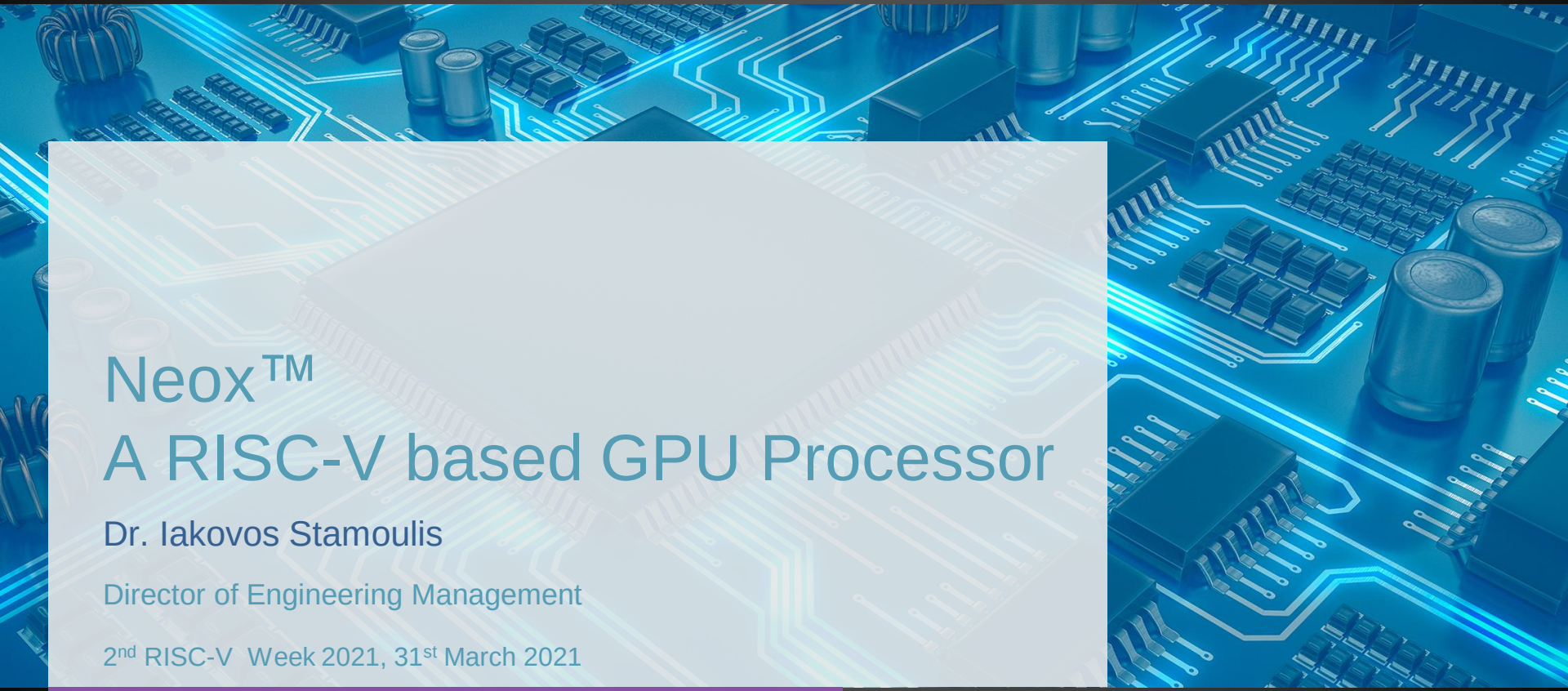


Neox™
A RISC-V based GPU Processor

Dr. Iakovos Stamoulis

Director of Engineering Management

2nd RISC-V Week 2021, 31st March 2021

Company

- **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

Atmel Microchip
ambiq micro
ST life.augmented
Lattice Semiconductor
dialog Semiconductor
General Dynamics
Faraday
Sequans Communications
ARC Synopsys
清微智能 TSING MICRO

2 x Undisclosed Tier1 US

2 x Undisclosed Tier1 China

Technology Partnerships

Samsung
LPGPU2
Horizon 2020 European Union Funding for Research & Innovation
codeplay
Technische Universität Berlin
ARC Synopsys

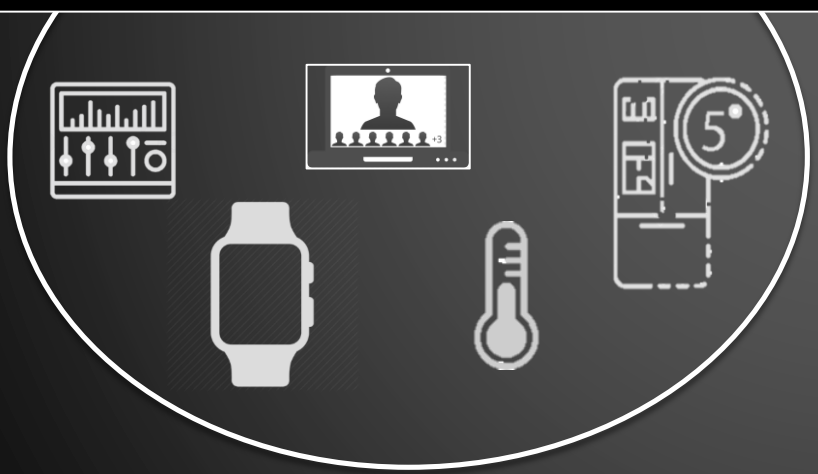
Different architectures for different markets & applications

