USER'S MANUAL

Nema Pico Graphics Processing Unit

A-SOCAPG-UMGA06EN v1.0

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. "TYPICAL" PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING "TYPICALS" MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER'S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Revision History

Rev	Date	Description
1.0	December 2025	Initial Release

Reference Documents

Document ID	Description
A-SOCAPG-UMGA01EN	Nema DC API Library User's Manual
A-SOCAPG-UMGA02EN	Nema GFX API Library User's Manual
A-SOCAPG-ANGA01EN	Nema DC MiP Panels Configuration Application Note
A-SOCAPG-ANGA02EN	Nema GFX Extensions TSVG Supported Elements List Application Note
A-SOCAPG-SGGA01EN	Nema Pixpresso Starting Guide
A-SOCAPG-ANGA03EN	Nema GFX API Debugging Application Note
A-SOCAPG-UMGA04EN	Nema GUI-builder User's Manual
A-SOCAPG-UMGA05EN	Nema GFX Benchmark Suite User's Manual
A-SOCAPG-UMGA03EN	Nema GFX Extension Vector Graphics User's Manual
A-SOCAPG-UMGA07EN	Nema Pico Platform Drivers User's Manual

Table of Contents

List of Figures

SECTION

1

Introduction to Graphics

Computer graphics is the science of communicating visually via a display and its interaction devices. It is a cross-disciplinary field in which physics, mathematics, human perception, human-computer interaction and engineering blend, towards creating artificial images with the help of programming. It heavily involves computations, creation and manipulation of data and is based on a set of well-defined principles. There are several structural elements that computer graphics are built upon, the most significant of which can be found in the following list:

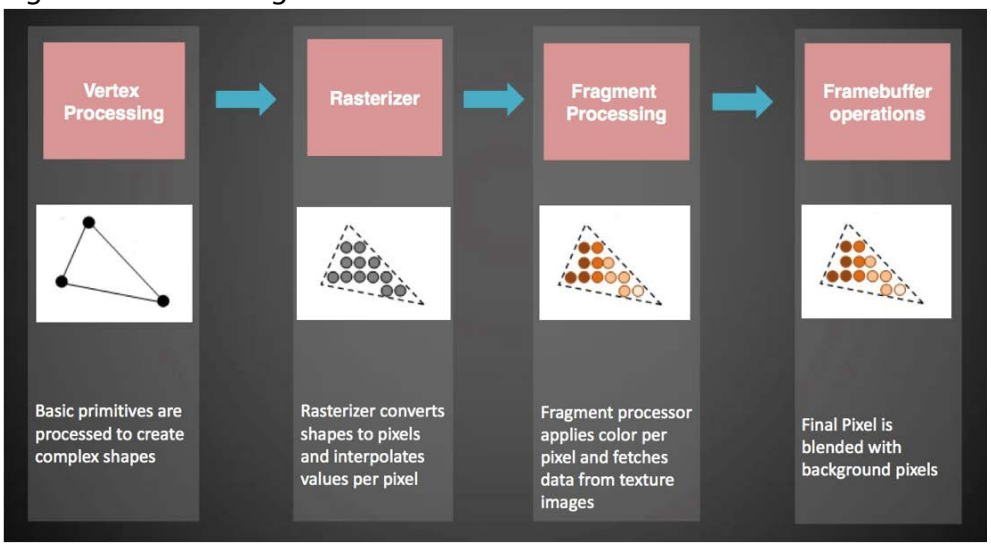
- **Pixel** in digital imaging is the smallest addressable element in an all points addressable display device. A pixel is generally considered as the smallest single component of a digital image and is often used as a measurement unit.
- **Raster** image (or bitmapped image) is a matrix data structure representing the actual image content. Raster graphics are resolution-bound therefore unable to scale up without apparent loss of quality.
- **Vector** graphics is a technique of using polygons, plane figures bound by a finite chain of straight-line segments closing a loop, to represent images. Vector graphics have inherent scale up abilities, only depending on the rendering device capability.
- **Rasterization** is the process of converting an image described in a vector graphics format to a raster image consists of pixels for output on a video display or for storage in a bitmap format.
- **Texture** is the digital representation of an object's surface. In addition to two-dimensional qualities such as color and transparency, a texture also incorporates three dimensional ones such as reflectiveness. Well-defined textures are very important for realistic three-dimensional image representation.
- **Texture mapping** is the process of wrapping a pre-defined texture around any two or three dimensional object. Through this process, digital images and objects obtain a high level of detail.
- **Texel** is the fundamental unit of texture space. Textures are represented by arrays of texels in the same way that pictures are represented by arrays of pixels.

