A RISC-V-based, Multi-threaded General Purpose GPU Core

Tamer EREN, Ömer GÜZEL, Mehmet Eyyüp ERGİN, Muhammed Mustafa ÖNEY, Hasan Erdem YANTIR
Integrated Circuits Design and Training Laboratory (TÜTEL), TÜBİTAK BİLGEM

MOTIVATION

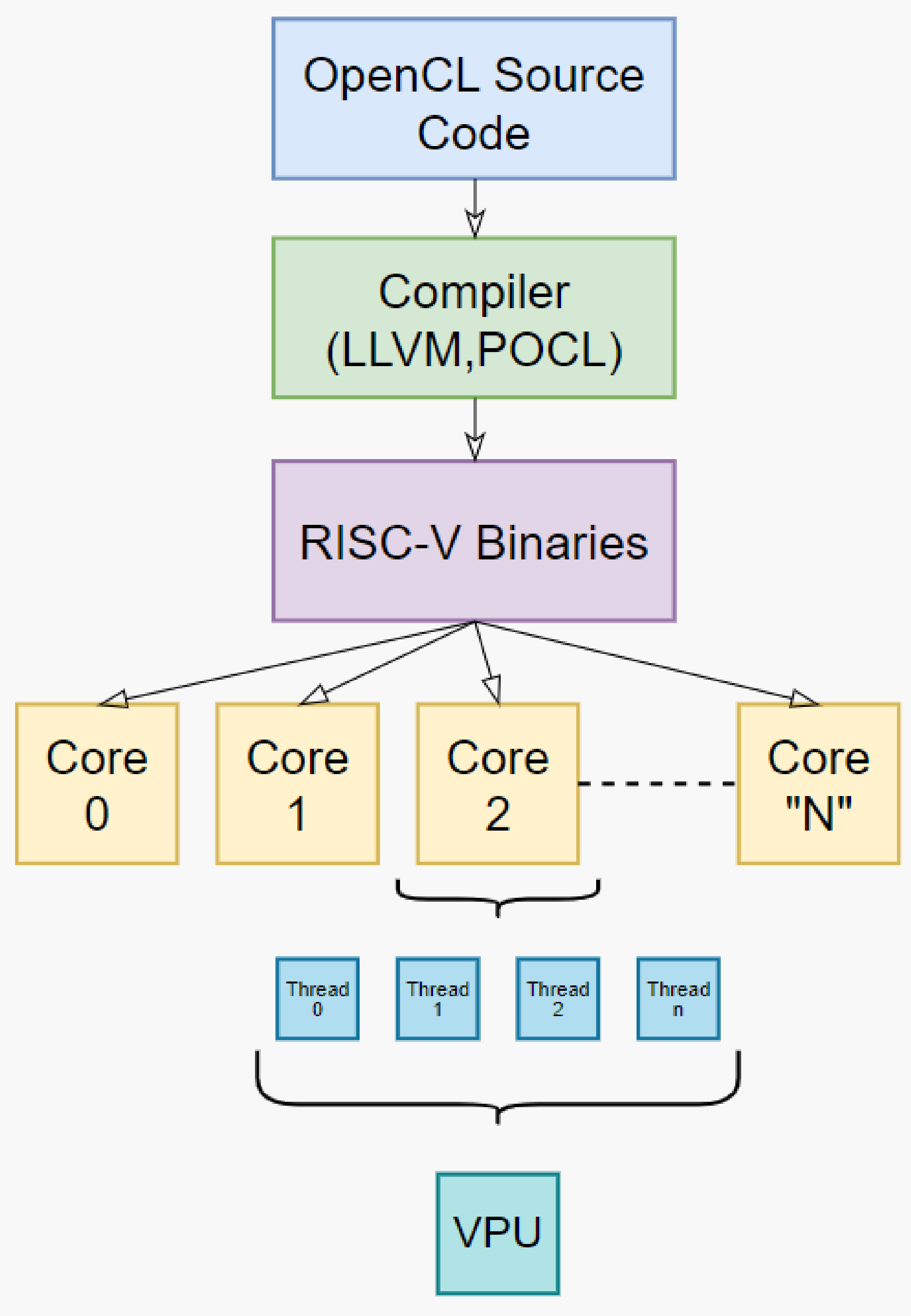
- GPGPUs become more popular due to their parallel processing and complex calculation capabilities.
- Typical implementations are;
  - artificial intelligence,
  - biophysics,
  - gaming industry,
  - generation of crypto currencies,
  - scientific computations [1] .
- Designing a RISC-V based GP-GPU will be good alternative to other GPUs because of its variety of advantages including its openness, simplicity, modularity, extensibility, and stability.

