Neox™
A RISC-V based GPU Processor

Dr. Iakovos Stamoulis
Director of Engineering Management

2nd RISC-V Week 2021, 31st March 2021
- **HQ and development center**: Athens/Patras, Greece
- **Sales & Tech-support offices**: USA, Canada, Germany, Taiwan, Japan
- **Technology licensing**: graphics solutions including: HW design, SW Libraries, SDK
- **IP cores**: graphic processing units (GPUs), Display controllers delivering low system power, low system cost and high performance
- **Target markets**: small-mid size display devices (Wearables, Embedded) using 32bit MCU

### Licensees
- 2 x Undisclosed Tier1 US
- 2 x Undisclosed Tier1 China

### Technology Partnerships
- Samsung
- MLPGPU²
- Hong Kong University of Science & Technology
- Technische Universität Berlin
- Faraday
- Lattice Semiconductor
- STMicroelectronics
- Digilen
- General Dynamics
- ARC Synopsys
- AMIQ Micro
- Sequans Communications
- Synopsys
- Codeplay
- Qualcomm
- Atmel Microchip
Different architectures for different markets & applications

**Markets/Applications**

**Graphics acceleration/Video Overlay**
Small displays (1.5” – 6”) 1024x768
Home Control, Appliances, Wearable / IoT / Embedded, Video Overlay (4k)
- RTOS based, Bare Metal

**Markets/Applications**

**Connected Endpoints/EDGE**
Mid/large displays >4k
AI Inference, Security/Surveillance, Augmented Reality, Smart Factory, Entertainment, Auto
Linux, GPGPU, Compute

**CPU**
32-bit

**NEMA® Pico-Series**
XS & XL
2D / 2.5D GPU

**CPU**
32/64-bit

**NEOX™-Series**
AI & GFX
AI Accelerator / 3D GPU

**Markets/Applications**

Small displays (1.5” – 6”) 1024x768
Home Control, Appliances, Wearable / IoT / Embedded, Video Overlay (4k)
- RTOS based, Bare Metal

**Markets/Applications**

Connected Endpoints/EDGE
Mid/large displays >4k
AI Inference, Security/Surveillance, Augmented Reality, Smart Factory, Entertainment, Auto
Linux, GPGPU, Compute
Different architectures for different markets & applications

<table>
<thead>
<tr>
<th></th>
<th>Nema®XL</th>
<th>NEOX™-Series</th>
<th>CPU 32/64-bit</th>
<th>NEMA®</th>
<th>Pico-Series</th>
<th>XS &amp; XL</th>
<th>2D / 2.5D GPU</th>
<th>CPU 32-bit</th>
<th>NEOX™-Series</th>
<th>AI &amp; GFX</th>
<th>AI Accelerator / 3D GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>2.5D GPU</td>
<td>3D GPU / Compute / AI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corescore</td>
<td>1-4</td>
<td>1-64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock Range @28nm</td>
<td>300 MHz</td>
<td>300 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>1 pixel/clock/cycle/core</td>
<td>1 pixel/clock/cycle/core</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISA</td>
<td>Nema VLIW</td>
<td>RV64IMFC + extensions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shader Processor</td>
<td>Limited programmability</td>
<td>Fully Programmable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fragment processor</td>
<td>GCC / LLVM C/C++ RISCV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FB Compression</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Texture Compression</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory System</td>
<td>AHB 32-bit</td>
<td>AXI4 64/128-bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AI Framework</td>
<td>NemaGFX + SDK</td>
<td>NemaGFX + SDK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3D Extension</td>
</tr>
<tr>
<td></td>
<td>GUI Builder</td>
<td>GUI Builder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GUI Builder</td>
</tr>
<tr>
<td>Graphics Framework</td>
<td>NemaGFX + SDK</td>
<td>NemaGFX + SDK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3D Extension</td>
</tr>
<tr>
<td></td>
<td>GUI Builder</td>
<td>GUI Builder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GUI Builder</td>
</tr>
<tr>
<td></td>
<td>PixPresso</td>
<td>PixPresso</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NEOX™, a RISC-V based application specific processor

CPU and GPU/Al cores represent the bulk of Silicon Real Estate in a modern SoC!