- **Vertex** is a data structure that describes the location of an object by properly define its corners as positions of points in two or three-dimensional space.
- **Primitives** in computer graphics are the simplest geometric objects a system can handle. Common sets of two-dimensional primitives include lines, points, triangles and polygons while all other geometric elements are built up from these primitives.

  In three-dimensions, properly positioned triangles or polygons can be used as primitives to model more complex forms.
- **Blending** is the process in which two or more images are combined per-pixel and weights to create new pictures.
- **Fragment** is the data necessary to generate a single-pixel primitive. This data is possible to include raster position, color or texture coordinates.
- **Interpolation** in computer graphics is the process of generating intermediate values between two known reference points to give the appearance of continuity and smooth transition. Several distinct interpolation techniques are used in both computer graphics and animation, such as linear, bilinear, spline and polynomial interpolation.
- **Graphics Pipeline** is an abstract sequence that incorporates the basic operations of generic rasterizer implementations, in particular:
  - (Vertex) Per-vertex transformation to screen space
  - (Rasterize) Per-triangle iteration over pixels with perspective-correct interpolation
  - (Pixel) Per-pixel shading
  - (Output Merge) Merging the output of shading with the current color and depth buffers

As stated in the term **pipeline**[1] is used due to the sequential steps that are used for the actual transformation from mathematical model to pixels; the results of the one stage are pushed on to the next stage so that the first stage can begin processing the next element immediately.

Figure 1-1: Rendering Flow



[1] John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner and Kurt Akeley, Computer graphics: principles and practice (3rd ed.), Addison-Wesley Professional, Boston, MA, USA, 2013.

SECTION

2

Nema Pico Graphics Pipeline

Nema Pico has been designed for graphics efficiency in ultra-compact silicon area. Its fixed-point data path and instruction set architecture (ISA) are tailored to GUIs acceleration and small display applications leading to substantial improvements in power consumption and silicon area. Nema Pico micro architecture is based on a lean version of Nema ISA and it combines hardware-level support for multi-threading, VLIW and low level vector processing in the most power efficient way. Nema Pico is connected to the host SoC processor via a 32 bits AHB or 32/64 bits AXI4 bus. The internal architecture of Nema Pico and the main components of the graphics pipeline are shown in Figure 2-1 on page 10.

2.1 Configuration Register File

Nema GPUs are programmed through a set of registers. This set is called the Configuration Register File (CRF) and each sub-module of the GPU is programmed through a subset of the CRF. The CRF can be memory mapped to the CPU address space, thus making it directly accessible. Writing the CRF directly is considered inefficient since it consumes a large volume of the CPU resources and ties the CPU execution to the GPU. For this reason, it can also be accessed indirectly through the Command List Processor (CLP).

2.2 Command List Processor

In order to decouple CPU and GPU execution and achieve both better performance and lower power consumption, Nema GPUs incorporate an advanced Command List Processor (CLP), capable of reading entire lists of commands from the main memory and relay them to the Configuration Register File.

The CPU pre-assembles Command Lists (CL) prior to submitting them to the Command List Processor for execution, while a single Command List can be submitted

multiple times. This approach alleviates the CPU from recalculating drawing operations for repetitive tasks, resulting in more efficient resource utilization.

The steps for writing commands to the Configuration Registers through the Command List Processor are the following:

1. The CPU assembles a Command List, through the Nema GFX Library.
2. The CPU submits the Command List for execution. The Command List Processor is informed of a pending Command List.
3. The Command List Processor reads the Command List from the System Memory.
4. The Command List Processor relays the commands to the Configuration Register File.

2.3 Rasterizer

Nema GPUs are capable of drawing a multitude of geometrical shapes called Geometric Primitives, such as lines, rectangles, triangles and quadrilaterals. The Rasterizer Unit 2 Nema Pico Graphics Pipeline reads the coordinates of the primitives' vertices and feeds the rest of the graphics pipeline with the fragments contained in the geometry. A fragment contains information concerning a single pixel. This information includes raster position (coordinates), texture coordinates, interpolated color and alpha values.