Central Processing Unit (CPU)[2]

- Handles main processing functions of a computer and runs operating systems
- Number of cores is generally 2-64 (most CPUs)
- Runs processes serially
- Better at processing one big task at a time (as fast as possible)

Graphics Processing Unit (GPU)

- Specialized component for graphic and video rendering.
- State-of-the-art models have thousands of cores
- Runs processes in parallel at the same time
- Better at processing several smaller tasks at a time (throughput is extremely high)



Features

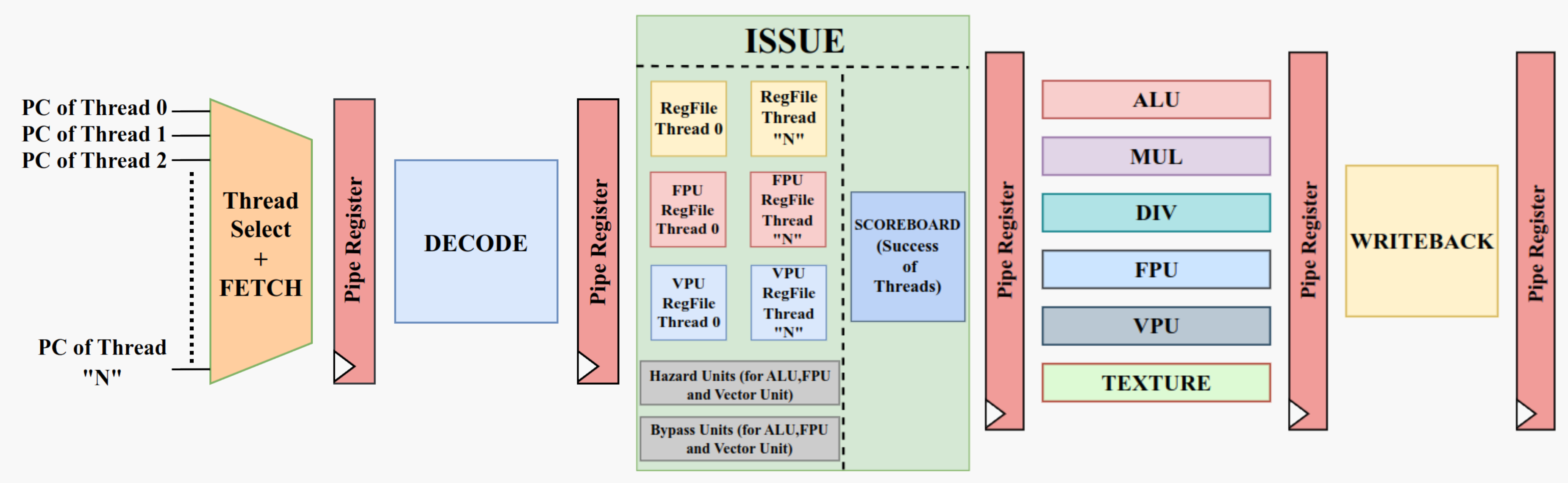
- Each core has its own L1 Cache
- 4 Cores share common L2 Cache
- All L2 Caches are connected to Shared Memory (L3 Cache)
- Each thread executes same instruction on different data (which means for every thread there are different register files)

There are 3 levels of **data parallelization**
- OpenCL (Multicore)
- Multithread
- VPU