CPU
32-bit

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

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

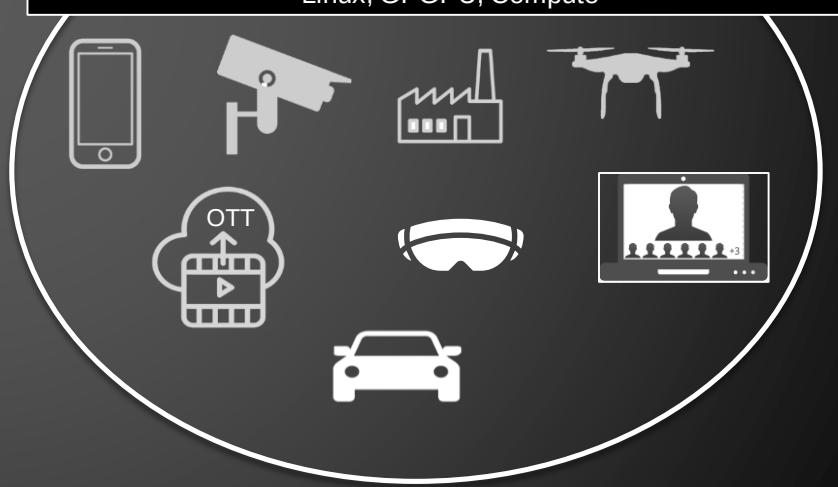


CPU
32/64-bit

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

Markets/Applications

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

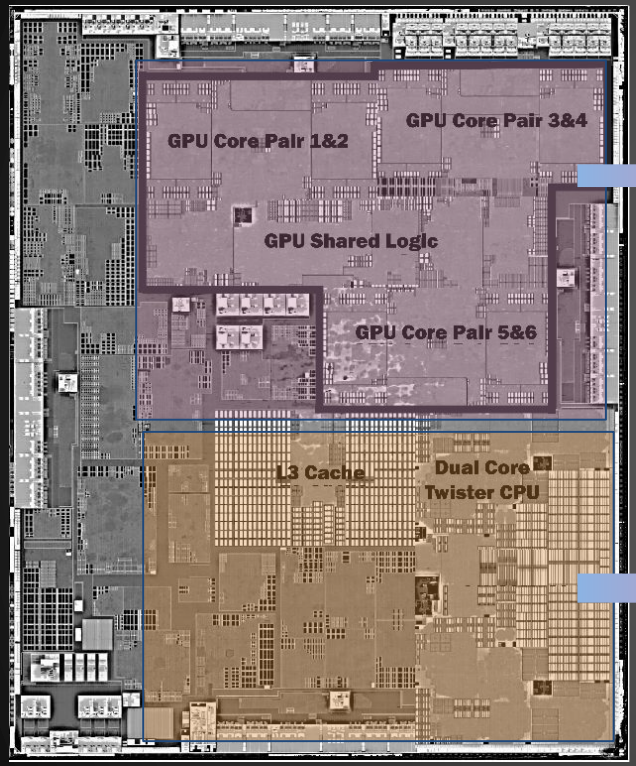


Different architectures for different markets & applications

CPU 32-bit	NEMA®| Pico-Series XS & XL 2D / 2.5D GPU	CPU 32/64-bit	NEOX™-Series AI & GFX AI Accelerator / 3D GPU

	Nema®XL	NEOX™
Applications	2.5D GPU	3D GPU / Compute / AI
Corescore	1-4	1-64
Clock Range *@28nm*	300 MHz	300 MHz
Performance	1 pixel/clock/cycle/core	1 pixel/clock/cycle/core
ISA	Nema VLIW	RV64IMFC + extensions
Shader Processor	Limited programmability Fragment processor	Fully Programmable GCC / LLVM C/C++ RISCV
FB Compression	yes	yes
Texture Compression	yes	yes
Memory System	AHB 32-bit AXI4 64/128-bit	AXI4 64/128-bit
AI Framework	-	Tensorflow Lite for MCU
Graphics Framework	NemaGFX + SDK GUI Builder PixPresso	NemaGFX + SDK 3D Extension GUI Builder PixPresso

NEOX™, a RISC-V based application specific processor



Think Silicon
NEOX™
GPU

CPU
Core(s)