2.4 Texture Map Unit

The Texture Map Unit produces texels that sends to the Fragment Processing Core. It is fed with texture's attributes (base address, dimensions, color format) and the required coordinates. The Texture Map Unit performs some internal processing and outputs the corresponding texel. Generating a texture element requires a series of operations like wrapping (clamp, mirror, repeat etc.), reading corresponding color values from memory, converting the color values to RGBA8888 format and performing filtering if necessary. If Think Silicon's proprietary texture compression technique is used, then on-the-fly decompression is performed.

2.5 Fragment Processing Core

The Fragment Processing Core is the main processing unit of the Nema GPU's architecture. It is a 64-bit VLIW processor which performs computations on the fragments coming from the Rasterizer Unit and on the texels coming from the Texture Map Unit and calculates the final color to a fragment. The Core is programmable through binary executables called Fragment Shaders.

2.6 Render Output Unit

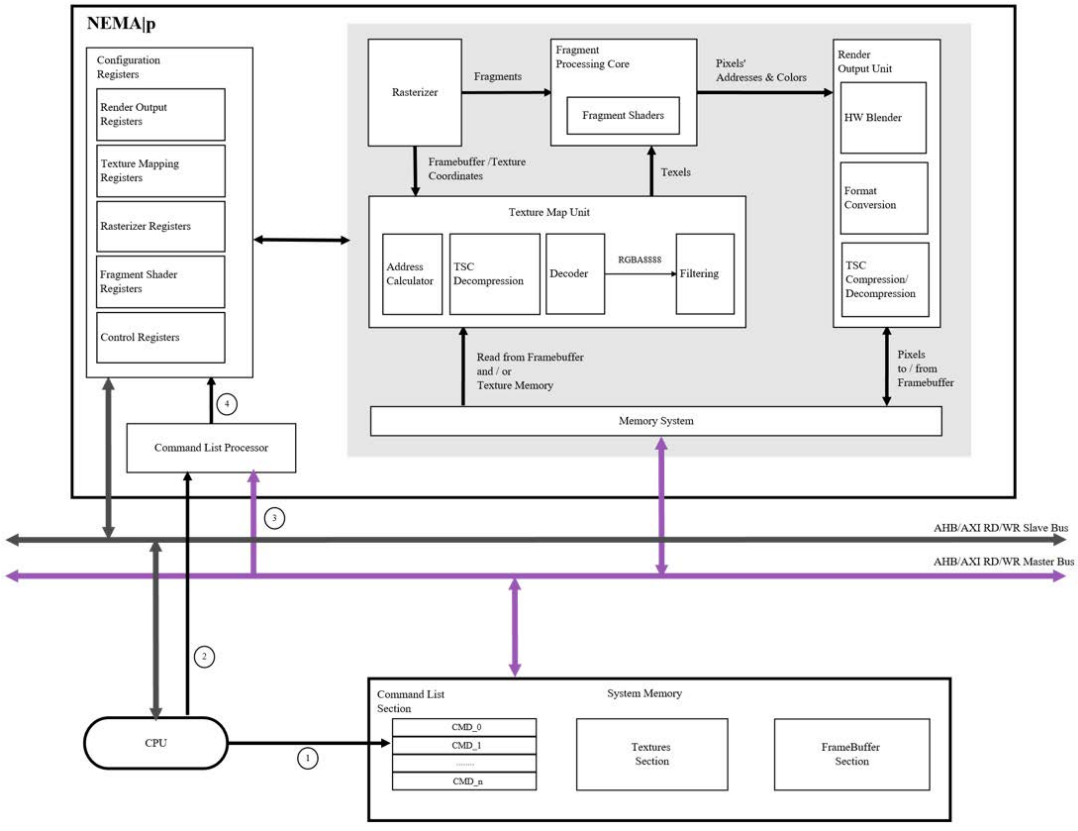
The Render Output Unit (ROP) is the last stage of the Graphics Pipeline. The Fragment Processing Core feeds the Render Output Unit with the pixel's coordinates and color value. Before the color value is written to the memory, the color is converted to the Frame Buffer's format. If Think Silicon's proprietary texture compression technique is used, then decompression is performed while reading from the Frame Buffer and compression is performed while writing to the Frame Buffer.

If H/W Blender is enabled, the Render Output Unit reads pixels from the Fragment Processing Core (source) and pixels from the Frame Buffer (destination) to perform blending. Blending requires a series of calculations between the source (foreground) and destination (background) color fragments to produce the final color, which is written back to memory.