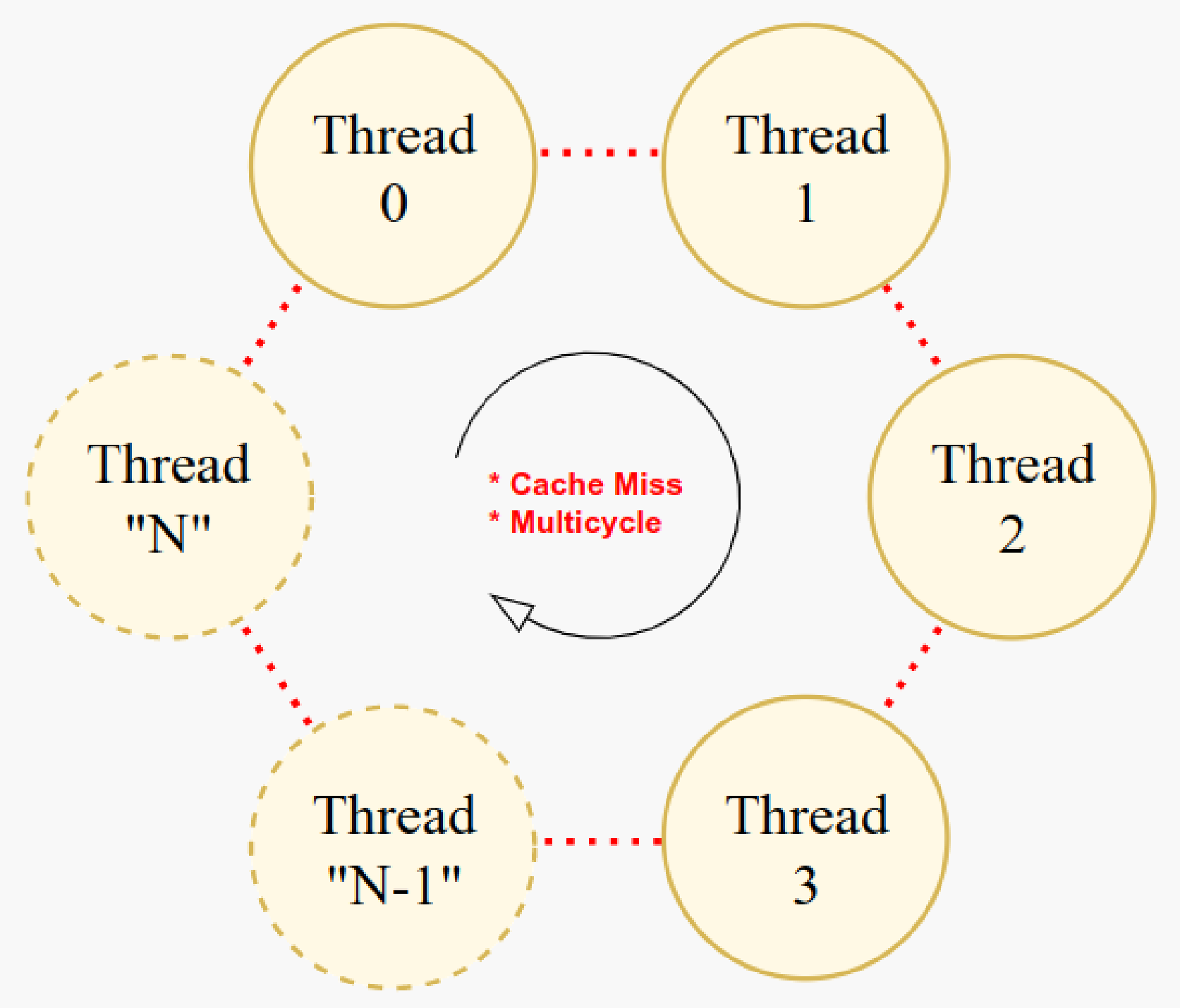
Performance Criteria	Digilent Nexys A7 FPGA
Working Frequency (MHz)	100
Power (W)	0.47
Slack (ns)	0.12
Number of LUTs	27255
Number of FFs	15028
Number of BRAMs	80

- The first prototype was synthesized and simulated with Digilent Nexys A7 FPGA at 100 MHz.
- This version incorporates 5 threads, I-M-F extensions, and a texture unit to handle the provided instructions.
- Moreover, it was synthesized using 28nm TSMC technology and the Synopsys IC Compiler II at a clock frequency of 2 GHz.

