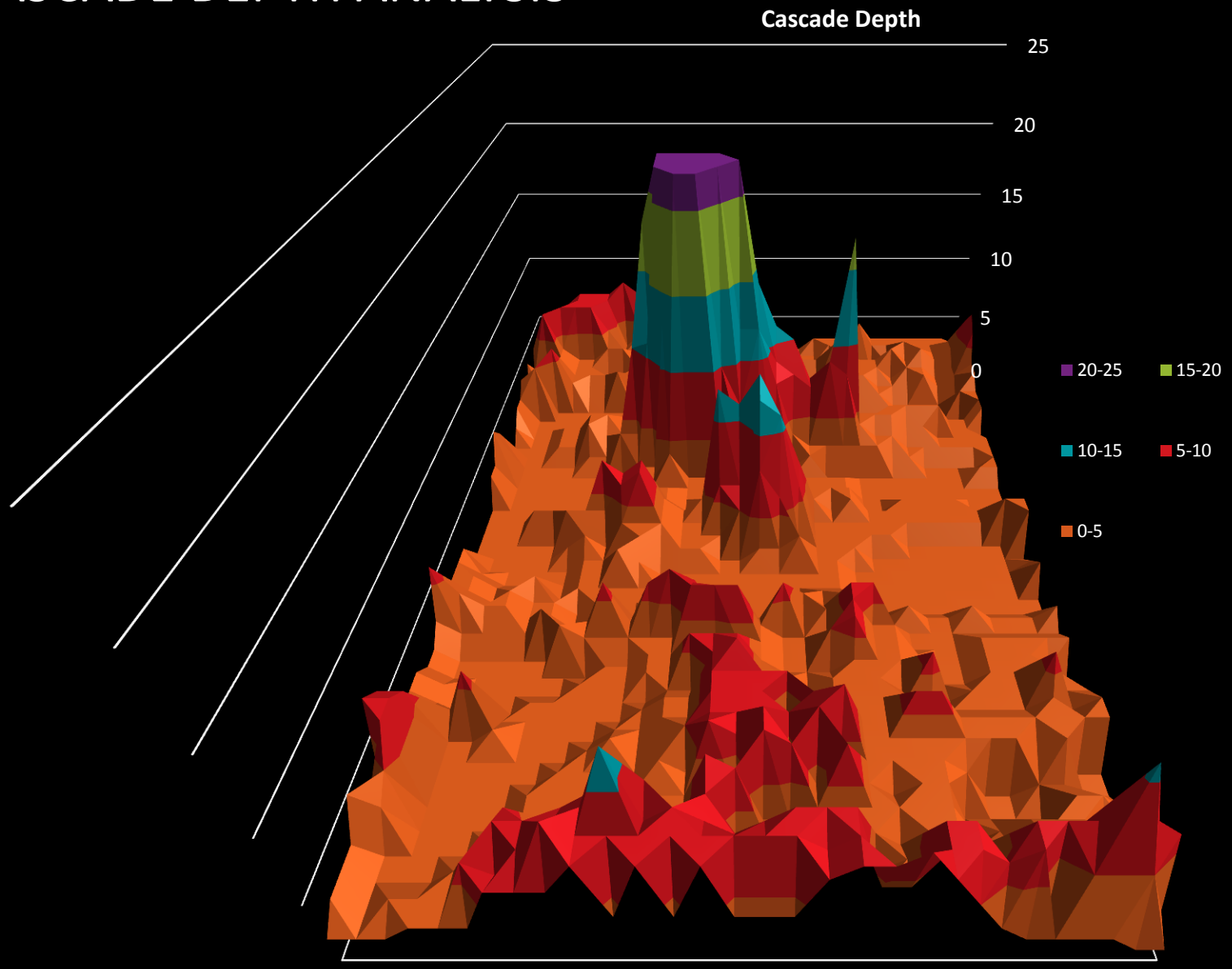
Calculation Rate

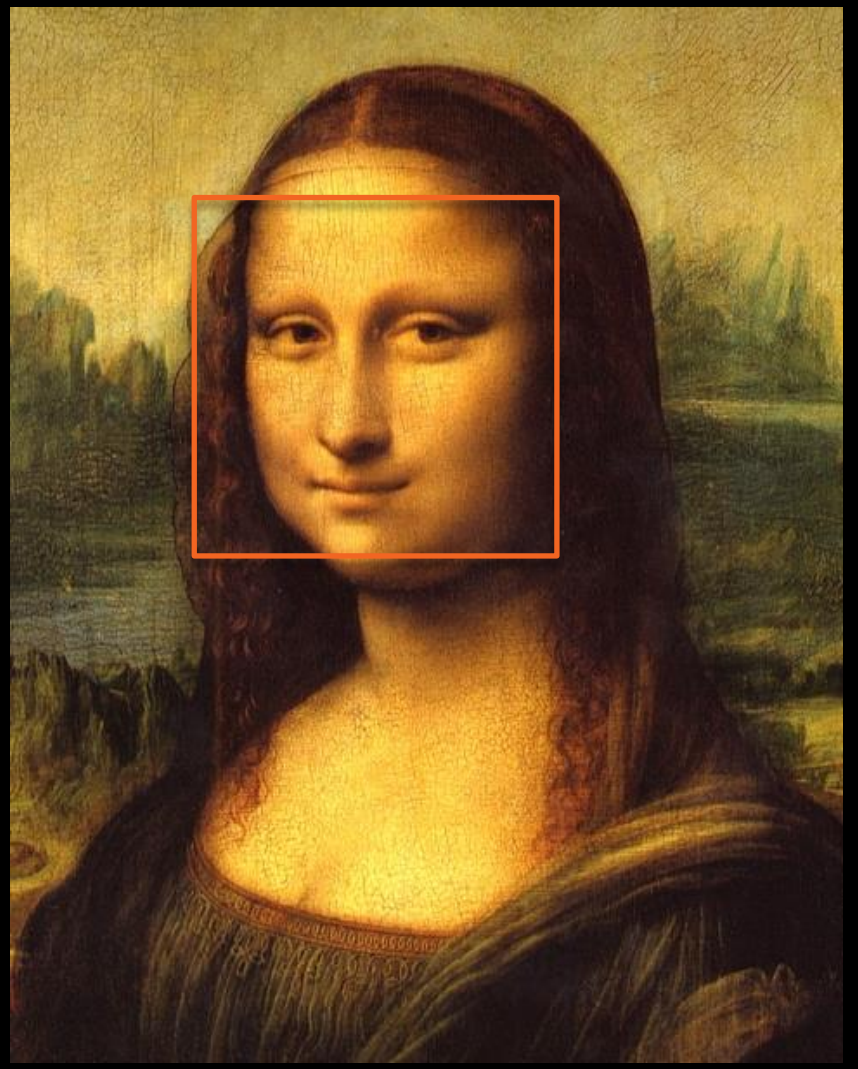
30 frames/sec = 1.4TCalcs/second

60 frames/sec = 2.8TCalcs/second

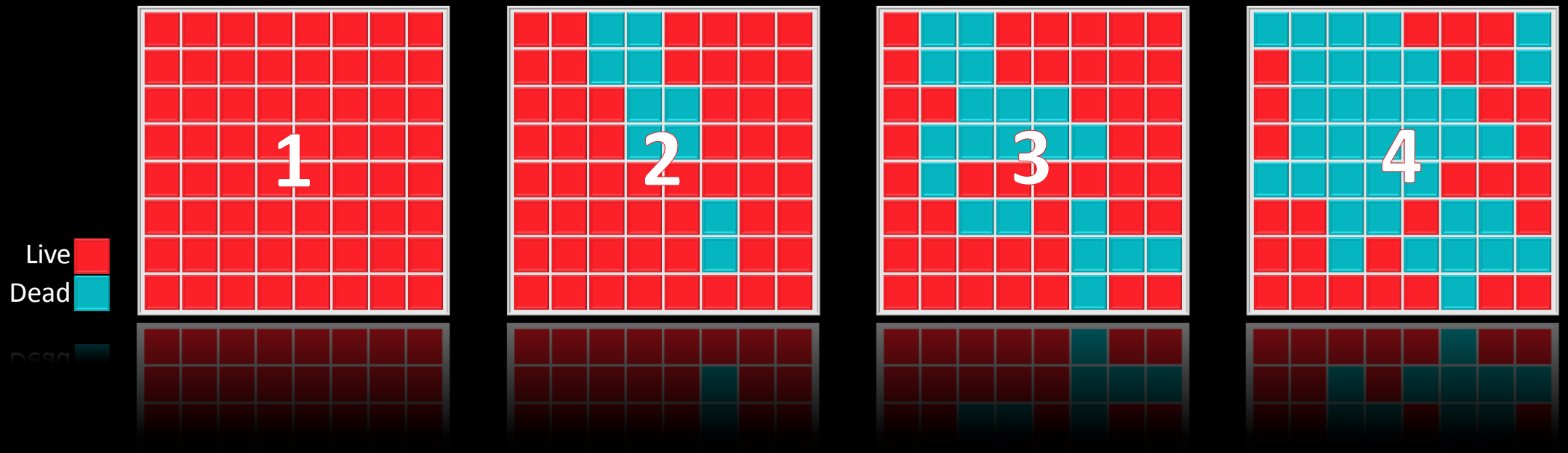
…and this only gets front-facing faces

CASCADE DEPTH ANALYSIS



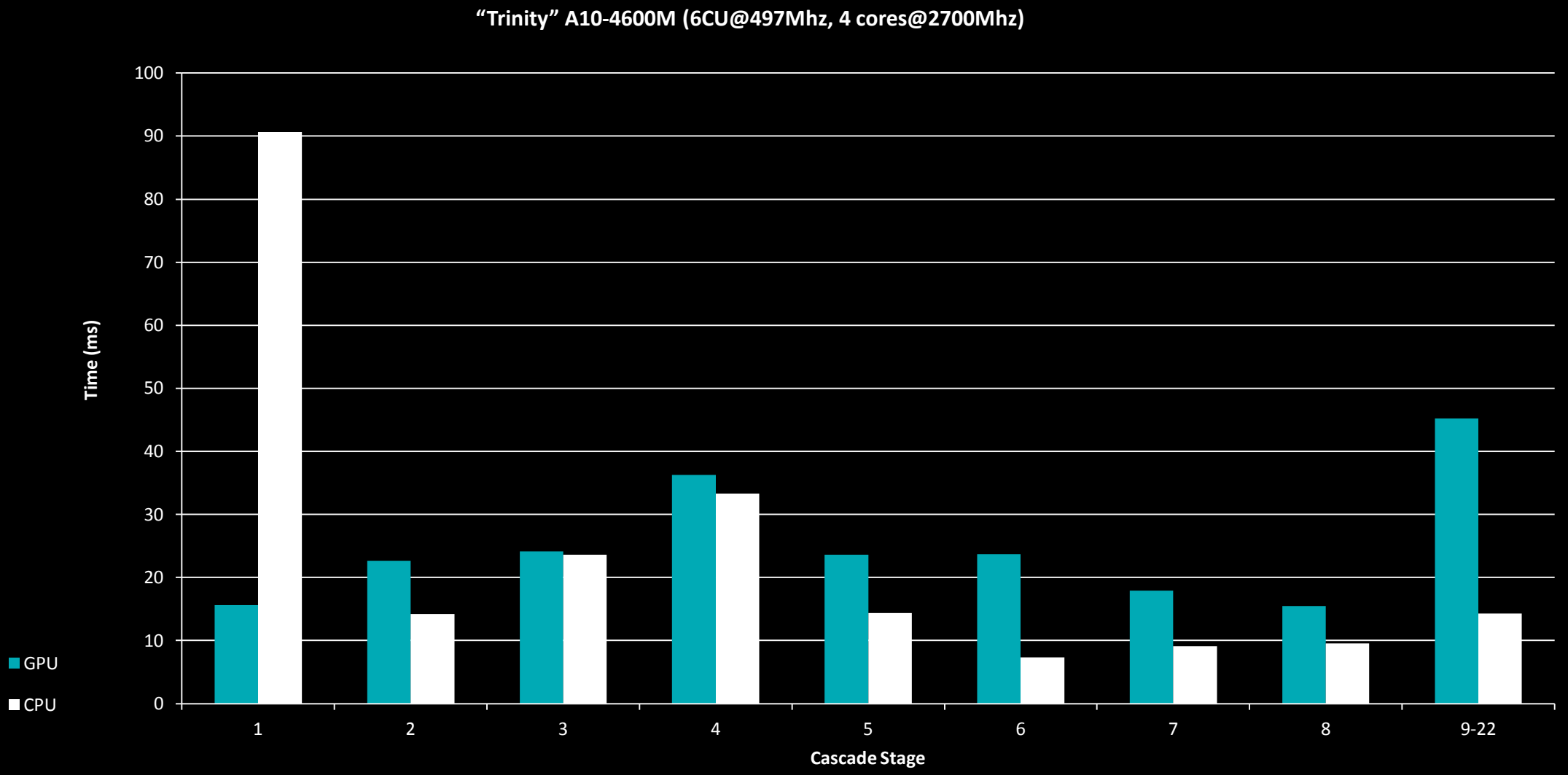


UNBALANCING DUE TO EXITS IN EARLIER CASCADE STAGES