CPU and GPU/AI 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

- **Neox Architecture**

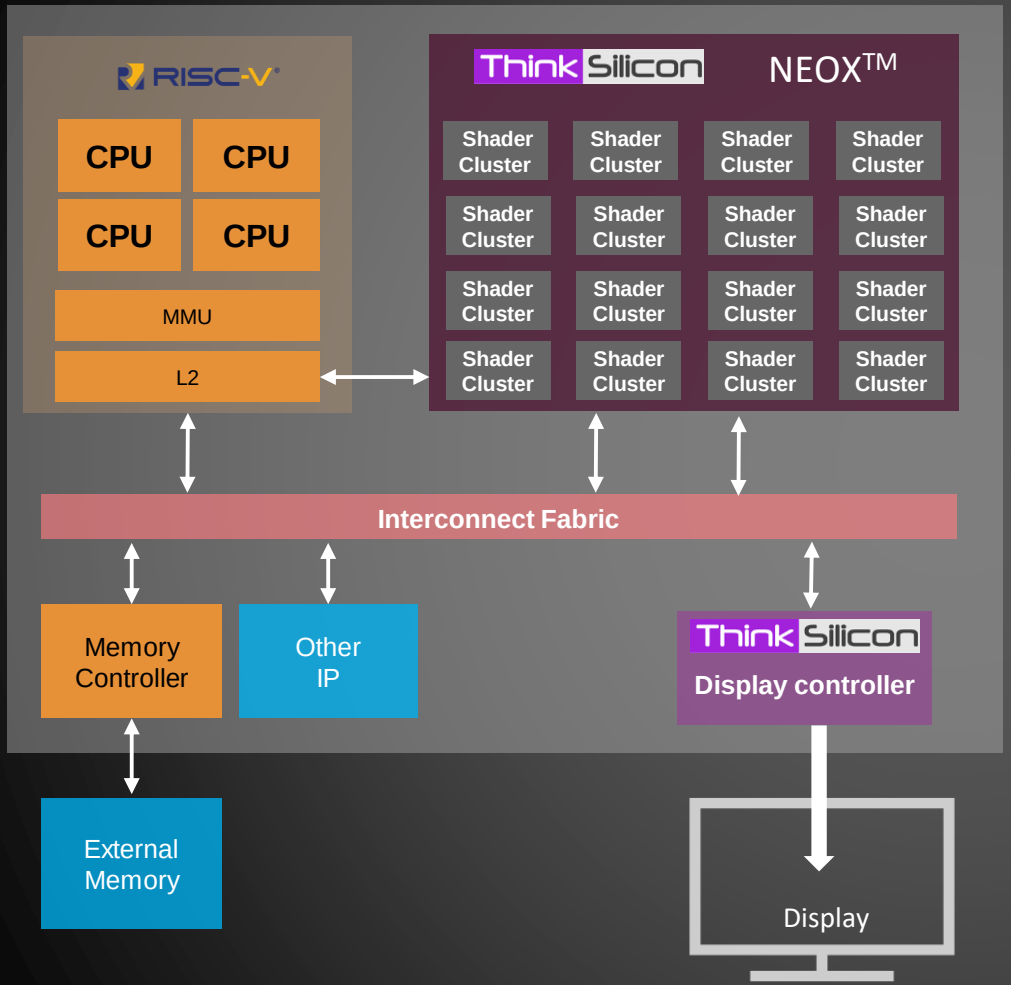
- **Software Stack**
  - SDK Tools
  - AI Inference
  - Graphics

CPU vs GPU

	CPU Core(s)	Think Silicon NEOX™ GPU
Workload	▪ Lots of instructions / Less data ▪ Task Parallel	▪ Few Instructions/Lots of Data ▪ Data Parallel
Features	▪ Out of order exec ▪ Pipeline Interlocks ▪ Branch prediction ▪ Complex sync	▪ Hardware Threading ▪ Barriers for synchronization
Data Reuse	▪ Data Reuse and Locality ▪ Latency Optimized	▪ Little Data Reuse ▪ Throughput Optimized
ISA	▪ Scalar Instruction ▪ +SIMD Extensions	▪ 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

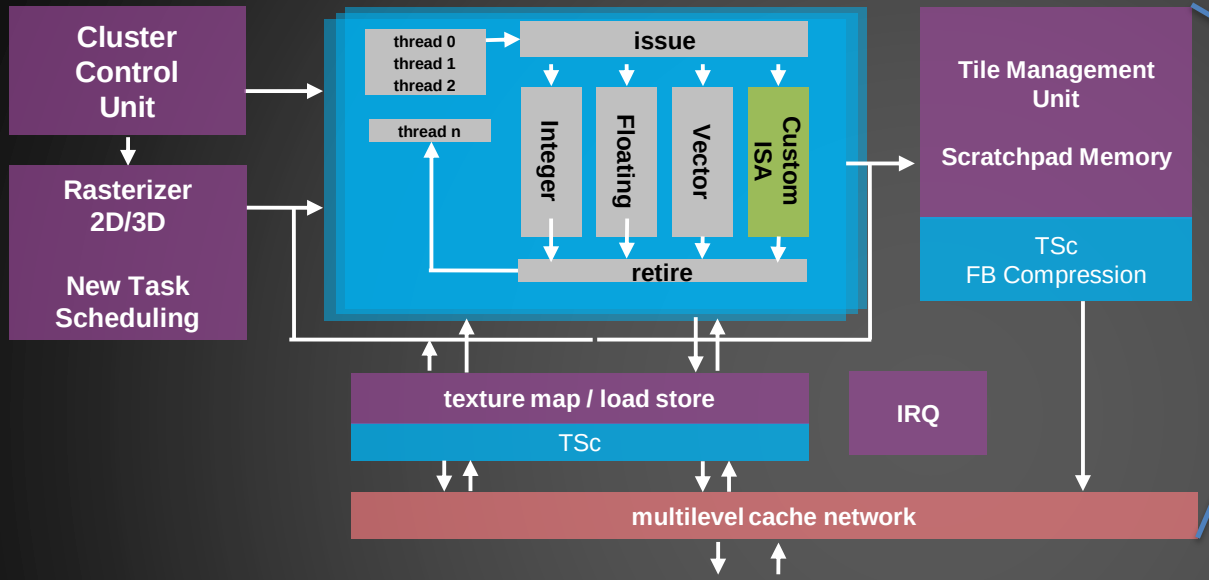
Graphics Rendering: Think Silicon graphics Libraries