Think Silicon’s NEOX™, a RISC-V based GPU together with RISC-V CPUs can serve a wide range of markets that require:

- Artificial Intelligence
- GPGPU Compute
- Graphics Rendering

**System Advantages** when used with RISC-V CPU
- Leveraging common compiler technologies
- Unique workload balancing
Agenda

• Neox Architecture

• Software Stack
  - SDK Tools
  - AI Inference
  - Graphics
## CPU vs GPU

### Workload
- **CPU**
  - Lots of instructions / Less data
  - Task Parallel
- **GPU**
  - Few Instructions/Lots of Data
  - Data Parallel

### Features
- **CPU**
  - Out of order exec
  - Pipeline Interlocks
  - Branch prediction
  - Complex sync
- **GPU**
  - Hardware Threading
  - Barriers for synchronization

### Data Reuse
- **CPU**
  - Data Reuse and Locality
  - Latency Optimized
- **GPU**
  - Little Data Reuse
  - Throughput Optimized

### ISA
- **CPU**
  - Scalar Instruction
  - +SIMD Extensions
- **GPU**
  - SIMD Instructions
  - +Scalar/Integer for GPGPU

---

**RV64GC + Application Extensions**
Example SoC CPU + NEOX™GPU

- **A RISC-V ISA coprocessor array suitable for AI/ Graphics /Imaging workloads**

- **Scalable Design:** 1-64 cores targeting from embedded market to high end solutions

- **Ecosystem:** Leverage RISC-V ecosystem and Tooling (GCC/LLVM)

- **AI Inference:** TensorFlow Lite/MCU

- **Graphics Rendering:** Think Silicon graphics Libraries

- **Low Power:** Small Design with Ultra Low area and Gate Count

- **Extensible:** User Defined Instructions
NEOX™: Cluster

NEOX™1-4 Cores per Cluster

Cluster Control Unit

Rasterizer 2D/3D

New Task Scheduling

Tile Management Unit

Scratchpad Memory

TSc FB Compression

multilevel cache network

NEOX™1-16 Cluster

NEOX™ Array

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Shader Cluster

Graphics ISA Extensions/Coprocessors
Unified Shader Architecture
Tile Based Rendering
Color/Vertex Vector Support

Dedicated HW Modules:
- Rasterizer
- Texture Unit / Caches
- Tile Unit (Blending /Z Depth/Stencil Test)

User Defined Instruction
Dedicated Interface to support user provided extensions
Agenda

- Neox Architecture

- Software Stack
  - SDK Tools
  - AI Inference
  - Graphics
Neox® | SDK - A Complete Ecosystem Of Tools

**Neox®|Bits**
An EVK Kit for technology evaluation and pre-silicon application development

**Neox AI-SDK**
Import networks in ONNX /Tensorflow Format and optimize them for Hardware Acceleration with TF Lite/MCU Runtime Libraries

**Neox Compiler**
GCC/LLVM compiler RISC-V with added support for all Neox custom ISA extensions with support for C/C++, SPIR-V and GLSL

**NEMA®|GFX**
Lightweight Graphics API for embedded Graphics applications

**NEMA®|GUI-Builder**
GUI Design Toolkit that allows drag & drop creation of advanced GUI, in minutes instead of months.

**NEMA®|PIX-Presso**
Asset management and image optimization, for optimal visual appearance and efficient memory utilization
Neox Software Stack

- **Hardware**
  - Neox

- **Kernel Space**
  - Platform Backend

- **User Space**
  - Application
    - NemaGFX API
    - TFLite/MCU

- **Software Stack**
  - RISC-V C/C++ Compiler
    - GCC/LLVM
NEMA® | GUI-Builder allows the rapid creation of GUIs for Bare/RTOS or Linux embedded System within minutes instead of weeks.
NEMA®|Pix-Presso is a utility for converting images to formats suitable for low power embedded devices. It is an easy to use tool for graphics developers in order to adapt the best image by file-size and quality for its dedicated application requirements.