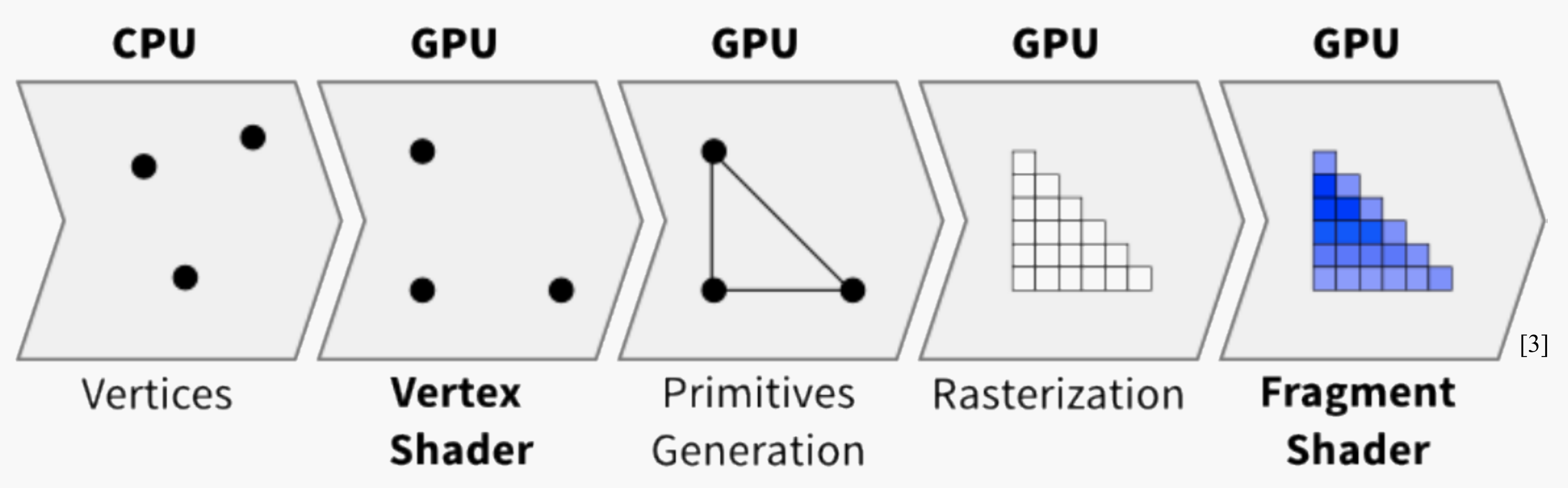
CORE DIAGRAM OF MULTI-THREADED GPGPU CORE



- Five-stage pipeline
- Designed with SystemVerilog
- In-Order Execution
- Supports 32 bits I – M – F – V (partial) extensions
- No Branch Prediction
- Core never stops the cache
- Multi-cycle operations are pipelined (including floating-point operations)
- Each thread has its own score in Scoreboard
- Cache miss and other multicycle operations cause thread circulation