- When running on the GPU, we run each search rectangle on a separate work item
- Early out algorithms, like HAAR, exhibit divergence between work items
  - Some work items exit early
  - Their neighbors continue
  - SIMD packing suffers as a result

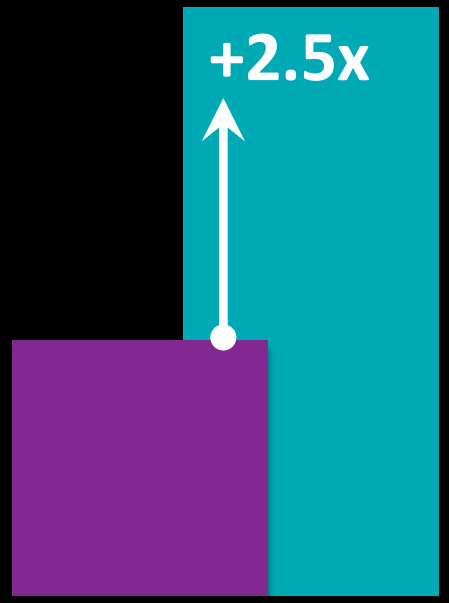
PROCESSING TIME/STAGE



AMD A10 4600M APU with Radeon™ HD Graphics; CPU: 4 cores @ 2.3 MHz (turbo 3.2 GHz); GPU: AMD Radeon HD 7660G, 6 compute units, 685MHz; 4GB RAM; Windows 7 (64-bit); OpenCL™ 1.1 (873.1)

HAAR SOLUTION – RUN DIFFERENT CASCADES ON GPU AND CPU

By seamlessly sharing data between CPU and GPU,
HSA allows the right processor to handle its appropriate workload



INCREASED PERFORMANCE

DECREASED ENERGY PER FRAME

REFERENCES

AMD

- Introduction to GCN
  - http://developer.amd.com/wordpress/media/2013/06/2620_final.pdf
- GCN white paper
  - http://www.amd.com/us/Documents/GCN_Architecture_whitepaper.pdf
- CodeXL home page
  - http://developer.amd.com/tools-and-sdks/heterogeneous-computing/codexl/
- AMD OpenCL programmers guide
  - http://developer.amd.com/download/AMD_Accelerated_Parallel_Processing_OpenCL_Programming_Guide.pdf

DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

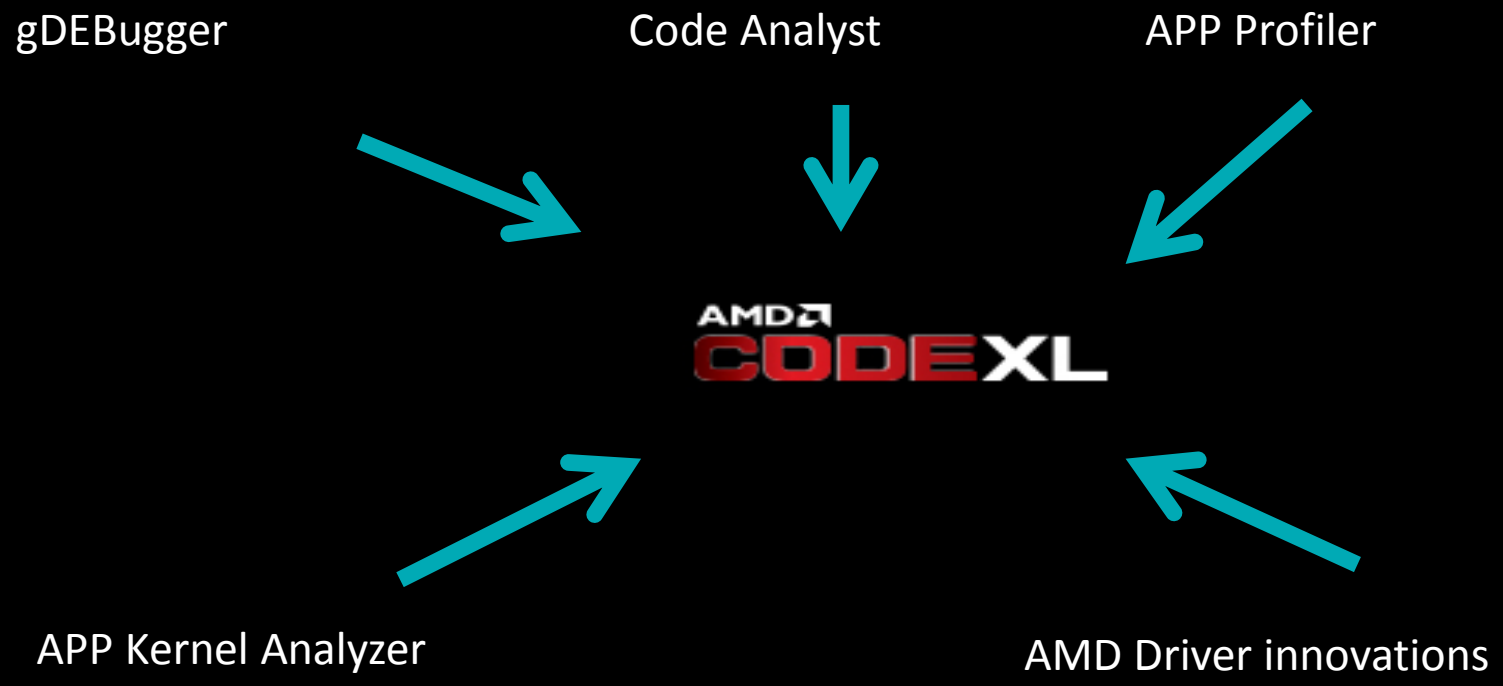
AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION

BACKUP

AMD CodeXL

CONVERGE TOOLS AND TECHNOLOGIES



AMD CODE XL

- Coherent, innovative and unified developer tools suite
  - Debug, Profile, and Analyze applications
  - System-wide “white box” view
  - AMD CPUs, GPUs and APUs
  - Does not require source code modifications