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

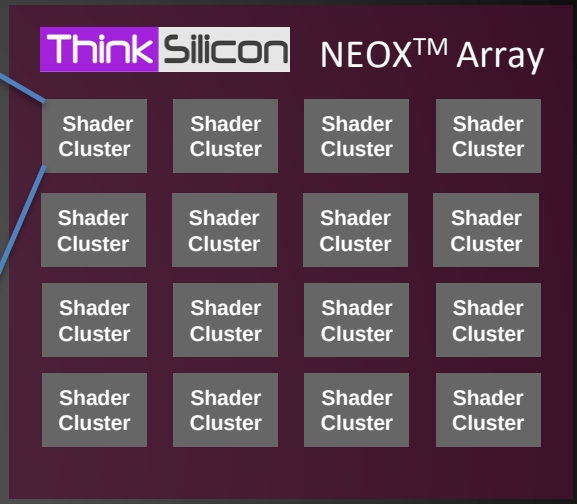
Extensible: User Defined Instructions

NEOX™: Cluster

NEOX™1-4 Cores per Cluster



NEOX™1-16 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

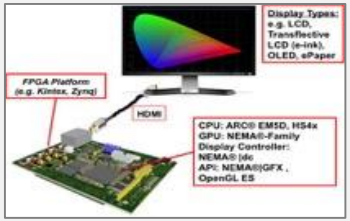
Dedicated Interface to support user provided extensions

Agenda

- **Neox Architecture**

- **Software Stack**
  - SDK Tools
  - AI Inference
  - Graphics

Neox®| SDK - *A Complete Ecosystem Of Tools*



Neox®|Bits

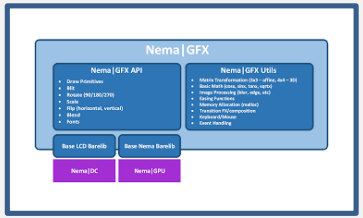
An EVK Kit for technology evaluation and pre-silicon application development

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

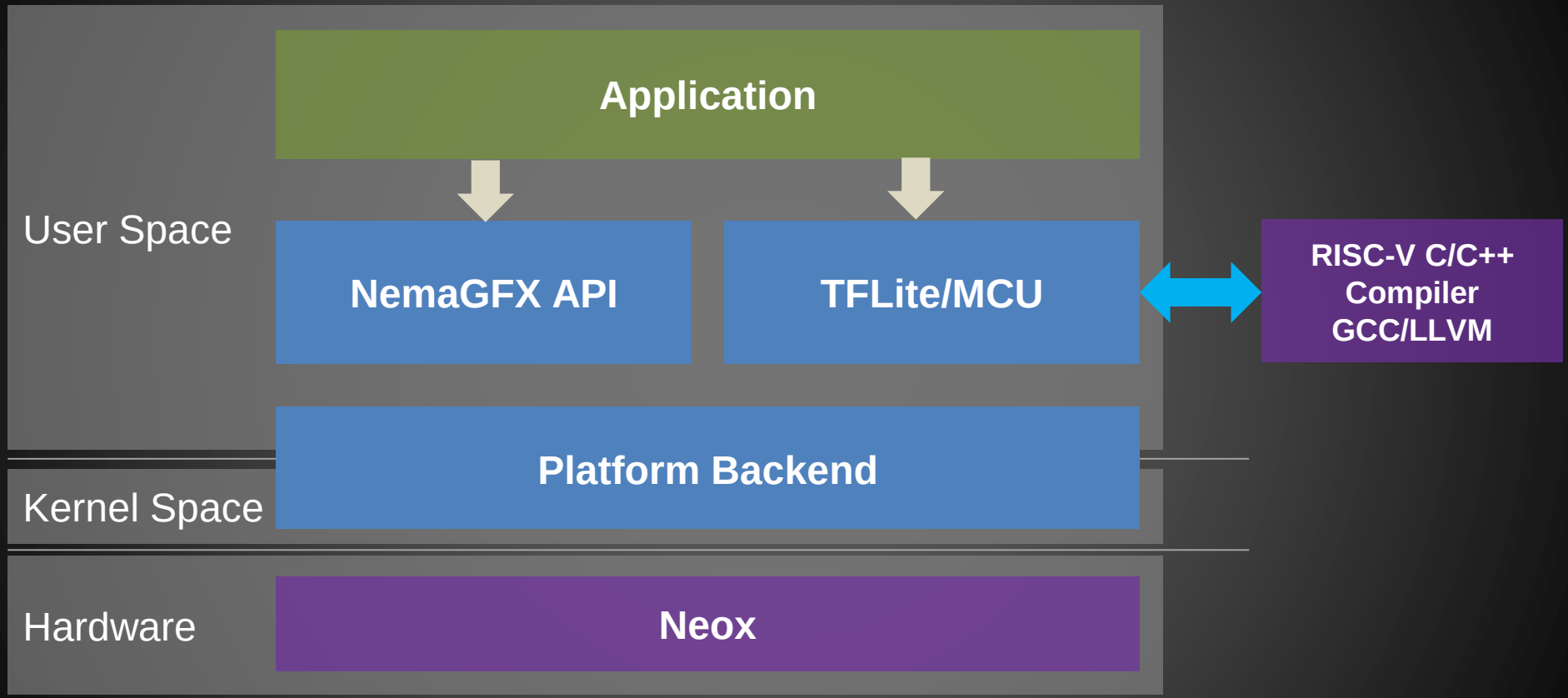
NEMA®|GUI-Builder

GUI Design Toolkit that allows drag & drop creation of advanced GUI, in minutes instead of months.