THE RENDERING PIPELINE IN TEXTURE UNIT

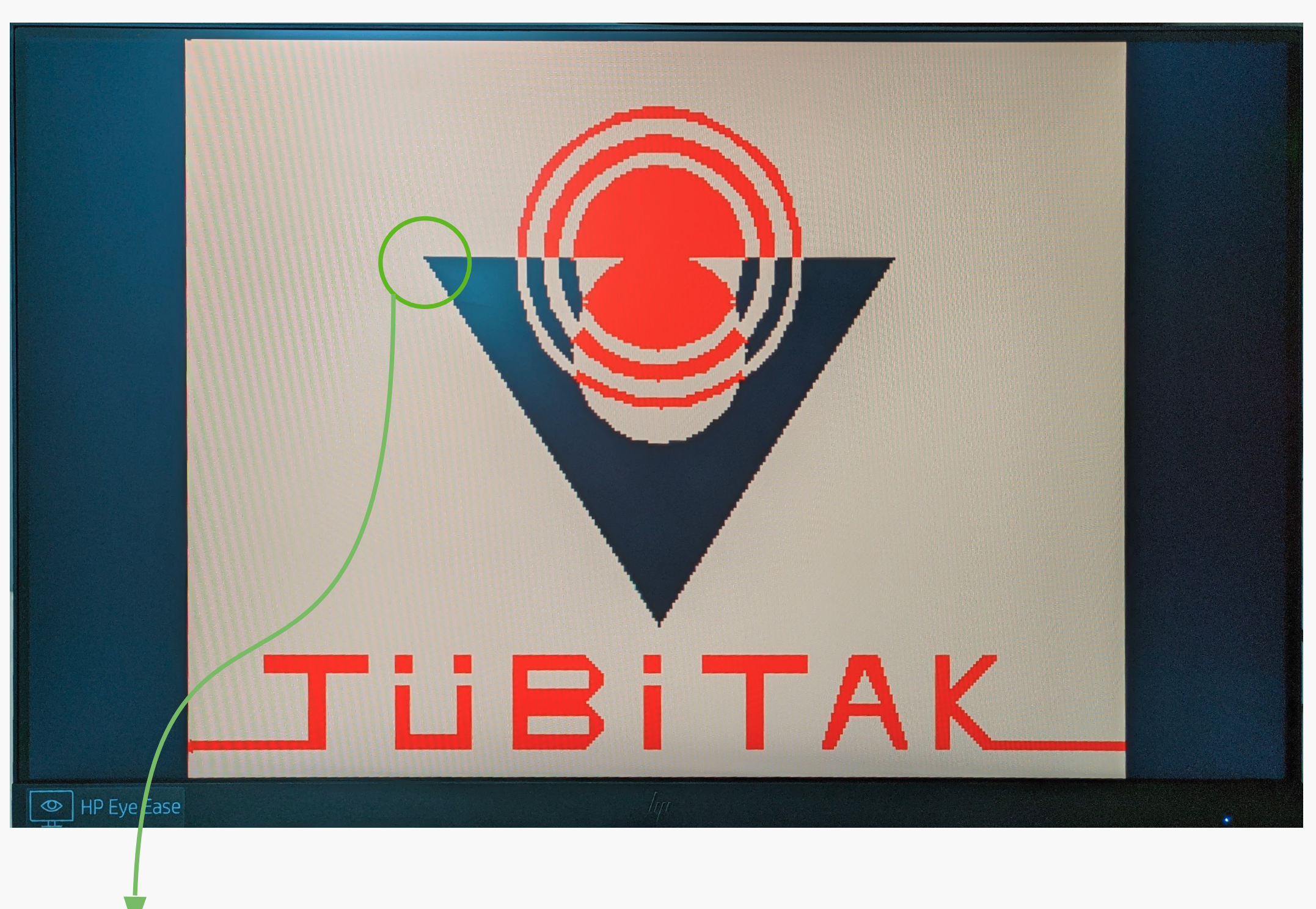


1) Determine the points[4]
2) Apply transformations if any (!)
3) Construct the triangle
4) Determine the colors of the points
5) Color the edges and pixels inside the triangle (or interpolate)
6) Apply masking or any other transformation
7) Send them via HDMI / VGA

There are 2 distinct approaches to form a triangle;

The first one is directly uploding the coordinates of the triangles from memory as a Load-Store Unit. **The second one** is calculating the new coordinates inside the GPU and generate a triangle in Texture Unit.