Platforms:
Windows 10
Linux
Neox Toolchain API can accept code in multiple source languages and generate executable binaries for Neox.
Simple 3D demo with a vertex and fragment shader. Workload is split in dynamically in multiple threads for each code type.
Neox Compiler: Vertex Shader

```
#version 400
layout(location = 1) in vec4 a_pos;
layout(location = 2) in vec2 a_texCoord;
layout(std140, binding = 0) uniform uni0 {
  mat4 MVP;
};

layout(location = 0) out vec2 TextureCoord;

void main() {
  gl_Position = MVP * a_pos;
  TextureCoord = a_texCoord;
}
```

```
vec4 a_pos;
vec2 a_textCoord;
void main_shader(uint32_t rast_index){
  struct uni0 _25;
  struct gl_PerVertex _19;
  auto _2500 = neox_read_consts_vec2(0);
  auto _2501 = neox_read_consts_vec2(1);
  auto _2510 = neox_read_consts_vec2(2);
  auto _2511 = neox_read_consts_vec2(3);
  auto _2520 = neox_read_consts_vec2(4);
  auto _2521 = neox_read_consts_vec2(5);
  auto _2530 = neox_read_consts_vec2(6);
  auto _2531 = neox_read_consts_vec2(7);
  _25.MVP = {_2500[0], _2500[1], _2501[0], _2501[1],
              _2510[0], _2510[1], _2511[0], _2511[1], _2520[0],
              _2520[1], _2521[0], _2521[1], _2530[0], _2530[1],
              _2531[0], _2531[1]};
  _19.gl_Position = _25.MVP*a_pos;
  neox_write_hw_vec2(a_textCoord, gl_varying_0);
  neox_write_hw_vec2(_19.gl_Position.xy, gl_PerVert_gl_Position_XY_0);
  neox_write_hw_vec2(_19.gl_Position.zw, gl_PerVert_gl_Position_ZW_0);
}
```

```
.globl __Z11main_shaderj
__Z11main_shaderj:
    lui a1, %hi(a_pos)
    addi a2, a1, %lo(a_pos)
    addi a3, a0, 352
    addi a4, a0, 360
    addi a5, a0, 368
    ld  v2, 8(a2)
    ld  v0, %lo(a_pos)(a1)
    lui a6, %hi(a_textCoord)
    ld  v6, %lo(a_textCoord)(t5)
    writer64.hw v6, t5
    mul.v2 v4, cv6, v3.yy
    mul.v2 v3, cv7, v3.yy
    madd.v2 v4, cv4, v2.xx, v4
    madd.v2 v2, cv5, v2.xx, v3
    madd.v2 v4, cv2, v3.yy, v4
    madd.v2 v2, cv3, v3.yy, v2
    madd.v2 v3, cv0, v0.xx, v4
    madd.v2 v0, cv1, v0.xx, v2
    writer64.hw v3, t3
    writer64.hw v0, t4
    yield
```
vec4 a_pos;
void main_shader(uint32_t rast_indx){
    auto TextureCoord000 = neox_read_reg_vec2(0);
    vec2 TextureCoord = (vec2){TextureCoord000};
    hvec4 gl_FragColor = neox_texture(TextureCoord, 1);
    gl_FragColor = _19;
    __builtin_neox_pixout(gl_FragColor, 0);
    yield();
}
Neox Compiler Future Feature

- GLSL
- SPIRV
- C/C++ OpenCL

GLSLANG ➔ SPIRV-Cross ➔ LLVM with SPIRV converter ➔ NeoxLib

Executable binary
NEOX™ Graphics Instructions

- RV64GC
- RVV
- Graphics Extensions (optional)
- AI Extensions (optional)
- Custom Instructions

- Thread management (fork, yield etc.)
- Load from Texture Units (readtex)
- Store color / Z value to Framebuffer/Tile Unit
- Barrier synchronization for threads (vertex, assembly)
- Half Float FP16/BFLOAT16 support
- Vector V2/V4 FP32
- Linear interpolation, dot product etc.
- Reciprocal/Inverse Square Root
Neox AI SDK

Datasets → Graph → Encapsulated Trained Network

TensorFlow Graph

AI SDK

Target HW validator

Analyzer

Quantizer / Compression

TensorFlow Lite graph

AI Inference Kernels

Customized Kernels

RISC-V C/C++ Compiler

AI Inference Runtime

TFLite / MCU

Runtime

Neox HW

Neox AI SDK allows to perform various iterative steps in model compression and model analysis, until the desired balance between "accuracy-performance-memory" is achieved.
AI Demonstrator

- Neox Human Presence Detection Demo
  - USB Camera on Linux
  - Human detection pretrained model
  - TensorFlow Lite /MCU Backend
  - Neox Kernels custom Kernels

No Person Detected  Person Detected
- A **RISC-V** based ISA suitable for Graphics /AI/CNN and Vision Tasks
- **Flexible**: Leverage RISC-V ecosystem and OSS Tooling (GCC/LLVM)
- **Graphics**: Support for Vector and 3D Graphics
- **AI Inference**: TensorFlow Lite/MCU
- **Futureproof**: Support for common existing ML operations and datatypes and full programmability to accommodate future compute needs
- **Scalable Design**: 1-64 cores targeting from tiny to embedded market
- **Multithreading**: Lightweight and Efficient pipeline, high bandwidth throughput
- **Low Power**: Small Core Design with Very Low area and Gate Count