NEMA®|PIX-Presso

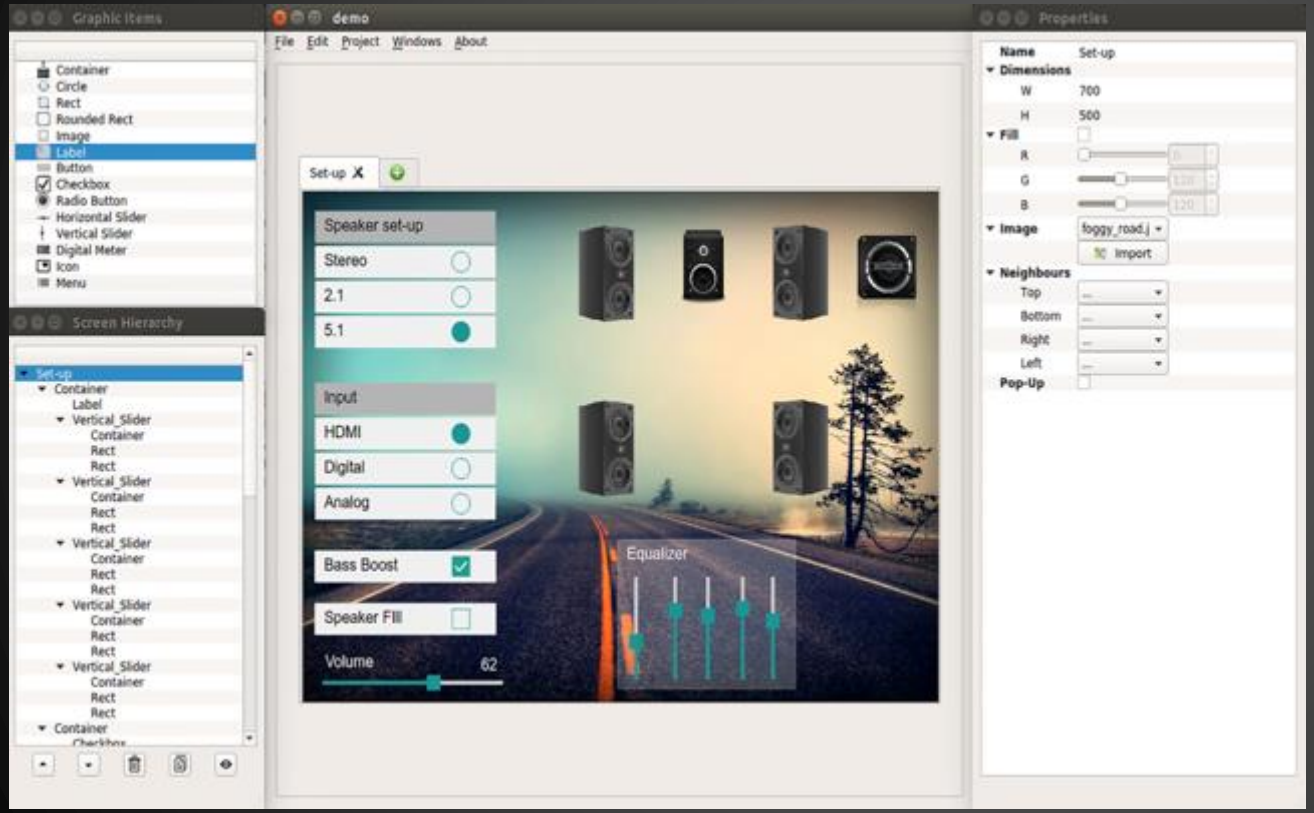
Asset management and image optimization, for optimal visual appearance and efficient memory utilization

Neox Software Stack



SDK:NEMA®|GUI-Builder

NEMA®|GUI-Builder allows the rapid creation of GUIs for Bare/RTOS or Linux embedded System within minutes instead of weeks



Optimized Graphics Assets (images, icons, fonts)