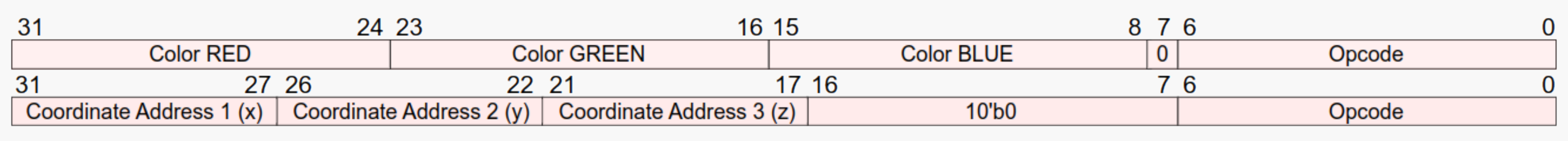
DEMO RESULTS OF THE GPGPU CORE WITH NEXSYS A7

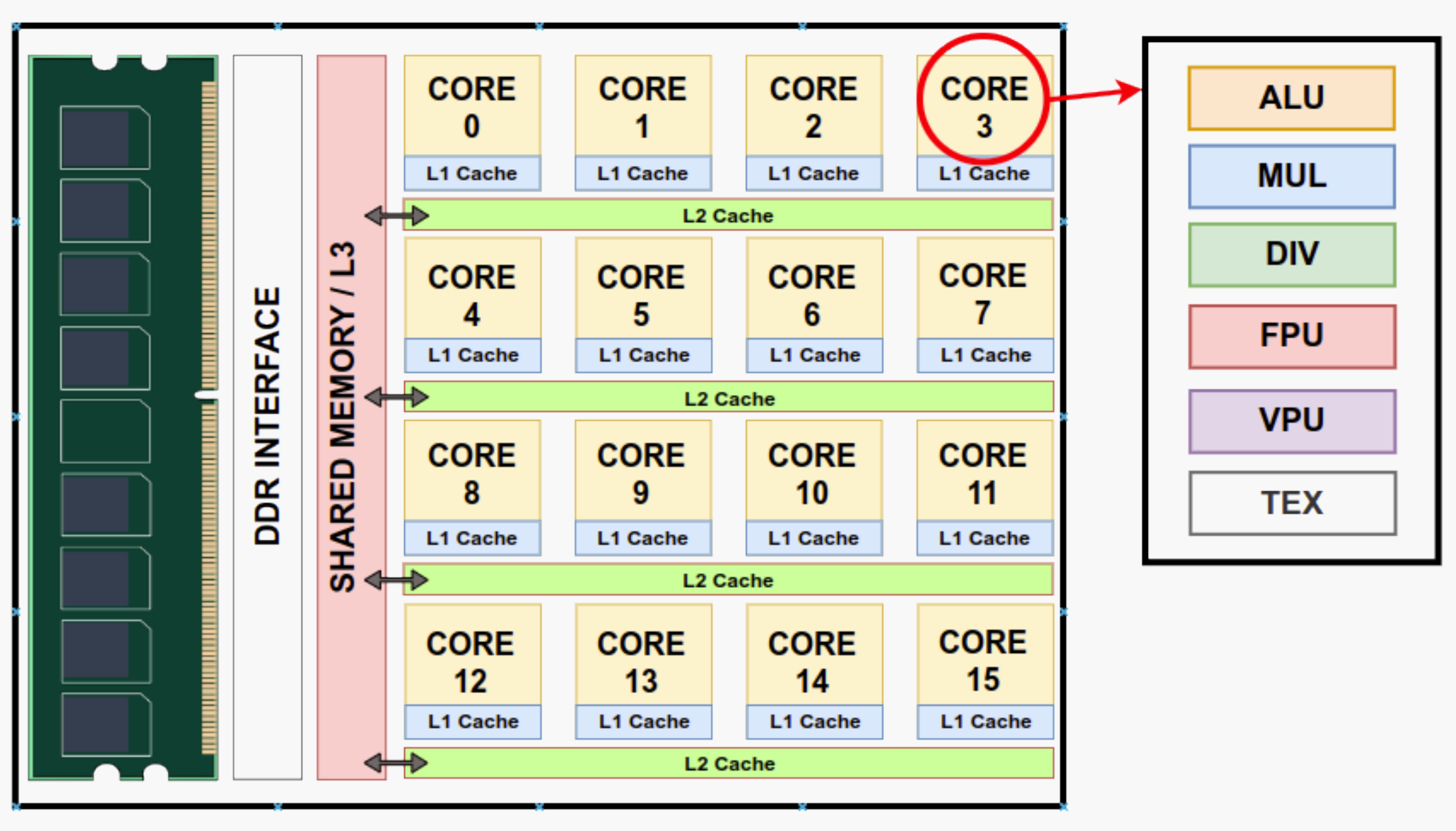


Shapes are composed of smaller triangles, which become more prominent along the edges.

1) Verification process of the I-M-F instructions were completed.
2) A dataset consisting of 2000 triangles capable of forming the 480x640 pixels TUBITAK Logo was constructed.
3) The coordinates (x, y, z) of each triangle were uploaded individually to the texture unit as a LSU.
4) Upon completion of the rendering pipeline, the aforementioned logo was successfully generated.

FUTURE WORKS





- 128/256 Cores, 16 Threads
- Complete support for Vector extension
- Chip level place and route with 28nm or below technologies.
- Dividing the screen into sections, and driving them with different cores
- Developing a complete set of instructions for all of the texture applications.

REFERENCES

1) Ghorpade, Jayshree, et al. "GPGPU processing in CUDA architecture." arXiv preprint arXiv:1202.4347 (2012).
2) Asano, Shuichi, Tsutomu Maruyama, and Yoshiki Yamaguchi. "Performance comparison of FPGA, GPU and CPU in image processing." 2009 international conference on field programmable logic and applications. IEEE, 2009.
3) Andreussi, Bauhaus–Universit¨at Weimar, "Shading with GLSL Vertex, Fragment Shaders & the Shading Equation"https://www.uni-weimar.de/fileadmin/user/fak/medien/professuren/Computer_Graphics/CG_WS_18_19/CG/03_Shaders.pdf ,2018.
4) Baek, Nakhoon, and Kuinam J. Kim. "Design and implementation of OpenGL SC 2.0 rendering pipeline." Cluster Computing 22.Suppl 1 (2019): 931-936.

For detailed questions and further explanations, you can contact Tamer EREN: tamer.eren@tubitak.gov.tr