2.7 Memory System

The Memory System Unit is responsible for the interconnection of the Nema GPU's internal channels to the AHB/AXI master buses depending on the system configuration. For more information regarding Nema Pico graphics pipeline refer to the *Nema Pico Hardware User Manual*.

Figure 2-1: Nema Pico Graphics Pipeline



SECTION

3

Linux Kernel Driver

Think Silicon provides the GPU device driver for the Linux operating system, which enables user processes to use Nema Pico. The driver configures hardware parameters and allows communication between the host CPU and Nema Pico. The main responsibility of the driver is to properly initialize the kernel module, detect the GPU, perform the necessary setup, and handle hardware interrupts.

3.1 Requirements

- Nema Pico driver has been tested and optimized for Linux Kernel 3.10.0.
- Nema Pico driver has been tested on Ubuntu 16.04 OS running on top of 32-bit Arm processor

3.2 Folder Structure

Nema Pico driver folder structure and a brief description are shown below:

- include
  - nema_incl.h: Nema Pico driver public header file.
- src
  - **build_nema_driver.sh**: Nema Pico driver building script
  - Makefile
  - **nema.c:** Nema Pico driver implementation file
  - **nema.h**: Nema Pico driver internal header file
  - **README**: Nema Pico driver building instructions

SECTION

4

Software Stack

Nema Pico can be utilized either via Nema GFX drivers.

4.1 Nema GFX

Nema GFX is a low overhead software library that interfaces directly with the Nema GPUs and provides a software abstraction layer to organize and employ drawing commands with ease and efficiency. The main goal of Nema GFX is to be able to be used as a backend to existing APIs (such as OpenGL or any proprietary one) but also to expose higher level drawing functions, so as to be used as a standalone Graphics API.

Its small footprint, efficient design and lack of any external dependencies, makes it ideal for use in embedded applications. By leveraging Nema's sophisticated architecture, it allows great performance with minimum CPU/MCU usage and power consumption.

Nema GFX includes a set of higher level calls, forming a complete standalone Graphics API for applications in systems where no other APIs are needed. This API is able to carry out draw operations from as simple as lines, triangles and quadrilaterals to more complex ones like blitting and perspective correct texture mapping.

Nema GFX is supported in Linux and bare-metal based systems. For further information please refer to the Nema GFX Library User Manual.

4.1.1 Requirements

Nema GFX is implemented in pure C and therefore it can be compiled with any compiler that supports C99. A small abstraction layer is the only prerequisite to make Nema GFX run on bare-metal, RTOS and Linux based systems.

4.1.2 Directory Structure

Nema GFX Library directory structure is the following:

```
NemaGFX_SDK
├── common
├── examples
├── include
├── NemaGFX
└── utils
```

Folder common contains the necessary files for the memory management as well as some utility functions. examples include some sample applications that make use of Nema GFX. Folder include contains the header files required by Nema GFX. Nema GFX includes the Nema GFX source code and utils includes a utility application used for converting TTF fonts to raster fonts.

SECTION

5

Additional Software Tools for Nema Pico

Think Silicon developed two tools to ease the software development for Nema Pico GPUs and also to increase the capabilities of its hardware solutions.

5.1 Nema GUI Builder

Nema GUI Builder is a GUI Design Toolkit that allows drag and drop creation of advanced GUIs. With the Nema GUI Builder a software developer can automatically generate performance optimized executable code and API calls, with small memory footprint while making extensive use of the Nema Pico hardware acceleration features.

More details about Nema GUI Builder can be found at https://www.think-silicon.com/guibuilder, from where you can also download a non-commercial version of the toolkit.

5.2 Nema Pixpresso

Nema Pixpresso is a utility software for converting images to/from formats suitable for low power embedded devices. It supports both uncompressed and various compressed image formats that allows the developers to explore image quality versus performance/power consumption trade-offs. The tool also supports Think Silicon's proprietary and patented compressed formats (TSC4, TSC6 and TSC6a) that offer high compression ratios (4 and 6 bits-per-pixel respectively). TSC texture and framebuffer compression formats are ideal solutions for devices with scarce on-chip storage and memory bandwidth resources.

For further information, download a non-commercial version of the Nema Pixpresso User Manual and the Nema Pixpresso for Windows and Linux from: https://www.think-silicon.com/nema-pix-presso

6500 River Place Boulevard, Building 7, Suite 200, Austin, TX 78730
www.ambiq.com
sales@ambiq.com
+1 512. 879.2850

A-SOCAPG-UMGA03EN v1.0
December 2025