Automatic generation of optimized code

Application Code

NEMA®|GFX Library

HAL OS

Platforms:
Windows 10
Linux

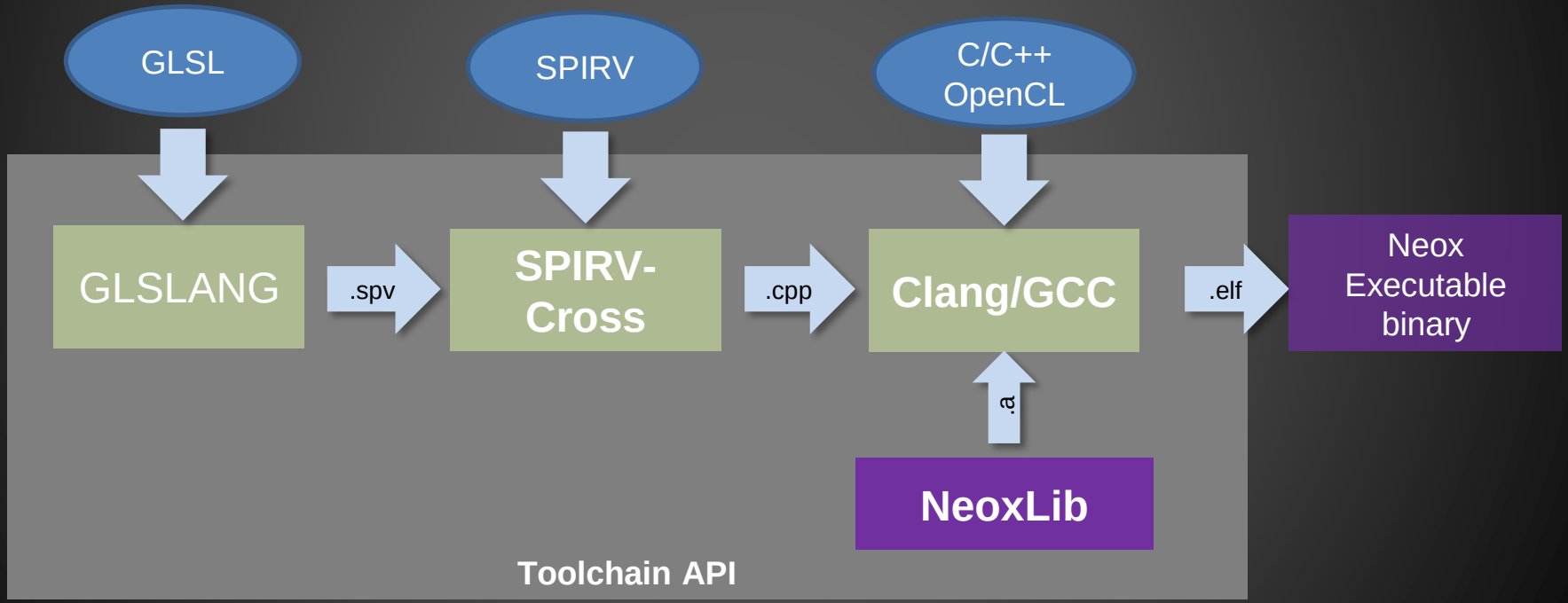
SDK:NEMA®|Pix-Presso

NEMA®|Pix-Presso is a utility for converting images to formats suitable for low power embedded devices. Its 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

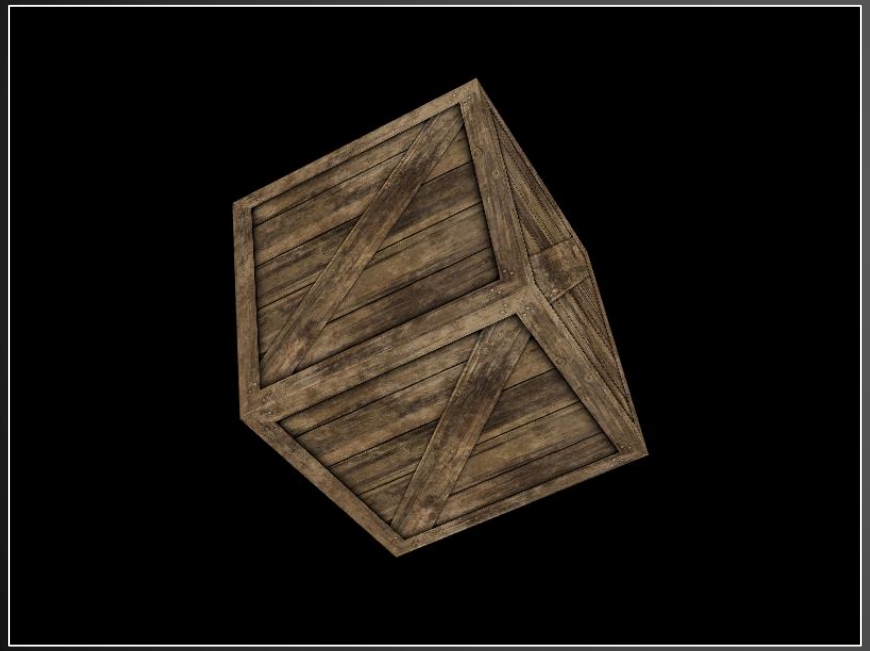
Neox Toolchain

Neox Toolchain API can accept code in multiple source languages and generate executable binaries for Neox



3D example

Simple 3D demo with a vertex and fragment shader.
Workload is split in dynamically in multiple threads for each code type



Vertex Shader

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

layout(location = 0) out vec2 TextureCoord;

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

Fragment Shader

```
#version 400

layout(binding = 2) uniform sampler2D MaterialTexture0;
layout(location = 0) in vec2 TextureCoord;

void main(){
gl_FragColor = Texture2D(MaterialTexture0,
  TextureCoord);
}
```

Neox Compiler: Vertex Shader

GLSL → C/C++ → Neox Assembly

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

layout(location = 0) out vec2
TextureCoord;

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

```
vec4 a_pos;
vec2 a_textCoord;
void main_shader(uint32_t rast_indx){
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 t3, a0, 352
 addi t4, a0, 360
 addi t5, a0, 368
 ld  v2, 8(a2)
 ld  v0, %lo(a_pos)(a1)
 lui t5, %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
```

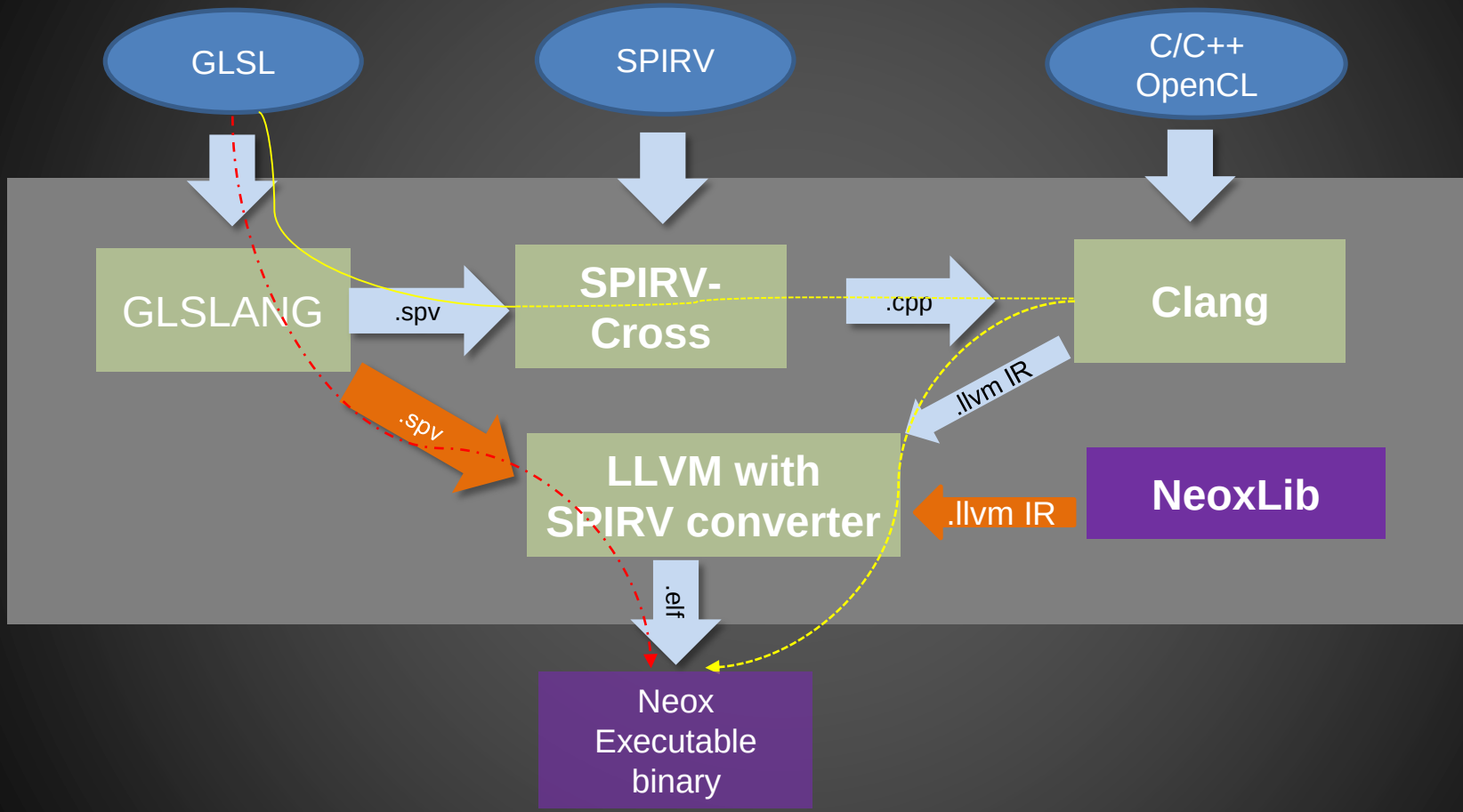
Neox Compiler: Fragment Shader


```
#version 400

layout(binding = 2) uniform sampler2D
MaterialTexture0;
layout(location = 0) in vec2 TextureCoord;
void main(){
gl_FragColor =
Texture2D(MaterialTexture0,
  TextureCoord);
}
```

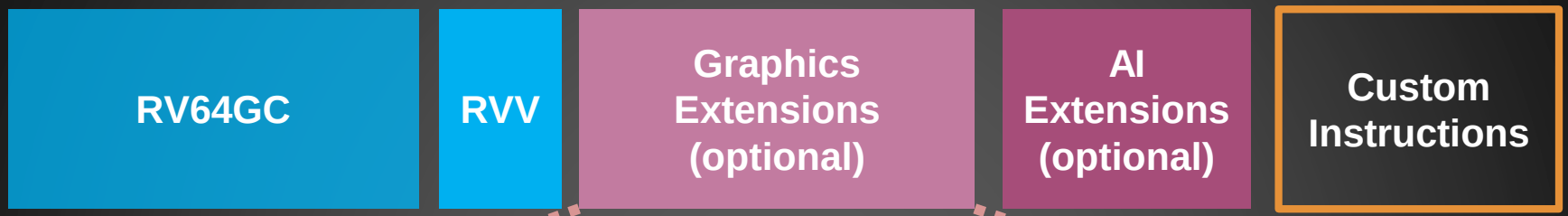
```
vec4 a_pos;
void main_shader(uint32_t rast_indx){

  auto TextureCoord000 = neox_read_reg_vec2(0);
  vec2 TextureCoord;
  TextureCoord = (vec2){TextureCoord000};
  hvec4 gl_FragColor;
  hvec4 _19 = neox_texture(TextureCoord,  1);
  gl_FragColor = _19;
  __builtin_neox_pixout(gl_FragColor, 0);
  yield();
}
```

```
.globl  _Z11main_shaderj
_Z11main_shaderj:
  readtex.txty  v0, v0, 1
  pixout v0, 1
  yield
```

Neox Compiler Future Feature



NEOX™ Graphics 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

TensorFlow

Caffe2

Microsoft CNTK

Keras

Cloud & Desktop Training

Encapsulated Trained Network

ONNX

TensorFlow Lite /MCU Runtime

Import and execute AI Graphs From various industry formats e.g. ONNX

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

TensorFlow Graph

AI SDK

Target HW validator

Analyzer

Quantizer / Compression

TensorFlow Lite graph

TensorFlow Lite

AI Inference Kernels

Customized Kernels

RISC-V C/C++ Compiler

AI Inference Runtime

TFLite / MCU

Runtime

Neox HW

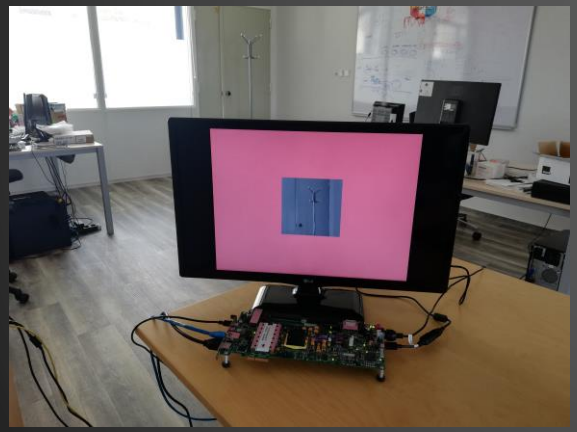
"Offline" - Training

"Offline" - Optimizations

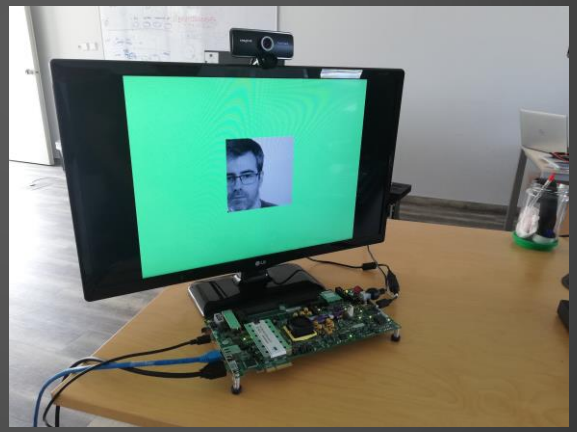
"Online" - Inferencing

AI Demonstrator

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

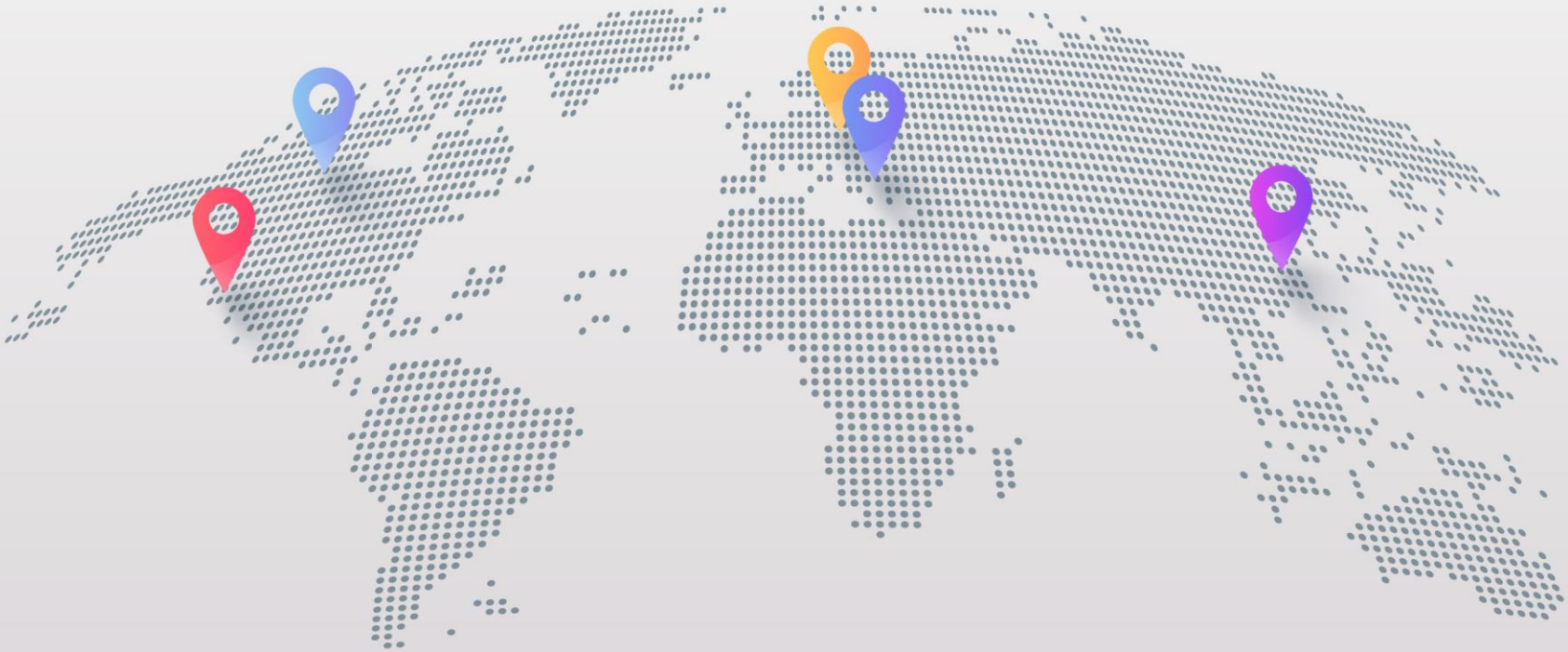


No Person Detected



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



HQ & Development Center
Patras Science Park
Rion Achaias | 26504 | Greece
Tel: +30 2610 910650

Development Center
121, Tatoiou Av. | 14452
Metamorfosi Attiki | Athens | Greece
Tel: +30 210 6895191

Sales EMEA
Think-Silicon S.A.
Email: sales_EMEA(at)think-silicon.com
Tel: +49 170 63 65 370
Tel: +49 173 57 06 378

Business Development
Think-Silicon S.A.
Canada
Tel: +1 437-236-8873

Sales NA
Think-Silicon S.A
Email: sales_NA(at)think-silicon.com
Tel: +1-5126239571

Sales APAC
Think-Silicon S.A
Email: sales_APAC(at)think-silicon.com
Tel: +886-903-962863

Think Silicon®
AN APPLIED MATERIALS COMPANY


Think Silicon®
AN APPLIED MATERIALS COMPANY